

Chapter 11

Memory

This section describes the memory structure of the HP 49G. It describes how to create backup objects of data that you want to save, and how to use libraries to add functionality to the calculator.

How memory is structured

The HP 49G contains a total of 2.5 Mb of memory. Of this memory:

- 1 Mb is used to hold the operating system.
- 1.5 Mb is used for performing the operations that you specify, and for storing data that you want to keep. The HP 49G needs a minimum of approximately 200 bytes free in order to perform operations.

The memory of the HP49G is divided into 4 areas:

- The Home directory
- Port 0, labelled IRAM in File Manager

The Home directory and port 0 share the same area of memory. That is, the more data you store in the HOME directory, the less data you can store in port 0 and vice versa. The total size of objects in these two areas cannot exceed 241 Kb.

- Port 1, labelled ERAM in File Manager
Port 1 can contain up to 255KB of data.
- Port 2, labelled FLASH in File Manager
Port 2 can contain up to 1085KB of data.

The Home directory, port 0 and port 1 are part of RAM. Leaving the calculator without batteries can cause RAM to lose its data. When changing batteries, ensure that you do not leave the calculator without batteries for longer than two minutes.

The data in port 2 is stored in the Flash ROM. This data does not depend on batteries. It is preserved even if you leave the calculator without batteries for a long time.

Unlike the HOME directory, port memory cannot be subdivided into directories. A port can only contain two types of objects:

- backup objects
- library objects.

Accessing port contents

In order to access the contents of the variables stored in the ports you can use File Manager. For details on how to use File Manager, refer to the *User's Guide*.

Alternatively, you can access the contents of libraries and ports, by pressing   A function key menu of the attached libraries and the available ports is displayed.

- To view the contents of a port, press the function key that corresponds to the port. The contents are displayed as a function key menu.
- To access the functions in an attached library, press the function key that corresponds to the library. The functions it contains are displayed as a function key menu. To run a function, press the function key that corresponds to the function.

Backup objects

You generally use backup objects to save data that you want to keep for a long time. Backup objects are special copies of objects from your HOME directory. Note the following points regarding backup objects:

- Backup objects can only exist in port memory.
- You cannot modify the contents of a backup object.
- You can store either a single calculator object as a backup object or an entire directory. You cannot create a backup object for selected objects in a directory.

The HP 49G monitors the integrity of backup objects. When you create a backup object in a port, the HP 49G calculates a CRC value (cyclic redundancy check) based on the binary data that the object contains. A CRC value is a checksum calculated from the data. This value is stored with the object.

When you restore a backup object, the calculator re-calculates the CRC value and compares it to the original value. If the CRC values are not the same, the calculator warns you that the data is corrupt.

You can use File Manager to copy and delete backup objects in a similar fashion to normal calculator objects. In addition, there are specific commands for manipulating backup objects.

Backing up and restoring HOME

You can back up and restore the contents of the entire HOME directory in a backup object. This includes all variables, and any key assignments and alarms that you have created.

Backing up

To backup the HOME directory in RPN mode, enter the following:

`:Port_Number: Backup_Name ARCHIVE`

To backup the HOME directory in algebraic mode, enter the following:

`ARCHIVE(:Port_Number: Backup_Name)`

Restoring

To restore the HOME directory in RPN mode, enter the following:

`:Port_Number: Backup_Name RESTORE`

For example, to backup the home directory as MYHOME in port 2, use the following command in algebraic mode:

`ARCHIVE(:2: MYHOME)`

To restore the HOME directory in algebraic mode, enter the following:

`RESTORE(:Port_Number: Backup_Name)`

To restore the HOME directory backup from the previous example, use the following command:

`RESTORE(:2:MYHOME)`



When you restore a HOME directory backup, the backup directory overwrites the current HOME directory and its contents. Any data not backed up is lost.

After you restore a HOME directory, the calculator restarts. The contents of history or the stack is lost.

Storing and deleting backup objects

There are three ways to create a backup object:

- Use File Manager to copy the object to a port. With this method, the backup object has the same name as the original object.
- Use the STO command to copy the object to a port, and assign it a name. See *Command Reference Part D* for details of the STO command.
- Use the ARCHIVE command to create a backup of the HOME directory, and assign a name to the backup object. See *Command Reference, Part B* for details of the ARCHIVE command.

There are two ways to delete a backup object from a port:

- Use File Manager to delete the object as you would a normal calculator object.
- Use the PURGE command
 - In RPN mode, use the following syntax:
 :Port_Number: Backup_Name PURGE
 - In algebraic mode, use the following syntax:
PURGE(:Port_Number: Backup_Name)
For example, to delete a backup object in port 1 named
“MYBACKUP”:
PURGE(:1: MYBACKUP)

You can restore a backup object by using File Manager to copy it to the Home directory. When you restore a backup object, the HP 49G checks the object's integrity by calculating the CRC value and comparing it to the one stored with the backup object. An error message appears if the data has been corrupted.

Using data in backup objects

You cannot modify the contents of backup objects. You can, however, use the contents in calculator operations. For example, if your backup object contains a program, you can run the program. If your backup object contains a value assigned to a variable, you can use the variable in calculator operations.

To view a value saved in a backup object, or to run a program stored in a backup object, you can use File Manager as you would with a normal object.

You can also execute an object from the command line as follows:

- In RPN mode:
 - To evaluate a backup object, enter:
 :Port_Number: Backup_Name variable_name EVAL
 - To recall a backup object to the command line, enter:
 :Port_Number: Backup_Name variable_name RCL
- In algebraic mode:
 - To evaluate a backup object, enter:
 EVAL(:Port_Number: Backup_Name variable_name)
 - To recall a backup object to the command line, enter:
 RCL(:Port_Number: Backup_Name variable_name)

You can also use   as described in “Accessing port contents” on page 11-2.

If you do not remember in which port a backup object is stored, you can use the & symbol in place of the port number. This causes the HP49G to search all the ports, and then the HOME directory.

You can also refer to a port by tags:

“E” and “ERAM” refer to port 1

“F” and “FROM” refer to port 2

Library objects

A library is a collection of objects that extend the calculator's functionality. You can execute objects in a library, but you can neither view, nor edit them. You can obtain libraries from various web sites.

Installing and attaching a library

To install a library, perform the following:

1. Copy the library to your HOME directory.

- from a PC, use the Connectivity Kit.
- From another calculator, use the calculator-to-calculator connection cable.

2. Install the library in a port.

To install the library in a port, use the following commands. In the commands, *n* is the number of the port in which you want to store your library, and `Library_variable` is the name of the variable that contains the library. This is the variable that you created when you copied the library to HOME.

- In RPN mode, with `Library_variable` on stack level 1:
`port_number STO`
- In algebraic mode:
`STO(Library_variable, port_number)`

3. Attach the library. A library generally contains a special routine to attach it. This routine runs when the calculator re-boots. To reboot, press simultaneously **ON** and **F3**, then release them.

You can use the `ATTACH` command to attach a library without re-booting. See *Command Reference Part B* for details on the `ATTACH` command.

Some libraries need to be stored in port 0. Check the documentation for a library for more information.

A library has a library number. The calculator uses this number to refer to the library. In File Manager, the library's number appears as "Lxxxx", where *xxxx* is the library number.

Deleting a library

To delete a library from a port, use the following commands. In the commands, `port_number` is the number of the port you stored your object in and `lib_number` is the library number of the library you want to purge.

- In RPN mode:

```
:port_number: lib_number PURGE
```

- In algebraic mode:

```
PURGE( :port_number: lib_number )
```

How the HP 49G manages memory

The following section explains how the calculator manages memory in the various ports.

Port 0

Port 0 and the HOME directory share the same memory space. This means that objects stored in port 0 decrease the amount of user memory available, and objects stored in user memory decrease the amount of memory available in port 0.

Port 1

Port 1 contains 255 Kb of storage space. Because of the way port 1 memory is structured, you cannot store objects larger than 128 Kb. This is because port 1 memory is in fact 2 discrete areas of 128 Kb. Objects must be stored in either one of the two areas.

Because of this, when storing large objects it may appear that you have enough memory in port 1 to store your object, but the calculator will not allow you to store the object if it is bigger than the free space in either area.

Port 2

Port 2 is part of the Flash ROM. As with port 1, it is not possible to store objects larger than 128KB. Flash ROM is organized as 8 areas of 128KB each and one area of 64KB.

The method that the system uses to manage Flash ROM can sometimes affect operations in port 2. When it erases data, the system must erase 128Kb areas at a time. It cannot erase single objects. When you delete an object from flash ROM, the object is simply flagged as deleted. It still occupies memory space.

Occasionally when you want to store an object in Flash ROM, the system needs to remove files flagged as deleted. It does this by copying all the non-flagged object to RAM, erasing the entire Flash ROM, and then moving the non-tagged objects back to flash ROM.

Depending on how much data you have stored in flash ROM, up to 128KB of RAM might be needed for this operation. If there is not enough RAM, an error occurs, and the following message is displayed:

Ins. Mem. Need *x* bytes.

In this event, to store your object, you need to free up at least *x* bytes of RAM.

To list a port's contents, and find free memory

You can use the PVARS command to list the contents of a port. The PVARS command returns a list containing the port number and name of each object in the port, and the amount of free memory in the port.