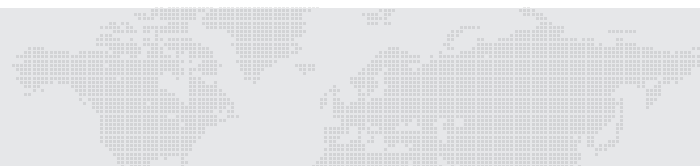


Results Report

REPORT

Post-Processor:
Visualization & Analysis

Version 2018



This publication and the application described in it are furnished under license exclusively to the licensee, for internal use only, and are subject to a confidentiality agreement. They may be used only in accordance with the terms and conditions of that agreement.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic, mechanical, or otherwise, including photocopying, recording, or by any information storage/retrieval system, to any party other than the licensee, without the written permission of Computer Modelling Group.

The information in this publication is believed to be accurate in all respects. However, Computer Modelling Group makes no warranty as to accuracy or suitability, and does not assume responsibility for any consequences resulting from the use thereof. The information contained herein is subject to change without notice.

Copyright © 2018 Computer Modelling Group Ltd.

All rights reserved.

The license management portion of this program is based on:

Reprise License Manager (RLM)

Copyright © 2006-2018, Reprise Software, Inc.

All rights reserved

Results Report, CMG, and *Computer Modelling Group* are registered trademarks of Computer Modelling Group Ltd. All other trademarks are the property of their respective owners.

Computer Modelling Group Ltd.

3710 33 Street N.W.

Calgary, Alberta Canada T2L 2M1

Tel: (403) 531-1300

Fax: (403) 289-8502

E-mail: cmgl@cmgl.ca

Contents

Tutorial Section	1
How to Run Results Report	1
Running from the CMG Technologies Launcher	1
Running from an MS-DOS Prompt	2
Notes on Using the Keyword Input System.....	4
How to Specify Information for a Column	5
How to Create Many Tables with Many Columns	6
How to Modify Table and Column Attributes	8
Obtaining Information for Layers, Groups, Sectors and Special Histories.....	9
How to Produce a Table Comparing Simulation Runs	11
How to Produce a Table for Economic Analysis Packages	12
How to Produce Simulator Well Data from Field Data	13
How to Extract Grid Information from an SR2 File	14
 List of Keywords	 17
BLANKS / NO-BLANKS	17
COLUMN-FOR	19
COMPONENTS	20
DATE.....	21
DATES-FOR	22
FILES.....	23
GRID-FOR.....	25
GROUPS.....	26
LAYERS	28
LEASES.....	30
LINES-PER-PAGE.....	32
LIST-PARAMETERS	33
LIST-PROPERTIES	34
MULTIPLY	35
OUTPUT.....	36
PARAMETERS	37
PRECISION.....	39
PROPERTY-FOR.....	40
SECTORS.....	43
SPECIALS	45

SPREADSHEET	47
TABLE-FOR / TABLE-END.....	48
TABLE-WIDTH	49
TIMES-FOR.....	50
TIMES-FROM	51
TIME.....	52
TITLE1 / TITLE2 / TITLE3	53
UNITS	54
WELL-KEYWORDS	55
WELL-SUMMARY	56
WELLS.....	58
WIDTH.....	60
Appendix	61
List of Parameter Names	61
Index	63
Keyword Definitions	63

Tutorial Section

How to Run Results Report

Before running Results Report, it is first necessary to create a Results Report command file. The command file is a keyword input file, similar in many respects to the simulator keyword input files. By default, these files should have the file name extension .rwd (Results Report Data). This manual describes the keywords that are used to control report output.

Several sample Results Report command files may be found in the template directory \$CMG_HOME/BR/2018.10/TPL/Report (the location of \$CMG_HOME will vary depending on where the CMG software was installed on your system or network). After you have modified one of the template command files to suit your needs, you may run Results Report in one of the following manners.

Running from the CMG Technologies Launcher

Results Report can be run from the CMG Technologies Launcher (“Launcher”). Launcher can be started from the Windows Start menu or from the Launcher icon installed on your desktop.

The Launcher should display an icon for each of the simulators and support tools that have been installed on your system. One of the icons being shown will be for Results Report. In Launcher, browse to the location of your command file and then double-click on the Results Report icon. After a few seconds, a new window should pop up on your workstation, showing the Results Report header, and prompting you for the Results Report command file name. After you have entered a file name, you will be prompted for the name of an output file. Results Report will then read your input command file and produce the output file. The command sequence is shown in the following example:

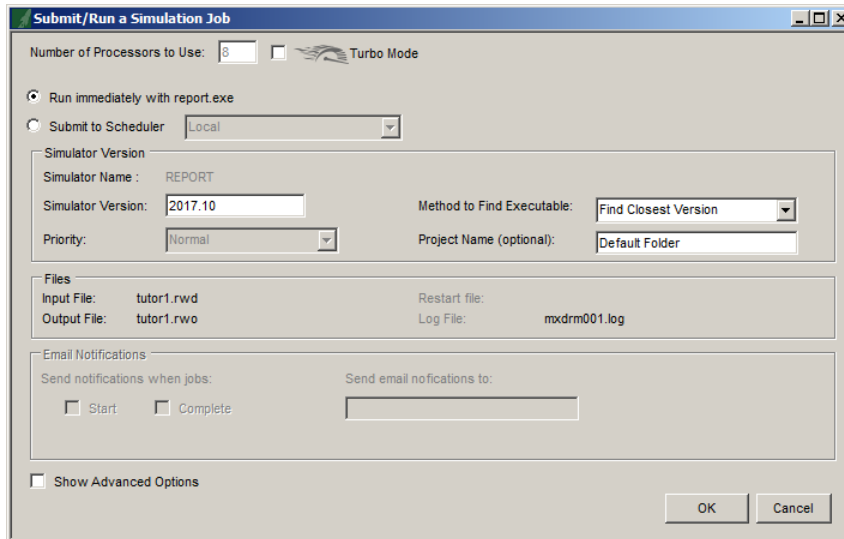
```
RESULTS Report (2018.10 Jun 18 2018)
Enter Report Command filename or 'q' to quit:
stt01c.rwd
Enter output filename, 'q' to quit or <enter> for STANDARD OUT:
stt01c.out
Input file used   : stt01c.rwd
Output file used  : stt01c.out

C:\Program Files
(x86)\CMG\BR\2018.10\TPL\Report>C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2018 Microsoft Corporation. All rights reserved.
```

Note: If the OUTPUT keyword is encountered in the command file, it will take precedence over any output file specified on the command line. For more information, refer to the OUTPUT keyword on page 36.

Another method of running Results Report takes advantage of the list of the files in the current project directory shown in the Launcher window. If you drag and drop a Results Report command file onto the Results Report icon, you will be presented with the **Submit/Run a Simulation Job** dialog box:



You can then select **Run immediately with report.exe**. Unless the Results Report command file specifies an output file, the name of the file will be used to generate an associated output file with a .rwo extension.

Running from an MS-DOS Prompt

The WIN32 version of Results Report may be run directly from the command prompt.

Usage for the WIN32 version of Results Report is one of:

```
report.exe /f command_file /o output_file
    - read commands from command_file and write output to output_file

report.exe /f command_file
    - read commands from command_file and write to standard output

report.exe
    - prompt for names on input command file and output file

report.exe /n < command_file > output_file
    - read commands from standard input and write output to standard output
```

Note: If you are running Results Report from the command line in a terminal not instantiated through the CMG Technologies Launcher, you may get a message stating that the SR2 file

could not be found. To fix this, use an explicit path for the FILES keyword in your Results Report command file. For more information, refer to the FILES keyword on page 23.

Notes on Using the Keyword Input System

As mentioned, before running Results Report, you first have to create an input file, referred to as a command file. By default, these files should have the file name extension of "rwd" (Results Report Data). Results Report command files are read by Results Report using a "keyword input system" like that used by CMG's simulators. The following rules apply to all input files:

1. Comments

Comments are indicated by a pair of asterisks ("**"). The part of any line in the input file that follows the pair of asterisks is ignored by Results Report.

2. Include files

If you wish to include another file within a master input file, the *INCLUDE 'file_name' keyword may be used.

3. Keyword indicator and upper/lower case

A single asterisk immediately before a keyword is optional. Keywords are case insensitive. Thus, the following are all equivalent:

```
*WELLS
WELLS
Wells
wells
```

The keyword indicator is used in all examples and occurrences of keywords in this document for clarity only.

4. Name strings

Names of wells, parameters and files are input surrounded by single quotes ('). Names within single quotes are case sensitive and sensitive to blanks. Thus, 'poola.irf' and 'PoolA.irf' are different, as is 'Oil Rate SC' and 'Oil Rate SC' (there are two blanks between "Rate" and "SC" in the second). Failure to pay attention to these differences can lead to error messages such as "file not found" and "parameter not found". Well names used in Results Report must be exactly the same as those used in the simulator input data set.

How to Specify Information for a Column

The following example shows how to obtain the oil rate at surface conditions for well 'W-1'. This example will produce one table containing two columns. The first column contains the time since the beginning of the simulation, and the second column contains the oil rate at surface conditions for the well 'W-1'.

Example 1 (tutor1.rwd):

```
*TABLE-FOR
  *COLUMN-FOR
    *FILES      'mxdrm005.irf'
    *WELLS      'W-1'
    *PARAMETERS 'Oil Rate SC'
*TABLE-END
```

The pair of keywords *TABLE-FOR and *TABLE-END are used to request a table. At least one pair of these keywords must be present in every report input file. The keyword *COLUMN-FOR is used to request a column, and at least one *COLUMN-FOR keyword must be present between each pair of *TABLE-FOR and *TABLE-END keywords.

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name. In this example, the file name is specified following the keyword *FILES, the origin name is specified using *WELLS followed by a well name and the parameter name is specified using the keyword *PARAMETERS.

The *FILES, *WELLS, and *PARAMETERS can be located on the same line as the *COLUMN-FOR keyword or on the immediately following lines. Thus, the above example could also be written as:

```
*TABLE-FOR
  *COLUMN-FOR *FILES 'mxdrm005.irf' *WELLS 'W-1' *PARAMETERS 'Oil Rate SC'
*TABLE-END
```

How to Create Many Tables with Many Columns

The following examples show how to obtain 3 tables, one per well, such that each table contains 4 columns (a time column plus the 3 parameter columns).

Example 2A:

```
*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'    *WELLS 'W-1' *FILES 'mxdemo.irf'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'     *WELLS 'W-1' *FILES 'mxdemo.irf'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'   *WELLS 'W-1' *FILES 'mxdemo.irf'
*TABLE-END

*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'    *WELLS 'W-2' *FILES 'mxdemo.irf'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'     *WELLS 'W-2' *FILES 'mxdemo.irf'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'   *WELLS 'W-2' *FILES 'mxdemo.irf'
*TABLE-END

*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'    *WELLS 'W-3' *FILES 'mxdemo.irf'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'     *WELLS 'W-3' *FILES 'mxdemo.irf'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'   *WELLS 'W-3' *FILES 'mxdemo.irf'
*TABLE-END
```

Clearly there is too much repetitive typing here. However, the same report can be obtained using a much more compact notation. As all the information is obtained from the same file, you can move the file name outside the TABLE-FOR TABLE-END pairs, and specify it only once.

Information that is specified outside TABLE-FOR or TABLE-END pairs, is said to be in "report scope" and will apply to all subsequent tables in the report.

With this modification Example 2A can be rewritten as follows:

Example 2B:

```
*FILES 'mxdemo.irf'
*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'    *WELLS 'W-1'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'     *WELLS 'W-1'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'   *WELLS 'W-1'
*TABLE-END

*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'    *WELLS 'W-2'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'     *WELLS 'W-2'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'   *WELLS 'W-2'
*TABLE-END

*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'    *WELLS 'W-3'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'     *WELLS 'W-3'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'   *WELLS 'W-3'
*TABLE-END
```

Furthermore, as all the columns in each table correspond to the same well, you can specify the well name following *TABLE-FOR and preceding the first *COLUMN-FOR keyword.

Information that is specified between a *TABLE-FOR and the first *COLUMN-FOR in a table is said to be in "table scope" and will apply to all the columns in the table. Information that follows a *COLUMN-FOR keyword and preceding the next *COLUMN-FOR is said to be in "column scope" and will apply only to that column.

In this way Example 2B can be modified as follows:

Example 2C:

```
*FILES 'mxdrm005.irf'
*TABLE-FOR *WELLS 'W-1'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'
*TABLE-END
*TABLE-FOR *WELLS 'W-2'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'
*TABLE-END
*TABLE-FOR *WELLS 'W-3'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC'
  *COLUMN-FOR *PARAMETERS 'Water Rate SC'
*TABLE-END
```

As no column attributes are being modified, we can take advantage of the fact that more than one parameter name can follow the *PARAMETERS keyword, and more than one well name can follow the *WELLS keyword. Thus, we can further compact this Results Report input data file as follows:

Example 2D (tutor2.rwd):

```
*FILES 'mxdrm005.irf'

*TABLE-FOR *WELLS 'W-1' 'W-2' 'W-3'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC' 'Oil Rate SC' 'Water Rate SC'
*TABLE-END
```

In this example, *WELLS 'W-1' 'W-2' 'W-3' was used to generate three tables, and similarly *PARAMETERS followed by 3 parameter names was used to generate four columns for each table (time is the first column).

How to Modify Table and Column Attributes

The examples in the previous section demonstrate that the same tables can be obtained using a variety of notations. This flexibility allows the user to modify the attribute of a certain table or column. The compact notation can be used when all the attributes for the tables and columns are the same. The following example will show how less compact notations are used to modify the attributes of a given table or column.

Assume for example, that we have performed two different runs of mxdemo.dat (for two different prediction runs), and we would like the information for well 'W-3' to be taken from the second file. Then, instead of the compact notation shown in Example 2D, we will use the following:

Example 3A:

```
*FILES 'mxdemo1.irf'
*TABLE-FOR *WELLS 'W-1' 'W-2'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC' 'Oil Rate SC' 'Water Rate SC'
*TABLE-END
*FILES 'mxdemo2.irf'
*TABLE-FOR *WELLS 'W-3'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC' 'Oil Rate SC' 'Water Rate SC'
*TABLE-END
```

Next assume that we would like the column specifying the gas rates to be printed with higher precision, while the other columns are printed using the default precision. This will also require a less compact notation.

Example 3B (tutor3.rwd):

```
*FILES      'mxdrm005.irf'

*TABLE-FOR  *WELLS 'W-1' 'W-2' 'W-3'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC' *PRECISION 8 *WIDTH 16
  *COLUMN-FOR *PARAMETERS 'Oil Rate SC' 'Water Rate SC'
*TABLE-END
```

Individual columns can be multiplied by some constant (*MULTIPLY), printed in different units (*UNITS), printed in different precisions (*PRECISION), and be printed with different column widths (*WIDTH).

For details on these keywords, see the keyword reference section.

These keywords can be used in "table scope" (immediately after the *TABLE-FOR keyword) if they are to apply to all columns in the table.

They can be used in "report scope" (outside of a *TABLE-FOR *TABLE-END pair) if they are to apply to all columns in all subsequent tables.

Obtaining Information for Layers, Groups, Sectors and Special Histories

The above examples have shown how to obtain information for wells.

We will now show you how to obtain columns of information from layers, groups and sectors.

The simulators can output information to the SR2 file (*.irf) for well layers, well groups, and sectors of the reservoir. The amount of information written to the SR2 file is controlled by keywords input to the simulator. You will need to look at the *WSRF and *OUTSRF keywords in the simulator user guides to ensure that the information that you want is written to the SR2 files. Well layer information is not written to the SR2 file unless specifically requested in the simulator data set.

This example shows how to obtain a table of information from a well group (the IMEX default group for all producing wells), a table of information from a sector and from "special histories", and a table of information from three layers of a well.

Example 4A (tutor4.rwd):

```
FILE 'mxdrm005.irf'

*TABLE-FOR
  *COLUMN-FOR *PARAMETERS 'Cumulative Oil SC' *GROUPS 'Default-Field-PRO'
*TABLE-END

*TABLE-FOR
  *COLUMN-FOR *SECTORS 'Entire Field' '1' *PARAMETERS 'Ave Pres HC
POVO SCTR'
  *COLUMN-FOR *SECTORS 'Entire Field' '1' *PARAMETERS 'Oil Ave Sat
SCTR'
  *COLUMN-FOR *SPECIALS 'PRES Average Reservoir Pressure.'
  *COLUMN-FOR *SPECIALS 'WINFLUX Total Water Influx.'
*TABLE-END

** *TABLE-FOR *PARAMETERS 'Cumulative Gas SC'
**   *COLUMN-FOR *LAYERS 'W-11{9,7,3}' 'W-11{9,7,5}' 'W-11{9,7,6}'
** *TABLE-END
```

Well group names are defined in the well and recurrent data section of the simulator input data set. The group names used in the report writer input data must match exactly the names used in the simulator input. Similarly, sector names must exactly match those used in the simulator input. In the above example, 'Entire Field' is produced by IMEX in all cases (note the two spaces between "Entire" and "Field"), while '1' is defined in the input data set.

Special history names differ from simulator to simulator. The best way of obtaining the exact string for the special history name is to look in the "irf" file of the SR2 file pair. Open the "irf" file in read-only mode in an editor. Search for the SR2 keyword "SPHIST-NAMES", and copy the entries on the immediately following lines.

Layer names are formed by combining the well name and the I, J, K indices of the layer completion block. Thus, the layer of well "W-11" that is completed in block I=9, J=7, K=3, is specified by 'W-11{9,7,3}'. Do not put any spaces between the well name and the "{" or between the IJK indices.

How to Produce a Table Comparing Simulation Runs

The following example compares two different simulation runs and field history. A table is produced for each production well (using the *ALL-PRODUCERS subkeyword of *WELLS) with side by side columns from the two runs and field history for oil rate, GOR and water cut (a total of nine parameter columns plus time and date columns).

Notice the use of the *FILES keyword following the *COLUMN-FOR keyword. In this example, multiple file numbers are used. A "1" following the *FILES keyword indicates the first file defined in the input file, a "2" the second file defined, etc. Using "*FILES 4" when only three files have been defined will generate an error message.

Example 5A:

```
*FILES 'poola1.irf'      ** Open the first simulation output file
*FILES 'poola2.irf'      ** Open the second simulation output file
*FILES FHF 'poola.fhf'   ** Open the field history file
*DATE ON                 ** Add a column for the date
** Produce a table for each production well
*TABLE-FOR *WELLS *ALL-PRODUCERS
  *COLUMN-FOR *FILES 1 2 3 *PARAMETERS 'Oil Rate SC'
  *COLUMN-FOR *FILES 1 2 3 *PARAMETERS 'Gas Oil Ratio SC'
  *COLUMN-FOR *FILES 1 2 3 *PARAMETERS 'Water Cut SC - %'
*TABLE-END
```

How to Produce a Table for Economic Analysis Packages

Often the results of the prediction portion of a simulation run will be used as input to an economic analysis package. In this case, a single table containing oil rates for all producers may be appropriate. In addition, the economic package requires that the oil rates be reported at regular intervals (i.e. monthly, quarterly, or yearly). The package also requires the average oil rate over the interval - not the instantaneous rate at a point in time. Monthly, quarterly, and yearly oil, water, gas, solvent, and polymer rates, reported on the last day of each period for the period, are available from Results Report. See Appendix A for a list of the parameter names.

When importing a table into an economics package or spreadsheet, partially empty columns may cause a problem. The *NO-BLANKS keyword causes Results Report to write a "0.0" into any column where the well is not yet defined or open for the time/date of the row.

The following example shows how to create tables with monthly average oil and gas rates with a column for each producer well.

Example 6A (tutor6.rwd):

```
*FILE 'mxdrm005.irf'
*LINE-PER-PAGE 5000      ** Don't have any page breaks in a table.
*TABLE-WIDTH 300        ** Produce a wide table if many producers,
*NO-BLANKS               ** Always have a value in every column,
*PRECISION 4             ** with four significant digits.
*TIME OFF                ** The tables will have no time column,
*DATE ON                 ** but will have a date column
*TABLE-FOR *PARAMETERS 'Oil Rate SC - Monthly'
  *COLUMN-FOR *WELLS *ALL-PRODUCERS
*TABLE-END
*TABLE-FOR *PARAMETERS 'Gas Rate SC - Monthly'
  *COLUMN-FOR *WELLS *ALL-PRODUCERS
*TABLE-END
```

If you are using a spreadsheet to do your economic analysis, or if your economic analysis package is having trouble reading the file output from Results Report, you may wish to try outputting the file in a "tab separated" format. To do this, put a *SPREADSHEET keyword before the first *TABLE-FOR keyword. The *LINE-PER-PAGE, *TABLE-WIDTH and *NO-BLANKS keywords are not needed. The following example shows the input data after these changes:

Example 6B (tutor6b.rwd):

```
*FILE 'mxdrm005.irf'
*SPREADSHEET
*PRECISION 4             ** Four significant digits.
*TIME OFF                ** The tables will have no time column,
*DATE ON                 ** but will have a date column
*TABLE-FOR *PARAMETERS 'Oil Rate SC - Monthly'
  *COLUMN-FOR *WELLS *ALL-PRODUCERS
*TABLE-END
*TABLE-FOR *PARAMETERS 'Gas Rate SC - Monthly'
  *COLUMN-FOR *WELLS *ALL-PRODUCERS
*TABLE-END
```

How to Produce Simulator Well Data from Field Data

When beginning a history match simulation, historical well production data needs to be converted to simulator input format. If the well data is in one of the supported Results Report input formats (see Appendix A Field History File Formats and Appendix B Production Analyst Load Format in the *Results Graph User Guide*), then Results Report can convert the data to simulator input format.

The following example shows how to convert well data in CMG's field history file (fhf) format to simulator input format. The data in 'sample.fhf' is monthly production totals. Three production wells exist. To speed up the simulation, the simulator well changes are to occur on a quarterly basis - so the parameter 'Oil Rate SC - Quarterly' is used. The well named 'INJECTOR' is a water injector.

Example 7A (tutor7.rwd):

```
*FILES *FHF 'sample.fhf'
*UNITS *FIELD
*WELL-KEYWORDS
*TABLE-FOR
  *COLUMN-FOR  *PARAMETERS 'Oil Rate SC - Quarterly'
                *WELLS 'PROD 1' 'PROD 2' 'PROD 3'
  *COLUMN-FOR  *PARAMETERS 'Water Rate SC - Quarterly'
                *WELLS 'INJECTOR'
*TABLE-END
```

The output will produce a comment line giving an *OPERATE keyword for each well:

```
** PRODUCER 'PROD 1' or INJECTOR 'PROD 1'
** OPERATE MAX STO          100.000 ** Oil Rate SC - Quarterly
```

The output will produce one or more *ALTER keywords (depending on the number of wells and the *TABLE-WIDTH), followed by a simulator DATE keyword:

```
DATE      1993 01 02
*ALTER          'PROD 1'          'PROD 2'          'PROD 3'          'INJECTOR'
                100.0             100.0             100.0             100.0

*DATE      1993 04 02
*ALTER          'PROD 1'          'PROD 2'          'PROD 3'          'INJECTOR'
                100.0             100.0             100.0             100.0
```

How to Extract Grid Information from an SR2 File

At times, you may wish to extract grid properties from one simulation run and use them as initial conditions for another run. For example, you may run a history match with IMEX, then wish to run STARS to do a prediction case using a thermal process.

If you will be using the same grid for the second simulation study, the process is straightforward. The `*PROPERTY-FOR` command will extract grid property information from an SR2 file. For example, to extract pressure, oil saturation and water saturation at 2586.0 days, use the following:

```
*FILE          'watfld1.irf'
*OUTPUT        'pres.inc'
*PROPERTY-FOR  'PRESS' 2586.0
*OUTPUT        'so.inc'
*PROPERTY-FOR  'SO' 2586.0
*OUTPUT        'sw.inc'
*PROPERTY-FOR  'SW' 2586.0
```

Each of the three files will contain the indicated property in simulation input format.

If grid property information is required for many properties, the `*ALL-PROPERTIES` and `*ALL-TIMES` keywords can be used. The `*ALL-PROPERTIES` keyword option will extract grid information for all available properties. The `*ALL-TIMES` keyword option will extract grid information for all available times. Consider the following examples:

```
*FILE          'watfld1.irf'
*OUTPUT        'pres_all.inc'
*PROPERTY-FOR  'PRESS' *ALL-TIMES
*OUTPUT        'all_prop_t0.inc'
*PROPERTY-FOR  *ALL-PROPERTIES 2586.0
*OUTPUT        'all_prop.inc'
*PROPERTY-FOR  *ALL-PROPERTIES *ALL-TIMES
```

The output file 'pres_all.inc' will contain pressure data for each time in the SR2 file. The output file 'all_prop_t0.inc' will contain data for each property at the time closest to 2586.0 days. The output file 'all_prop.inc' will contain data for all properties at all available times.

If you are planning on changing the grid, more work is required. Each 'K' layer of the original grid may be extracted as a Mesh Map (a grid of data points that may be used in Builder like a contour map). Suppose, in the above example, we wish to extract the oil saturation for simulation layers 1 and 2:

```
*FILE          'watfld1.irf'
*OUTPUT        'sol.dig'
*PROPERTY-FOR  'SO' 2586.0 *MESH LAYER 1
*OUTPUT        'so2.dig'
*PROPERTY-FOR  'SO' 2586.0 *MESH LAYER 2
```

As the SR2 files, produced by simulator versions prior to 98.00, do not contain information on how the simulation grid has been positioned relative to the origin of any structure contour maps, it is necessary to add this information to the output mesh map files. In each mesh map file produced by Results Report, you will see three lines as follows:

```
RESULTS ROTATION 0
RESULTS XOFFSET 0
RESULTS YOFFSET 0
```

If the history match simulation data file was produced with Builder, you should find three similar lines in your data file. In each mesh map file, change the rotation value of "0" to the rotation value found in your simulator data and similarly for the X offset and Y offset values. You should now be able to open the mesh map files in Builder.

List of Keywords

BLANKS / NO-BLANKS

***BLANKS, *NO-BLANKS**

PURPOSE:

*BLANKS specifies that blanks be used in individual columns to indicate that the value for the column is not defined for the time specified for the row. *NO-BLANKS specifies that a "0.0" be used instead of blanks when the value for the column is not known at the time of the row.

FORMAT:

*BLANKS
*NO-BLANKS

DEFAULTS:

Optional keyword. The default value is *BLANKS.

CONDITIONS:

This keyword, if present, must be in "report scope" (it must be outside any *TABLE-FOR *TABLE-END pair). It can appear several times in the Results Report input file.

EXPLANATION:

Often a Results Report output will be imported into a spreadsheet. Some spreadsheets, particularly older versions, have problems correctly reading in reports unless there is a placeholder in each column. Using the *NO-BLANKS keyword causes Results Report to insert a "0.0" in each column.

For example, suppose you are producing a report on the cumulative oil production from three wells. In the report, time is in the first column. "Well 1" is in the second column and starts having values at 30 days. "Well 2" is in the third column but doesn't start having values until 90 days. (It is shut in for the first two months of the simulation.) "Well 3" is in the fourth column and starts having values at 30 days.

If you use *BLANKS (the default) your report will look something like this:

TIME (DAYS)	Well 1 (m3)	Well 2 (m3)	Well 3 (m3)
30.0	2670.0		2345.1
60.0	4920.3		4134.8
90.0	6840.9	1894.5	5698.0
.	.	.	.
.	.	.	.
.	.	.	.

If you put the *NO-BLANKS keyword before the *TABLE-FOR keyword for the table, you will get the following report:

TIME (DAYS)	Well 1 (m3)	Well 2 (m3)	Well 3 (m3)
30.0	2670.0	0.0	2345.1
60.0	4920.3	0.0	4134.8
90.0	6840.9	1894.5	5698.0
.	.	.	.
.	.	.	.
.	.	.	.

COLUMN-FOR

***COLUMN-FOR**

PURPOSE:

*COLUMN-FOR is used to specify the information for a column or a group of columns.

FORMAT:

*COLUMN-FOR

DEFAULTS:

Required keyword.

CONDITIONS:

At least one *COLUMN-FOR must appear between each pair of *TABLE-FOR and *TABLE-END keywords, i.e. each table must contain at least one column.

EXPLANATION:

This keyword is used to request the output of a column or a group of columns. The specification of the information for these columns should follow the keyword *COLUMN-FOR.

All the information (keywords) that follows a *COLUMN-FOR and which precede a subsequent *COLUMN-FOR apply to that column and is said to be in "column-scope". This information determines whether one column or a group of columns will be printed.

For example:

```
*TABLE-FOR *FILES 'run1.irf' *WELLS 'w1'  
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'  
*TABLE-END
```

will produce one table containing one column while

```
*TABLE-FOR *FILES 'run1.irf'  
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC' *WELLS 'w1' 'w2' 'w3'  
*TABLE-END
```

will produce one table containing three columns.

COMPONENTS

***COMPONENTS**

PURPOSE:

*COMPONENTS is used to specify the component names for which information will be printed in tables and columns.

FORMAT:

*COMPONENTS 'comp name'

DEFINITION:

'comp name'

One or more component names used in the simulation runs.

DEFAULTS:

Optional keyword. Required when 'parameter name (COMPONENTS)' is used.

CONDITIONS:

*COMPONENTS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one component name. When it appears in "table scope" or "column scope", it can be followed by more than one component names. Thus, it can be used to generate several tables or columns.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name. For parameters that depend on a component, the keyword *COMPONENTS, followed by one or more component names, is used to specify the component name.

When *COMPONENTS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one component name. This component name will be in effect for all subsequent tables and columns.

When *COMPONENTS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more component names – it will apply only to that table or group of tables. Thus, it will override any previous *COMPONENTS in "global scope".

When *COMPONENTS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more component names – it will apply only to that column or groups of columns. Thus, it will override any previous *COMPONENTS in "global scope" or "table scope".

DATE

***DATE**

PURPOSE:

*DATE controls the printing of a column containing the dates.

FORMAT:

*DATE (*ON)
 (*OFF)

DEFINITION:

*ON

Print a column containing the dates.

*OFF

Do not print a column containing the dates.

DEFAULTS:

Optional keyword. Default *DATE *OFF

CONDITIONS:

This keyword, if present, must be in "report scope"; i.e. it must be outside any TABLE-FOR TABLE-END pair. It can appear several times in the Results Report input file. The *DATES keyword has no effect when the *WELLS-KEYWORD keyword is used.

EXPLANATION:

This keyword controls the printing of a column containing the dates corresponding to the simulation times. These dates are obtained from the initial simulation date and the times in the time array (See *TIME for an explanation of how the times in the time array are chosen).

If both keywords *TIME *ON and *DATE *ON are used, the time and date columns will be the first and the second columns respectively in all the pages corresponding to a given table.

The data columns will follow these two columns.

If only *DATE *ON is specified, the date column will be the first column in the page followed by the data columns.

DATES-FOR

***DATES-FOR**

PURPOSE:

*DATES-FOR is used to specify the list of dates for which information will be printed.

FORMAT:

*DATES-FOR 'date-string1' 'date-string2' ...

DEFINITION:

'date-string'

One or more string dates. One of the following formats must be used:

'yyyy mm dd' or 'yyyy:mm:dd' or 'yyyy/mm/dd'

DEFAULTS:

Optional keyword.

If neither *TIMES-FOR, *DATES-FOR, nor *TIMES-FROM are used, the program will internally generate a default list of times and dates. (As the information for the different columns can be printed to the simulation result files at different times, the program takes the default times for the times array from the times in the column with the minimum time, and the column with the maximum time.)

CONDITIONS:

*DATES-FOR must appear in global scope, i.e. outside any pairs of *TABLE-FOR

*TABLE-END keywords. The list of string dates must be monotonically increasing in time and must follow exactly one of the allowed formats. yyyy must be greater than 1901.

EXPLANATION:

This keyword allows the user to specify a list of dates for which information will be printed.

*DATES-FOR can only appear outside *TABLE-FOR *TABLE-END pairs and applies to all tables that follow. The list of dates can be overridden by any subsequent use of *DATES-FOR, *TIMES-FOR, or *TIMES-FROM.

For example:

```
*DATES-FOR
'1990:01:01' '1990:04:01' '1990:07:01' '1990:10:01'
'1991:01:01' '1991:04:01' '1991:07:01' '1991:10:01'
'1992:01:01' '1992:04:01' '1992:07:01' '1992:10:01'
'1993:01:01' '1993:04:01' '1993:07:01' '1993:10:01'
'1994:01:01' '1994:04:01' '1994:07:01' '1994:10:01'
```

will produce a report containing 20 rows per table – one row for each of the specified dates.

FILES

***FILES**

PURPOSE:

*FILES is used to specify the simulation results file or field history file from which the information for a column will be obtained.

FORMAT:

*FILES (*SR2 | *FHF | *PAF) ('file name'| file_int)

DEFINITION:

***SR2**

*SR2 indicates that the file is a simulation results file.

***FHF**

*FHF indicates that the file is a field history file. (See Appendix A Field History File Formats and Appendix B Production Analyst Load Format in the *Results Graph User Guide*.)

***PAF**

*PAF indicates that the file is in PA Load format. (See Appendix A Field History File Formats and Appendix B Production Analyst Load Format in the *Results Graph User Guide*.)

'file name'

The file name of a simulation results file or the field history file.

'file_int'

An integer used to refer to a previously defined file name.

DEFAULTS:

Required keyword. There is no default. The keyword *SR2, *FHF, or *PAF is optional, with the default *SR2.

CONDITIONS:

*FILES can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it can be followed by only one file name. When it appears in "table scope" or "column scope", it can be followed by more than one file name. In such a case, it will generate several tables or columns.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name.

The keyword *FILES, followed by one or more file names, is used to specify the file name. *FILES can be followed by the keyword *SR2 and one or more simulation results files in quotes or by the keyword *FHF followed by one or more field history files in quotes.

When *FILES appears in "report scope", it can be followed by only one file name. This file name will be in effect for all subsequent tables and columns.

When *FILES appears in "table scope", it can be followed by one or more file names – it will apply only to that table or group of tables. Thus, it will override any previous *FILES in "global scope".

When *FILES appears in "column scope", it can be followed by one or more file names – it will apply only to that column or groups of columns. Thus, it will override any previous *FILES in "global scope" or "table scope".

*FILES can be followed by one or more integers, which refer to previously specified file names. For example:

```
*FILE 'run1.irf'
*FILE 'run2.irf'
*FILE 'run3.irf'
*TABLE-FOR
    *FILES 1 3
    *COLUMN-FOR *PARAMETERS 'Gas Rate SC' *WELLS 'w1'
*TABLE-END
```

can be used to generate two tables, one for file 'run1.irf' and the other for 'run3.irf'.

GRID-FOR

***GRID-FOR**

PURPOSE:

*GRID-FOR specifies the printing of a simulation grid in simulator input file (dataset) format. This keyword is primarily intended for use in debugging by CMG staff but may be of some use to simulation users.

FORMAT:

*GRID-FOR time (*VERBOSE (*ON | *OFF)) (*ONETOP)

DEFINITION:

time

Indicates which grid to print if the grid changes over the simulation.

*VERBOSE

Controls the amount of comments to print.

*ONETOP

Forces the printing of the simulator dataset *DTOP keyword (depths to tops of the upper most layer of grid blocks) rather than the simulator dataset *PAYDEPTH keyword. Use of the *ONETOP keyword for a grid with separated layers will result in a *GRID-FOR printout that is different than the grid in the input file.

DEFAULTS:

Optional keyword. The default for time is 0.0 days. The grid will be obtained from the current "global scope" file, which must be an SR2 file.

CONDITIONS:

This keyword, if present, must be in global scope. The printing routine will ignore the *TABLE-WIDTH and *LINE-PER-PAGE keywords.

EXPLANATION:

Results Report will obtain and print the simulation grid from the current "global scope" file.

GROUPS

***GROUPS**

PURPOSE:

*GROUPS is used to specify the group names for which information will be printed in tables and columns.

FORMAT:

*GROUPS ('group name')
 (*ALL)

DEFINITION:

'group name'

One or more group names used in the simulation runs.

DEFAULTS:

Optional keyword. However, one of the following keywords:

*WELLS, *LAYERS, *GROUPS, *SECTORS, *LEASES, *SPECIALS

is required to specify an origin name.

CONDITIONS:

*GROUPS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one group name. When it appears in "table scope" or "column scope", it can be followed by more than one group names. Thus, it can be used to generate several tables or columns.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name.

The keyword *GROUPS, followed by one or more group names, is one of the keywords (the others are *WELLS, *LAYERS, *SECTORS, *LEASES and *SPECIALS) that can be used to specify an origin name. The group name must be in quotes and must be spelled exactly as in the simulation run.

When *GROUPS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one group name. This group name will be in effect for all subsequent tables and columns.

When *GROUPS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more group names, and it will apply only to that table or group of tables. Thus, it will override any previous *GROUPS which appeared in "global scope". It will also override any other keyword that was used to specify an origin name.

When *GROUPS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more group names, and it will apply only to that column or group of columns. Thus it will override any previous *GROUPS in "global scope" or "table scope". It will also override any other keyword that was used to specify an origin name. When *ALL follows *GROUPS in table (column) scope, the program will print one table (column) for each of the groups used in the simulation runs. Thus, you can easily obtain the same information for all the groups without having to type all the group names.

LAYERS

***LAYERS**

PURPOSE:

*LAYERS is used to specify the layer names for which information will be printed in tables and columns.

FORMAT:

*LAYERS ('layer name')
 (*ALL)

DEFINITION:

'layer name'

One or more layer names used in the simulation runs. The layer name is well name{i,j,k} where well name is one of the well names specified in the simulation file and i,j,k are the layers indices.

DEFAULTS:

Optional keyword. However, one of the following keywords: *WELLS, *LAYERS, *GROUPS, *SECTORS, *LEASES, *SPECIALS is required to specify an origin name.

CONDITIONS:

*LAYERS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one layer name. When it appears in "table scope" or "column scope", it can be followed by more than one layer names. Thus, it can be used to generate several tables or columns.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name. The keyword *LAYERS, followed by one or more layer name, is one of the keywords (the others are *WELLS, *GROUPS, *SECTORS, *LEASES and *SPECIALS) that can be used to specify an origin name. The layer name must be in quotes and is given by well name{i,j,k} where well name is one of the well names specified in the simulation file and i,j,k are the layer indices. For example, the layer name for well W-1 perforated in layer 10,7,3 is W-1{10,7,3}.

When *LAYERS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one layer name. This layer name will be in effect for all subsequent tables and columns.

When *LAYERS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more layer names and it will apply only to that table or group of tables. Thus, it will override any previous *LAYERS that appeared in "global scope". It will also override any other keyword which is used to specify an origin name.

When *LAYERS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more layer names, and it will apply only to that column or group of columns. Thus it will override any previous *LAYERS in "global scope" or "table scope". It will also override any other keyword that was used to specify an origin name. When *ALL follows *LAYERS in table (column) scope, the program will print one table (column) for each of the layers used in the simulation runs. Thus, you can easily obtain the same information for all the layers without having to type all the layer names.

LEASES

***LEASES**

PURPOSE:

*LEASES is used to specify the lease line names for which information will be printed in tables and columns.

NOTE:

The simulators do not output lease-line-information to the simulation results file yet. This keyword is available once the simulators can output this information.

FORMAT:

```
*LEASES      ( 'lease-line-name' )  
              ( *ALL )
```

DEFINITION:

'lease-line-name'

One or more lease line names used in the simulation runs.

DEFAULTS:

Optional keyword. However, one of the following keywords: *WELLS, *LAYERS, *GROUPS, *SECTORS, *LEASES, *SPECIALS is required to specify an origin name.

CONDITIONS:

*LEASES can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one lease line name. When it appears in "table scope" or "column scope", it can be followed by more than one lease line name. Thus it can be used to generate several tables or columns.

EXPLANATION:

In order to specify the information for one column you must specify a file name, an origin name and a parameter name. The keyword *LEASES, followed by one or more lease line name, is one of the keywords (the others are *WELLS, *GROUPS, *LAYERS, *SECTORS and *SPECIALS) that can be used to specify a lease line name. The lease line name must be in quotes and must be spelled exactly as in the simulation run.

When *LEASES appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one lease line name. This lease line name will be in effect for all subsequent tables and columns.

When *LEASES appears in "table scope", i.e. following *TABLE-FOR and proceeding *COLUMN-FOR, it can be followed by one or more lease line names, and it will apply only to that table or group of tables. Thus, it will override any previous *LEASES that appear in "report scope".

When *LEASES appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more lease line names, and it will apply only to that column or group of columns. Thus, it will override any previous *LEASES in "global scope" or "table scope".

LINES-PER-PAGE

***LINES-PER-PAGE**

PURPOSE:

*LINES-PER-PAGE specifies the maximum number of lines (including titles) to be printed in a page.

FORMAT:

*LINES-PER-PAGE num_lines

DEFINITION:

num_lines

An integer specifying the maximum number of lines in a page.

DEFAULTS:

Optional keyword. The default value is 66.

CONDITIONS:

This keyword, if present, must be in "report scope", i.e. it must be outside any TABLE-FOR TABLE-END pair. It can appear several times in the Results Report input file.

EXPLANATION:

This keyword controls the total number of lines to be printed in a page. This number includes all titles and empty lines. Each table will be printed in a new page, and after num_lines are printed a new page will be created. All table titles and column titles will be printed in each of the pages.

*LINES-PER-PAGE can appear several times in the data set. Each value will override the previous value and will be in effect for all subsequent tables. In the absence of *LINES-PER-PAGE, a default value of 66 lines per page is used. In order to obtain all the rows in one page (prevent breaking the tables into pages), a large value of num_lines should be used.

The minimum number of lines per page allowed is the number required to print all the titles + 1 row of data per page.

LIST-PARAMETERS

***LIST-PARAMETERS**

PURPOSE:

*LIST-PARAMETERS causes Results Report to list all the allowed parameters and origins for columns in a table (for first opened file).

FORMAT:

*LIST-PARAMETERS

DEFINITION / DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword, if present, must be in "report scope", i.e. it must be outside any TABLE-FOR TABLE-END pair. It can appear several times in the Results Report input file.

EXPLANATION:

A list is sent to the current output file with lines in one of the following forms:

```
*PARAMETERS 'Cumulative Gas SC' *GROUPS 'TOT-PROD'  
*PARAMETERS 'Cumulative Gas Moles(OXYGEN) RC' *WELLS 'INJECTOR'  
*PARAMETERS 'Gas Inje Cum SCTR' *SECTORS 'Entire Field'  
*SPECIALS 'TEMP 1,1,10'
```

Here, the *PARAMETER or the *SPECIALS keyword is first, and the origin keyword (if needed) is next.

Each of the output lines may be cut and pasted after a *COLUMN-FOR keyword to produce a column in a table in a subsequent Results Report run.

LIST-PROPERTIES

***LIST-PROPERTIES**

PURPOSE:

*LIST-PROPERTIES causes Results Report to list all the allowed grid properties and the times for which they are available (for first opened file).

FORMAT:

*LIST-PROPERTIES

DEFINITION / DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword, if present, must be in "report scope", i.e. it must be outside any TABLE-FOR TABLE-END pair. It can appear several times in the Results Report input file.

EXPLANATION:

A list is sent to the current output file with lines in one of the following forms:

```
*PROPERTY-FOR 'Gas Saturation'          358.4926 *MATRIX
*PROPERTY-FOR 'Gas Mole Fraction(HEVY OIL)' 10 *MATRIX
```

These lines can be cut and pasted into a second command file and used to output the desired grid property. See the *PROPERTY-FOR keyword for more details on output format options for grid properties.

MULTIPLY

***MULTIPLY**

PURPOSE:

*MULTIPLY specifies a multiplication factor to apply to data values in columns

FORMAT:

*MULTIPLY mult-factor

DEFINITION:

mult-factor

A factor that will multiply the data values of the column.

DEFAULTS:

Optional keyword. The default value is 1.0

CONDITIONS:

This keyword, if present, can be either in "report scope", i.e. outside any TABLE-FOR TABLE-END pair, or in "column scope", i.e. following a *COLUMN-FOR keyword. It can appear several times in the Results Report input file.

EXPLANATION:

This keyword specifies a multiplicative factor that will be applied to the data values of a column.

When *MULTIPLY appears in "report scope", i.e. outside *TABLE-FOR and *TABLE-END pairs, it will apply to all the columns in all the following tables.

When *MULTIPLY appears in "column scope", i.e. following *COLUMN-FOR, it will apply to the data values of that particular column only. In this way, different columns can be multiplied by different factors.

When *MULTIPLY appears more than once, its effect is not accumulative. The multiplication factor used is the one that is in scope for the corresponding column.

OUTPUT

***OUTPUT**

PURPOSE:

*OUTPUT indicates the name of a new output file.

FORMAT:

*OUTPUT 'file name'

DEFINITION:

'file name'

A file name within quotes.

DEFAULTS:

Optional keyword. Prior to encountering this keyword, all output is to the output file name specified when starting Results Report (which is input at the prompt, or specified with the -o option).

CONDITIONS:

This keyword, if present, must be in "report scope", i.e. it must be outside any *TABLE-FOR *TABLE-END pair. It can appear several times in the Results Report input file.

EXPLANATION:

This keyword is used to set the file name into which the report will be written (the Results Report output file). The keyword *OUTPUT must appear outside any *TABLE-FOR and *TABLE-END pair and will apply to all the tables which follow. When *OUTPUT is encountered, a file is opened and all the Results Report output is printed to that file. If another output keyword is encountered, the existing file will be closed and the output will be directed to the new file.

If the file corresponding to the file name following *OUTPUT cannot be opened, the output will be printed to the Results Report output file which is in effect, i.e. the standard out or file that was previously specified using *OUTPUT.

WARNING: If the file name following *OUTPUT corresponds to an existing file, or to a file which was created previously using another *OUTPUT keyword, the file will be overwritten and thus its contents will be lost.

PARAMETERS

***PARAMETERS**

PURPOSE:

*PARAMETERS is used to specify the parameter names for which information will be printed in tables and columns.

FORMAT:

*PARAMETERS 'parameter name'

or

*PARAMETERS 'parameter name(component name)'

or

*PARAMETERS 'parameter name(COMPONENT)'

DEFINITION:

parameter name

One or more parameter name(s) obtained from the parameter list, given in Appendix A.

component name

A component name used in the simulation runs.

DEFAULTS:

Required keyword. There is no default.

CONDITIONS:

*PARAMETERS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it can be followed by only one parameter name. When it appears in "table scope" or "column scope", it can be followed by more than one parameter name. Thus it can be used to generate several tables or columns.

For parameters that depend on a component, one can obtain the values of the parameter 'parameter name' for the component 'component name' using either

*PARAMETERS 'parameter name(component name)'

or the following combination of keywords

*PARAMETERS 'parameter name (COMPONENTS)'

*COMPONENTS 'component name'

The parameters, in appendix A, that depend on a component are those in which (COMPONENT) appear in the parameter name.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name. When *PARAMETERS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one parameter name. This parameter name will be in effect for all subsequent tables and columns.

When *PARAMETERS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more parameter names, and it will apply only to that table or group of tables. Thus, it will override any previous *PARAMETERS in "report scope".

When *PARAMETERS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more parameter names, and it will apply only to that column or groups of columns. Thus, it will override any previous *PARAMETERS in "report scope" or "table scope".

For parameters that depend on a component, both a parameter name and a component name should be specified. This can be done in two ways:

- a) *PARAMETERS 'parameter name(component name)' where parameter name is one of the parameter names in Appendix A, which contains (COMPONENT) as part of the parameter name.

For example:

```
*TABLE-FOR *WELLS 'INJ1'  
  *COLUMN-FOR 'Cumulative Gas(N2) SC'  
  *COLUMN-FOR 'Cumulative Gas(O2) SC'  
*TABLE-END
```

- b) *PARAMETERS 'parameter name(COMPONENT)' and *COMPONENT 'component name'

For example:

```
*PARAMETERS 'Cumulative Gas(COMPONENT) SC'  
*TABLE-FOR *WELLS 'INJ1'  
  *COLUMN-FOR *COMPONENT 'N2' 'O2'  
*TABLE-END
```

PRECISION

***PRECISION**

PURPOSE:

*PRECISION specifies the number of significant figures to be printed.

FORMAT:

*PRECISION precision

DEFINITION:

precision

An integer specifying the number of significant figures used for printing the data of a given column.

DEFAULTS:

Optional keyword. The default value is 6.

CONDITIONS:

This keyword, if present, can be either in "report scope" (outside any *TABLE-FOR and *TABLE-END pair) or in "column scope" (following a *COLUMN-FOR keyword). It can appear several times in the Results Report input file. The minimum allowed value is 4. As numbers in Results Report are stored as single precision, the maximum number of significant figures will be 7.

EXPLANATION:

This keyword specifies the number of significant figures that will be used to print the data values for a given column.

When *PRECISION appears in "report scope" (outside *TABLE-FOR *TABLE-END pairs), it will apply to all the following columns in all the following tables.

When *PRECISION appears in "column scope" (following *COLUMN-FOR), it will apply to the data values of that particular column only. In this way, different columns can be printed using a different number of significant figures.

PROPERTY-FOR

***PROPERTY-FOR**

PURPOSE:

*PROPERTY-FOR specifies the printing of a simulation grid property (such as oil saturation) in simulator input file (dataset) format, mesh map format, XYZ format, or SRF format.

FORMAT:

```
*PROPERTY-FOR
('prop_name' | 'prop_name(comp)' | *ALL-PROPERTIES)
(*COMPONENT 'comp')
(time | *ALL-TIMES)
(*MATRIX | *FRACTURE)
(*REFINED (*ON | *OFF))
(*MESH LAYER layer | *XYZ LAYER layer1 | *SRF-FORMAT |
  *ONE-BLOCK 'uba')
(*STATISTICS)
```

DEFINITION:

*ALL-PROPERTIES

Print all available properties from the SR2 file. The *MATRIX and *FRACTURE keywords will be ignored if the *ALL-PROPERTIES keyword is specified.

'prop_name'

Print the property corresponding to the indicated name. See Appendix A for a list of valid names.

'prop_name(comp)'

Print the property corresponding to the indicated property name for the indicated component.

'comp'

Component name. Only valid for component properties. If used, this string must be used in conjunction with the keyword *COMPONENT. Alternatively, the string 'prop_name(comp)' may be used to specify component properties.

time

Print the property for the nearest available time (in the SR2 file) to the specified time.

*ALL-TIMES

If the *ALL-TIMES key is specified, data for all available times will be printed in sequence.

layer

K layer number to print (integer).

layer1

K layer number to print (integer) or "0" (zero) indicating to print values for all layers in one row.

uba

A user block address. Must be in single quotes.

DEFAULTS:

Optional keyword. The default for time is 0.0 days. The property will be obtained from the current "global scope" input file, which must be an SR2 file.

The default print format is the same as the simulator input file format (simulator dataset), unless *XYZLAYER, *MESHLAYER, or *SRF-FORMAT are specified.

CONDITIONS:

This keyword, if present, must be in the global scope. The printing routine will ignore the *TABLE-WIDTH and *LINE-PER-PAGE keywords. You are not allowed to use "corner" properties (i.e., 'Grid Top' and 'Grid Thickness' for corner point grids) with the *PROPERTY-FOR keyword and *MESHLAYER or *XYZLAYER format. This is due to the possibility of having two values at a corner mesh point (i.e., one for each block with a fault - discontinuity - between blocks).

EXPLANATION:

Results Report will either obtain and print the specified grid property for the closest available time to the indicated time or all available times if the *ALL-TIMES key is indicated.

The *MESHLAYER format may be used to take a grid property from an SR2 file (say oil saturation at the end of a history match) and produce a mesh map for use by CMG's Builder (so it can be contoured and sampled for a different grid).

The *XYZLAYER format may be used to produce output of the form:

```
x(1) y(1) value(1)
x(2) y(2) value(2) ...
or
x(1) y(1) value(layer 1) value(layer 2) value(layer 3) ... ..
```

The *SRF-FORMAT keyword may be used to produce output similar to the old SR2 to SRF converter. No keywords or in-line comments will appear in the stream of block values. Output will include (if applicable) values for refined grid blocks. To specify component properties, the following notations may be used:

Notation (1):

PROPERTY-FOR 'Gas Mole Fraction(OXYGEN)

Notation (2):

PROPERTY-FOR 'Gas Mole Fraction(COMPONENT)' COMPONENT 'OXYGEN'

Notation (3):

PROPERTY-FOR 'Gas Mole Fraction\$C' COMPONENT 'OXYGEN'

The *ONE-BLOCK format will output the value for a single grid block for the specified property and time.

*STATISTICS will cause some aggregate statistics for the property to be output.

SECTORS

***SECTORS**

PURPOSE:

*SECTORS is used to specify the sector names for which information will be printed in tables and columns.

FORMAT:

```
*SECTORS      ( 'sector name' )  
              ( *ALL )
```

DEFINITION:

'sector name'

One or more sector names used in the simulation runs.

DEFAULTS:

Optional keyword. However, one of the following keywords: *WELLS, *LAYERS, *GROUPS, *SECTORS, *LEASES, *SPECIALS is required to specify an origin name.

CONDITIONS:

*SECTORS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one sector name. When it appears in "table scope" or "column scope", it can be followed by more than one sector names. Thus, it can be used to generate several tables or columns.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name. The keyword *SECTORS followed by one or more sector names, is one of the keywords (the others are *WELLS, *GROUPS, *LAYERS, *LEASES and *SPECIALS) that can be used to specify an origin name. The sector name must be in quotes and must be spelled exactly as in the simulation run.

When *SECTORS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one sector name. This sector name will be in effect for all subsequent tables and columns. When *SECTORS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more sector names, and it will apply only to that table or group of tables. Thus, it will override any previous *SECTORS that appears in "report scope". It will also override any other keyword that is used to specify an origin name.

When *SECTORS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more sector names, and it will apply only to that column or group of columns. Thus it will override any previous *SECTORS in "global scope" or "table scope". It will also override any other keyword that was used to specify an origin name.

When *ALL follows *SECTORS in table (column) scope, the program will print one table (column) for each of the sectors used in the simulation runs. Thus, you can easily obtain the same information for all the sectors without having to type all the sector names.

SPECIALS

***SPECIALS**

PURPOSE:

*SPECIALS is used to specify the special history names for which information will be printed in tables and columns.

FORMAT:

*SPECIALS ('special history name')
 (*ALL)

DEFINITION:

'special history name'

One or more special history names used in the simulation runs.

DEFAULTS:

Optional keyword. However, one of the following keywords: *WELLS, *LAYERS, *GROUPS, *SECTORS, *LEASES, *SPECIALS is required to specify an origin name.

CONDITIONS:

*SPECIALS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one special history name. When it appears in "table scope" or "column scope", it can be followed by more than one special history names. Thus, it can be used to generate several tables or columns.

EXPLANATION:

In order to specify the information for one column, you must specify a file name, an origin name and a parameter name. The keyword *SPECIALS, followed by one or more special history names, is one of the keywords (the other are *WELLS, *GROUPS, *LAYERS, *LEASES and *SPECIALS) that can be used to specify an origin name. The special history name must be in quotes and must have the same format as the special history names following SPHIST-NAMES in the .irf file. Therefore, the simplest method to obtain a valid special history name is to find SPHIST-NAMES in the .irf file, and cut and paste one of the following special history names from the .irf file to the Results Report input file.

When *SPECIALS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one special history name. This special history name will be in effect for all subsequent tables and columns. When *SPECIALS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more special history names and it will apply only to that table or group of tables. Thus it will override any previous *SPECIALS that appears in "report scope". It will also override any other keyword that was used to specify an origin name.

When *SPECIALS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more special history names, and it will apply only to that column or group of columns. Thus, it will override any previous *SPECIALS in "report scope" or "table scope". It will also override any other keyword that was used to specify an origin name. When *ALL follows *SPECIALS in table (column) scope, the program will print one table (column) for each of the special histories used in the simulation runs. Thus, you can easily obtain the same information for all the special histories without having to type all the special history names.

SPREADSHEET

***SPREADSHEET**

PURPOSE:

*SPREADSHEET tells Results Report to print out a table as a "tab separated" file, which makes importing into a spreadsheet (such as Microsoft Excel) easier.

FORMAT:

*SPREADSHEET

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword, if present, must be in "report scope" (it must be outside any *TABLE-FOR *TABLE-END pair).

EXPLANATION:

The table will be printed with tabs separating the column titles and the columns of information. Page breaks and form feeds will not be put in the output file, regardless of the settings of *LINES-PER-PAGE and *TABLE-WIDTH.

TABLE-FOR / TABLE-END

***TABLE-FOR, *TABLE-END**

PURPOSE:

*TABLE-FOR and TABLE-END are used to specify the information for a table or a group of tables.

FORMAT:

*TABLE-FOR

*TABLE-END

DEFAULTS:

Required keywords.

CONDITIONS:

At least one pair of *TABLE-FOR and *TABLE-END keywords must appear in the Results Report input file.

EXPLANATION:

This pair of keywords are used to request the output of a table or a group of tables (the examples explain how a group of tables can be printed using one pair of *TABLE-FOR and *TABLE-END keywords). These keywords must appear in pairs and can appear more than once in the Results Report input file. Thus, you can specify many groups of tables to be printed.

All the information (keywords) that follows *TABLE-FOR and precede the first *COLUMN-FOR keyword apply to all the columns in the table or group of tables and is referred to as being in "table scope". This information also determines whether one table or a group of tables will be printed.

For example:

```
*TABLE-FOR
  *FILES 'run1.irf' *WELLS 'w1'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'
*TABLE-END
will produce one table containing one column while
*TABLE-FOR
  *FILES 'run1.irf' 'run2.irf' 'run3.irf'
  *WELLS 'w1' 'w2'
  *COLUMN-FOR *PARAMETERS 'Gas Rate SC'
*TABLE-END
```

will produce 6 tables (for 2 wells from 3 different files) each containing one column.

TABLE-WIDTH

***TABLE-WIDTH**

PURPOSE:

*TABLE-WIDTH specifies the width in characters of a page.

FORMAT:

*TABLE-WIDTH width

DEFINITION:

width

An integer specifying the width in characters of a page.

DEFAULTS:

Optional keyword. The default value is 132

CONDITIONS:

This keyword, if present, must be in "report scope" (it must be outside any *TABLE-FOR *TABLE-END pair). It can appear several times in the Results Report input file.

EXPLANATION:

This keyword controls the maximum number of characters that can be printed in a line. The program will print as many columns as possible within the specified table width and will continue printing the remaining columns in a new page. This keyword can appear several times in the data set. Each value will override the previous value and will be in effect for all subsequent tables. In the absence of *TABLE-WIDTH, a default value of 132 characters per line is used.

In order to prevent breaking the tables into pages, a large value of width should be used (larger than the width required to print all the columns).

The minimum width allowed is 40 characters per line. The maximum width is system dependent but usually larger than 8000 characters.

TIMES-FOR

***TIMES-FOR**

PURPOSE:

*TIMES-FOR is used to specify the list of times for which information will be printed.

FORMAT:

*TIMES-FOR time-val

DEFINITION:

time-val

One or more positive reals or integers. DEFAULTS: Optional keyword.

If neither *TIMES-FOR, *DATES-FOR, nor *TIMES-FROM are used, the program will internally generate a default list of times and dates. (As the information for the different columns can be printed to the simulation result files at different times, the program takes the default times for the times array from the times in the column with the minimum time, and the column with the maximum time.)

CONDITIONS:

*TIMES-FOR must appear in global scope, i.e. outside any pairs of *TABLE-FOR and *TABLE-END keywords. The list of times must be monotonically increasing and all values must be positive. Time zero corresponds to the beginning of the simulation run.

EXPLANATION:

This keyword allows the user to specify a list of times for which information will be printed.

*TIMES-FOR can only appear outside *TABLE-FOR and *TABLE-END pairs and applies to all tables that follow. The list of dates can be overridden by any subsequent use of *DATES-FOR, *TIMES-FOR, or *TIMES-FROM.

For example:

```
*TIMES-FOR
  30 60 90 120 150 180 210 240 270 300 360
```

will produce a report containing 12 rows per table - one row for each of the specified times.

TIMES-FROM

***TIMES-FROM**

PURPOSE:

*TIMES-FROM specifies a column to be used to determine the times that information in all columns will be reported.

FORMAT:

*TIMES-FROM column-number

DEFINITION:

column-number

A positive integer to specify the column to be used.

DEFAULTS:

Optional keyword. If neither *TIMES-FOR, *DATES-FOR, nor *TIMES-FROM are used, the program will internally generate a default list of times and dates. (As the information for the different columns can be printed to the simulation result files at different times, the program takes the default times for the times array from the times in the column with the minimum time, and the column with the maximum time.)

CONDITIONS:

*TIMES-FROM must appear in the global scope, i.e. outside any pairs of *TABLE-FOR *TABLE-END keywords.

EXPLANATION:

The specified column will determine the times that information in all columns will be reported. If the specified column number is not defined in the table, the default time list will be generated for the report.

Since *TIMES-FROM must appear in the global scope, it will apply to all Tables that follow. The list of times can be overridden by any subsequent use of *TIMES-FOR, *DATES-FOR, or *TIMES-FROM.

For example:

```
*TIMES-FROM 4
```

will produce a report such that data in all columns will be reported at times corresponding to available data for the fourth column in the table.

TIME

***TIME**

PURPOSE:

*TIME controls the printing of a column containing the times.

FORMAT:

*TIME (*ON)
 (*OFF)

DEFINITION:

*ON

Print a column containing the times.

*OFF

Do not print a column containing the times.

DEFAULTS:

Optional keyword. Default *TIME *ON

CONDITIONS:

This keyword, if present, must be in "report scope", i.e. it must be outside any *TABLE-FOR and *TABLE-END pair. It can appear several times in the Results Report input file.

EXPLANATION:

This keyword controls the printing of a column containing the simulation times. As the information for the different columns can be printed to the simulation result files at different times (for example, different wells come into production at different times), the program takes the times for the time array from the times which correspond to the first column.

If both keywords *TIME *ON and *DATE *ON are used, the time and date columns will be the first and the second columns respectively in all the pages corresponding to a given table. The data columns will follow these two columns. If only *TIME *ON is specified, the time column will be the first column in the page followed by the data columns.

TITLE1 / TITLE2 / TITLE3

***TITLE1, *TITLE2, *TITLE3**

PURPOSE:

*TITLE1 specifies a title string to be printed in the title block of all following tables.

FORMAT:

*TITLE1 'first title string'
*TITLE2 'second title string'
*TITLE3 'third title string'

DEFAULTS:

Optional keyword.

DEFINITION:

string

A string in single quotes giving the title.

CONDITIONS:

These keywords, if present, must be in "report scope" (it must be outside any *TABLE-FOR *TABLE-END pair). They can appear several times in the Results Report input file with each subsequent appearance overriding previous appearances.

EXPLANATION:

These keywords may be used to add extra annotations to the reports. They can be used to add the simulation study name to the report.

UNITS

***UNITS**

PURPOSE:

*UNITS specifies the output units for data

FORMAT:

*UNITS (*SI)
 (*FIELD)
 (*MODSI)
 (*LAB)

DEFAULTS:

Optional keyword. The default value is to use the output units specified in the simulator or to use SI if the output units are not known.

CONDITIONS:

This keyword, if present, can be either in "report scope", i.e. outside any TABLE-FOR TABLE-END pair or in "column scope", i.e. following a *COLUMN-FOR keyword. It can appear several times in the Results Report input file.

EXPLANATION:

This keyword can be used to change the output units for a report to something different than was used in the simulator. Results Report will convert the parameter values to the units appropriate for the specified unit system.

WELL-KEYWORDS

***WELL-KEYWORDS**

PURPOSE:

*WELL-KEYWORDS tells Results Report to print out a table in the form of simulator input well keywords (*OPERATE and *ALTER).

FORMAT:

*WELL-KEYWORDS

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword, if present, must be in "report scope" (it must be outside any *TABLE-FOR *TABLE-END pair). *WELL-KEYWORDS causes *LINES-PER-PAGE to be set to 20000 and forces *DATE *ON and *TIME *OFF.

EXPLANATION:

When beginning a history match simulation, historical well production data needs to be converted to simulator input format. If the well data is in one of the supported Results Report input formats (see Appendix A Field History File Formats and Appendix B Production Analyst Load Format in the *Results Graph User Guide*), then Results Report can convert the data to simulator input format.

The following example shows how to convert well data in CMG's field history file (fhf) format to simulator input format. The data in 'sample.fhf' is monthly production totals. Three production wells exist. To speed up the simulation, the simulator well changes are to occur on a quarterly basis - so the parameter 'Oil Rate SC - Quarterly' is used. The well named 'INJECTOR' is a water injector.

```
*FILES *FHF 'sample.fhf'
*UNITS *FIELD
*WELL-KEYWORDS
*TABLE-FOR
  *COLUMN-FOR  *PARAMETERS 'Oil Rate SC - Quarterly'
                *WELLS 'PROD 1' 'PROD 2' 'PROD 3'
  *COLUMN-FOR  *PARAMETERS 'Water Rate SC - Quarterly'
                *WELLS 'INJECTOR'
*TABLE-END
The output will produce a comment line giving an *OPERATE keyword
for each well:
** PRODUCER 'PROD 1' or INJECTOR 'PROD 1'
** OPERATE MAX STO          100.000  ** Oil Rate SC - Quarterly
The output will produce one or more *ALTER keywords
(depending on the number of wells and the *TABLE-WIDTH),
followed by a simulator DATE keyword:
*ALTER   'PROD 1'  'PROD 2'  'PROD 3'  'INJECTOR'
          100.000   100.000   100.000   100.000
*DATE 1993 03 31
```

WELL-SUMMARY

***WELL-SUMMARY**

PURPOSE:

*WELL-SUMMARY tells Results Report to print out a table in a form similar to the old style simulator output file well summary table. At a specified time interval, a table is produced for each of the injection groups, injection wells, production groups and production wells. Each table has columns for phase rates, cumulatives, pressures, etc., and a row for each group or well. Layer information may also be requested.

FORMAT:

*WELL-SUMMARY report_interval info table_style

DEFINITIONS:

report_interval

Indicates the time period for reporting information. You must select one of the following:

- *YEARLY report information at the end of each calendar year (December 31)
- *QUARTERLY report information at the end of each calendar quarter
- *MONTHLY report information at the end of each calendar month

Normally, the phase rates are averaged over the reporting interval. If you want rates at the end of the reporting interval, add the keyword *INSTANT-RATES.

- *SR2TIMES report information at all times that the requested information is available in the SR2 file

info

Indicates which information to put in tables. You may select one or more of the following:

- *GROUPS information from injection and production groups
- *WELLS information from injection and production wells
- *LAYERS information from well layers (if available in SR2 file)
- *SECTORS sector information (if available in SR2 file)
- *LEASE-PLANES lease plane information (if available in SR2 file)

table_style

Indicates how the tables will be arranged. You must select one of the following:

- *TWP Produce a table for each time, a row for each origin (well, etc.) and a column for each parameter.
- *TPW Produce a table for each time, a row for each parameter and a column for each origin.

*WTP	Produce a table for each origin, a row for each time and a column for each parameter.
*WPT	Produce a table for each origin, a row for each parameter and a column for each time.
*PWT	Produce a table for each parameter, a row for each origin and a column for each time.
*PTW	Produce a table for each parameter, a row for each time and a column for each origin.

DEFAULTS:

Optional keyword. By default, only Group and well information tables are written.

*YEARLY is the default reporting interval. The default format is TWP (table for each time, row for each well, column for each parameter). For each origin type (GROUPS, WELLS, LAYERS, SECTORS, and LEASE-PLANES) there is an internally fixed list of parameters to be output. This list may be reduced if some of the parameters are not available in the SR2 file. There is currently no way to change the list.

CONDITIONS:

This keyword, if present, must be in "report scope" (it must be outside any *TABLE-FOR *TABLE-END pair). Any *DATE, *WIDTH, *PRECISION, and *SPREADSHEET keywords prior to *WELL-SUMMARY are honored, other keywords have no effect on the table format.

EXPLANATION:

This keyword can cause the printing of a set of tables similar to those that were produced in pre-98 versions of the IMEX simulator. However, the reporting periodicity can be set to yearly, quarterly, monthly, or all the times that well information is written to the SR2 file (usually controlled by the simulator *WSRF *WELLS keywords in the simulator). Period average rates are available for yearly, quarterly and monthly periods.

For example:

```
*FILE *SR2 'dualfld2.irf'
*DATE *ON

** Produce a table similar to the pre-98 version of IMEX
*WELL-SUMMARY *YEARLY
```

WELLS

*WELLS

PURPOSE:

*WELLS is used to specify the well names for which information will be printed in tables and columns.

FORMAT:

*WELLS ('well name')
 (*ALL)
 (*ALL-PRODUCERS)
 (*ALL-INJECTORS)

DEFINITION:

'well name'

One or more well names used in the simulation runs. Each well name must be enclosed in separate single quotes (i.e. *WELLS 'well 1' 'well 2' 'well 3').

*ALL

Print one column or one table for each well.

*ALL-PRODUCERS

Print one column or one table for each producer.

*ALL-INJECTORS

Print one column or one table for each injector.

DEFAULTS:

Optional keyword. However, one of the following keywords: *WELLS, *LAYERS, *GROUPS, *SECTORS, *LEASES or *SPECIALS is required to specify an origin name.

CONDITIONS:

*WELLS can appear in either "report scope", "table scope" or in "column scope". When it appears in "report scope", it must be followed by only one well name. When it appears in "table scope" or "column scope", it can be followed by more than one well names. When more than one well name is used, several tables or columns will be generated.

EXPLANATION:

In order to specify the information for one column you must specify a file name, an origin name and a parameter name. The keyword *WELLS, followed by one or more well names, is one of the keywords (the other are *GROUPS, *LAYERS, *SECTORS, *LEASES and *SPECIALS) that can be used to specify an origin name. The well name must be in quotes and must be spelled exactly as in the simulation run.

When *WELLS appears in "report scope", i.e. outside any *TABLE-FOR *TABLE-END pairs, it can be followed by only one well name. This well name will be in effect for all subsequent tables and columns.

When *WELLS appears in "table scope", i.e. following *TABLE-FOR and preceding *COLUMN-FOR, it can be followed by one or more well names, and it will apply only to that table or group of tables. Thus, it will override any previous *WELLS that appears in "report scope". It will also override any other keyword that was used to specify an origin name.

When *WELLS appears in "column scope", i.e. following a *COLUMN-FOR keyword, it can be followed by one or more file names, and it will apply only to that column or group of columns. Thus, it will override any previous *WELLS in "global scope" or "table scope". It will also override any other keyword that was used to specify an origin name.

When *ALL follows *WELLS in table (column) scope, the program will print one table (column) for each of the wells used in the simulation runs. Thus, you can easily obtain the same information for all the wells without having to type all the well names.

Similarly, the keywords *ALL-PRODUCERS or *ALL-INJECTORS can be used to print tables or columns for all the producers or all the injectors.

WIDTH

***WIDTH**

PURPOSE:

*WIDTH specifies the width in characters for a given column.

FORMAT:

*WIDTH width

DEFINITION:

width

An integer specifying the width in characters for a give column.

DEFAULTS:

Optional keyword. The default value is 14.

CONDITIONS:

This keyword, if present, can be either in "report scope" (outside any *TABLE-FOR and *TABLE-END pair) or in "column scope" (following a *COLUMN-FOR keyword). It can appear several times in the Results Report input file. The minimum allowed value is 12. The program will ignore the value specified by the user when $\text{width} < \text{precision} + 8$. In such a case, $\text{width} = \text{precision} + 8$ will be used.

EXPLANATION:

This keyword specifies the width in characters of the space in which a column can be printed.

When *WIDTH appears in "report scope" (outside *TABLE-FOR and *TABLE-END pairs), it will apply to all the following columns in all the following tables.

When *WIDTH appears in "column scope" (following a *COLUMN-FOR keyword), it will apply to the data values of that particular column only. In this way the width of the space, in which the columns will be printed, can be given different values for different columns.

In order to guarantee that there will be always enough space to print the data with the required number of significant figures, the program will override the value of width when $\text{width} < \text{precision} + 8$. In such a case $\text{width} = \text{precision} + 8$ will be used. Furthermore, the minimum allowed value for width is 12.

Appendix

List of Parameter Names

A complete list of all the parameters, components and origins available in a given SR2 file may be obtained by running the following command file:

```
*FILE 'sr2-file-name.irf'  
*LIST-PARAMETERS
```

where you should substitute your SR2 file name between the single quotes. See [LIST-PARAMETERS](#) for more details. The output from this command file will contain lines that can be cut and pasted into command files.

In a similar manner, a complete list of all the grid properties and times at which they are available, for a given SR2 file, can be obtained using the [*LIST-PROPERTIES](#) keyword.

Index

Keyword Definitions

B

BLANKS 17

C

COLUMN-FOR 19
COMPONENTS 20

D

DATE 21
DATES-FOR 22

F

FILES 23

G

GRID-FOR 25
GROUPS 26

L

LAYERS 28
LEASES 30
LINES-PER-PAGE 32
LIST-PARAMETERS 33
LIST-PROPERTIES 34

M

MULTIPLY 35

N

NO-BLANKS 17

O

OUTPUT 36

P

PARAMETERS 37
PRECISION 39
PROPERTY-FOR 40

S

SECTORS 43
SPECIALS 45
SPREADSHEET 47

T

TABLE-END 48
TABLE-FOR 48
TABLE-WIDTH 49
TIME 52
TIMES-FOR 50
TIMES-FROM 51
TITLE1 53
TITLE2 53
TITLE3 53

U

UNITS 54

W

WELL-KEYWORDS 55
WELLS 58
WELL-SUMMARY 56
WIDTH 60