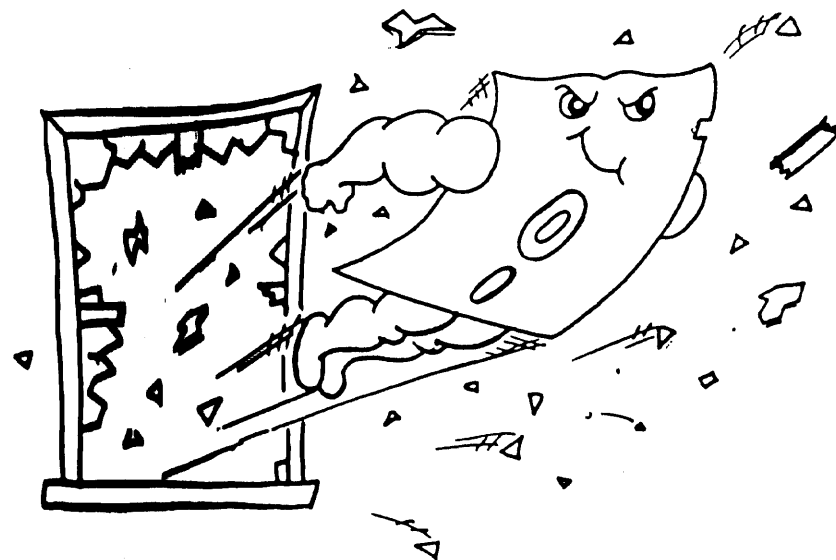


# *DATA WINDOWS*



**REFERENCE MANUAL**

## TABLE OF CONTENTS

<b>Quick Reference Guide</b>	1
Creating a Database System	1
Creating a Database	1
Creating a Windows	1
Creating a Report	2
Creating Mailing Labels	2
Using a Database System	2
Opening Windows	2
Printing Reports and Labels	3
Miscellaneous	3
Merging Data	3
Purging Data	3
Packing Databases	3
Importing Data	4
Exporting Data	4
 <b>Reference</b>	 5
File Menu	6
Create Database	6
Adding fields	6
Changing fields	7
Deleting fields	7
Create Window	7
Window database information	8
Sizing and Placing the window	8
Coloring the window	8
Defining window fields	8
Defining GET fields	9
Defining PUT expressions	9
Defining TEXT fields	9
Deleting window fields	10
Completing the window	10
Create report	10
Report database information	10
Report width/height	10
Report lines/record	11
Editing the report fields	11
Create Labels	11
Label database information	12
Label width/height	12

Editing the mailing labels	12
Open window	12
Close window	13
Print report	13
Print labels	14
Quit	14
Edit Menu	15
Undo	15
Cut	15
Copy	15
Paste	15
Edit Database	16
Edit Window	16
Edit Report	16
Edit Labels	16
Records Menu	17
Find Record	17
Browsing Records	17
First Record	17
Last Record	17
Next Record	17
Previous Record	17
Edit Record	18
Delete Record	18
Add Records	18
Database Menu	19
Merge Databases	19
Purge Databases	19
Pack Databases	19
Import Data	20
Export Data	21
Appendix A: Configuring Your System	22
Appendix B: DB9 - 90 File Specifications	24

## Quick Reference Guide

This section will quickly describe the use of the Data - Windows database development package. This is not a complete description of Data - Windows, it is only intended to allow users to get up and running quickly.

### Creating a Database System

A Data - Windows database system consists of at least 2 parts, a database and a data window. The database is where the data is stored, and a data window is where the user accesses the data in the database.

In addition to a database and a data window, a database system may contain reports, and labels. A report allows you to print a formatted listing of all the records in a given database. Labels allow you to print mailing labels from the data in the database.

All of the above can be created from the FILE menu. This is the first menu in the Data - Windows menu bar. Simply select either: Create Database, Create Window, Create Report, or Create Label.

The creation of each of these items is fairly straightforward. The program will prompt for all necessary information.

### Creating a Database

The creation of a database will require the following information for each field:

Field name - Name of the data field

Field type - Type of field C = character, N = numeric

Field size - Width of the field (in characters. NOTE: numbers are stored as characters)

Index file - A file in which to store the index for a field. This field is optional and is only used if you would like the field to be indexed (a key field).

### Creating a Window

The creation of a window requires several pieces of information.

The database associated with this window.

The field to be used as a key (must be an index field).  
The size and placement of the window (using mouse or arrow keys).  
The background, foreground, and border color of the window.  
The placement of each database field (arrow keys/mouse).  
The edits (picture) associated with each field, defined as follows:  
'9' – any digit, or decimal point, or minus sign.  
'X' – any character.  
'I' – uppercase characters only

any other character is automatically placed in the field and cannot be edited.

Examples: (999)999-9999 – phone number  
XXXXXXXXXXXXXXXXXX – 15 character name

And finally the access to the field as follows:

GET – Data in this field can be edited.  
PUT – Data in this field can only be viewed.

## Creating a Report

The creation of a report requires the following information:

The database to use for this report.  
The field to use as a key. (report will be sorted by this field)  
The width of the report.  
The number of lines per record for the report.  
The placement of each field in the report.  
The editing (picture) of each field in the report (same editing as in windows).

## Creating Mailing Labels

The creation of labels requires the following information:

The database to use for these labels  
The field to use as a key (labels will be sorted by this field).  
The height and width of each label.  
The placement of each field in each label.  
The editing (picture) of each field in each label (same editing as in windows).

## Using a Database System

## Opening Windows

Once you have created the database system (at least 1 database and 1 window), you will want to use it. To do this, simply select the OPEN WINDOW option from the FILE menu. You will then be prompted for the name of the window to open. Once you have selected the window, it will be opened for you.

On your screen you will see the window you created, with the fields you have given. From here you may browse, search, edit, add and delete records from the database. To do all of these, simply select the correct option from the RECORDS menu.

You can also CUT, COPY, and PASTE records from the EDIT menu. This allows you to move data from one database to another easily. Simply select these options from the EDIT menu.

## Printing Reports and Labels

After using the database, you will want to see some results. To do this you will want to either print a report or some mailing labels. On the FILE menu there is a PRINT REPORT, and a PRINT LABELS option. Select either of these options and you will be prompted for the name of the report or label to print. Select the correct report or label file, select the output device (Example: /p), and the report will be printed.

## Miscellaneous

There are several other options available from within Data – Windows.

## Merging Data

You may merge data from another database by selecting the MERGE DATABASE option from the DATABASE menu. The MERGE DATABASE option will prompt you for a source and destination database. All fields names that match the destination database from the source database will be copied over.

## Purging Data

You may purge a database by selecting the PURGE DATABASE option from the DATABASE menu. When you purge a database you remove ALL records from that database. **WARNING:** Selecting this option will clear out all data in a database. Do not select this option if you have valuable data in the database!

## Packing Databases

From time to time you may need to pack databases. Packing a database removes all deleted entries (they are only flagged for deletion) and re-generates all database indexes. This option can be used when index files get corrupted and need to be re-generated.

**NOTE:** Packing a database is simply an EXPORT followed by anIMPORT (see below). For this reason there must be sufficient space on the disk for the import and export of the data.

## Importing Data

It may be necessary, from time to time, to retrieve data from another database system. For this you can use the IMPORT DATA from the DATABASE menu. This option will prompt you for the name of a text file from which to import the data. The text file should be of the following format:

"field 1 data","field 2 data","field 3 data",...

**NOTE:** the quotes are optional, but are recommended. They are needed if the field contains a comma.

## Exporting Data

Data can be sent to other applications via the EXPORT DATA option on the DATABASE menu. Data is exported in the same format as it is imported (see above).

## Reference

The Data – Windows database system consists of several menus. On each of these menus is a set of options. Each option performs some special function.

The Data – Windows database system is organized into 4 menus: FILE, EDIT, RECORDS, and DATABASE.

The FILE menu performs all file handling and print handling features of the system. This includes creating files, opening files, printing files, and quitting the system.

The EDIT menu performs all edit functions. This includes cut, copy, paste, and any file editing.

The RECORDS menu performs all database record handling functions. These include find record, first record, last record, next record, previous record, edit record, delete record, and add record.

The DATABASE menu performs all database-wide functions. These include merge database, purge database, pack database, import data, and export data.

In this section of the manual we will step through each of these operations and describe in detail how each works and what each is for.

## The FILE Menu

The file menu, as mentioned before, is where all file handling operations are. The file operations are for creating, opening, and printing files. The quit option is also stored on this menu. Let's take a close look at each of these options.

### Create Database

The CREATE DATABASE option allows you to create a new database for storing data. This option will prompt you for all information necessary to create a new database.

After selecting the Create Database option, you are prompted for the name of the new database file. Enter a valid OS9 filename in which to store the database. After you enter the OS9 filename you will be put into the database field editing mode.

The database editing mode presents you with a window that looks like this:

Field name	Type	Length	Index file (optional)
------------	------	--------	-----------------------

The database field editing mode allows you to perform the following operations:

- Add fields to the database
- Delete fields from the database
- Modify fields in the database

### Adding fields:

To add a field to the database, press the A key. A blank field line will be inserted into the database and you will be allowed to edit it.

The first prompt is for the Field name. The field name is a 10 character name that you will use to identify this particular field. Later you will need this field name to refer to this piece of information.

The second prompt is for the field type. Two types of fields are supported, character and numeric. If you would like a character field enter a C, if you want a numeric field use an N. A field should only be numeric if some calculations will be performed on that field, like totals, etc.

The next prompt is for the field length. The field length is the number of bytes that will need to be used to store your data. For character data it is simply the most characters that will be used for the field. Numeric data is stored as characters, therefore the length field should be equal to the total number of characters that will need to be used to represent the desired number. For example: the number 100.00 would require 6 characters. The number -450.459 would require 8 characters.

The final prompt is for the field index file. This information is NOT REQUIRED and should only be used if you want this field to be a key field. Key fields are fields that will be used for searching and sorting records. Most databases have only one or two key field, but there is no limit.

### Changing fields:

To change a field you must press the C key while the cursor is on the field that you wish to edit. The cursor can be moved from field to field with the arrow keys or the mouse. If you are using a mouse you may double click (<click><click>) on a field to change it.

After you select to change a field you will be prompted for the same information as when adding a field, but the information will already be keyed in for you. The arrow keys may then be used to edit the data. The <ENTER> or <DOWN ARROW> keys will move you to the next prompt. The <UP ARROW> key will move you to the previous prompt.

When the last prompt (the field index filename) is entered, you will return to normal operation.

### Deleting fields:

To delete a field from the database, press the 'D' key while the cursor is on the field to delete. As with changing field, the cursor can be moved with either the arrow keys or the mouse.

### Create Window

The CREATE WINDOW option allows you to create a data window. A data window is a window in which to view a database. The data window acts as the communication between the computer and the user. It is how the user stores, retrieves, and manipulates the data.

There is quite a bit of information in a data window. There is window positioning data, window size data, window color data, window field data, etc. You must supply all of this information to Data - Windows so that it can create the window.

Luckily Data - Windows makes entering this information logical and easy. You will not simply type the data in, but you will graphically design the window using either the keyboard or the mouse.

### Window Database Information:

After selecting the CREATE WINDOW option for the menu, you will be prompted for the filename in which to store the window's information. Simply enter the name and press <ENTER>.

After giving the window's name, you will be prompted for the database to use for this window, and the key field for this window. The database name should be the name of a previously created database. The key field name should be the name of a field in that database for which an index file has been defined. If no index is to be used, leave the index field name blank.

### Sizing and Placing the window:

The first bit of information required is the size and placement of the data window. For this there are two modes of input, keyboard and mouse.

If you are in the graphical mouse-driven mode, you will see a box on your screen. Your mouse pointer will point to the lower right corner of the box. NOTE: you may have to move your mouse before you can see the box, this means that the window is off of the screen and therefore invalid.

Sliding the mouse will change the position of the window on the screen. Move the mouse until the window is in the desired position. The size of the window can also be changed by pressing and holding the left mouse button and sliding the mouse. The window will change size as the mouse moves. When the window is in the proper position and is the right size, press either the <ENTER> key or the right mouse button. This will terminate the window placement prompt.

If you are using the text mode, the window can be repositioned and re sized with the arrow keys. The arrow keys by themselves position the window, and the SHIFTED arrow keys size the window. When the window is properly positioned and sized, pressing the <ENTER> key will terminate this mode.

### Coloring the Window:

Once the window has been sized and positioned, it will be displayed on the screen. You can now select the colors that you would like to use for the window. To select the colors you can press 'B' to change the background color, 'F' to change the foreground color, and 'O' to change the border color.

If you are using the graphical mouse-driven mode, you can use these keys, or you can click on either the background, the border, or the text name of the window displayed in the window. This will change the color of the background, the border, and the foreground respectively.

NOTE: changing the color cycles through the 16 color values (0 - 15). These colors may not look the same in different types of windows.

After coloring the window, press the <ENTER> key or the right mouse button, and you will terminate this mode and enter the define window fields mode.

### Defining Window Fields:

Defining the window fields is the heart of defining the data window. The window fields are the pieces of information from the database that you wish to display. A database may have more than one data window, so all of the database info does not have to be in the window.

There are three types of fields in a data window, GET fields, PUT fields, and TEXT fields. GET fields are fields in which the user may edit and enter data. PUT fields are expressions that the user can only view and not manipulate. TEXT fields are simply text that is placed in the window and does not change. TEXT fields are used for labeling the GET and PUT fields.

#### Defining GET fields

To define a GET field, position the cursor (using the arrow keys or the mouse) to the position in the window where you would like the field to be, and press 'G'. A window will pop up prompting you for the name of the field, and the field picture. The name of the field is the name that you gave the field when the database was defined. The picture is a string that defines how the data will be displayed in the window.

A picture contains 1 character for each character that will be displayed in the window. The picture character can be any character, but certain characters have special meanings. The 'X' character means that ANY character from the database

field can be displayed/edited in that position. The '9' character means that only numbers can be displayed/edited in that position. And finally the '!' character means that characters will be converted to uppercase before being displayed.

Here are some example pictures:

(999)999-9999 (phone number)  
999-99-9999 (soc. sec. number)  
!XXXXXXXXXXXX (a name (first letter capitalized))

### Defining PUT expressions

To define a PUT field, position the cursor (using the arrow keys or the mouse) to the correct position and press 'P'. A window will pop up prompting you for the expression to put and a picture to use when displaying the data.

Notice that for a PUT field you are allowed to enter an expression instead of just a field name. This expression is any regular expression that you would use in a programming language such as BASIC. In the expression you can use +, -, \*, /, ^, and (). for numeric values, and + for string values. The expression may contain field names, constants, or any of the listed operators. Note that a field name by itself is a valid expression.

Expression outputs are useful in situations such as invoice processing, where some formula may determine some important data. Take for example an invoice. On this invoice you have a field for quantity, and a field for unit cost. You could then have a put field that displays the expression "(quantity \* unitcost)". This PUT field could then be labeled "Total cost".

### Defining TEXT fields

To define a TEXT field, position the cursor to the place where you want the text to be displayed and press 'T'. You can then begin typing the text into the window. You will see the text as it is typed. When finished, press the <ENTER> key and the text will be placed in the window.

### Deleting Window Fields:

To delete fields in a window, position the cursor to the field that you would like to delete and press 'D'. The field will disappear from the screen.

### Completing the window:

When you are finished with the data window, press the <ENTER> key or the right mouse button. You will be given the simple prompt "[S]ave or [A]bort". If you type 'S' the window will be saved. If you type 'A' ALL DATA FOR THE WINDOW WILL BE LOST.

### Create Report

To create a report, select the CREATE REPORT option from the FILE menu. You will then be prompted for the information necessary to create the report. This includes the following:

Description	Prompt
Name in which to store the report data	(Report name)
Name of the database file for the report	(Database name)
Name of the key field for the report	(Key Field)
Width of the report	(Width)
Lines per record for the report	(Lines/rec)

The report name is simply the file name in which the report data will be stored.

### Report Database Information:

The database name is the name of the database that will be used for this report.

The key field is the name of the field in the database that will be used as a key for the report. The report will be sorted by its key. NOTE: THE KEY FIELD MUST BE A KEY FIELD IN THE DATABASE FILE.

### Report Width / Height:

The width of the report is the maximum width of the report to be printed. This is the width (in characters) of your printer.

### Report Lines / Record:



The lines per record is the number of lines on the report given for each record. This will usually be 1, but may be more if a lot of data must be placed on the report.

## Editing The Report Fields:

After the above data is entered, a screen will appear that allows you to edit the layout of the report. Editing a report is very similar to editing a window. The biggest difference is that only PUT fields are allowed on the report (obviously data cannot be retrieved from a report!). The only other difference is the information retrieved for each PUT field. The data is as follows:

Description	Prompt
Expression to be put in report	Expression
Flag to determine report breaks	Break Expr?
Flag to determine report totals	Total Expr?
Picture to use for report output	Picture
Header to place on report	Header

The expression is an expression like that defined for windows. See the section on expressions in windows for more information.

The break flag determines whether or not this expression defines a report break. A report break expression generates subtotals in the report whenever it changes. Break expressions are usually the key field for the report, but it is not required that they be.

The total flag determines whether or not totals will be generated for this expression. A total field has totals printed for every report break, and a grand total at the end of the report. Total fields must be numeric.

The picture is defined as it is for windows. See the section on window pictures for more information on pictures.

The header is text to be displayed at the top of the report.

## Create Labels

To create mailing labels, select the CREATE LABELS option from the FILE menu. After selecting this option, you will be prompted for several pieces of information. This information is defined as follows:

## Description

## Prompt

Name of file to store label data	Label Name
Name of database file to use for labels	Database Name
Name of key field in database to use	Key Field
Width of mailing label in characters	Width
Height of mailing label in lines	Height

The Label Name is the name of the file in which the label data will be stored. Whenever the label data needs to be used or edited, it will be referred to by this file.

## Label Database Information:

The Database Name is the name of the database file to use for this set of labels.

The Key Field is the field in the database file that will be used as the key. The labels will be sorted by key field. This field is optional, if omitted labels will be printed in the order they were entered. NOTE: THIS FIELD MUST BE A KEY FIELD IN THE DATABASE.

## Label Width / Height:

The Width is the character width of the mailing labels. The width of the label can be calculated by measuring the label's width (with a ruler) and multiplying the inches by the characters per inch of the printer. Most printers are 10 characters per inch. This means that a 3 1/2 inch wide label (3.5 inches) would be 35 characters wide.

The Height is the number of lines on the mailing labels. The height can be calculated by measuring a label with a ruler, and multiplying the inches by the lines per inch. Most printers are 6 lines per inch. This means that a 1 inch label would be 6 lines high.

## Editing the Mailing Labels:

After giving this information, the label layout can be edited. Editing the label is exactly like editing a window, except that only put fields are allowed. See the

section on editing windows for more information on editing labels.

## Open Window

To open a window, select the OPEN WINDOW option from the FILE menu. You will be prompted, with a filename window, for the name of the window to open. Once you have selected a window, it will be opened and you will be able to manipulate the database associated with that window.

Windows are the basic user interface to the database in Data Windows. Most manipulations of data must be accomplished through the window. This includes searching, browsing, editing, adding, and deleting records. All of these functions are performed from the RECORDS menu. No options on the RECORDS menu can be selected until a window has been opened.

NOTE: Opening a data window opens a new DEVICE window. This means that the <CLEAR> key can be used to switch from data window to data window. This also means that you must have enough device windows available to open a new one.

## Close Window

To close a data window, select the CLOSE WINDOW option from the FILE menu. This will close the data window and return the user to the window from which this data window was opened. If the old window is no longer available, you will be placed in some random window. The <CLEAR> key can then be used to go to the desired window.

NOTE: The CLOSE WINDOW option will only be available from open data windows.

## Print Report

To print a report, select the PRINT REPORT option from the FILE menu. You will be prompted for the name of the report file to use. After you enter the report file name, you will be prompted for the device name to use for output (printer path). This can be a printer name (like /p) or it can be a file name if you would like the report printed to a file. You will then be prompted for a test

condition.

The test condition is a condition that is tested before the printing of each record. If the condition is true for the record, the record will be printed. Otherwise the record will not be printed or processed. You are allowed to enter 3 lines of text for the test condition.

Valid conditions can contain any expression that is legal in PUT statements, plus the relational operators '<', '>', '<=', '>=', and '<>'. In addition, the logical operators 'AND' and 'OR' may be used to combine several relations. Here are some example conditions:

```
field1 + field2 < 1000
field1 <= field2
field1 + field2 < 1000 or field2 + field3 > 1000
(field1 + field2) * 2 < 1000 or (field2 + field3) * 2 > 1000
zipcode = "90125" or zipcode = "90124" or zipcode = "90123"
lastname > "BBBBBBBBBBBB" and lastname < "CCCCCCCCCCCC"
```

These are all valid conditions. Basically, the conditions are the same as that used in BASIC IF statements, except that field names are used for variables.

## Print Mailing Labels

To print mailing labels, select the PRINT LABELS option from the FILE menu. You will be prompted for the name of the labels file to use. After you enter the labels file name, you will be prompted for the device name to use for output (printer path). This can be either a printer port name (like /p) or it can be a file name. You will then be prompted for a test condition.

The test condition is a condition that is tested before the printing of each record. If the condition is true for the record, the record will be printed. Otherwise the record will not be printed or processed.

Valid conditions can contain any expression that is legal in PUT statements, plus the relational operators '<', '>', '<=', '>=', and '<>'. In addition, the logical operators 'AND' and 'OR' may be used to combine several relations. Here are some example conditions:

field1 + field2 < 1000  
field1 <= field2  
field1 + field2 < 1000 or field2 + field3 > 1000  
(field1 + field2) \* 2 < 1000 or (field2 + field3) \* 2 > 1000  
zipcode = "90125" or zipcode = "90124" or zipcode = "90123"  
lastname > "BBBBBBBBBBB" and lastname < "CCCCCCCCCCC"

These are all valid conditions. Basically, the conditions are the same as that used in BASIC IF statements, except that field names are used for variables.

## Quit

To quit data windows select the QUIT option from the FILE menu.

## The EDIT Menu

The EDIT menu is where the editing operations are located. All editing operations EXCEPT the editing of data records. The editing of data records is on the RECORDS menu. The EDIT menu does allow the cutting, copying, and pasting of records. It also allows the editing of database formats, window formats, report formats, and mailing label formats.

### Undo

The UNDO operation is not implemented in the current version of Data Windows. The option is left on the menu for historical significance as well as for future versions of the software.

### Cut

To CUT a record from a database, select the CUT option from the EDIT menu. When a record is CUT a copy of the record is stored in a special CUT file, and the record is deleted from the database. The record can then be PASTED into another database, or the same database.

### Copy

To COPY a record, select the COPY option from the EDIT menu. When a record is COPIED, a copy of the record is stored in the CUT file (as with CUT), but the record is not deleted from the database.

### Paste

To PASTE a record, select the PASTE option from the EDIT menu. The PASTE option adds a new record, and stores data from the CUT file into that record. This can be used to duplicate records (by using COPY then PASTE within the same database), to move records from one database to another (by using CUT in one database, and PASTE in another), or to copy records from one database to another (by using COPY in one database and PASTE in another).

NOTE: The CUT, COPY and PASTE options move records into a special file called "/dd/sys/clipboard.txt". This is a non-indexed file that has the database format of the last CUT/COPY operation. When a PASTE operation is performed, only data fields with the same name are PASTED.

NOTE: The CUT, COPY and PASTE options are only valid in an open data window. If no data window is open on the current screen, these options will be invalid.

## Edit Database

To edit the format of a database, select the EDIT DATABASE option from the EDIT menu. After selecting this option you will be prompted for the name of the database file to edit. After selecting the file name you will be able to edit the database.

Editing the database format is exactly like creating the database format, except that the old field definitions, etc. are all still valid. See the section on creating databases for more information on editing them.

## Edit Window

To edit the format of a data window, select the EDIT WINDOW option from the EDIT menu. After selecting this option you will be prompted for the name of the window file to edit. Once you select the window file, editing of the window can begin.

Editing a data window is exactly like creating a data window except all of the window size, coloring, field placements, etc. are already in place. You will be allowed to change these. For more information on editing the data window, see the section on creating a data window.

## Edit Report

To edit the format of a report, select the EDIT REPORT option from the EDIT menu. As with editing databases and windows, you will be prompted for the file name then allowed to edit.

As with databases and windows, editing a report is exactly like creating one. See the section on creating reports for more information.

## Edit Labels

The procedure for editing mailing labels follows the same course as databases, windows, and reports. Select the EDIT LABELS option from the FILE menu, give a file name, and edit the labels as you did when you created them. See the section on creating mailing labels for more information.

## The RECORDS Menu

The RECORDS menu is where all database record handling options are. Note that these options will only be available for open data windows. If no data window is open on the current screen, these options will not be available.

The RECORDS menu contains options for FINDing records, browsing records (NEXT, PREVIOUS, FIRST, LAST), EDITing records, DELeting records, and ADDing records.

### Find Record

To find a record in an open data window, select the FIND RECORD option from the RECORDS menu. After selecting this option, the field defined as the KEY field for this data window will be highlighted. You can then enter the key field data to be used for the search. After pressing <ENTER>, the database will be searched for the given key. If no record is found that matches the key, an error message will appear. If more than one matching record is found, the first will be displayed.

NOTE: The entire key does not have to be entered. A partial key can be used for the search. The first record meeting the partial key will be retrieved. (EXAMPLE: "AL" will match "ALPHA").

NOTE: A KEY field must be defined for the window if the FIND RECORD option is to be used.

### Browse Records

Browsing records refers to the forward and backward paging through records. There are four (4) options on the RECORDS menu to allow browsing, FIRST RECORD, LAST RECORD, PREV RECORD, and NEXT RECORD. Each of these options can also be accomplished with the arrow keys. The UP/DOWN arrow keys will move to previous/next records respectively. And the shifted UP/DOWN arrow keys will move to the first/last records respectively.

The FIRST RECORD option will bring you to the first record in the database. Note that if there is a key defined for this data window, the first record will be the sorted first record for the key.

The LAST RECORD option will bring you to the last record in the database. As with the FIRST option, the data window's key field (if any) will be used for

sorting.

The NEXT RECORD option will bring you to the next record in the database. As with all data window operations, the records are sorted by the data window's key field, therefore the next record will be the next record in key sequence order.

The PREV RECORD option will (of course) bring you to the previous record (sorted by key order) in the database.

### Edit Record

To edit the current record, select the EDIT RECORD option from the RECORDS menu. When you select this option, all of the fields in the data window will be highlighted, and you will be able to edit them.

The left and right arrow keys can be used to maneuver within each field. The up and down arrow keys can be used to maneuver from field to field. After the last field is completed, record editing will be terminated and record viewing mode will become active again.

### Delete Record

To delete the current record, select the DELETE RECORD option from the RECORDS menu. When you select this option, the currently displayed record will be deleted. You will then be moved to the next record in the database for viewing.

### Add Records

To add records to the database, select the ADD RECORDS option from the RECORDS menu. When you select this option a blank record will be appended to the database and you will be placed in edit mode so that you can edit the new record.

After entering the data for the new record, a simple prompt will ask if you want to enter more records. Pressing the 'Y' key will allow you to add another record.

## The DATABASE Menu

The database menu provides database wide functions. Note that the options on this menu are only available when no data window is open on the current screen. If the screen you are on has an open data window, use the <CLEAR> key until the screen without any data windows appears.

The DATABASE menu has options for MERGING, PURGING, PACKING, IMPORTING, and EXPORTING databases.

### Merge Database

To merge two databases together, select the MERGE DATABASE option from the DATABASE menu. After selecting this option you will be prompted to enter the destination database. This will be the database into which the data will be merged. You will then be prompted for the name of the source database file. This is the file from which the data will come. After this file is selected, all data in the source database will be merged into the destination database. Note that nothing will be changed in the source database, only the destination.

The merging of two databases is the copying of all records from one database into another. This copying of records only applies to fields that have the same name. Any fields in the source database that don't match fields in the destination database will not be merged into the destination database. (where would it put them!).

### Purge Database

To purge a database, select the PURGE DATABASE option from the DATABASE menu. After selecting this option you will be prompted for the name of the database to purge. When the database name has been selected, a simple prompt will ask "Are You Sure?". If you are sure you want to purge the database, press the 'Y' key.

Purging a database will DELETE ALL RECORDS in that database. You should only perform this option if you are sure that you want to lose all data in the database!

### Pack Database

To pack a database, select the PACK DATABASE option from the DATABASE

menu. After selecting the option you will be prompted for the name of the database to pack. After selecting the database, the pack operation will begin.

Packing a database is a process that removes the data being used by deleted records and re-generates all indexes. For largedatabases this operation can be lengthy. It can also be disk consuming. For this reason you should make sure that there is enough disk space for a complete duplicate of the database file to pack. If there is not enough space, much time could be wasted only to get an ERROR 249 before finishing. This will not lose any data, but can be a real pain!!

The PACK operation is simply and EXPORT DATA (described below), followed by a PURGE DATABASE (described above), followed by an IMPORT DATA (described below). For this reason, data loss is easily preventable, even in the worst circumstances.

If a lockup or power fail occurs during a PACK operation, the system could have been in either of several states, exporting, purging, or importing. If the problem occurred during exporting, all data will still be in the database, and an incomplete temporary file will exist. Simply delete the temporary file and re-perform the pack. If the problem occurred during purging of the data, the database may still have records, but may be erroneous. To fix this simply purge the database and import the temporary file. If the problem occurred during importing of data, the database will be incomplete and may be erroneous. To fix this simply purge the database and import the temporary file.

### Import Data

To import data from a text file, select the IMPORT DATA option from the DATABASE menu. You will be prompted for the destination database file name. This will be the database into which the data will be imported. You will then be prompted for the name of the source data file. After selecting this file, the data will be imported.

Importing data allows you to retrieve data from other database systems as well as other programs. The import file format is fairly simple and should be easy to generate with any other database program.

The import file should be as follows:

"data for field1", "data for field2", "data for field3" ...

The data is put into the database in the order that the fields are listed when editing or creating the database. This is the only instance in which the ordering of fields in the database is important.

The quotes in the file are not necessary, as long as there are not commas in the text data itself. It is, however, recommended that the quotes be used if possible, as this will alleviate any possible problems.

If there is an extra comma that is not in quotes in the file it could throw off the entire import. It is for this reason that the quotes should be used whenever possible.

To get data from other database systems, the report generator can be used to create the import file. Simply define the report as looking like the import file and have it output to a file. If the database system does not provide a report facility, or does not allow you to make reports in the necessary format, you may have to write a program in BASIC09 or some other language to format the data properly.

## Export Data

To export data, select the EXPORT DATA option from the DATABASE menu. You will then be prompted for the name of the database to export data from, and the name of the file to store the export in. The data will then be exported in the following format:

"data for field1", "data for field2", ...

Note that this is the format that is used for importing data (makes sense, right?).

This appendix is designed to help users that would like to use this product, but are having trouble with getting their OS9 up and running properly.

OS9 is quite intricate and can tend to be unfriendly to newcomers. The biggest reason for this is an improperly set up OS9 system. If OS9 is not set up properly, it can be slow, tedious, and aggravating.

The main configuration problems with OS9 have to do with devices. OS9 supports many different devices, and has a standard way of dealing with these different devices. Unfortunately this is not well documented by Microware (so what else is new??).

All OS9 devices are described by device descriptors. In fact, whenever you refer to an OS9 device you refer to its descriptor. For example, when you say "dir /d0", the "/d0" refers to the "d0" device descriptor in memory. Device descriptors are stored in the OS9Boot file and are loaded into memory at boot time.

If you want to change something about a device, you must only change the device descriptor for that device. This means that to make a single sided disk drive double sided, you must make the descriptor double sided. Of course the real hardware must also be double sided.

To change a descriptor, we can use the modpatch program given to us with OS9. In modpatch you can say link to a descriptor, change this to that, verify (fix the CRC), and your done. But this only changes things in memory. If you want the changes to be permanent, you must save them to disk by putting them in your boot file.

Usually this would mean you would have to make a new boot file, but we have included a special program called "ReCobbler" to help you. If you make changes using modpatch, you can save those changes to your original file with ReCobbler. ReCobbler will be in the CMDS directory of the Data - Windows disk #2.

Now we have to determine, what is the THIS and THAT of the change. Believe it or not, all the THISES and THATs that you can change are listed in your OS9 manual, but they are hidden. We will give them to you here so that you won't have to work for it. The following patches can be used to make changes to your devices.

command	Description
modpatch	Run modpatch

1 /d0      Link to d0 (use whatever drive you want to change)

c 14 00 03 Change byte 14 from 0 to 3. Byte 14 is the disk drive step rate. This can be crucial as a fast step rate will make disk access about 100 times faster. Most drives will allow a 6ms step rate. For this step rate the value is 3. The value of 0 is used for 65ms step rates. This line therefore changes the step rate from 65ms (the default) to 6ms (FAST). If you get the warning message "byte does not match" ignore it. Just make sure the second value is a 3.

c 19 01 02 Change byte 19 from 1 to 2. Byte 19 is the number of sides byte. This line would change a single sided drive to double sided.

c 18 23 28 Change byte 18 from 23 to 28. Byte 18 is the number of tracks on the disk. Note that numbers are all in HEX so \$23 is 35 and \$28 is 40. This line therefore changes a 35 track disk drive into a 40 track disk drive.

vv verify (fixes the CRC) of a module. OS9 uses CRCs to make sure that the system does not get corrupted. This helps to prevent lockups, but can be annoying when trying to change a device descriptor.

NOTE: You can find more of these change values in your OS9 Technical Reference Manual page 5-9 (RBF-type device descriptor modules).

Once these changes have been made for all necessary floppy drives, you can use the "ReCobbler" command to save them to your boot file. To do this:

```
Insert your Data - Windows disk #2 in drive 0
type:load /d0/cmds/recobbler
Insert your boot disk in drive 0
type:ReCobbler /d0
```

This will save all changes to your boot disk.

NOTE: All changes made with Xmode, Dmode, and ModPatch can be saved to your boot file in this manner. (Dmode is a command listed in Rainbow somewhere and on various BBSs).

Another common problem with OS9 users is their neglect to install a /dd device. /dd is a special device descriptor that is really a duplicate of another device, but with a different name. This allows software to access a default device for things like the SYS directory.

Data - Windows uses the /dd device...in fact it uses the /dd/SYS directory. It uses this for the CUT file. If you do not have a /dd device, you must get one. You can do this by either modifying a current device to make it /dd, or by adding a new device for /dd. To add a new device, use CONFIG.

If you want to change an existing device to be /dd, that can be done with modpatch.

OS9 stores the name of each device descriptor inside of the device descriptor. Because of this we can change the name if we want.

In the floppy drive descriptors distributed with OS9, the name of the descriptor is stored in bytes 21 and 22. If we want to change the name, we can therefore change these bytes. The procedure is:

```
modpatch
1 dx                   (dx is the drive to link to)
c 22 b0 c4 (change "0" to "D")
vv verify module
```

This patch will change the d0 device descriptor name to dd. The "b0" is the HEX value of an ASCII "0" + 128. "c4" is the HEX value of an ASCII "D" plus 128. The 128 is because it's the last byte in the name. The last characters of names in OS9 are given by adding 128 to the character value (setting the high bit).

I hope these notes will help to make the OS9 configuration easier.



## Appendix B:DB9 - 90 File Specifications

### DB9 - 90 File Format Specifications Ver. 1.0

#### Background

DB9 - 90 is a database file format specification for OS9. The DB stands for "DataBase", the 9 is for OS9, and the 90 is for the year it was created (1990).

This file specification has been developed by Mr. Keith J. Alphonso of Alpha Software Technologies. These file format specifications are hereby donated by him to the public domain for public use. Anyone may write software that uses and accesses files of this format.

The purpose of this file specification is an attempt at creating a standard data storage format that everyone can use. There are several requirements for such a data storage specification. The requirements taken into account are as follows:

This file specification is to be completely generic in order that it can be used for storing any data. This is indeed the idea behind a generic DataBase format.

The file specification should be written in such a way as to be expandable without alienating previous revisions in the specification. This will facilitate backward compatibility (older software can use newer files).

The file specification should provide a means of quickly accessing data in random, sequential, and sorted order quickly and easily.

The file specification should provide a means of quickly locating desired records, without resorting to sequential search.

All of these factors are accounted for in the DB9 - 90 format.

#### The Specification

The DB9 - 90 specification provides for the storage of data as a set of files instead of a single file. One main file is to be used for the storing of the actual data. This file will have an extension of ".dba". Other separate index files will be used to store index data into the main file. These files will have an extension of ".ndx".

#### DB9 - 90 Main file specification (".dba")

The main data file consists of a variable length header record followed by sequential fixed length data records. The header record specifies the length and format of the fixed length data records.

#### Header Record:

The header record is variable length. This will facilitate the easy addition of information into the header record. The header record contains several fixed and several variable length fields as follows:

4 byte file offset to the first data record.

4 byte file offset to the database field definition section of the header.

2 byte number of fields in each record.

6 byte application signature. This application signature allows an application to determine the format of the application specific section that follows.

variable length application specific data field.

48 byte field definition for each field.

#### C Example:

```
struct DB9_header
{
    long first;
    long fields;
    unsigned num_flds;
    char app_sig[6];
}
```

#### ASM Example:

```
DB9_hed equ .
first    rmb 4
fields   rmb 4
numflds  rmb 2
         appsig      rmb 6
appdep   rmb ?
```

char app\_dep[?];

#### Application Specific Data Field:

Immediately following the 6 character application signature is any application specific data that must be stored in the file. The field is variable length and the data stored is wildcard data. All other information in the file is pointed to by the beginning of the header record (in first and fields). What is placed here is to be determined by the application signature.

NOTE: I am reserving for myself 10 applications signatures as: "ALPHA0" ... "ALPHA9". All other applications must battle it out for their own personal signatures. Any signature conflicts will have to be resolved by the application developers.

#### Header Record Field Definition:

The field definition offset points a set of field definitions. Each field definition is 48 bytes long and is specified as follows:

- 10 byte field name
- 32 byte field index file (zero filled if no index file)
- 2 byte field type
- 2 byte field length
- 2 byte reserved

The 10 byte field name is a name given to the field in order that the data in that field can be accessed by any software. This allows for portable data files across various products.

The 32 byte field index file name specifies the name of the file in which the index for this field is stored. Each field can have an associated index file, but it is not required to. An index file is only necessary if the programmer wants to do searching or sorting on that field. If there is no index file, this field should be zero (NULL) filled.

The 2 byte field type provides for 65536 different field types. Currently there are only 2 specified field types, numeric and character. A character field is specified as 1. A numeric field is specified as 2. Although a field is specified as numeric, the data is to be stored in ASCII format. This will solve any problems of numeric representation across languages. (Example: 5.4 is stored as "5.4")

The 2 byte field length provides for fields with a length up to 65535 bytes long. Note that for numeric fields this INCLUDES the decimal point as well as all digits following the decimal point. The field can be considered the length of the ASCII representation of the data (be it numeric or character).

The final 2 byte field is reserved for future use.

#### C Example:

```
struct DB9_field {
    char fld_nam[10];
    char fld_ndx[32];
    unsigned fld_type;
    unsigned fld_len;
    char fld_res[2];
};
```

#### ASM Example

```
DB9_fld equ .
fld_nam rmb 10
fld_ndx rmb 32
fld_type rmb 2
fld_len rmb 2
fld_res rmb 2
```

#### Expandability note:

Note that this header specification may be expanded as the first data record is pointed to by the header. This allows any amount of information to be stored from the end of the field specifications to the first data record. This can include extra field/database information.

#### Data Records:

The DB9 data records are sequentially stored fixed length records. The length of each record is calculated by adding the length of each field in the record. A 1 byte deletion flag field is then added giving the total length of each record. The records are then stored one after another in the order they are received. All searching and sorting of records will be done through associated index files.

The data for each field is stored in the position given by the addition of all prior fields listed in the database header. The data is stored as specified length character strings. The length is specified in the field definition.

The deletion flag is the last byte in the record. A 0 indicates a NON-DELETED record. Any value besides 0 makes the record deleted. A new record can then be placed in that slot.

#### Example:

Suppose a record contains a 10 character name and 3 5 character numbers. That record would look as follows:

```
NAME      00.3001.3010.25(NULL)
|         |   |   |   |
0         10  15  20  25(deletion flag)
```

The three number are .3, 1.3, and 10.25. Note that the numbers are zero filled, this is NOT required. Neither is it required that the decimal be in the same position for each number (although it is good practice). A VAL command should be used to obtain the numeric value of the data. In C this would be "atof".

## DB9 - 90 Index File Specification (".ndx")

The indexing method used for DB9 - 90 index files will be Btrees. The reasons for this include:

The wide acceptance of Btrees as an indexing scheme.

The ability retrieve records in key order using a Btree.

The ability to quickly search for keys in a Btree.

The wide amount of documentation available on Btrees (look at ANY data structures book).

### Btree description:

A Btree is NOT a binary tree. A Btree is a set of nodes, each of which has some number of keys (N), and N+1 pointers. Each key has a left and right pointer. The left pointer of key 2 is the right pointer of key 1. This gives the N+1 pointers.

Example:

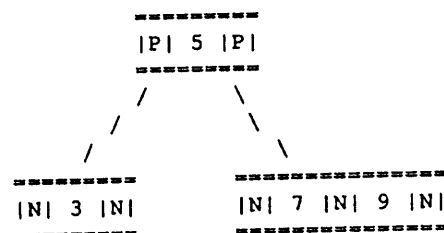
N = 5

```
-----  
|P| key |P| key |P| key |P| key |P| key |P|  
-----
```

Note that there are 6 P's and 5 keys.

All keys in a node are in sorted order. Each pointer points to a node in which all keys in that node are greater than its left key and less than its right key.

Example:



Note that not all entries in a node must be filled. Also note that the P's are replaced by N's in the leaf nodes. This is to show that these do not point to any further nodes, they are simply NIL pointers.

#### DB9 - 90 Btree file specification:

The DB9 - 90 Btree file specification defines a variable length index file header followed by a set of fixed length nodes.

The index header is defined as follows:

- 4 byte file offset to the root node of the btree
- 2 byte duplication flag
- 250 byte reserved space
- variable length application specific data

NOTE: although the index file header record is variable length, it is recommended that a multiple of 256 bytes be used for efficiency (file accesses will never cross sector boundaries). The application signature given in the main file should be used to determine the format of the application specific data section.

The btree nodes are stored after the btree file header, but are not necessarily stored in any specific order. In fact, the way Btrees are handled mandates that the nodes not be accessed in order, but by traversing them through the pointers. This is why a pointer is provided in the file header to the root node.

The index file nodes are defined as follows:

- 6 32 byte key entries (keys are limited to 32 bytes)
- 7 4 byte node pointers
- 6 4 byte main file pointers
- 1 4 byte previous node pointer (for reverse tree traversal)
- 6 1 byte node deletion flags
- 2 bytes of reserved space

NOTE that this makes for a total of 256 bytes per node.

The 6 32 byte key entries are character string keys with a maximum length of 32 bytes each. If the key is shorter than 32 bytes it should be NULL terminated (C style). If the key is exactly 32 bytes, no terminator is required. The 32 byte maximum key size does not mean that the key field must be 32 bytes or less, only that the only the first 32 bytes are significant. If the key field is larger than 32 bytes, it must be truncated to 32 bytes when stored into the index. This should be reasonable. If the node entry is empty, the first character in the key should be a 0.

The 7 4 byte node pointers are the left and right key pointers. pointer 2 is the left pointer for key 2 and the right pointer for key 1. For leaf nodes and for

empty node entries, these pointers should be set to NIL (-1 or \$FFFFFFFF).

The 6 4 byte main file pointers associate the index key to the main file data record. These are 4 byte file offsets into the main database file. They point to the record associated with each key.

The 4 byte previous node pointer is a file offset to the previous Btree node. If this node is the root node, this pointer should be NIL (-1). This pointer allows reverse tree traversal.

The 6 1 byte node deletion flags deem a node entry as deleted. The node entry must, however, still be kept as the left and right pointers still point to other nodes. The key must also be kept for comparison.

There are 2 exceptions to this rule. The first is when a node is a leaf node. In this case the pointers are NIL and the node entry can simply be deleted.

The second exception is when another key is to be added that fits (by order) into the space of a deleted node. The new key can then be copied directly into the node entry.

For a more complete understanding of Btrees, consult any data structures textbook. There should be a chapter or so on this widely used data structure. The main difference between this structure and textbook Btree structures is the deletion flag. This flag was added, obviously, to facilitate the quick and easy deletion of keys.

### Closing Remarks

The data file formats documented here are hereby donated to the public domain. Anyone desiring to use these file formats is completely free to do so, and is encouraged to do so. No royalty fee, distribution fee, or any other monetary concession can be imposed for the use of these file formats.

All that would be asked is that applications using this file format document the fact that they do use the format. This is to promote the format as a standard, and to allow persons using the product to know that they can integrate their data with other products.

As creator and designer of the file specification, Alpha Software Technologies, will take responsibility for the management of the specification. This is to say that any desired revisions to the specification should be routed to me at:

DB9 Data Specs.  
c/o Alpha Software Technologies  
210 Bluefield Dr.  
Slidell, LA. 70458

I can also be reached on Delphi as "ALPHASOFT".

I will designate, and document, any new revisions to this specification. The current version is 1.0.

It is intended that this database format specification become the OS9 standard for database management. This is the purpose of making the standard completely documented and public domain. This is believed to be the reason for the unbelievable success of the Xmodem file transfer protocol.

If this specification is accepted as the standard, OS9 may become the first operating system to have a well standardized database format. This could greatly help promote the growth of OS9 and OSK.

## Alpha Software Technologies

As designer of this format, Alpha Software Technologies will, of course, be producing software for the format. These products will be officially announced later, but a description of the planned products will be listed here.

**Data – Windows:**Data – Windows will be a full featured generic database system. The Data–Windows system will allow a user to create, modify, and use DB9 – 90 databases. The system will come complete with a report generator, label maker, and database facilities provided for by the DB9 – 90 specification.

**Data – App:**Data – App will be an add on system to Data –Windows that will allow non-programmers to create complete applications from Data – Windows window definitions.

**Data – Developer:**Data – Developer will be a complete toolbox consisting of BASIC09, RMA, and C routines for performing all facilities provided in the DB9 – 90 specification. This will include creating databases with multiple key fields, automatic sorting and searching, and much more.

**Data – Merger:**Data – Merger will be a mail merge system that will allow the user to use any DB9 – 90 database for mail merging. Documents can be created on any word processor, then run through the data – merger and printed.

**Data – Forms:**Data – Forms will be a form writing system that will use DB9 – 90 databases to create and print user definable forms.

These are just a few of the possible applications of the DB9 – 90 database specification.

The Data – Windows product is scheduled to be release in September, 1990.

Alpha Software Technologies  
210 Bluefield Dr.  
Slidell, LA. 70458

## Copyright Notice

Data - Windows is Copyright (c) 1990, by Keith J. Alphonso and is licensed to Alpha Software Technologies. The unauthorized duplication of this software or the accompanying documentation is **PUNISHABLE UNDER FEDERAL LAW**. Anyone in violation of this copyright agreement will be prosecuted to the full extent of the law.

The only authorized copies of this software are those made for backup purposes only. Backup copies may only be retained by the owner of the software, and cannot be distributed in any fashion.

**WARNING:** Your serial number is attached to the Data - Windows executable files. This serial number cannot be changed, and can be linked directly to you in the event that an unauthorized copy is found. **DO NOT DISTRIBUTE THIS SOFTWARE!**

## Disclaimer

This software is designed to be reliable, easy to use, and safe; however, the loss of data is always a possibility. The maker of this software, namely Alpha Software Technologies and Keith J. Alphonso, are not responsible for the loss of data caused directly, or indirectly by the use of this software.

© 1991 Keith Alphonso  
Licensed to and distributed by

**COCO PRO!**

All rights reserved