REALM Development

REALM MACROS

10/11/2008

Contents

1. Introduction

The inclusion of a macro language for the Resource Allocation Model (REALM) has been undertaken at the request of Victoria University and the Department of Sustainability and Environment. The introduction of macros to REALM allows the user to automate tasks by creating a macro script file which includes instructions to execute REALM commands. It allows tasks to be undertaken such as loading a scenario file, changing aspects of the scenario file, including log file name, input file names, start and end dates of simulations and the specified system file. It also allows system file information to be changed, such as the equation in any variable capacity carrier or the maximum capacity of any carrier within the given system file. This report documents the REALM macro capability.

2. REALM Macro Capabilities

The REALM macro language allows the user to make inquiries and change specified REALM variables and output options. It also allows the user to save changes to both system and scenario files and run scenario files through REALM.

The change, save and run capabilities of the program can be used to run REALM scenarios in batch mode. REALM macros are capable of handling scenario files with multiple associated system files.

An overview of the types of commands included in the macro language is listed below.

Inquiries:

- River System Node;
- River System Carrier;
- Associated Files;
- Content of associated Files;
- Single node or carrier inquiry;
- Output variables;
- Scenario variables.

System parameter changes:

- Associated Files;
- Single node;
- Output variable selection;
- Scenario variables;
- Save the amended system & scenario files;
- Run the model.

System Output Manipulation:

- Node outputs;
- Carrier Outputs.

An inquiry of REALM variables may include:

- Node numbers and associated name, node type and auxiliary inputs;
- Carrier number and associated name, starting node and ending node;
- Names of flow and demand time series associated with each input file;
- Associated system file lists.

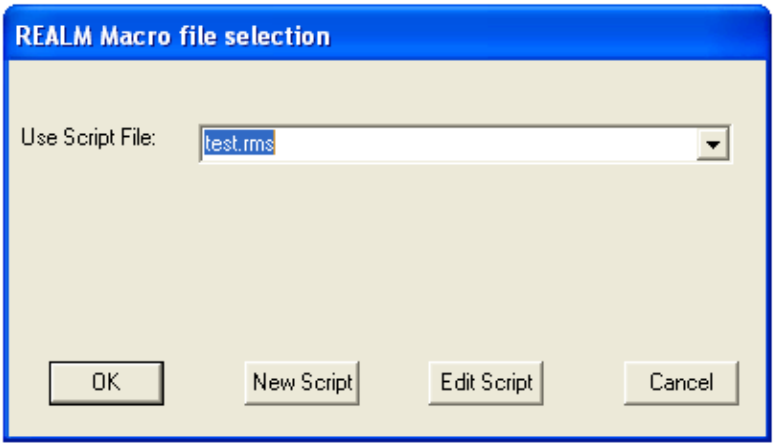
REALM variables that can be inquired and changed include:

- Carrier penalties and losses;
- Names of input flow/climate and demand files;
- Output options for either a specified node or carrier or for a specified type including reservoirs, demand centres, stream junctions and carriers;
- The capacity associated with a specified carrier;
- Log file name;
- Log file description;
- Start year and seasons;
- End year and seasons;
- System file name;
- Initial volumes of all or specified reservoirs.

The details of these features is described in the sections 2.2.1 to 2.4.2 below.

2.1. Getting Started

REALM macros are run from within the main screen of the REALM user interface by selecting the Macros menu item from the Run Menu. This action displays the Run Macros Dialog box as shown below in Figure 1.



- **Figure 2-1 REALM Macro File Selection Dialog Box**

This dialog box allows REALM macro script files (*.rms) in the current directory to be selected and run by using the drop down box and pressing OK. Alternatively, existing script files can be edited or new script files created using the “New Script” or “Edit Script” buttons.

The form of the script file is described below under the three categories of script commands: Inquiries, Changes and Operational Commands. Typically, the first line of the script file will instruct the macro which scenario file to load. Example script files are presented at the end of this report.

2.2. Inquiries

The following inquiry commands, unless otherwise stated require the load scenario command to be used prior to any of the inquiry commands being used. The load scenario command is “*Load : Example.scn*”.

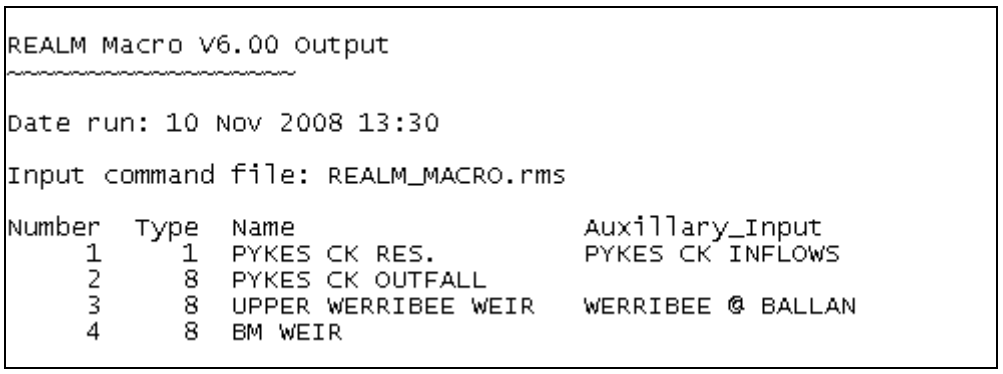
2.2.1. River System Node Inquiry

This inquiry lists all the nodes in the river system, together with key attributes. The format of the macro command is: "`WriteNodes file:nodes.txt",` where `"WriteNodes file:"` is the inquiry command and `"nodes.txt"` is the output file name (can be a *.txt of any name). This command must be preceded by a “*Systemfile ExampleSystemFile.sys*” command. This loads the system file into the memory to enable the extraction of the node attributes. The output is written to a text file and includes the following information:

- Node Number

- Node Name
- Node type
- Auxiliary Input.

Figure 2-2 outlines and example output from the WriteNodes command.



```
REALM Macro V6.00 Output
~~~~~~~~~~~~~~~~~~~~~~~

Date run: 10 Nov 2008 13:30

Input command file: REALM_MACRO.rms

Number  Type  Name                    Auxillary_Input
     1     1  PYKES CK RES.           PYKES CK INFLOWS
     2     8  PYKES CK OUTFALL
     3     8  UPPER WERRIBEE WEIR     WERRIBEE @ BALLAN
     4     8  BM WEIR
```

- **Figure 2-2 Example WriteNodes command output**

Note: This information is currently listed in the *.lis file which is created by the system listing utility.

2.2.2. River System Carrier Inquiry

This inquiry lists all the carriers in the river system, together with key attributes. The format of the macro command is: "`WriteCarriers file:carrier.txt",` where `"WriteCarriers file:"` is the inquiry command and `"carriers.txt"` is the output file name (can be a *.txt of any name). This command must be preceded by a "*Systemfile ExampleSystemFile.sys*" command. This loads the system file into the memory to enable the extraction of the carrier attributes. The output is written to a text file which includes the following information:

- Number
- Name
- Type
- From Node
- To Node
- Penalty
- Loss

Figure 2-3 outlines an example output from the WriteCarriers command.

```
REALM Macro V6.00 Output
~~~~~~~~~~~~~~~~~~~~~~~

Date run: 10 Nov 2008 13:30
 |
Input command file: REALM_MACRO.rms

Input system file: WERRIRRG2.SYS

Number  Type  Name                  From Node  To Node    Penalty       Loss
     1     1  PYKES DIV1                    3        5         10          0
     2     1  PYKES DIV2                    5        1         10          0
     3     2  UPPER WEIR OVERFLOW           3        2        100          0
     4     2  PYKES CK SPILL               29        2         10          0
     5     2  WERRIBEE D/S PYKES            2       12         10          1
     6     2  TO BM IRRIGATION              8        6         10         15
```

- **Figure 2-3 Example WriteCarriers output**

Note: This information is currently listed in the *.lis file which is created by the system listing utility.

2.2.3. Associated Files Inquiry

There are two types of time series files specified in the REALM scenario file, streamflow and demand files. This inquiry uses the two inquiry statements for each of these:

`Inquire parameters: Fileflw?` : This command returns a list of all streamflow files in the following format. Eg:

```
Streamflow file 1 = goul1.fdy
Streamflow file 2 = goul2.fdy
Streamflow file 3 = c:\realm\work\goulburn\data\goul3.fdy
Streamflow file 4 = goul4.fdy
```

The “*Inquire parameters:* `Filedmd?`” command returns a list of all demand files in the scenario file.

```
Demand file 1 = DirrectIrrigation.dem
Demand file 2 = c:\realm\work\goulburn\data\FarmDams.dem
```

2.2.4. Content of associated Files Inquiry

This command lists the total number of variables and the name of each variable within each flow or demand time series which is specified in the *.scn file.

The two commands are:

- `Inquire parameters: noflowfiles?` and
- `Inquire parameters: nodemdfiles?`

Note: This feature does not have an equivalent function or utility currently in REALM.

Figure 2-4 and Figure 2-5 show the outputs from the “noflowfiles?” and “nodemfiles?” commands.

```
Processing Full Command: inquire parameters: noflowfiles?

Number of input files:   1
                   Flow or Climate Input File Name   No.   Input Name
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                       werrflow.sf    1   WERRIBEE @ BALLAN
                                       werrflow.sf    2   PYKES CK INFLOWS
                                       werrflow.sf    3   LERDERDERG U/S DIV
                                       werrflow.sf    4   GOODMAN CK U/S DIV
                                       werrflow.sf    5   COIMADIA CK INFLOWS
                                       werrflow.sf    6   DJERRIWARRH INFLOWS
                                       werrflow.sf    7   MERRIMU RES INFLOWS
                                       werrflow.sf    8   LOWER LERD INFLOWS
                                       werrflow.sf    9   PARWON CK
                                       werrflow.sf   10   TOOLERN CK
                                       werrflow.sf   11   INFLOW BET.N WEIRS
                                       werrflow.sf   12   INFLOW BM TO MELTON
                                       werrflow.sf   13   INFLOW
                                       werrflow.sf   14   RAINFALL(87039)
                                       werrflow.sf   15   EVAPORATION
                                       werrflow.sf   16   RAINFALL(87002)
```

Figure 2-4 Output from the "noflowfiles" command

```
Processing Full Command: inquire parameters: nodemdfiles?

Number of input files:   1
                    Demand Input File Name           No.  Input Name
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                          WERRIRR.DEM   1  LERDERDERG DIV
                                          WERRIRR.DEM   2  MID RIVER DIV
                                          WERRIRR.DEM   3  LOWER DIV
                                          WERRIRR.DEM   4  WERRIBEE IRR
                                          WERRIRR.DEM   5  WERRIBEE O/S SALES
                                          WERRIRR.DEM   6  WERRIBEE OUTFALLS
                                          WERRIRR.DEM   7  BM IRRIGATION
                                          WERRIRR.DEM   8  BM OUTSIDE SALES
                                          WERRIRR.DEM   9  CSR FACTORY
                                          WERRIRR.DEM  10  BM URBAN
                                          WERRIRR.DEM  11  MELTON URBAN
                                          WERRIRR.DEM  12  WERIRR
                                          WERRIRR.DEM  13  BMIRR
                                          WERRIRR.DEM  14  DIVERSION DEM
                                          WERRIRR.DEM  15  NOTIONAL DEMAND
                                          WERRIRR.DEM  16  NOTIONAL2000
```

- **Figure 2-5 Output from "nodemfiles" command**

2.2.5. Single node or carrier inquiry

This inquiry is designed to return information relevant to a particular node or carrier. It only includes parameters which modellers are able to modify using REALM in interactive mode. The command only returns the node or carrier attribute applicable to the specified node or carrier. The following example outlines the results from inquiring about a reservoir node.

The "`Inquire parameters: nd(XXX) %output?" command` outputs a list of all relevant output types for that node type and whether it is selected for output. (Where XXX is the node number).

Eg: If node 010 is a reservoir the command "`nd(010) %output`?" gives the following results in the format:

```
Output options for node 10
==========================
%lvls  True
%stor  True
%rels  True
%evap  False
%targ  True
%spil  False
%infw  True
```

Similarly, the command "`ca(XXX) %output?`" provides a list relating to carrier output properties, however in the case of a carrier there is one additional line of information output. As carriers can be individually selected for output, the inquiry also outputs a line with the carrier number and whether it is selected for output. For example:

```
Output options for Carrier 3
============================
%flws    True
%capc    True
%loss    True
%eqns    False
%3       False
```

Table 1 contains a list of inquiries for each node or carrier type. All of these must be preceded by the command: `"Inquire parameters: "`

- **Table 1 Inquiry commands**

Command	Type	Comment
`nd(XXX) %output?`	All Nodes	Returns the status of all valid output properties of the node type.
`nd(XXX) %flwinput?`	SJ, RV	Returns the time series streamflow name input at the node.
`Ca(XXX) %output?`	All carriers	Returns the status of all valid output properties of the carrier type
`Ca(XXX) %loss?`	All carriers	Returns the carrier loss (+ve = % loss, -ve = constant loss)
`Ca(XXX) %capc?`	All carriers	Returns the maximum carrier capacity for fixed capacity carriers

2.2.6. Output variables inquiry

This inquiry provides a list of the output selections for all nodes or carriers of a given type. The list of commands are:

`Inquire parameters: quanout (rv)?` : This command lists all reservoir node output selections;

`Inquire parameters: quanout (dc)?` : This command lists all demand node output selections;

`Inquire parameters: quanout (sj)?` : This command lists all demand node output selections;

`Inquire parameters: quanout (ca)?` : This command lists which output properties of carriers have been selected (ie flow, capc, loss) and then a list of all carriers showing whether they have been selected for output. Eg:

```
%flow     True
%capc     True
%loss     True
%eqns     False
%   1     True
%   2     False
%   3     True
etc…
```

`quanout (al)?` : This command lists all the information that is output from each of the individual inquiries listed above.

This information is currently shown in the output dialog box in the setup menu of REALM.

2.2.7. Scenario Variables Inquiry

This inquiry returns certain information specified in the scenario file. The commands and parameters are as follows:

`Inquire parameters: Scen%LogName? :` Log file name;

`Inquire parameters: Scen%LogDesc? :` Log file description;

`Inquire parameters: Scen%StartYr? :` Start Year;

`Inquire parameters: Scen%StartSeas? :` Start Season;

`Inquire parameters: Scen%EndYr? :` End Year;

`Inquire parameters: Scen%EndSeas? :` End Season;

`Inquire parameters: Scen%System? :` Lists all system files in the scenario file;

`Inquire parameters: Scen%InitRes? :` Reservoir initial volumes. This command outputs a list including the node number of each reservoir in the following format:

```
No    Node No   Init. Vol.
=========================
  1         1       14419
  2        15        5000
```

```
3          22          6735
4          24           341
5          37         16098
6          65             0
7          82          5537
```

2.3. Parameter Changes

Unless otherwise specified the following commands need to be preceded by the “Load : *example.scn*” command.

2.3.1. Associated Files parameter changes

This command allows each of the input files to be changed. It is based on the format of the associated file inquiry. The following example changes the first flow input file to n2.fdy.

```
Change parameters: Fileflw%1 = c:\realm\work\goulburn\data\n2.fdy
```

2.3.2. Single node or carrier parameter changes

The parameter change statements are based on the format of the output of the single node inquiry. Table 2 contains a list of parameter change statements, which are preceded by "`Change parameters:` " command.

Table 2 Change commands

Command	Type	Comment
`Quanout(RV)`	All Nodes	Sets the status of all valid output properties of the node type.
`nd(XXX) %flwinput =`	SJ, RV	Sets the time series streamflow name input at the node.
`Ca(XXX) %loss =`	All carriers	Sets the carrier loss (+ve = % loss, -ve = constant loss)
`Ca(XXX) %capc =`	All carriers	Sets the maximum carrier capacity for fixed capacity carriers

Examples of each of these statements are presented below:

```
Change parameters: quanout(rv)%STOR = True
Change parameters: Nd(012) %flwinput = BIG RIVER
Change parameters: Ca(001) %loss = 0.05
```

2.3.3. Output variable selection changes

The parameter change statements are based on the format of the output variable inquiry. Examples of these statements are provided below.

```
Change parameters: quanout  (rv) %lvls = False
Change parameters: quanout  (dc) %unrs = True
Change parameters: quanout  (ca) %flow = True
Change parameters: quanout  (ca) %capc = True
Change parameters: quanout  (ca) %loss = False
Change parameters: quanout  (ca) %N001 = False
Change parameters: quanout  (ca) %N002 = False
```

2.3.4. Scenario Variables Parameter Changes

These commands are based on the output format of the inquiry statements. Examples are shown in the list below:

```
Change parameters: Scen%LogName = A001.log : Log file name;
Change parameters: Scen%LogDesc = Test macro scenario 1 : Log file description;
Change parameters: Scen%StartYr = 1950 : Start Year;
Change parameters: Scen%EndYr = 2000 : End Date;
Change parameters: Scen%System = c:\realm\work\goulburn\Goulburn.sys : System File;
Change parameters: Scen%InitRes(010) = 50000 : Reservoir initial volumes.
```

2.4. Operational Commands

2.4.1. Save the amended system & scenario files

The two commands save either the REALM system file or the REALM scenario file. These need to be used to save any changes made through the “Change parameters” command.

```
Change parameters: SaveAs system: NewGoulb.sys
Change parameters: SaveAs scenario: A001.scn
```

2.4.2. Run the model

This command runs the specified scenario file.

```
Run : A001.scn
```

2.4.3. Execute a DOS Command

This command allows a dos command to be executed, as if at the DOS prompt. Two examples are as follows.

```
doscmd : copy gsm_test.txt gsm_testspring.txt
```

```
doscmd : testdata.exe
```

3. REALM Macro Operation

The files required and the process to run REALM macros is described in Sections 3.1 to 3.3 below.

3.1. Files required to run REALM Macros

To run REALM Macros you will need the following files in the same directory:

- REALM_SO.exe;
- A REALM.set file;
- A REALM Macro Script File (*.rms); and
- REALM.exe.

In addition to the above files, the other files that may need to be accessed but which may be in different directories, provided there is a REALM.set file accompanying them are:

- Scenario Files;
- Associated system files (.sys); and
- Associated flow/climate and demand input files.

3.2. Command Input files

The functions that REALM Macro executes are supplied by a user created command script file. Script files should be saved as text files with the extension .rms and contain a series of REALM Macro commands, see Table 3.

Commands start on the first line of the input file, and one line is used for each command (blank lines are not required between commands).

3.3. Running REALM Macros

Once the REALM macro script file has been created and all the necessary files are in the relevant directories, the macro script file is ready to be run. When the user selects OK from the “REALM macro file selection” dialog box, REALM then executes each line of the input file. The program creates a log file called of the same name as the input *.rms file with end of the file and extension being “_log.txt”. This log file keeps track of the progress and stores inquiry information, with the exception of writenode and writecarrier inquiries which are stored in separate files. It should be noted that this log file is different from the log file created when REALM is run and only refers specifically to REALM Macro commands.

4. Summary of REALM Macro commands

Table 3 lists all commands recognised by the REALM macros program and gives a brief description of the command's function.

Table 3 REALM Macro commands and descriptions

Example of Command	What it does
Systemfile sysname.sys	Selects a system file. All subsequent commands will apply to this system file until it is respecified. The command comprises of 'systemfile' followed by the system file, with path if necessary. If this command is not given, the program will use the first system file name specified in the scenario file.
Load : QQQ.scn	Loads system file (QQQ.scn) into the memory to enable it to be altered
WriteCarriers file: carrier.txt	Lists all the carriers in the river system together with key attributes: number, name, type, from node, to node, penalty and loss. Information is output to a text file. The output file name is given in the command after 'file:'.
WriteNodes file: nodes.txt	Same as WriteCarriers except outputs key attributes of nodes, including node number, name, type and auxiliary input.
Inquire parameters: Fileflw?	Returns a list of all streamflow files.
Inquire parameters: Filedmd?	Returns a list of all demand files.

Example of Command	What it does
Inquire parameters: noflowfiles? Inquire parameters: nodemdfiles?	Lists the total number of variables and the name of each variable with each flow or demand time series specified in the scenario file.
Inquire parameters: nd(XXX) %output? Inquire parameters: ca(XXX) %output?	Outputs a list of all relevant output types for that node or carrier type and whether it is selected for output. For carriers, it also outputs whether the particular carrier is selected for output. XXX is the node or carrier number.
Inquire parameters: nd(XXX) %flwinput?	This command is relevant for stream junction and reservoir nodes and returns the time series streamflow name input at the node.
Inquire parameters: ca(XXX) %loss?	Relevant for all carriers and returns the carrier loss (+ve=% loss, -ve=constant loss).
Inquire parameters: ca(XXX) %capc?	Relevant for all carriers and returns the maximum carrier capacity for fixed capacity carriers.
Inquire parameters: quanout (rv)? Inquire parameters: quanout (dc)? Inquire parameters: quanout (sj)? Inquire parameters: quanout (ca)? Inquire parameters: quanout (al)?	Provides a list of the output selections for all nodes or carriers of a given type. rv = reservoir, dc = demand centre, gd = gravity diversion, sj = stream junction, ca = carrier. quanout (al) provides output selections for all node and carrier types.
Inquire parameters: Scen%logName?	Outputs log file name.

Example of Command	**What it does**
Inquire parameters: Scen%LogDesc?	Outputs log file description.
Inquire parameters: Scen%StartYr?	Outputs start year.
Inquire parameters: Scen%StartSeas?	Outputs start season.
Inquire parameters: Scen%EndYr?	Outputs end year.
Inquire parameters: Scen%EndSeas?	Outputs end season.
Inquire parameters: Scen%System?	Outputs all system files associated with scenario file.
Inquire parameters: Scen%InitRes?	Lists reservoir node numbers and initial volumes for all reservoirs.
Change parameters: Fileflw%1=QQQ Change parameters: Filedmd%1=QQQ	Allow each of the flow and demand input files to be changed. X = which input file to be changed (remember 'Inquire Parameters: FileFlw?' and 'Inquire Parameters: Filedmd?' can be used as a reference for this number). QQQ = the new input file name (with path if necessary).
Change parameters: nd(XXX) %flwinput= QQQ	Relevant for reservoir and stream junction nodes, sets the time series streamflow name input at the node to QQQ.
Change parameters: ca(XXX) %loss= QQQ	Relevant to all carriers, sets the carrier loss (+ve=% loss, -ve=constant loss) to QQQ.
Change parameters: ca(017) %capc= QQQ	Relevant for all carriers, sets the maximum carrier capacity for fixed capacity carriers

Example of Command	What it does
	to QQQ.
Change parameters: quanout(TT) %HHHH= QQQ	Changes output options. Where, TT is a two letter node/carrier type- rv = reservoir, dc = demand centre, sj = stream junction, ca = carrier. quanout (al) provides output selections for all node and carrier types. HHHH= four letter output code, see Table 4. QQQ is either True of False To set an individual carrier, specify HHHH in the form of the carrier number (ieN002).
Change parameters: Scen%Logname=QQQ.log	Changes log name to QQQ.
Change parameters: Scen%LogDesc=QQQ	Changes scenario description to QQQ
Change parameters: Scen%StartYr=QQQ	Changes the start year of the selected system file to QQQ.
Change parameters: Scen%StartSeas= QQQ	Changes the start season of the selected system file to QQQ.
Change parameters: Scen%EndYr=QQQ	Changes the end year of the selected system file to QQQ.
Change parameters: Scen%EndSeas=QQQ	Changes the end season of the selected to QQQ.
Change parameters: Scen%System=QQQ.sys	Changes the system file specified in the scenario file to QQQ.
Change parameters: Scen %InitRes(XXX)=	Relevant only to reservoir nodes, changes the initial reservoir volume of node XXX to

Example of Command	**What it does**
QQQ	QQQ.
SaveAs system: QQQ.sys	Saves system file changes in a system file named QQQ.sys.
SaveAs scenario: QQQ.scn	Saves scenario file changes in a scenario file named QQQ.scn.
Run: QQQ.scn	Runs the saved scenario file named QQQ.scn.
`doscmd : X`	Executes a dos command X, as if at the command prompt

- **Table 4 Output descriptions and code names**

Name	Sub-Group	Output Code	Type Description
Node	Reservoir (rv)	STOR	Start season storage
		LVLS	Start water levels (via RV rating table)
		INFW	Inflow data from the streamflow file *
		EVAP	Net Evaporation
		TARG	REALM calculated target value
		SPILL	Uncontrolled release over spillway **
		RELS	Sum of water released
	Demand Centre (dc)	UNRS	Unrestricted demand
		REST	Restricted demand
		SHRT	Demand shortfall
		RATN	Rationed demand
		LVLS	Demand restriction levels / allocations
		SUPP	Demand actually supplied
	Stream Junction (sj)	INFW	Inflow data from the streamflow file
Carrier (ca)		FLOW	Carrier flow
		CAPC	Carrier capacity
		LOSS	Carrier loss

* does not include inflow from upstream carriers

** External spills only (ie when there are no downstream carriers)

5. Notes on REALM Macro Use

REALM Macros has been designed to allow some flexibility when recognising user commands, for example, commands are not case sensitive. However, there are a few necessities that need to be remembered.

- If the scenario, system or any of the input files are not in the same directory, complete or relative path names must be used (e.g. c:\work\Flow.fdy or ..\Inputs\Demand.dem);
- 'Inquire' statements always finish with a question mark '?';
- Names such as log file names, input names and system file names do not have be surrounded by inverted commas;
- Percentage signs, '%' are used in several command statements and are important in identifying output types;
- Although the Change commands assign new values to specified variables, these changes are not reflected in the system or scenario file until the changes have been saved. Always follow up a series of Change statements with a Save As statement;
- REALM does not specify an end date for each system file. The 'EndYr' and 'EndSeas' variable refers to the end year and season of the entire simulation period. However, the 'StartYr' and 'StartSeas' variables do refer to specific system files and the value returned will be relevant for the active system file. REALM assumes system files run back to back, so if the user wants to change the end year or season of a specific system file, it must change the start date of the next system file (unless it is the last system file, then the 'EndYr' and 'EndSeas' dates can be changed);
- When changing the name of input flow variables to nodes, it will always include the new name in upper case.

Appendix A Example Macro Script Files

Example A – Inquiries, changes and running the model

```
writenodes  file:nodelist.txt
writecarriers file:carrlist.txt
inquire parameters: scen%logname?
inquire parameters: filedmd?
inquire parameters: fileflw?
inquire parameters: noflowfiles?
inquire parameters: nodemdfiles?
inquire parameters: quanout (rv)?
inquire parameters: quanout (ca)?
inquire parameters: quanout (sj)?
inquire parameters: quanout (dc)?
inquire parameters: quanout (al)?
inquire parameters: nd(3) %flwinput?
inquire parameters: nd(25) %output?
inquire parameters: ca(3) %output?
inquire parameters: ca(3) %capc?
inquire parameters: ca(100) %loss?
change parameters:  scen%logname = tjs1.log
change parameters:  scen%system = t3.sys
inquire parameters: scen%system?
change parameters: quanout (rv) %lvls= true
change parameters: quanout (rv) %stor= false
change parameters: quanout (ca) %capc= false
change parameters: quanout (ca) %n003= true
change parameters: nd(3) %flwinput= LERDERDERG U/S DIV
change parameters: ca(3) %loss= 23
change parameters: ca(3) %capc = 900 200 300 400 500 600 700 800 900 1000 1100
1300
inquire parameters: scen%initres?
saveas scenario: tttt.scn
saveas system: t2.sys
```

Example B – Load scenario and undertake one year runs using the same initial storage levels but different climate, each with different log file and scenario file names

```
load : C074.scn
Run : C074.scn
load : C074.scn
Change parameters: Scen%StartYr = 1975
Change parameters: Scen%EndYr = 1976
Change parameters : Scen%LogName = C075.log
SaveAs scenario :  C075.scn
Run : C075.scn
load : C074.scn
Change parameters: Scen%StartYr = 1976
Change parameters: Scen%EndYr = 1977
Change parameters : Scen%LogName = C076.log
SaveAs scenario :  C076.scn
Run : C076.scn
load : C074.scn
Change parameters: Scen%StartYr = 1977
Change parameters: Scen%EndYr = 1978
Change parameters : Scen%LogName = C077.log
SaveAs scenario :  C077.scn
Run : C077.scn
load : C074.scn
Change parameters: Scen%StartYr = 1978
Change parameters: Scen%EndYr = 1979
Change parameters : Scen%LogName = C078.log
SaveAs scenario :  C078.scn
Run : C078.scn
load : C074.scn
Change parameters: Scen%StartYr = 1979
Change parameters: Scen%EndYr = 1980
Change parameters : Scen%LogName = C079.log
SaveAs scenario :  C079.scn
Run : C079.scn
load : C074.scn
Change parameters: Scen%StartYr = 1980
Change parameters: Scen%EndYr = 1981
Change parameters : Scen%LogName = C080.log
SaveAs scenario :  C080.scn
Run : C080.scn
```

Example C – Load and run base scenario and then load base and change input files and run scenarios with different log file and scenario file names

```
Load : base.scn
run : base.scn
Load : base.scn
change parameters: fileflw%9=D:\ReadyReckoner\InputFiles\K\WandOut\C1Acapc.ar
change parameters: fileflw%10=D:\ReadyReckoner\InputFiles\K\WandOut\C1Aec.ar
change parameters: fileflw%11=D:\ReadyReckoner\InputFiles\K\WandOut\C1Aflow.ar
change parameters: scen%logname= C1A.log
saveas scenario:C1A.scn
run : C1A.scn
Load : base.scn
change parameters: fileflw%9=D:\ReadyReckoner\InputFiles\K\WandOut\C2Acapc.ar
change parameters: fileflw%10=D:\ReadyReckoner\InputFiles\K\WandOut\C2Aec.ar
change parameters: fileflw%11=D:\ReadyReckoner\InputFiles\K\WandOut\C2Aflow.ar
change parameters: scen%logname= C2A.log
saveas scenario:C2A.scn
run : C2A.scn
Load : base.scn
change parameters: fileflw%9=D:\ReadyReckoner\InputFiles\K\WandOut\C3Acapc.ar
change parameters: fileflw%10=D:\ReadyReckoner\InputFiles\K\WandOut\C3Aec.ar
change parameters: fileflw%11=D:\ReadyReckoner\InputFiles\K\WandOut\C3Aflow.ar
change parameters: scen%logname= C3A.log
saveas scenario:C3A.scn
run : C3A.scn
Load : base.scn
change parameters: fileflw%9=D:\ReadyReckoner\InputFiles\K\WandOut\C4Acapc.ar
change parameters: fileflw%10=D:\ReadyReckoner\InputFiles\K\WandOut\C4Aec.ar
change parameters: fileflw%11=D:\ReadyReckoner\InputFiles\K\WandOut\C4Aflow.ar
change parameters: scen%logname= C4A.log
saveas scenario:C4A.scn
run : C4A.scn
Run : C080.scn
```

Appendix B Example Macro Log File

```
REALM Macro V1.00 Output
~~~~~~~~~~~~~~~~~~
Date run:  28 Aug 2003 08:41
Input command file: tjslr.txt
Input scenario file: mact.scn
Processing writenodes  file:nodelist.txt
   Information written to nodelist.txt
Processing writecarriers file:carrlist.txt
   Information written to carrlist.txt
Processing inquire parameters: scen%logname?
   scen%logname=
mac1.log
Processing inquire parameters: filedmd?
   Filedmd  1 = C:\mac_test\DEMD20-00b.PRN
   Filedmd  2 = C:\werrmb43\DEMACCNT.prn
Processing inquire parameters: fileflw?
   Fileflw  1 = C:\mac_test\fTJS20-00.prn
   Fileflw  2 = C:\werrmb43\CLIM20-00.PRN
   Fileflw  3 = C:\werrmb43\BELowTun.fmy

Processing inquire parameters: noflowfiles?
Number of input files:   3
```

Flow or Climate Input File Name	No.	Input Name
C:\mac_test\fTJS20-00.prn	1	WERRIBEE @ BALLAN
C:\mac_test\fTJS20-00.prn	2	PYKES CK INFLOWS
C:\mac_test\fTJS20-00.prn	3	LERDERDERG U/S DIV
C:\mac_test\fTJS20-00.prn	4	GOODMAN CK U/S DIV
C:\mac_test\fTJS20-00.prn	5	COIMADIA CK INFLOWS
C:\mac_test\fTJS20-00.prn	6	DJERRIWARRH INFLOWS
C:\mac_test\fTJS20-00.prn	7	MERRIMU RES INFLOWS
C:\mac_test\fTJS20-00.prn	8	LOWER LERD INFLOWS
C:\mac_test\fTJS20-00.prn	9	TOOLERN CK
C:\mac_test\fTJS20-00.prn	10	INFLOW BET.N WEIRS
C:\mac_test\fTJS20-00.prn	11	INFLOW BM TO MELTON
C:\mac_test\fTJS20-00.prn	12	INFLOW
C:\mac_test\fTJS20-00.prn	13	F8+F11+F12 INFLOW
C:\mac_test\fTJS20-00.prn	14	F13 INFLOW
C:\werrmb43\CLIM20-00.PRN	1	EVAPORATION
C:\werrmb43\CLIM20-00.PRN	2	RAINFALL
C:\werrmb43\CLIM20-00.PRN	3	RAIN - MERRIMU
C:\werrmb43\CLIM20-00.PRN	4	RAIN - PYKES
C:\werrmb43\CLIM20-00.PRN	5	RAIN - MELTON
C:\werrmb43\CLIM20-00.PRN	6	EVAP - PYKES
C:\werrmb43\CLIM20-00.PRN	7	EVAP - MERRIMU
C:\werrmb43\CLIM20-00.PRN	8	EVAP - MELTON

```
C:\werrmb43\BELowTun.fmy   1  UPPER LERD ENV

C:\werrmb43\BELowTun.fmy   2  GOODMAN ENV

C:\werrmb43\BELowTun.fmy   3  LOWER LERD ENV

C:\werrmb43\BELowTun.fmy   4  LERD TUNNEL FLOW

C:\werrmb43\BELowTun.fmy   5  GOOD TUNNEL FLOW

Processing inquire parameters: nodemdfiles?


Number of input files:   2

Demand Input File Name           No.  Input Name

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

C:\mac_test\DEMD20-00b.PRN   1  LERD DIV US GOODMAN

C:\mac_test\DEMD20-00b.PRN   2  LERD DIV DS GOODMAN

C:\mac_test\DEMD20-00b.PRN   3  MID RIVER DIV

C:\mac_test\DEMD20-00b.PRN   4  WERRIBEE IRR

C:\mac_test\DEMD20-00b.PRN   5  BM IRRIGATION

C:\mac_test\DEMD20-00b.PRN   6  BM URBAN

C:\mac_test\DEMD20-00b.PRN   7  MELTON URBAN

C:\mac_test\DEMD20-00b.PRN   8  LERD WINTERFILL

C:\mac_test\DEMD20-00b.PRN   9  TOOLERN WINTERFILL

C:\mac_test\DEMD20-00b.PRN  10  LERD D&S

C:\mac_test\DEMD20-00b.PRN  11  MELTON RES D&S

C:\mac_test\DEMD20-00b.PRN  12  WERRIBEE D&S

C:\mac_test\DEMD20-00b.PRN  13  MERRIMU GENERAL

C:\mac_test\DEMD20-00b.PRN  14  LERD GENERAL

C:\mac_test\DEMD20-00b.PRN  15  NOTIONAL DEMAND
```

```
C:\mac_test\DEMD20-00b.PRN  16  CSR FACTORY

C:\mac_test\DEMD20-00b.PRN  17  BMID DEMAND NEXT YR

C:\mac_test\DEMD20-00b.PRN  18  WID DEMAND NEXT YR


C:\werrmb43\DEMACCNT.prn   1  EX BM ACCOUNT

C:\werrmb43\DEMACCNT.prn   2  EX WERR ACCOUNT

C:\werrmb43\DEMACCNT.prn   3  NOTIONAL DEMAND

Processing inquire parameters: quanout (rv)?


Reservoir output options

========================

 %stor         True

 %spil         False

 %targ         False

 %infw         False

 %evap         True

 %awat         False

 %lvls         False


Processing inquire parameters: quanout (ca)?
```

Carrier output options

%flws	True
%capc	True
%loss	True
% 1	True
% 2	False
% 3	False
% 4	False
% 5	False
% 6	False
% 7	True
% 8	False
% 9	False
% 10	False
% 11	False
% 12	False
% 13	False
% 14	False
% 15	False
% 16	False
% 17	False
% 18	True
% 19	False
% 20	False

```
Processing inquire parameters: quanout (sj)?

Stream junction output option
==============================
 %infw         False

Processing inquire parameters: quanout (dc)?

Demand centre output options
============================
 %unrs         True
 %rest         True
 %shrt         True
 %ratn         False
 %lvls         True
 %supp         True

Processing inquire parameters: quanout (al)?

Reservoir output options
========================
 %stor         True
 %spil         False
 %targ         False
```

```
 %infw         False
 %evap         True
 %awat         False
 %lvls         False

Demand centre output options
============================
 %unrs         True
 %rest         True
 %shrt         True
 %ratn         False
 %lvls         True
 %supp         True

Stream junction output option
==============================
 %infw         False

Carrier output options
======================
 %flws         True
 %capc         True
 %loss         True
 %   1         True
 %   2         False
```

```
 %   3         False
 %   4         False
 %   5         False
 %   6         False
 %   7         True
 %   8         False
 %   9         False
 %  10         False
 %  11         False
 %  12         False
 %  13         False
 %  14         False
 %  15         False
 %  16         False
 %  17         False
 %  18         True
 %  19         False
 %  20         False

Processing inquire parameters: nd(3) %flwinput?

Nd(3) %flwinput = WERRIBEE @ BALLAN

Processing inquire parameters: nd(25) %output?
```

```
Output options for node 25
==========================
%unrs   True
%rest   True
%shrt   True
%ratn   False
%lvls   True
%supp   True

Processing inquire parameters: ca(3) %output?

Output options for Carrier 3
============================
%flws   True
%capc   True
%loss   True
%3      False

Processing inquire parameters: ca(3) %capc?

Capacity of Arc 3
=====================
 Jan      99999999
 Feb      99999999
 Mar      99999999
```

```
 Apr      99999999
 May      99999999
 Jun      99999999
 Jul      99999999
 Aug      99999999
 Sep      99999999
 Oct      99999999
 Nov      99999999
 Dec      99999999

Processing inquire parameters: ca(100) %loss?

Ca(100) %loss =           0

Processing change parameters:  scen%logname = tjs1.log

Processing change parameters:  scen%system = t3.sys

Processing inquire parameters: scen%system?

Associated System Files
~~~~~~~~~~~~~~~~~~~~~~
 1  t3.sys

Processing change parameters: quanout (rv) %lvls= true
```

```
Processing change parameters: quanout (rv) %stor= false

Processing change parameters: quanout (ca) %capc= false

Processing change parameters: quanout (ca) %n003= true

Processing change parameters: nd(3) %flwinput= lerderderg u/s div

new flow input for node 2 is: lerderderg u/s div

Processing change parameters: ca(3) %loss= 23

Processing change parameters: ca(3) %capc = 900 200 300 400 500 600 700
800 900 1000 1100 1300

Processing inquire parameters: scen%initres?

Initial Reservoir Volumes:

No    Node No   Init. Vol.

==========================

  1         1       14419

  2        15        5000

  3        22        6735

  4        24         341

  5        37       16098

  6        65           0

  7        82        5537


Processing saveas scenario: tttt.scn

Processing saveas system: t2.sys
```