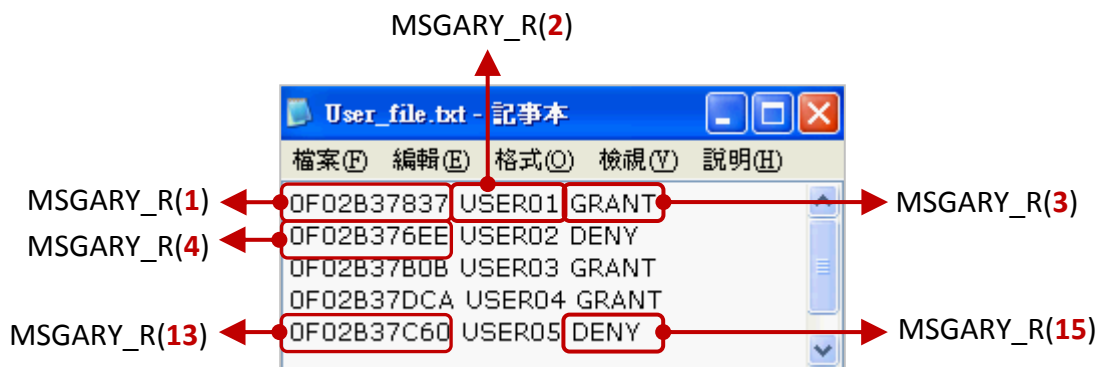


| | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|----------------|
| Classification | ISaGRAF English FAQ-172 | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page 1 / 10 |

How to use ISaGRAF functions to read multiple strings from a file?

This FAQ-172 provides an ISaGRAF demo program to show the way to use the “MSGARY_R(addr)” and the “MSGARY_W(addr,msg)” functions to read multiple strings from a file. For instance, the “User_file.txt” file has five rows and three columns for a total of fifteen strings. Then, user can use MSGARY_R(1) in the ISaGRAF program to get the first string “0F02B37837”, using MSGARY_R(2) to get the 2nd string “USER01”, using MSGARY_R(3) to get the 3rd string “GRANT” and using MSGARY_R(15) to get the 15th string “DENY”, and so on.



1.1. Download / Upgrade the ISaGRAF Driver

The following ISaGRAF WinCE PACs support the MSGARY_R(addr) and the MSGARY_W(addr,msg) functions, and the new ISaGRAF driver versions are shown as below.

| ISaGRAF WinCE PAC | ISaGRAF Driver Version |
|-------------------|------------------------|
| XP-8xx7-CE6 | Ver. 1.46 or later |
| XP-8xx7-Atom-CE6 | Ver. 1.03 or later |
| WP-8x47/8x37 | Ver. 1.66 or later |
| WP-5147 | Ver. 1.12 or later |
| VP-25W7/23W7/4137 | Ver. 1.59 or later |

Download the ISaGRAF Driver:

If your ISaGRAF does not support these functions, go to the website to download the new ISaGRAF driver.

[ISaGRAF Web Page](#) > Download – [Driver](#)

http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/isagraf/isagraf.html

Upgrade:

Refer to the PDF included in the downloaded ZIP file (e.g., “vp-2xw7-1.59.zip”) to upgrade the ISaGRAF driver.

| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 2 / 10 |

1.2. Download / Restore the ISaGRAF Project

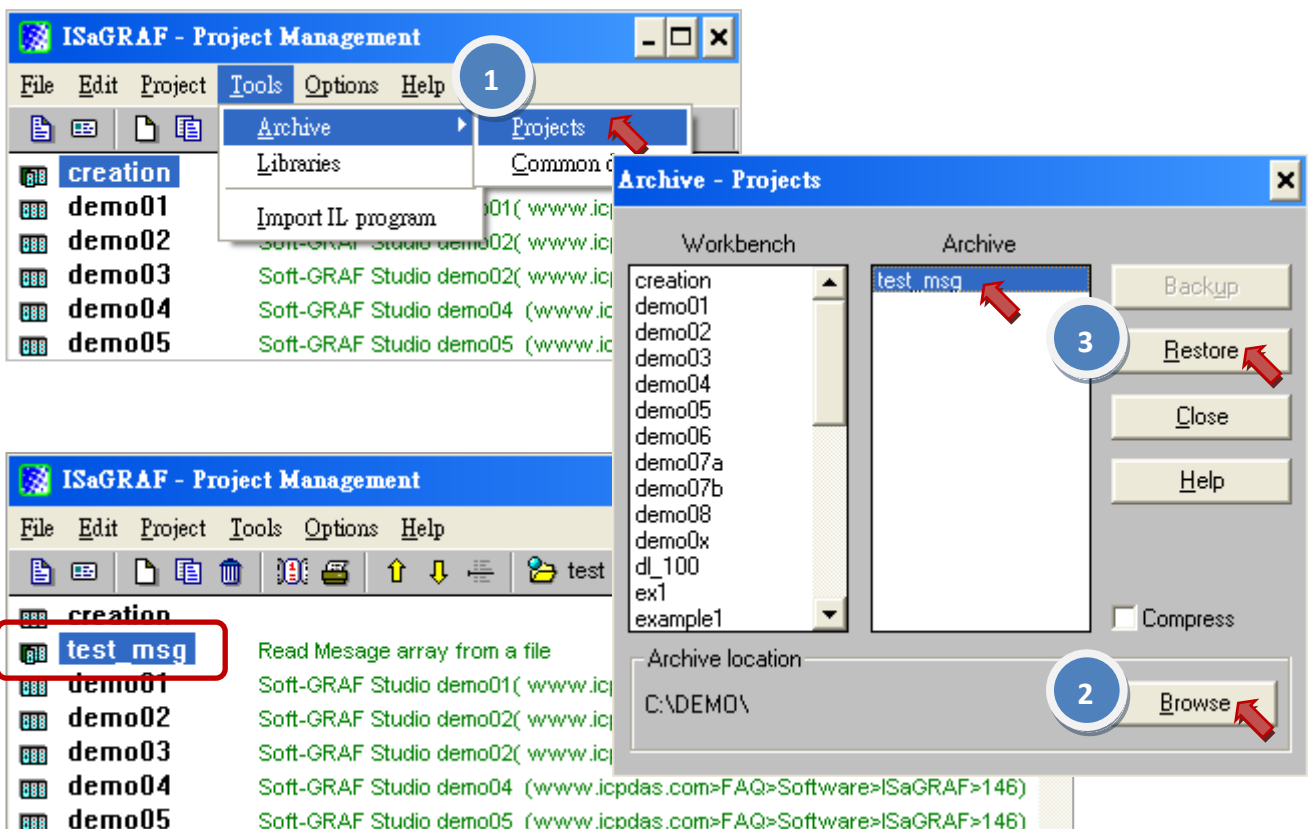
Download the demo program:

Go to [ISaGRAF Web Page](#) > Download – [FAQ](#) > FAQ-172 to download the “faq172_demo_chinese.zip” file. This file includes this document, the ISaGRAF demo program (“test_msg.pia”) and the data file (“User_file.txt”).

Restore the demo program:

Unzip the file and then restore the ISaGRAF demo program (“test_msg.pia”) to the PC/ISaGRAF.

1. Click “Tools → Archive → Projects” to open the “Archive – Projects” window.
2. Restore the “test_msg.pia” project to the ISaGRAF Workbench from an assigned folder (e.g., “C:\DEMO”).



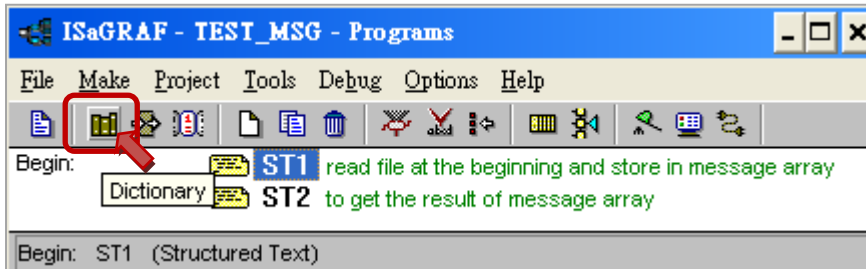
3. Then, copy the “User_file.txt” data file to the “\Micro_SD\” folder of the ISaGRAF PAC by using the FTP or USB Disk. (In this example, the “\Micro_SD\” is the default file location for this ISaGRAF project to read strings).

| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 3 / 10 |

1.3. Introduction of the demo program

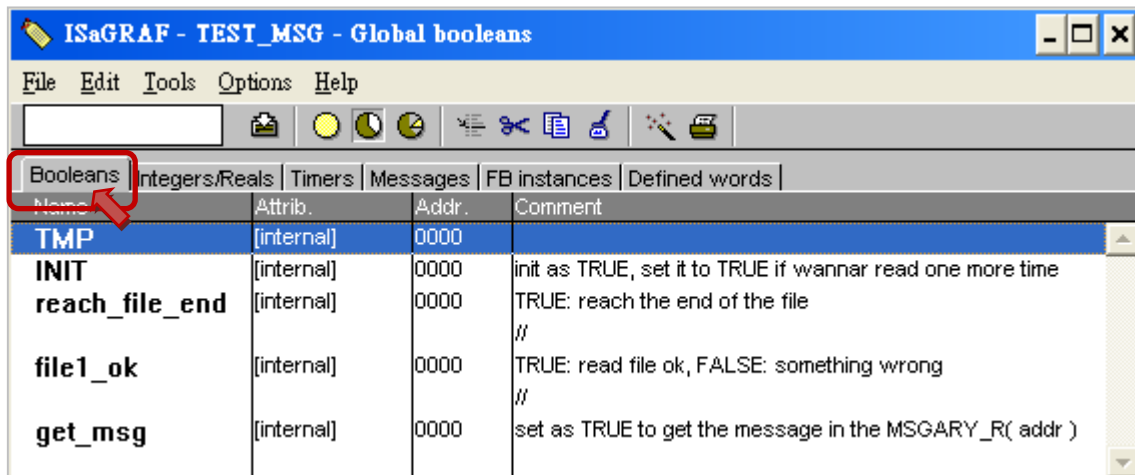
Project Architecture:

There are two ST programs in this project. Click the “Dictionary” button to see the variables that used in these programs.



ISaGRAF Variables Table:

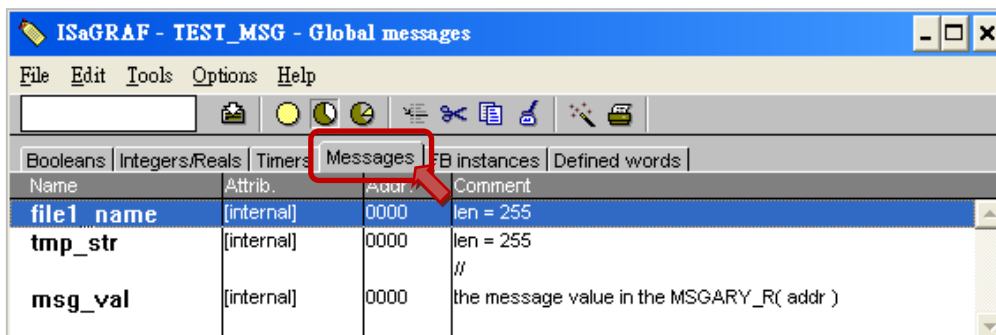
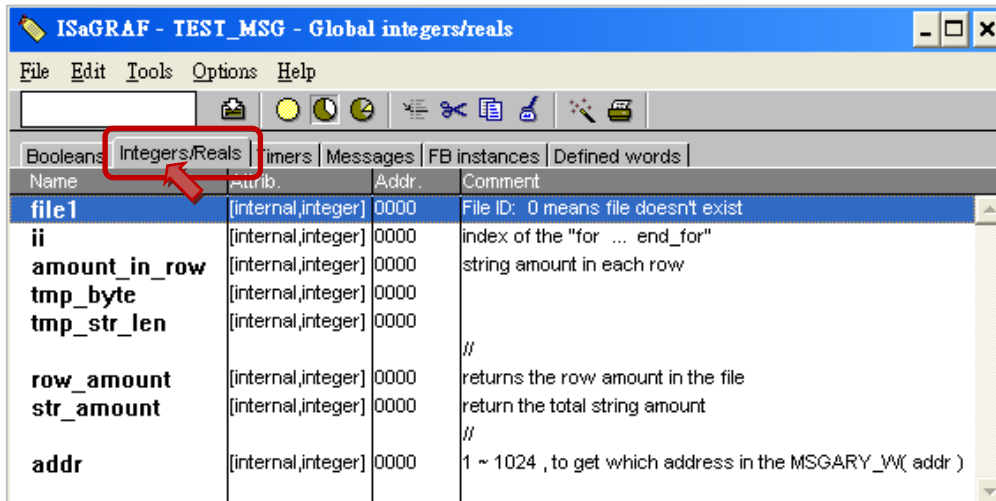
“Booleans” variables:



| Name | Type | Attribute | Description |
|----------------|---------|-----------|--|
| TMP | Boolean | Internal | For temporary usage. |
| INIT | | | Set this "True" to read the data file again. To declare its initial value as "True". |
| reach_file_end | | | When the status is "True", it means reach the end of the file. |
| file_ok | | | When the status is "True", it means read file OK. When the status is "False", it means read file error. |
| get_msg | | | Set this "True" to get message by using the “MSGARY_R(addr)” function. |

| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 4 / 10 |

“Integers/Reals” , “Messages” variables:



| Name | Type | Attribute | Description |
|---------------|---------|-----------|--|
| file1 | Integer | Internal | File ID: 0 means file doesn't exist. |
| ii | | | For the use of "For" loop in this program. |
| amount_in_row | | | It stands for the string amount in each row. (In this example, the value is 3) |
| tmp_byte | | | For temporary usage. |
| tmp_str_len | | | It stands for string length. (For temporary usage) |
| row_amount | | | It stands for the total amount of rows. (In this example, the value is 5) |
| str_amount | | | It stands for the total amount of strings. (In this example, the value is 15) |
| addr | | | The string address wish to get by the "MSGARY_R(addr)" function. (Address Range: 1 ~ 1024) |
| file_name | Message | | It stands for the file location. (In this example, the file is located at '\Micro_SD\User_file.txt') |
| tmp_str | | | For the temporary usage in the "MSGARY_W()". |
| msg_val | | | For the temporary usage in the "MSGARY_R(addr)". |

| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 5 / 10 |

ST1 Program:

(* Read string array in a file and store them by "MSGARY_W(addr, string)" .

File format : each row end with <CR><LF>, each row has 3 strings

0F02B37837 USER01 GRANT

0F02B376EE USER02 DENY

0F02B37B0B USER03 GRANT

0F02B37DCA USER04 GRANT

0F02B37C60 USER05 DENY

Then MSGARY_R(1) will return '0F02B37837'.

MSGARY_R(2) will return 'USER01'.

MSGARY_R(3) will return 'GRANT'.

MSGARY_R(4) will return '0F02B376EE'.

MSGARY_R(5) will return 'USER02'.

MSGARY_R(6) will return 'DENY'.

...

MSGARY_R(13) will return '0F02B37C60'.

MSGARY_R(14) will return 'USER05'.

MSGARY_R(15) will return 'DENY'.

The addr parameter can be 1 to 1024 in the MSGARY_R(addr) and MSGARY_W(addr,msg) .

*)

(* INIT is declared with initial value TRUE *)

if INIT then

INIT := FALSE ;

file1_name := '\Micro_SD\User_file.txt' ;

file1_ok := FALSE ; (* Init as "not ok" *)

reach_file_end := FALSE ; (* Init as no reach the end of file yet *)

amount_in_row := 3 ; (* This example has 3 strings in each row *)

row_amount := 0 ; (* Init total row amount as 0 *)

str_amount := 0 ; (* Init total string amount as 0 *)

file1 := f_ropen(file1_name) ; (* Try to open file1 *)

if file1 = 0 then

(* File doesn't exist *)

else

| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 6 / 10 |

```

(* File does exist, open it and read and then close it at the end *)
while (reach_file_end = FALSE) and ( row_amount < 1024/amount_in_row ) do
  for ii := 1 to amount_in_row do    (* Each row has 3 strings in this example *)
    if reach_file_end then
      exit ; (* Reach file end, exit "for" loop *)
    end_if ;
    tmp_str := "" ;
    tmp_str_len := 0 ; (* Init string length as 0 *)
    while tmp_str_len < 255 do (* Max string length is 255 in ISaGRAF *)
      TMP := f_eof(file1) ;
      if TMP then
        (* Reach the end of the file *)
        reach_file_end := TRUE ;
        if tmp_str_len > 0 then
          (* The current read message is complete , store this string by MSGARY_W() *)
          str_amount := str_amount + 1 ;
          TMP := MSGARY_W(str_amount , tmp_str) ;
          if ii = amount_in_row then
            row_amount := row_amount + 1 ;
          end_if ;
        end_if ;
        if (row_amount > 0) and ( MOD(str_amount , amount_in_row) = 0 ) then
          file1_ok := TRUE ; (* File read successfully *)
        end_if ;
        exit ; (* Exit "while" loop *)
      end_if ;
      (* It hasn't reach file end *)
      tmp_byte := f_read_b(file1) ; (* Read 1 byte *)
      if (tmp_byte = 0) (* Wrong string character or f_read_b() fail *)
        or (tmp_byte = 32) (* CHAR(32) = ' ', space *)
        or (tmp_byte = 9) (* CHAR(9) = HT , Horizontal Tab *)
        or (tmp_byte = 44) (* CHAR(44) = ',' , Comma *)
        or (tmp_byte = 13) (* CHAR(13) = CR *)
        or (tmp_byte = 10) (* CHAR(10) = LF *)
      then
        if tmp_str_len > 0 then

```

| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 7 / 10 |

```

        (* The current read message is complete , store this string by MSGARY_W() *)
        str_amount := str_amount + 1 ;
        TMP := MSGARY_W(str_amount , tmp_str) ;
        if ii = amount_in_row then
            row_amount := row_amount + 1 ;
        end_if ;
        exit ; (* Exit "while" loop *)
    end_if ;
else
    (* Other character , add it into the tmp_str *)
    tmp_str := tmp_str + CHAR(tmp_byte) ;
    tmp_str_len := tmp_str_len + 1 ;
end_if ;
end_while ;
end_for ;
end_while ;
TMP := f_close(file1) ;
end_if ;

(* If the final result is file1_ok = TRUE, then the "row_amount" is the total row amount in this
file *)

end_if ;

```

ST2 Program: To get the result of message array.

```

(* Read one message in the MSGARY( addr ) *)
if get_msg then
    get_msg := FALSE ;
    msg_val := MSGARY_R( addr ) ;
end_if ;

```

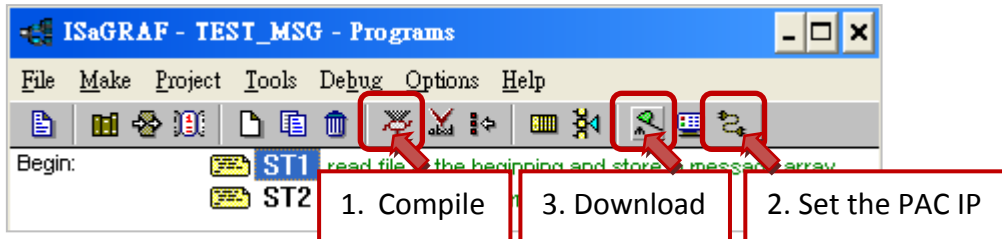
| | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|-------------|
| Classification | ISaGRAF English FAQ-172 | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page 8 / 10 |

1.4. Test the demo program

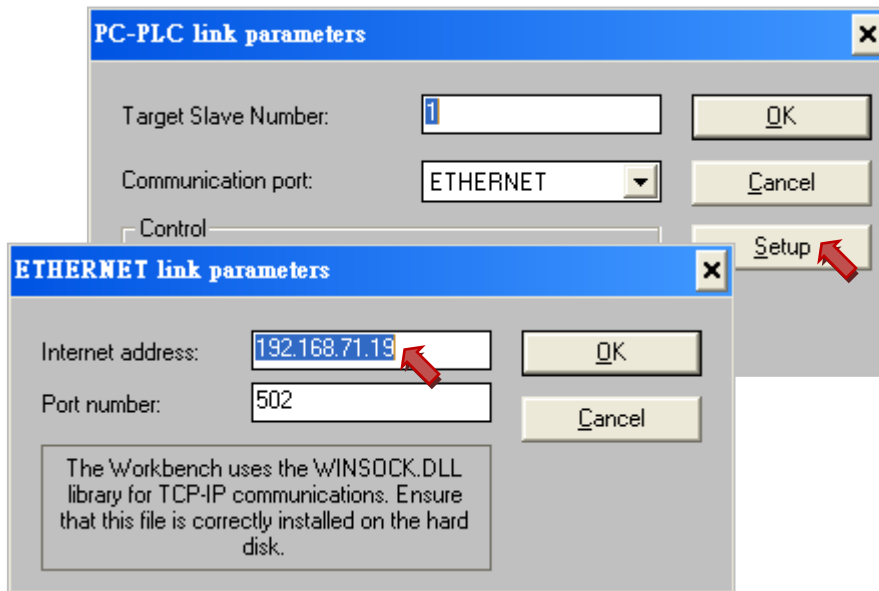
Users must download the ISaGRAF project (“TEST_MSG”) to the PAC before test it.

Download the ISaGRAF project:

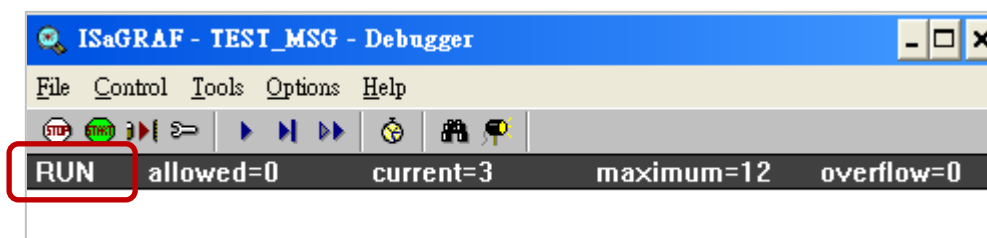
1. Click the menu bar “Make → Make application” or the tool button to re-compile the ISaGRAF project.



2. Click the menu bar “Debug → Link setup” or the tool button to set the download IP (i.e., PAC IP).



3. Click the menu bar “Debug → Debug” or the tool button to download the “TEST_MSG” project. If the process is successful, the “Debugger” window will show as below.



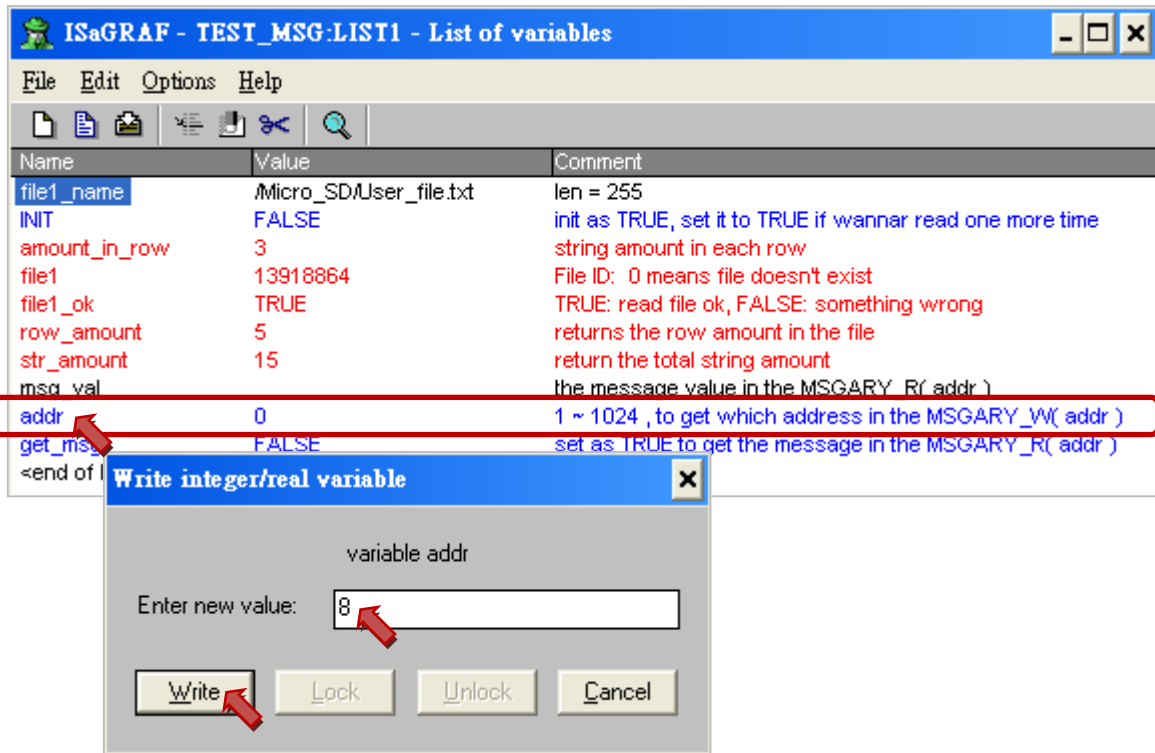
| | | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|------|--------|
| Classification | ISaGRAF English FAQ-172 | | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page | 9 / 10 |

Test Way:

Users will see the “Spy lists” after downloading the ISaGRAF project (or click “Tools → Spy lists” in the “Debugger” window).

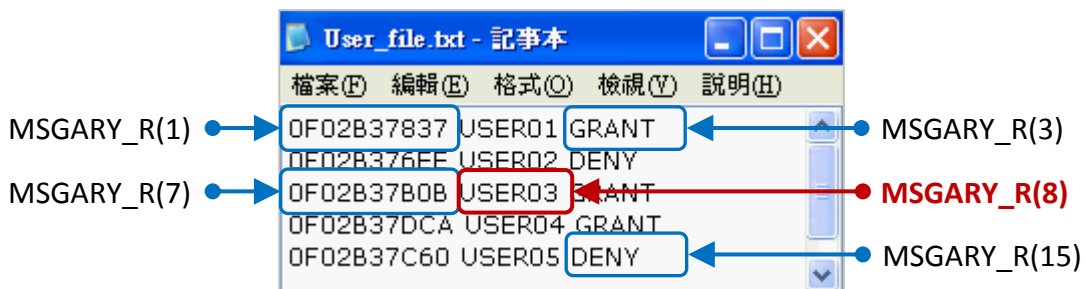
Notice: This project will try to read the data file once while running on a PAC. In this “Spy lists”, If the “file_ok” shows “FALSE” that means error reading from file, verify that the file exists in the specified location (e.g., “\Micro_SD\User_file.txt”) and the file format is correct (e.g., If the “amount_in_row” is specified as 3 that means there are three strings in each row. So, if there are only two strings in a row, it will cause a read error.) After the confirmation, set the “INIT” to “TRUE” to read the file again.

1. Double click the “addr” variable and assign a value (e.g., 8) to read the Nth string in the data file.



In this “Spy lists”, the value of the “row_amount” is 5 and the value of the “str_amount” is 15 that means there are total five rows and fifteen strings in this data file.

The corresponding sequence of these strings:



| | | | | | | |
|----------------|-------------------------|---------|-------|------|-----------|-----------------|
| Classification | ISaGRAF English FAQ-172 | | | | | |
| Author | Janice Hong | Version | 1.0.0 | Date | Feb. 2014 | Page 10 / 10 |

2. Double click the “get_msg” variable to set it as “TRUE” to get the string.
(Note: It will auto reset to “FALSE”)

| Name | Value | Comment |
|---------------|------------------------|---|
| file1_name | Micro_SD/User_file.txt | len = 255 |
| INIT | FALSE | init as TRUE, set it to TRUE if wannar read one more time |
| amount_in_row | 3 | string amount in each row |
| file1 | 13918864 | File ID: 0 means file doesn't exist |
| file1_ok | TRUE | TRUE: read file ok, FALSE: something wrong |
| row_amount | 5 | returns the row amount in the file |
| str_amount | 15 | return the total string amount |
| msg_val | | the message value in the MSGARY_R(addr) |
| addr | 8 | 1 ~ 1024 , to get which address in the MSGARY_W(addr) |
| get_msg | FALSE | set as TRUE to get the message in the MSGARY_R(addr) |

<end of list>

Write boolean variable ✕

variable get_msg

0 FALSE TRUE 1

3. Now, the return value of the “msg_val” variable is shown as below (e.g., “USER03”)

| Name | Value | Comment |
|---------------|------------------------|---|
| file1_name | Micro_SD/User_file.txt | len = 255 |
| INIT | FALSE | init as TRUE, set it to TRUE if wannar read one more time |
| amount_in_row | 3 | string amount in each row |
| file1 | 13918864 | File ID: 0 means file doesn't exist |
| file1_ok | TRUE | TRUE: read file ok, FALSE: something wrong |
| row_amount | 5 | returns the row amount in the file |
| str_amount | 15 | return the total string amount |
| msg_val | USER03 | the message value in the MSGARY_R(addr) |
| addr | 8 | 1 ~ 1024 , to get which address in the MSGARY_W(addr) |
| get_msg | FALSE | set as TRUE to get the message in the MSGARY_R(addr) |

<end of list>

Note: If the value of the "addr" variable exceeds the total amount of strings (e.g., 20 > 15), It will show null string.

| Name | Value | Comment |
|---------------|------------------------|---|
| file1_name | Micro_SD/User_file.txt | len = 255 |
| INIT | FALSE | init as TRUE, set it to TRUE if wannar read one more time |
| amount_in_row | 3 | string amount in each row |
| file1 | 13918864 | File ID: 0 means file doesn't exist |
| file1_ok | TRUE | TRUE: read file ok, FALSE: something wrong |
| row_amount | 5 | returns the row amount in the file |
| str_amount | 15 | return the total string amount |
| msg_val | | the message value in the MSGARY_R(addr) |
| addr | 20 | 1 ~ 1024 , to get which address in the MSGARY_W(addr) |
| get_msg | FALSE | set as TRUE to get the message in the MSGARY_R(addr) |

<end of list>