By order of:
Rijkswaterstaat/Rijksinstituut voor Kust en Zee

User's guide

DONSIM

SIMONA report number 97-01

Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

User's guide
DONSIM

DONSIM is an intermediate system, that effectuates the exchange of data between DONAR and SIMONA.

Version : 2.7, November 2007
Maintenance : see www.helpdeskwater.nl/waqua

Preface

Not yet available.

Contents

1 About this manual

The user's guide DONSIM contains a comprehensive description of the subsystems of DONSIM. It also describes principles and backgrounds of the DONSIM system.

This document is written for users primarily. A general introduction to the system is given in chapter 2 (About DONSIM) of this user's guide for DONSIM. The other chapters give more detailed descriptions of the DONSIM system.

The DONSIM system as described in this user's guide DONSIM only deals with SIMONA based subsystems.

For a brief description of the program (filesprocedure call), the user can refer to the corresponding chapter in the Quick Reference Guide of the User's Guide of DONSIM.

2 About DONSIM

2.1 Introduction to the system

DONSIM

DONSIM is a system, which effectuates the exchange of data between DONAR and SIMONA applications. Only the exchange of data from DONAR to SIMONA is realised.

DONSIM is based on DONAR interface modules and SIMONA routines.

DONAR

DONAR is a system for storage, management and presentation of water management data for RWS. Information is available in the various DONAR guides.

SIMONA

SIMONA defines an architecture for preprocessing, memory management, data storage and postprocessing. SIMONA aims for structured and controlled development of software, reducing cost for maintenance and support. Information is available in the SIMONA programmer's guide.

The major part of the system is written in FORTRAN77. The interface with the DONAR Interface Modules (DIM) is written in ANSI-C. The system is available on a HP9000s700 and on a SUN-SPARC station.

2.1.1 Function of the system

exchange

The system can be used for the exchange of data between DONAR and SIMONA applications. At the moment, only the exchange from DONAR to SIMONA is realized. As documented here, the system consists of a program for a one to one conversion of a DONAR Interface File (DIF) to a SIMONA Data Storage (SDS) file and a program for the generation of SIMONA include files from series extracted from an (observed series) SDS-file.

.

2.1.2 Features

conversion DIF to SDS

One of the applications implemented, program DIFTOSDS, performs a one to one conversion of a DONAR Interface File (DIF) to a SIMONA Data Storage (SDS) file. At the moment, only time series at singular points are accepted.

conversion SDS to INCLUDE

The other application implemented, program SDSTOINC, generates SIMONA include files from series extracted from an (observed series) SDS-file. In the first stage, only time series at singular points are accepted.

2.2 Data flow

A graphical impression of the global data flow of the system DONSIM can be seen in Fig. 2.1

2.2.1 Input

DONAR interface file

The input for the DONSIM system is a so-called DONAR Interface File (DIF) in either ASCII (file extension .dia) or binary form (file extension .dib). The DIF file is converted to a SIMONA Data Storage file (SDS) by means of the conversion program DIFTOSDS.

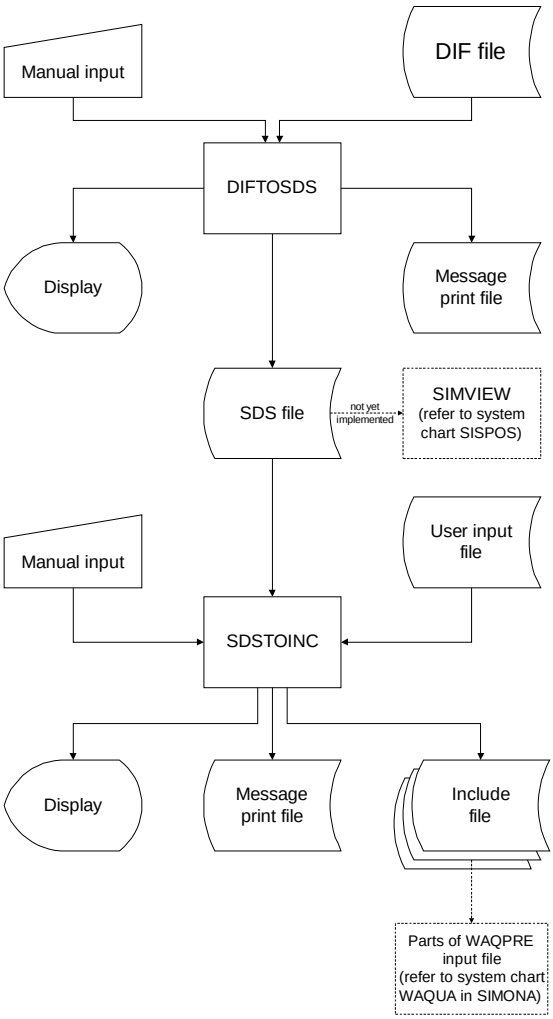

Fig. 2.1 System chart of DONSIM

2.2.2 Output

SDS-file, INCLUDE files and prints

At the moment, the output from the DONSIM system consists of a SIMONA Data Storage (SDS) file, SIMONA include files and printed diagnostic messages.

error and warning messages

All programs print run and error messages, as applicable. Run messages generally offer information when a program is running properly, while error messages give warnings concerning problems found. Therefore, the run messages generally are similar from run to run of the same program, but the error messages will differ, if they appear at all.

2.3 DONSIM programs

program DIFTOSDS

This program performs a one to one conversion of a DONAR Interface File (DIF) to a SIMONA Data Storage (SDS) file.

In the current implementation DIFTOSDS only accepts time series at singular points .

program SDSTOINC

This program generates series include files from series extracted from an (observed series) SDS-file.

In the current implementation SDSTOINC only accepts time series at singular points .

3 Quick reference guide

3.1 General

3.1.1 Introduction

This quick reference guide contains descriptions of several scripts to run DONSIM subsystems. First some general remarks are given.

3.1.2 Background runs and shells

The behaviour of a script depends on the shell being used. The descriptions of the format of the scripts in this quick reference guide are all based on the use of the Bourne shell. If the Korn shell or the C shell is used, there are some problems in relation to the use of the background options.

There are four ways to run a program in the background:

- by using *-back y* in the options list;
- by answering affirmative when (in an interactive run) the run procedure asks whether the program must be started in the background or not;
- by using *&* in the options list;
- by using a user script containing the options list and *&*.

The third possibility is not recommended when using the Korn shell or the C shell. In this case it is necessary to specify a complete options list. If one parameter is not given or is not correct, the run procedure will stop and wait for tty input. Even though *&* is specified, the run procedure will always produce screen output. When using the C shell or the Korn shell, the procedure will not be able to suppress screen output in case of a background run.

In case of an interactive run, a user script or the usage of the Bourne shell, none of these problems will appear.

3.2 Program DIFTOSDS

3.2.1 Function

This program converts a DONAR Interface File (DIF) into an SDS-file.

3.2.2 Files

data flow

A graphical impression of the data flow of the program DIFTOSDS is given in Fig. **3.1.**

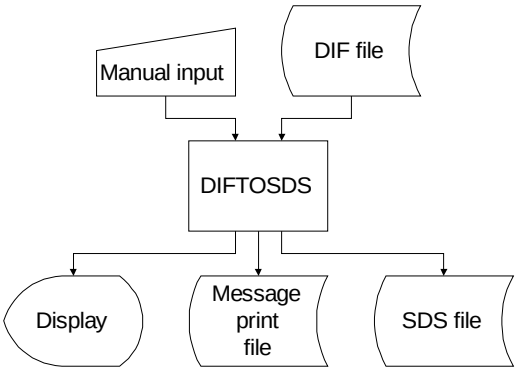

Fig. 3.1 Flow chart of DIFTOSDS

The following tables give the names of the input and output files as used by the run proceduree and the purpose of these files respectively.

System name

Logical name	System name
DIF file	<difid>.dia or <difid>.dib
Message print file	diftosds-m.<runid>
SDS-file	SDS-<runid>

purpose

Logical name	Purpose
DIF file	Interface file for the DONAR database system.
Message print file	Contains error messages and diagnostics.
SDS-file	Storage of permanent data in SIMONA.

3.2.3 Input and output

Refer to paragraph 4.1

3.2.4 Run data

format

diftosds.pl -runid <runid> -dif <dif> [-ins <ins>]
[-bufsize <bufsize>] [-back <back>] [&]

parameters

parameter	meaning and values
-runid	Code (alpha-numeric) to identify the output files; e.g.: -runid 001.
-dif	File name of DONAR Interface File.
-ins	Institute name, default = RIKZ.
-bufsize	The size of SIMONA blank common in million words, i.e. the space needed to store all SIMONA arrays; default: 5
-back	y(es): program is started in the background n(o): program is started in the foreground

In case of a background run (with &) runid and dif are obligatory parameters and back has no effect.

In an interactive start of the run the procedure prompts for above mentioned parameters (except those that have already been given in the run call) including back. All parameters are checked before the program is started.

Example: *diftosds.pl -runid t00 -dif test.dia -back no*

If an executable of the program (diftosds.exe) is present in the current directory, this executable will be used to run the program.

3.3 Program SDSTOINC

3.3.1 Function

This program generates SIMONA INCLUDE files from series extracted from an (observed series) SDS-file.

3.3.2 Files

A graphical impression of the data flow of the program SDSTOINC is given in Fig. 3.2

The following tables give the names of the input and output files as used by the run procedure and the purpose of these files respectively.

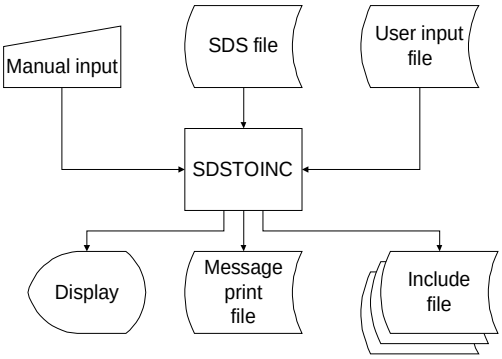

Fig. 3.2 Flow chart of SDSTOINC

System name

Logical name	System name
SDS-file	SDS-<runid>
Include file	<experiment name> _<quantity name> _<location name> _TS<sequence number>
User input file	free to choose
Message print file	sdstoinc-m.<runid>

purpose

Logical name	Purpose
SDS-file	SDS-file with observed series.
User input file	keyword organised user input to select observed series from an (observed series) SDS-file
Message print file	Contains error messages and diagnostics.

Include file	ASCII include form of observed series.

3.3.3 Input and output

Refer to paragraph 4.2.

3.3.4 Run data

format

*sdstoinc.run -runid <runid> -input <input> *
[-bufsize <bufsize>] [-back <back>] [&]

Parameters

parameter	**meaning and values**
-runid	Code (alpha-numeric) to identify the output files; e.g.: -runid 001.
-input	File name of user input file.
-bufsize	The size of SIMONA blank common in million words, i.e. the space needed to store all SIMONA arrays; default: 5
-back	y(es): program is started in the background n(o): program is started in the foreground

In case of a background run (with &) runid and input are obligatory parameters and back has no effect.

In an interactive start of the run the procedure prompts for above mentioned parameters (except those that have already been given in the run call) including back. All parameters are checked before the program is started.

Example: *sdstoinc.run -runid mux3 -input inpmux3.all -back no*

If an executable of the program (diftosds.exe) is present in the current directory, this executable will be used to run the program.

4 Extensive description of the programs

4.1 Program DIFTOSDS

4.1.1 Function

conversion

This program performs a one to one conversion of a DONAR Interface File (DIF) to a SIMONA Data Storage (SDS) file. Only time series at singular points are accepted.

DIFTOSDS reads the DIF file, which may be binary (file extension .dib) or ASCII (file extension .dia). If a set of requirements is met, the program will accept time series at a singular point and stores them consequently in an (observed series) SDS-file. Via a C interface the program makes use of DONAR Interface Modules (DIM) in order to read the DIF file. In the exceptional case that a DIM routine returns with a DIM error status (integrity error), the program DIFTOSDS will generate an appropriate error message and terminate immediately.

4.1.2 Input files

DIFTOSDS reads a control information file (with run procedure parameters SDS-file name, DIF file name, and institute name) and consequently the specified DIF file.

4.1.3 Output files

SDS-file

DIFTOSDS generates an observed series SDS-file and a diagnostics file for run information and error messages. If the SDS-file already exists it will be (destructively) overwritten.

4.1.4 Description

Once the binary/ASCII DIF file is opened, the program DIFTOSDS continues reading DIF series as long as no fatal DIM error occurs. Only DIF time series, which are defined at point locations, are candidates for acceptation. All other series will be ignored (after an appropriate error message). Both singular and multiplexed DONAR series are considered. Multiplexed series will fall apart in as many separate independent time series as there are multiplex channels, where each channel is a candidate for acceptation.

singular DONAR DIF series

A singular (non-multiplexed) DONAR DIF series is accepted and consequently stored on the series SDS-file if:

1 it is a time series;
2 it is defined on a singular point location (DONAR Locatiesoort = 'P') with resulting spatial coordinates without magic values;
3 the spatial coordinates system is 'Nieuw Rijks Driehoeksstelsel' (origin Paris);
4 in case of an equidistant time series, the time step is 'usable';

5 the parameter code is not blank.

Table 4.1 Parameter code table

DONAR quantity name (parameter code)	SIMONA quantity name	Number of different DONAR units to be accep-ted	DONAR units	SIMONA units	Units con-version factor	Specific SIMONA 'hiatus value' (magic value)
'BODHTE'	'depth'	1	'm'	'm'	1.	010101.
'LUCHTDK'	'atmpressure'	1	'pa'	'pa'	1.	020202.
'Q'	'discharge'	1	'm3/s'	'm3/s'	1.	030303.
'SALNTT'	'salinity'	1	'DIMSLS'	'kg/m3'	1.	040404.
'STROOMRTG'	'flowdirectn'	1	'graad'	'degrees'	1.	050505.
'STROOMSHD'	'flowspeed'	1	'm/s'	'm/s'	1.	060606.
'STROOMSHHRZT'	'flowhorizont'	1	'm/s'	'm/s'	1.	070707.
'STROOMSHVTCL'	'flowvertical'	1	'm/s'	'm/s'	1.	080808.
'T'	'temperature'	1	'°C'	'°C'	1.	090909.
'WATHTE'	'waterlevel'	1	'cm'	'm'	.01	101010.
'WINDRTG'	'wnddirection'	1	'graad'	'degrees'	1.	111111.
'WINDSHD'	'wndmagnitude'	1	'm/s'	'm/s'	1.	121212.
'Cl'	'Cl'	1	'mg/l'	'mg/l'	1.	131313.
'CONDTVTT'	'CONDTVTT'	1	'S/m'	'S/m'	1.	141414.

6 parameter code (identifying the quantity) of the series is part of the so-called 'parameter code table' (see Table 4.1), the units at the DONAR side must correspond to the units which are defined in the same parameter code table as the units to be accepted;
7 the combination of parameter code and location code is unique (it appears only once on the DIF file for a particular series);
8 the series contains 'usable numbers'; speaking in DONAR terms: the 'domain code' is either 'E' (E-notation in character form), 'F' (floating point) or 'I' (integer).

multiplexed DONAR DIF series

A multiplexed DONAR DIF series is accepted and consequently stored on the series SDS-file if:
1 it is a time series;
2 it is defined on a singular point location (DONAR Locatiesoort ='P') with resulting spatial coordinates without magic values;
3 the spatial coordinates system is 'Nieuw Rijks Driehoeksstelsel' (origin Paris);
4 in case of an equidistant time series, the time step is 'usable' (see below);
per MUX-channel:
5 the parameter code is not blanc;
6 if the parameter code (identifying the quantity) of the series is part of the so-called 'parameter code table' (see Table 4.1), the units at the

DONAR side correspond to the units which are defined in the same parameter code table as the units to be accepted;

7 the combination of parameter code and location code is unique (it appears only once on the DIF file for a particular series);

8 the series must contain 'usable numbers'; speaking in DONAR terms: the 'domain code' is either 'E' (E-notation in character form), 'F'(floating point) or 'I' (integer).

units

A time step for equidistant time series, called 'tijdstapeenheid' at the DONAR side, is usable if it is given in one of the following units:

1 's' (second);
2 'min' (minute);
3 'd' (day);
4 'mnd' (month);
5 'a'(year).

In case of a time step 'mnd' (month), the starting date of the series will be placed on the 16th of the starting month (with an exception for February. In that case the starting date will be the 15th).

In case of a time step 'a'(year), the starting date of the series will be placed on July 2nd.

Parameter code is part of the parameter code table

If a singular DONAR DIF series or a channel of a multiplexed DONAR DIF series has a parameter code (identifying the quantity) which is part of the parameter code table (see Table 4.1) and the series is 'accepted', it will be stored on the series SDS-file and:

a the quantity name at the DONAR side (parameter code) will be replaced by the SIMONA quantity name (refer to Table 4.1);
b the unit name at the DONAR side will be replaced by the SIMONA unit name (refer to Table 4.1), where a unit conversion, if any, will take place;
c the DONAR 'hiatus value' (magic value) will be replaced by a specific SIMONA 'hiatus value' (refer to Table 4.1);
d the spatial coordinates of the point location are converted from 'cm' into 'm'.

Parameter code is not part of the parameter code table

If a singular DONAR DIF series or a channel of a multiplex DONAR DIF series has a parameter code (identifying the quantity) which is not part of the parameter code table (see Table 4.1) and the series is 'accepted', it will be stored on the series SDS-file and:

a the quantity name at the DONAR side (parameter code) will be taken as the SIMONA quantity name;
b the unit name at the DONAR side will be taken as the SIMONA unit name;
c the DONAR 'hiatus value' (magic value) will be replaced by a neutral SIMONA 'hiatus value' (999999.);
d the spatial coordinates of the point location are converted from 'cm' into 'm'.

If a time series is stored on the series SDS-file, apart from the values of the quantity, the corresponding times are stored in terms of 'elapsed' minutes relative to the starting date of the series at midnight (00:00:00).

SDS-file

The following quantities will be stored on the SDS-file:

a for each of the series:
- quantity name;
- location code;
- coordinates of location (X, Y, Z);
- units;
- state;
- missing value (magic value);
- series length;
- series reference date;
- series start address;

b for all series:
- time instances for all series;
- values.

The DONAR system only makes use of X and Y coordinates. The values will be taken from the W3H administration. If the values in the time series administration differ from the W3H value (these refer to the exact point of measurement) the series administration values will be used. Only correct coordinates are allowed (no magic values). The value '0' is used for the Z coordinate.

4.2 Program SDSTOINC

4.2.1 Function

include file

This program generates SIMONA include files from series extracted from an (observed series) SDS-file. At this stage, only time series at singular points will be dealt with.

SDSTOINC reads the user input file containing the series specifications, accepts time series at a singular point, if a set of requirements is met, and stores the series consequently in an (observed series) include file.

4.2.2 Input files

user input file, DIF file

SDSTOINC reads a control information file first (with run procedure parameters, where input file name is one of them) and next the specified input file (user input file with key-word structure). Under control of the user input file the (observed series) SDS-file is read in order to extract the required series.

Note: *If the last keyword block in the input file contains a sequential keyword, the SIMONA application independent preprocessor is not*

able to check the correctness of the block. This can result in incorrect processing of the input file!

User input file

file structure

This file contains a key-word structure. For general information and conventions used, refer to the users's guide WAQPRE, Chapter 2.1. Under control of the user input file the (observed series) SDS-file is read in order to extract the required series.

Main key-words

The input is divided in two main key-words These key-words are (M = mandatory, O = optional, D = optional with default value):

IDENTIFICATION (M)
SELECTION (M)

The keywords are described in the following sections.

IDENTIFICATION (mandatory)

In the identification block the (observed series) SDS-file is identified.

IDENTIFICATION
SDSNAM = *[text]*

Explanation

SDSNAM = *[text]*

M In *[text]* the name of the SDS-file is given. The given file name may contain an explicit path name. The use any indication of a parent directory ('..') is allowed.

SELECTION (mandatory)

In the selection block the required series are specified.

SELECTION
TIMESERIES (O)
< TS: EXPNAM = *[text]* (M)
QUANTITY = *[text]* (M)
LOCATION = *[text]* (M)
STARTDATE = *[ival]* (M)
STARTTIME = *[ival]* (M)
ENDDATE = *[ival]* (M)
ENDTIME = *[ival]* (M)
TIMEZONE = *[ival]* (D)
COORDSYSTEM = *[ival]* (D)
>
CURVESERIES (O)
GRIDSERIES (O)

Explanation

Currently, it is only possible to select time series (at singular points), so no curve series and no grid series.

EXPNAM = *[text]* **M** In *[text]* the name of the experiment on the SDS-file is given. Maximum length of text is: 40 characters.

QUANTITY = *[text* **M** In *[text]* the name of the quantity is given. Maximum length of text is: 12 characters.
Note: the quantity name used here must be the SIMONA quantity name (refer to Table 4.1).

LOCATION = *[text]* **M** In *[text]* the name of the locationis given.
Maximum length of text is: 12 characters.

STARTDATE = *[ival]* **M** In *[ival]* the start date is given in the form YYYYMMDD.

STARTTIME = *[ival]* **M** In *[ival]* the start time is given in the form HHMMSS.

ENDDATE = *[ival]* **M** In *[ival]* the end date is given in the form YYYYMMDD.

ENDTIME = *[ival]* **M** In *[ival]* the end time is given in the form HHMMSS.

TIMEZONE = *[ival]* **M** In *[ival]* the time zone is given.
Default = 0 (MET).

COORDSYSTEM = *[ival]* **D** In *[ival]* the coordinates system code is given.
Default = 0 (Nieuw Rijksdriehoekstelsel (origin Paris)).

4.2.3 Output files

output file

SDSTOINC generates a SIMONA include file for each series and a diagnostics file for run information and error messages. if an SIMONA include file already exists, it will be (destructivly) overwritten.

SIMONA include file

include file

Each include file corresponds to one time series request in the user input file. The file is in ASCII format and contains 'irregular' time series. For (irregular) 'time series' refer to the user's guide WAQPRE, Chapter 2.1.

The name of each include is as follows:
<experiment name>_<quantity name>_<location name>_TS<sequence number>,

where the sequence number corresponds to the sequence number of the time series request in the user input file.

Besides the time series the include file also contains a starting comment block, where all relevant series parameters are printed out.

4.2.4 Description

Once the SDS-file is opened, for each required time series program SDSTOINC opens the experiment and checks whether it contains the

array 'OBS_SERIES', which tells us whether we are dealing with an observed series experiment or not.

Next a search is made for the required combination of quantity and location. If such a combination is found, this also implies that we are dealing with *the observed series* candidate on the SDS-file (since there is at most one time series present for each combination of quantity and location).

Assuming we have found such a candidate, the next requirement is that the period-span of the required series and the period-span of the observed series found (partly) overlap in such a way that the resulting series will contain one real non-magic (i.e. non-'hiatus value') value at least. The way in which the resulting series is obtained will be demonstrated next using symbolic schemes.

In the next scheme the axis that we use symbolises a time axis, 'O' symbolises the occurrence of an observed value in the original series found. An arrow marker is used to visualize the required series, 'R' symbolises an observed value in the resulting series and 'a' symbolises an 'artificial value'.

```
O--------O----------O---O--------O------O----O—O

    ↑------------------------------↑

    Ra----R----------R---R------R---Ra
```

The resulting series will contain an artificial value at the beginning and at the end. All intermediate values are just copied from the original series. The artificial value will be obtained by linear interpolation if the two embracing original values are real (non-magic) values. If both original values are magic, the artificial value will be magic as well. If only one original value is magic, the artificial value will be set equal to the other real original value.

The next scheme symbolises the case that the complete required period span is embraced by two original series values only.

```
O---------------------------O

    ↑---------------------↑

    Ra---------------------Ra
```

The resulting series will only contain an artificial value at the beginning and at the end. Each artificial value will be obtained by linear interpolation if the two embracing original values are real (non-magic) values. If both original values are magic, the artificial values will be magic as well. If only one original value is magic, the artificial values will be set equal to the real original value.

If the required starting point falls earlier than the first time instance in the original series, the first resulting value will be set equal to the first value in the original series (WAQUA-like extrapolation). For the required ending point the same strategy is used (mirror-image wise).

5 Examples

5.1 User input file for the program SDSTOINC

```
IDENTIFICATION
      sdsnam = 'SDS-par1'
SELECTION
      TIMESERIES
            TS1
                  expnam = 'expid'  quantity = 'waterlevel'
                  location = 'VLISSGN'
                  startdate = 19850101 starttime = 000000
                  enddate = 19850101 endtime = 000159
```

5.2 Include file resulting from SDSTOINC

```
#-------------------------------------------------------------
#
#Parameters of Timeseries TS001:
#
#  Observed Series Sds-file       : SDS-par1
#  Experiment name                : expid
#  Quantity name                  : waterlevel
#  Location name                  : VLISSGN
#  Series starting date           : 19850101
#  Series starting time           : 000000
#  Series ending date             : 19850101
#  Series ending time             : 000159
#  Time zone code (0=MET)         :  0
#  Coordinates system code        :  0
#  (0= Nieuw Rijksdriehoekstelsel (origin Paris))
#  Magic value (hiaatwaarde)      :  1.0101000E+005
#  X-coordinate                   :  3.0480000E+004
#  Y-coordinate                   :  3.8522000E+005
#  Z-coordinate                   :   .0000000E+000
#  Units                      : m
#  State (hoedanigheid) code      : NAP
#  Number of series values        :     120
#
#-------------------------------------------------------------
SERIES = 'irregular'
TIMES_AND_VALUES = (0000000000 00:00.000)  9.9999998E-003
TIMES_AND_VALUES = (0000000000 00:00.017)  2.0000000E-002
TIMES_AND_VALUES = (0000000000 00:00.033)  2.9999999E-002
TIMES_AND_VALUES = (0000000000 00:00.050)  3.9999999E-002
TIMES_AND_VALUES = (0000000000 00:00.067)  4.9999997E-002
TIMES_AND_VALUES = (0000000000 00:00.083)  5.9999999E-002
TIMES_AND_VALUES = (0000000000 00:00.100)  7.0000000E-002
TIMES_AND_VALUES = (0000000000 00:00.117)  7.9999998E-002
TIMES_AND_VALUES = (0000000000 00:00.133)  8.9999996E-002
TIMES_AND_VALUES = (0000000000 00:00.150)  9.9999994E-002
TIMES_AND_VALUES = (0000000000 00:00.167)  1.1000000E-001
TIMES_AND_VALUES = (0000000000 00:00.183)  1.2000000E-001
TIMES_AND_VALUES = (0000000000 00:00.200)  1.3000000E-001
TIMES_AND_VALUES = (0000000000 00:00.217)  1.4000000E-001
TIMES_AND_VALUES = (0000000000 00:00.233)  1.4999999E-001
TIMES_AND_VALUES = (0000000000 00:00.250)  1.6000000E-001
TIMES_AND_VALUES = (0000000000 00:00.267)  1.7000000E-001
TIMES_AND_VALUES = (0000000000 00:00.283)  1.7999999E-001
TIMES_AND_VALUES = (0000000000 00:00.300)  1.9000000E-001
TIMES_AND_VALUES = (0000000000 00:00.317)  1.9999999E-001
TIMES_AND_VALUES = (0000000000 00:00.333)  2.0999999E-001
TIMES_AND_VALUES = (0000000000 00:00.350)  2.2000000E-001
TIMES_AND_VALUES = (0000000000 00:00.367)  2.2999999E-001
TIMES_AND_VALUES = (0000000000 00:00.383)  2.3999999E-001
```

TIMES_AND_VALUES = (0000000000 00:00.400) 2.5000000E-001
TIMES_AND_VALUES = (0000000000 00:00.417) 2.5999999E-001
TIMES_AND_VALUES = (0000000000 00:00.433) 2.6999998E-001
TIMES_AND_VALUES = (0000000000 00:00.450) 2.8000000E-001
TIMES_AND_VALUES = (0000000000 00:00.467) 2.8999999E-001
TIMES_AND_VALUES = (0000000000 00:00.483) 2.9999998E-001
TIMES_AND_VALUES = (0000000000 00:00.500) 3.1000000E-001
TIMES_AND_VALUES = (0000000000 00:00.517) 3.1999999E-001
TIMES_AND_VALUES = (0000000000 00:00.533) 3.2999998E-001
TIMES_AND_VALUES = (0000000000 00:00.550) 3.4000000E-001
TIMES_AND_VALUES = (0000000000 00:00.567) 3.4999999E-001
TIMES_AND_VALUES = (0000000000 00:00.583) 3.5999998E-001
TIMES_AND_VALUES = (0000000000 00:00.600) 3.7000000E-001
TIMES_AND_VALUES = (0000000000 00:00.617) 3.8000000E-001
TIMES_AND_VALUES = (0000000000 00:00.633) 3.8999999E-001
TIMES_AND_VALUES = (0000000000 00:00.650) 3.9999998E-001
TIMES_AND_VALUES = (0000000000 00:00.667) 4.1000000E-001
TIMES_AND_VALUES = (0000000000 00:00.683) 4.1999999E-001
TIMES_AND_VALUES = (0000000000 00:00.700) 4.2999998E-001
TIMES_AND_VALUES = (0000000000 00:00.717) 4.4000000E-001
TIMES_AND_VALUES = (0000000000 00:00.733) 4.4999999E-001
TIMES_AND_VALUES = (0000000000 00:00.750) 4.5999998E-001
TIMES_AND_VALUES = (0000000000 00:00.767) 4.7000000E-001
TIMES_AND_VALUES = (0000000000 00:00.783) 4.7999999E-001
TIMES_AND_VALUES = (0000000000 00:00.800) 4.8999998E-001
TIMES_AND_VALUES = (0000000000 00:00.817) 5.0000000E-001
TIMES_AND_VALUES = (0000000000 00:00.833) 5.0999999E-001
TIMES_AND_VALUES = (0000000000 00:00.850) 5.1999998E-001
TIMES_AND_VALUES = (0000000000 00:00.867) 5.2999997E-001
TIMES_AND_VALUES = (0000000000 00:00.883) 5.3999996E-001
TIMES_AND_VALUES = (0000000000 00:00.900) 5.5000001E-001
TIMES_AND_VALUES = (0000000000 00:00.917) 5.6000000E-001
TIMES_AND_VALUES = (0000000000 00:00.933) 5.6999999E-001
TIMES_AND_VALUES = (0000000000 00:00.950) 5.7999998E-001
TIMES_AND_VALUES = (0000000000 00:00.967) 5.8999997E-001
TIMES_AND_VALUES = (0000000000 00:00.983) 5.9999996E-001
TIMES_AND_VALUES = (0000000000 00:01.000) 6.1000001E-001
TIMES_AND_VALUES = (0000000000 00:01.017) 6.2000000E-001
TIMES_AND_VALUES = (0000000000 00:01.033) 6.3000000E-001
TIMES_AND_VALUES = (0000000000 00:01.050) 6.3999999E-001
TIMES_AND_VALUES = (0000000000 00:01.067) 6.4999998E-001
TIMES_AND_VALUES = (0000000000 00:01.083) 6.5999997E-001
TIMES_AND_VALUES = (0000000000 00:01.100) 6.6999996E-001
TIMES_AND_VALUES = (0000000000 00:01.117) 6.8000001E-001
TIMES_AND_VALUES = (0000000000 00:01.133) 6.9000000E-001
TIMES_AND_VALUES = (0000000000 00:01.150) 6.9999999E-001
TIMES_AND_VALUES = (0000000000 00:01.167) 7.0999998E-001
TIMES_AND_VALUES = (0000000000 00:01.183) 7.1999997E-001
TIMES_AND_VALUES = (0000000000 00:01.200) 7.2999996E-001
TIMES_AND_VALUES = (0000000000 00:01.217) 7.4000001E-001
TIMES_AND_VALUES = (0000000000 00:01.233) 7.5000000E-001
TIMES_AND_VALUES = (0000000000 00:01.250) 7.5999999E-001
TIMES_AND_VALUES = (0000000000 00:01.267) 7.6999998E-001
TIMES_AND_VALUES = (0000000000 00:01.283) 7.7999997E-001
TIMES_AND_VALUES = (0000000000 00:01.300) 7.8999996E-001
TIMES_AND_VALUES = (0000000000 00:01.317) 7.9999995E-001
TIMES_AND_VALUES = (0000000000 00:01.333) 8.1000000E-001
TIMES_AND_VALUES = (0000000000 00:01.350) 8.1999999E-001
TIMES_AND_VALUES = (0000000000 00:01.367) 8.2999998E-001
TIMES_AND_VALUES = (0000000000 00:01.383) 8.3999997E-001
TIMES_AND_VALUES = (0000000000 00:01.400) 8.4999996E-001
TIMES_AND_VALUES = (0000000000 00:01.417) 8.5999995E-001
TIMES_AND_VALUES = (0000000000 00:01.433) 8.7000000E-001
TIMES_AND_VALUES = (0000000000 00:01.450) 8.8000000E-001
TIMES_AND_VALUES = (0000000000 00:01.467) 8.8999999E-001
TIMES_AND_VALUES = (0000000000 00:01.483) 8.9999998E-001
TIMES_AND_VALUES = (0000000000 00:01.500) 9.0999997E-001
TIMES_AND_VALUES = (0000000000 00:01.517) 9.1999996E-001
TIMES_AND_VALUES = (0000000000 00:01.533) 9.3000001E-001

```
TIMES_AND_VALUES = (0000000000 00:01.550)  9.4000000E-001
TIMES_AND_VALUES = (0000000000 00:01.567)  9.4999999E-001
TIMES_AND_VALUES = (0000000000 00:01.583)  9.5999998E-001
TIMES_AND_VALUES = (0000000000 00:01.600)  9.6999997E-001
TIMES_AND_VALUES = (0000000000 00:01.617)  9.7999996E-001
TIMES_AND_VALUES = (0000000000 00:01.633)  9.8999995E-001
TIMES_AND_VALUES = (0000000000 00:01.650)  1.0000000E+000
TIMES_AND_VALUES = (0000000000 00:01.667)  1.0100000E+000
TIMES_AND_VALUES = (0000000000 00:01.683)  1.0200000E+000
TIMES_AND_VALUES = (0000000000 00:01.700)  1.0300000E+000
TIMES_AND_VALUES = (0000000000 00:01.717)  1.0400000E+000
TIMES_AND_VALUES = (0000000000 00:01.733)  1.0500000E+000
TIMES_AND_VALUES = (0000000000 00:01.750)  1.0599999E+000
TIMES_AND_VALUES = (0000000000 00:01.767)  1.0699999E+000
TIMES_AND_VALUES = (0000000000 00:01.783)  1.0799999E+000
TIMES_AND_VALUES = (0000000000 00:01.800)  1.0900000E+000
TIMES_AND_VALUES = (0000000000 00:01.817)  1.1000000E+000
TIMES_AND_VALUES = (0000000000 00:01.833)  1.1100000E+000
TIMES_AND_VALUES = (0000000000 00:01.850)  1.1200000E+000
TIMES_AND_VALUES = (0000000000 00:01.867)  1.1300000E+000
TIMES_AND_VALUES = (0000000000 00:01.883)  1.1400000E+000
TIMES_AND_VALUES = (0000000000 00:01.900)  1.1500000E+000
TIMES_AND_VALUES = (0000000000 00:01.917)  1.1600000E+000
TIMES_AND_VALUES = (0000000000 00:01.933)  1.1700000E+000
TIMES_AND_VALUES = (0000000000 00:01.950)  1.1799999E+000
TIMES_AND_VALUES = (0000000000 00:01.967)  1.1899999E+000
TIMES_AND_VALUES = (0000000000 00:01.983)  1.1999999E+000
```

6 Appendices

6.1 Log-sheet

document version	date	program version	Changes with respect to the previous version
1.0	25-05-97	1.00	DONS02BO: initial version
1.1	25-03-98	1.00	P98043: adaptations to general standards
1.2	17-12-98	1.00	W98004: allow relative path's in SIMONA input files.
2.0	28-07-99	1.01	WPWORD02: conversion from WP5.1 to WORD 7.0
2.1	21-02-00	1.02	P98050: Logo changed to RWS
2.2	22-01-03	1.02	P02015: conversion from Word95 to Word2000
2.3	27-03-03	1.02	P03007: improvements (export 2003-01)
2.4	25-06-03	1.02	P03014: 'exportverbeteringen' (export 2003-01)
2.5	14-11-03	1.02	P03047: 'exportverbeteringen' (export 2003-02)
2.6	03-08-05	1.03	M04040: note added for BOX-restriction
2.7	21-11-07	?	c57213: eliminate usage of sdsexp

6.2 Trade marks

HP is a registrated trade mark of the Hewlett-Packard Company.

SUN is a registrated trade mark of SUN Microsystems.

6.3 Figures

6.4 Tables

6.5 Glossary of terms

DIF file	Conventional file for exchange of data between several applications.
Experiment	A complete run of an application including the input and output. So a change
DONAR	A system for storage, management and presentation of water management data.
MUX	A multiplexed observation type. For example, the individual measurements which make up a wave spectrum.
SDS-file	SIMONA permanent data storage file.
SIMONA	An architecture for preprocessing, memory management, data storage and postprocessing.
SIMONA include file	ASCII file that can be included in an input file for a SIMONA application

6.6 Index