

Modelit  
Elisabethdreef 5  
4101 KN Culemborg  
+31(345)531717



info@modelit.nl  
www.modelit.nl

# **Modelit Donar Interface Toolbox for Matlab**

Version: 2008\_01  
Date: August 18, 2008

# ModeliT

Version: 2008\_01  
Date: August 18, 2008  
Manual: Modelit Donar Interface Toolbox for Matlab  
Authors: Nanne van der Zijpp  
Kees-Jan Hoogland  
Copyright: 2008, Modelit  
Contact: [info@modelit.nl](mailto:info@modelit.nl)  
[www.modelit.nl](http://www.modelit.nl)

## Table of Contents

1	Introduction.....	1
2	Reference manual.....	2
2.1	bepaal_tijdstap.m .....	2
2.2	cmp_taxis.m.....	2
2.3	combineRKS.m .....	2
2.4	combineTPS.m .....	3
2.5	ComposeDialList.m .....	3
2.6	datenum2long.m .....	3
2.7	datenum2str.m .....	4
2.8	defaultdia.m .....	4
2.9	dia_merge.m .....	4
2.10	diafieldnames.m .....	5
2.11	dimspecs.m .....	5
2.12	displayStations.m .....	6
2.13	duration.m .....	6
2.14	emptydia.m .....	7
2.14.1	emptyblok.m .....	7
2.14.2	emptyMUX.m .....	7
2.14.3	emptyRGH.m .....	8
2.14.4	emptyRKS.m .....	8
2.14.5	emptyTPS.m .....	9
2.14.6	emptyW3H.m .....	9
2.14.7	emptyWRD.m .....	10
2.15	interp_blok.m .....	11
2.16	long2datenum.m .....	11
2.17	matroos2dia.m .....	11
2.18	mergeDias.m .....	12
2.19	meta2str.m .....	12
2.20	readdia_R14.m .....	12
2.21	readqinsy.m .....	13
2.22	readrwslod.m .....	13
2.23	readstruct.m .....	14
2.24	select_interval.m .....	14
2.25	setValidatieStatus.m.....	15
2.26	set_taxis.m .....	15
2.27	splitlongdate.m .....	16
2.28	writedia_R14.m.....	16



## 1 Introduction

DONAR is het centrale informatie systeem voor de gegevens van de 'natte' Rijkswaterstaat. Gegevens uit dit informatie systeem worden gebruikt door een groot aantal applicaties die door Modelit ontwikkeld zijn:

- Maria: Morfologische gegevens;
- WavixIV: Golfgegevens, windgegevens en waterstanden;
- Melissa: Waterstanden en –temperaturen;
- DialInspector: Golfgegevens, windgegevens, waterstanden, morfologische gegevens.

De uitwisseling van gegevens tussen DONAR en de door Modelit ontwikkelde applicaties vindt plaats door middel van Donar Interface Asciifiles (DIA's). Modelit heeft een aantal utilities ontwikkeld voor het importeren en exporteren van dit type bestanden. Binnen de door Modelit ontwikkelde applicaties wordt gewerkt met een Matlab representatie van deze bestanden. Daarbij blijven alle data en metadata die opgeslagen zijn in Donar Interface Files beschikbaar binnen deze applicaties. Op deze manier blijft het dus mogelijk om aangepaste data weer te exporteren naar Donar.

De import en exportroutines zijn gebaseerd op de zogenaamde DIM modules die RWS in onderhoud heeft. Dit is een verzameling C-functies die zijn gedocumenteerd in het document "Gebruikershandleiding DONAR Deel 7: Applicatie Programmatuur". De functionaliteit van deze programmatuur wordt via MEX files beschikbaar gemaakt. Het gebruik van deze C-programmatuur heeft als voordeel ten opzichte van zelfgeschreven importfilters in Matlab is dat zeer grote snelheden bereikt worden bij inlezen en wegschrijven en dat de gebruikte programmatuur grondig getest is.

Naast routines voor de import en export heeft Modelit ook routines ontwikkeld voor een aantal veel voorkomende bewerkingen zoals het samenvoegen, splitsen, inspecteren en aanpassen van Donar Interface Files. In sectie 2 wordt een overzicht gegeven van alle beschikbare routines. Per routine wordt daarbij een korte omschrijving gegeven van het gebruik van de routine, een beschrijving van de invoer en uitvoer argumenten en eventuele voorbeelden van gebruik.

## 2 Reference manual

### 2.1 bepaal\_tijdstap.m

SUMMARY	Determine timestep of given time axis.
CALL	<code>[tjdstapeenheid, tijdstap] = bepaal_tijdstap(taxis, mode)</code>
INPUT	<p>taxis: Vector of Matlab datenum.</p> <p>mode: (Optional) string with possible values:  “TE”: (default) Assume an equidistant timeseries, tijdstap is the smallest found timestep, this is useful when the timeseries has missing values.  Other: If timestep is always equal → TE timeseries.  Otherwise → TN timeseries.</p>
OUTPUT	<p>tjdstapeenheid: String with possible values, empty if TN timeseries:  “d”: days.  “min”: minutes.  “s”: seconds.  “cs”: centiseconds.</p> <p>tijdstap: Integer with timestep in tjdstapeenheid units, empty if TN timeseries.</p>
SEE ALSO	<code>cmp_taxis, set_taxis</code>

### 2.2 cmp\_taxis.m

SUMMARY	Compute time axis for Donar timeseries.
CALL	<code>taxis = cmp_taxis(s, N, SIGNIFIKANTIE)</code>
INPUT	<p>s: structure with the following relevant fields (Donar RKS block).  IBegdat: e.g. 19980101  iBegtyd: e.g. 1430  sTydehd: 'min'  iTydstp: 10  IEnddat: 19980228  iEndtyd: 2350</p> <p>N: (Optional) total number of datapoints datapunten, for checking.</p> <p>SIGNIFIKANTIE: (Optional) timeaxis precision, default value: 1440 (minutes), if necessary specify second argument N as [].</p>
OUTPUT	<p>taxis: Vector of Matlab datenum with the equidistant times.</p>
SEE ALSO	<code>select_interval</code>

### 2.3 combineRKS.m

SUMMARY	Combine two or more RKS (Reeksadministratie) blocks.
CALL	<code>RKS = combineRKS(oldRKS, newRKS)</code>
INPUT	oldRKS:

	Struct or struct array with one or more existing RKS blocks. newRKS: Struct or struct array with RKS block to be added.
OUTPUT	RKS: Structure with combined RKS blocks.
SEE ALSO	<b>emptyRKS</b>

## 2.4 combineTPS.m

SUMMARY	Combine two or more TPS (Tijdreeksperiodestatus) blocks.
CALL	<code>TPS = combineTPS(oldTPS, newTPS, RKS)</code>
INPUT	oldTPS: Struct or struct array with one or more existing TPS blocks. newTPS: Struct or struct array with TPS block to be added. RKS: RKS block to determine timestep.
OUTPUT	TPS: Structure with combined TPS blocks.
SEE ALSO	<b>emptyTPS</b>

## 2.5 ComposeDialList.m

SUMMARY	Make a list of DIA structures that can be displayed in a Java table.
CALL	<code>Contents = ComposeDialList(dialist, fields)</code>
INPUT	dialist: Struct array with Donar data blocks, see emptyblok for the format. fields: Cellstring, information to be displayed in table, possible values: - "Locatiecode", "Locatie" - "Parameter" - "Veldapparaat" - "Analysecode" - "Tijdstap" - "Begindatum" - "Einddatum"
OUTPUT	Contents: Structure with fields: - header: Cellstring with columnnames. - data: Cell array with data.
SEE ALSO	<b>jacontrol</b>

## 2.6 datenum2long.m

SUMMARY	Convert Matlab datenum to date with format YYYYMMDD, time with format HHmm and time with format HHmmSS.
CALL	<code>[Date, Time, LongTime] = datenum2long(D, timeunit)</code>
INPUT	D: Scalar, vector or matrix with datenum data. timeunit: Opional argument with possible values:

	<ul style="list-style-type: none"> <li>- "mnd": Donar uses different format for months.</li> <li>- otherwise: Use standard Donar date format.</li> </ul>
OUTPUT	<p>Date: Corresponding date(s) in YYYYMMDD.</p> <p>Time: Corresponding time(s) in HHmm.</p> <p>LongTime: Corresponding time(s) in HHmmSS.</p>
SEE ALSO	<b>long2datenum</b>

## 2.7 datenum2str.m

SUMMARY	Convert a Matlab datenum to a string with format: YYYYMMDDHHmm.
CALL	<code>str = datenum2str(datum)</code>
INPUT	<p>datum: Matlab datenum representation of date.</p>
OUTPUT	<p>str: String with date of form: YYYYMMDDHHmm.</p>
SEE ALSO	<b>datestr, datenum, datenum2long</b>

## 2.8 defaultdia.m

SUMMARY	Fill dia with default values.
CALL	<code>S = defaultdia(S)</code>
INPUT	<p>S: DIA structure.</p>
OUTPUT	<p>S: DIA structure with default values.</p>
SEE ALSO	<b>Dimspeccs, emptydia</b>

## 2.9 dia\_merge.m

SUMMARY	Merge two equidistant timeseries.
CALL	<code>[dia_new, missing, total] = dia_merge(dia_old, ... dia_new, SIGNIFIKANTIE, copyhiaat)</code>
INPUT	<p>dia_old: Structure with existing DIA.</p> <p>dia_new: Structure with DIA to be added (overwrite when necessary).</p> <p>SIGNIFIKANTIE: (Optional) integer with time axis precision, e.g. 1440 for minutes.</p> <p>copyhiaat: (Optional) True(default) → overwrite existing dia with missing values. False → do not overwrite existing dia with missing values.</p>
OUTPUT	<p>dia_new: Structure with merged timeseries.</p> <p>missing: Integer with total number of values which could not be filled in the new time axis.</p> <p>total: Integer with total number of new datapoints in new taxis.</p>



SEE ALSO	<code>mergeDias</code>
----------	------------------------

## 2.10 diafieldnames.m

SUMMARY	This function specifies the fields in the Matlab structure for all the 'regelidentificaties' in the Donar file.
CALL	<code>S = diafieldnames</code>
INPUT	No input required.
OUTPUT	<p>S:</p> <p>Structure with fields:</p> <pre> +----W3H     +----OGI (char array)     +----BHI (char array)     +----ANI (char array)     +----BMI (char array)     +----MUX (char array)     +----WNS (char array)     +----PAR (char array)     +----EHD (char array)     +----CPM (char array)     +----HDH (char array)     +----ORG (char array)     +----SGK (char array)     +----IVS (char array)     +----BTX (char array)     +----BTN (char array)     +----LOC (char array)     +----GBD (char array)     +----GRD (char array)     +----RAI (char array)     +----ANA (char array)     +----BEM (char array)     +----BEW (char array) +----RKS     +----TYD (char array)     +----PLT (char array)     +----SYS (char array)     +----VGS (char array)     +----STA (char array)     +----BGS (char array)     +----BLS (char array)     +----INV (char array)     +----VZM (char array)     +----SVZ (char array)     +----SSV (char array)     +----SSS (char array) </pre>
SEE ALSO	<code>readdia_R14</code>

## 2.11 dimspeccs.m

SUMMARY	Read fieldnames as used in the Donar Interface Modules.
CALL	<code>[veld_empty, veld_0, veld_999, veld_99, veld_NVT, veld_NietVanToepassing] = dimspeccs(blok)</code>
INPUT	<p>blok:</p> <p>Structure with Donar data blok, with supported fields: 'W3H', 'MUX', 'SGK', 'RGH', 'TYP', 'RKS', 'TPS', 'WRD'.</p>

OUTPUT	<pre>veld_empty:   fields with defaultvalue ''. veld_0:   fields with defaultvalue 0. veld_999:   fields with defaultvalue -99. veld_99:   fields with defaultvalue -999999999. veld_NVT:   fields with defaultvalue 'NVT'. veld_NietVanToepassing:   fields with defaultvalue 'Niet Van Toepassing'.</pre>
SEE ALSO	<b>defaultdia</b>

## 2.12 displayStations.m

SUMMARY	Display the stations contained in a dia block in a specified axis.
CALL	<code>displayStations(h_map, blok, labels, S)</code>
INPUT	<p><b>h_map:</b> Handle of axis in which to plot the stations.</p> <p><b>blok:</b> Struct array with location information, see emptyblok for format of a dia block.</p> <p><b>labels:</b> true → plot station labels and location. false → plot station location only.</p> <p><b>S:</b> Struct array with markup for stations, with fields:</p> <ul style="list-style-type: none"> <li>- color: Colour triple [r g b].</li> <li>- markerfacecolor: Colour triple [r g b].</li> <li>- marker: String, see Matlab plot function.</li> <li>- markersize Integer.</li> <li>- fontsize Integer.</li> <li>- legenda String to be displayed in legend.</li> <li>- linewidth Integer indicating the width of the marker edge.</li> <li>- callback Function handle of function to call when the station is clicked on.</li> <li>- locatie Char array with stationcodes(sLoccod), use “default” to specify default marker.</li> </ul>
OUTPUT	No direct output, the stations specified in the dia blocks are displayed in the axis with the specified markers.
SEE ALSO	<b>emptyblok</b>

## 2.13 duration.m

SUMMARY	Calculate duration of a timeunit in Matlab datenum units.
CALL	<code>d = duration(timeunit)</code>
INPUT	<p><b>timeunit:</b> String with possible values:</p> <ul style="list-style-type: none"> <li>• “mnd” months;</li> <li>• “d” days;</li> <li>• “min” minutes;</li> </ul>

	<ul style="list-style-type: none"> <li>• “uur” hours;</li> <li>• “cs” centiseconds.</li> </ul>
OUTPUT	d: Duration of the given timeunit in Matlab datenum units.
SEE ALSO	<b>cmp_taxis</b>

## 2.14 emptydia.m

SUMMARY	emptydia - Create an empty dia.
CALL	<code>S = emptydia(n)</code>
INPUT	n: Number of blocks filled with default values, default value: 0.
OUTPUT	S: Dia Structure, with fields: <pre> +----IDT     +----sFiltyp (char)     +----sSyscod (char)     +----lCredat (double)     +----sCmtrgl (char) +----blok +----W3H (struct): see emptyW3H +----MUX (struct): empty, see emptyMUX +----TYP (struct): empty +----RGH (struct): empty, see emptyRGH +----RKS (struct): see emptyRKS +----TPS (struct): empty, see emptyTPS +----WRD (struct): see emptyWRD </pre>
EXAMPLE	<code>s=emptydia(1);</code> <CHANGE STRUCTURE s> <code>writedia_R14(s, 'dia.dia');</code>
SEE ALSO	<b>readdia_R14, writedia_R14, emptyblok, emptyW3H, emptyWRD, emptyMUX, emptyTPS</b>

### 2.14.1 emptyblok.m

SUMMARY	Make an empty Donar data block.
CALL	<code>blok = emptyblok</code>
INPUT	No input required.
OUTPUT	blok: Donar data block, with the following required partial data blocks: <ul style="list-style-type: none"> <li>- W3H</li> <li>- RKS</li> <li>- WRD (must contain at least one row of data)</li> </ul> optional partial data blocks: <ul style="list-style-type: none"> <li>- MUX</li> <li>- TYP</li> <li>- TPS</li> </ul>
SEE ALSO	<b>readdia_R14, writedia_R14, emptyDia, emptyW3H, emptyWRD, emptyMUX, emptyTPS</b>

### 2.14.2 emptyMUX.m

SUMMARY	Make default MUX (Multiplex-administratie) block.
CALL	<code>MUX = emptyMUX</code>

INPUT	No input required.
OUTPUT	<p>MUX:</p> <p>Structure with fields:</p> <pre> +----iKannum (double) +----lWnsnum (double) +----sParcod (char) +----sParoms (char) +----sStaind (char) +----nCpmcod (double) +----sCpmoms (char) +----sDomein (char) +----sEhdcod (char) +----sOrgcod (char) +----sOrgoms (char) +----sHdhcod (char) +----sHdhoms (char) +----sSgkcod (char) </pre>
SEE ALSO	<b>emptyblok</b>

### 2.14.3 emptyRGH.m

SUMMARY	Make default MUX (Raaigeldigheid-administratie) block.
CALL	<code>RGH = emptyRGH(blok)</code>
INPUT	<p>blok:</p> <p>(Optional) struct array with DIA blocks, if specified create RGH blok based on known RGH blocks.</p>
OUTPUT	<p>RGH:</p> <p>Structure with fields:</p> <pre> +----lBegdat: begin datum +----lEnddat: eind datum (0: oneindig) +----Raaitype (sRaityp):     H: hulpstrandhoofdraai     S: strandhoofdraai     N: normale raai +----Raaiklasse (sRaikls):     L: landelijke raai     V: vaklodingsraai     P: ? </pre>
SEE ALSO	<b>emptyblok</b>

### 2.14.4 emptyRKS.m

SUMMARY	Make default RKS (Reeksadministratie) block.
CALL	<code>RKS = emptyRKS</code>
INPUT	No input required.
OUTPUT	<p>RKS:</p> <p>Structure with fields:</p> <pre> +----sRefvlk (char) +----lBemhgt (double) +----lBegdat (double) +----iBegtyd (double) +----sSyscod (char) +----sTydehd (char) +----iTdstp (double) +----lXcrdgs (double) +----lYcrdgs (double) </pre>

	<pre> +----lVakstp (double) +----lEnddat (double) +----iEndtyd (double) +----sRkssta (char) +----lBeginv (double) +----lEndinv (double) +----sVzmcod (char) +----sVzmoms (char) +----sSvzcod (char) +----sSvzoms (char) +----sSsvcod (char) +----sSsvoms (char) +----sSsscod (char) +----sSssoms (char) +----lXcrdwb (double) +----lYcrdzb (double) +----lXcrdob (double) +----lYcrdnb (double) +----lXcrdmn (double) +----lYcrdmn (double) +----lXcrdmx (double) +----lYcrdmx (double) </pre>
SEE ALSO	<b>emptyblok</b>

### 2.14.5 emptyTPS.m

SUMMARY	Make default TPS (Tijdreeksperiode-administratie) block.
CALL	<code>TPS = emptyTPS</code>
INPUT	No input required.
OUTPUT	<p>TPS:</p> <p>Structure with fields:</p> <pre> +----lBegdat (double) +----iBegtyd (double) +----lEnddat (double) +----iEndtyd (double) +----sRkssta (char) </pre>
SEE ALSO	<b>emptyblok</b>

### 2.14.6 emptyW3H.m

SUMMARY	Make default W3H (W3H administratie) block.
CALL	<code>W3H = emptyW3H</code>
INPUT	No input required.
OUTPUT	<p>W3H:</p> <p>Structure with fields:</p> <pre> +----sMuxcod (char) +----sMuxoms (char) +----lWnsnum (double) +----sParcod (char) +----sParoms (char) +----sStaind (char) +----nCpmcod (double) +----sCpmoms (char) +----sDomein (char) +----sEhdcod (char) +----sHdhcod (char) </pre>

	<pre> +----sHdhoms (char) +----sOrgcod (char) +----sOrgoms (char) +----sSgkcod (char) +----sIvscod (char) +----sIvsoms (char) +----sBtccod (char) +----sBtlcod (char) +----sBtxoms (char) +----sBttnam (char) +----sAnicod (char) +----sAnioms (char) +----sBhicol (char) +----sBhioms (char) +----sBmicod (char) +----sBmioms (char) +----sOgicod (char) +----sOgioms (char) +----sGbdcod (char) +----sGbdoms (char) +----sLoccod (char) +----sLocoms (char) +----sLocsrt (char) +----sCrdtyp (char) +----lXcrdgs (double) +----lYcrdgs (double) +----lGhoekg (double) +----lRhoekg (double) +----lMetrng (double) +----lStraal (double) +----lXcrdmp (double) +----lYcrdmp (double) +----sOmloop (char) +----sAnacod (char) +----sAnaoms (char) +----sBemcod (char) +----sBemoms (char) +----sBewcod (char) +----sBewoms (char) +----sVatcod (char) +----sVatoms (char) +----sRkstyp (char) </pre>
SEE ALSO	<b>emptyblok</b>

### 2.14.7 emptyWRD.m

SUMMARY	Make default WRD (Waarde) block.
CALL	<code>WRD = emptyWRD</code>
INPUT	No input required.
OUTPUT	<pre> WRD:   Structure with fields:     +----taxis (double)     +----lKeynr2 (double)     +----Wrd (double)     +----nKwlcod (double) </pre>
SEE ALSO	<b>emptyblok</b>

### 2.15 interp\_blok.m

SUMMARY	Interpolate Donar block to new time axis.
CALL	<code>blok = interp_blok(blok, taxis, mode)</code>
INPUT	<p><b>blok:</b> Structure with Donar data block, see emptyblok for format.</p> <p><b>taxis:</b> Vector of Matlab datenums.</p> <p><b>mode:</b> String with possible values: "all" - Estimate all points not in taxis AND missing values. other - Estimate only missing values.</p>
OUTPUT	<p><b>blok:</b> Structure with Donar data block, see emptyblok for format.</p>
SEE ALSO	<code>cmp_taxis</code> , <code>emptyblok</code>

### 2.16 long2datenum.m

SUMMARY	Convert two Longs with date with format YYYYMMDD and time with format HHmm to Matlab datenum format.
CALL	<code>taxis = long2datenum(taxisdatum, taxistime, timeunit)</code>
INPUT	<p><b>taxisdate:</b> Vector of Long with format YYYYMMDD.</p> <p><b>taxistime:</b> Vector of Long with format HHmm.</p>
OUTPUT	<p><b>taxis:</b> Vector with corresponding values in Matlab datenum format.</p>
SEE ALSO	<code>datenum2long</code>

### 2.17 matroos2dia.m

SUMMARY	Retrieve and convert timeseries from the matroos database.
CALL	<code>[dia message] = matroos2dia(stuurfilename, metafilename, diafilename)</code>
INPUT	<p><b>stuurfilename:</b> String with name of the file with timeseries to get from matroos.</p> <p><b>metafilename:</b> String with name of the file with metainfo (DIA).</p> <p><b>diafilename:</b> String with name of the file to which the DIA should be epxorted.</p>
OUTPUT	<p><b>dia:</b> Structure, for format see emptydia, empty on error.</p> <p><b>message:</b> String with message if error has occurred.</p>
EXAMPLE	<p>Format of the settingsfile:</p> <pre>sLoccod sParcod sVatcod source loc &lt;string&gt; &lt;string&gt; &lt;string&gt; &lt;string&gt; &lt;string&gt; HUIBGT WINDRTG FASTRCDR knmi_noos huibertgat  Unit          tstart          tstop &lt;string&gt;      &lt;string&gt;        &lt;string&gt;</pre>

	<code>Wind_direction 200701010000 200702010000</code>
SEE ALSO	<code>emptydia</code> , <a href="http://matroos2/direct/get_series.php?">http://matroos2/direct/get_series.php?</a>

## 2.18 mergeDias.m

SUMMARY	Merge two DIA structures and merge blocks with identical sLoccod, sParcod, sVatcod and sRkstyp == 'TE'.
CALL	<code>dia = mergeDias(dia, extraDia, SIGNIFIKANTIE)</code>
INPUT	dia: Structure with original DIA. extraDia: Structure with Dia to be added.
OUTPUT	dia: Structure with merged dia.
SEE ALSO	<code>emptydia</code> , <code>emptyblok</code> , <code>dia_merge</code>

## 2.19 meta2str.m

SUMMARY	Print BLOK header to string.
CALL	<code>str = meta2str(BLOK, Blokfields)</code>
INPUT	BLOK: Donar data block structure. Blokfields: Structure with fields to be printed, see routine diafieldnames.
OUTPUT	str: String with block header with metainfo.
SEE ALSO	<code>diafieldnames</code>

## 2.20 readdia\_R14.m

SUMMARY	Read a DIA file to a Matlab structure.
CALL	<code>data = readdia_R14(fname)</code>
INPUT	fname: String with the name of the DIA file to be read.
OUTPUT	data: Dia Structure (empty on error), with fields: +----IDT   +----sFilty (char)   +----sSyscod (char)   +----lCredat (double)   +----sCmtrgl (char) +----blok +----W3H (struct): see emptyW3H +----MUX (struct): empty, see emptyMUX +----TYP (struct): empty +----RGH (struct): empty, see emptyRGH +----RKS (struct): see emptyRKS +----TPS (struct): empty, see emptyTPS +----WRD (struct): see emptyWRD
SEE ALSO	<code>writedia_R14</code>



## 2.21 readqinsy.m

SUMMARY	Scan or read Qinsy datafile using mex-file
CALL	<code>locs = readqinsy(fname)</code> (scan file) <code>locs = readqinsy(fname, locs)</code> (read file)
INPUT	<code>fname</code> : <string> Name of qinsy file. If this is the only input argument, the output argument will only contain the header data. <code>locs</code> : <struct array> This value must be equal to or a subset of the output of an earlier call to <code>readqinsy</code> . It tells the function which data to retrieve from file.
OUTPUT	<code>locs</code> : Struct array with 1 record per location: +----sLoccod (char array) +----marker (double) +----aantal (double) +----lBegdat (double) +----iBegtyd (double) +----lEnddat (double) +----iEndtyd (double) +----xy (double) +----z (double)  If "readqinsy" was called with 1 argument only the header data will be nonempty (sLoccod, marker and aantal).  If called with 2 input arguments also the remaining (data) fields are returned nonempty.
EXAMPLE	<code>locs = readqinsy(fname) ;</code> <code>data= readqinsy(fname,locs(2:2:end)) ;</code>
SEE ALSO	<code>readrwslod</code>

## 2.22 readrwslod.m

SUMMARY	Scan or read rwslod data file using mex file
CALL	<code>locs = readrwslod(fname, locs)</code>
INPUT	<code>fname</code> : <string> Name of rwaslod file. If this is the only input argument, the output argument will only contain the header data. <code>locs</code> : <struct array> This value must be equal to or a subset of the output of an earlier call to <code>readrwslod</code> . It tells the function which data to retrieve from file.
OUTPUT	<code>locs</code> : <struct array> If called with only filename (scanning mode), <code>readrwslod</code> will only return the header fields. A struct array with the following fields is returned: +----longdate (double) +----sLoccod (char array) +----raailocatie (double array) +----aantal (double) +----marker (double)  If called with 2 arguments (read mode) a more elaborate struct array is returned: +----longdate (double) +----sLoccod (char array) +----raailocatie (double array)

	<pre> +----aantal (double) +----marker (double) +----xy (int32 array) +----z (int32 array) +----lBegdat (double) +----iBegtyd (double) +----lEnddat (double) +----iEndtyd (double) </pre>
EXAMPLE	<pre> locs = readrwslod(fname); data= readrwslod(fname,locs(2:2:end)); </pre>
SEE ALSO	<b>readqinsy</b>

### 2.23 readstruct.m

SUMMARY	Alternative for readdia_R14. Read only Donar MetaInfo.
CALL	Defaultopt = readstruct(fname)
INPUT	fname: (Optional) name of headerfile, default value = 'defaultheader.hdr'.
OUTPUT	Defaultopt: Structure with metainfo.
SEE ALSO	<b>readdia_R14</b>

### 2.24 select\_interval.m

SUMMARY	GUI for selecting, extending, narrowing, refining or coarsing the timeinterval for the current selected timeseries.
CALL	[begintijd, eindtijd, tijdstap, aggregatie] = select_interval(begintijd,eindtijd,tijdstap,tijdstapkeuzes)
INPUT	<p>begintijd: Datum with the original startingtime, is shown in GUI.</p> <p>eindtijd: Datum with the original endingtime, is shown in GUI.</p> <p>tijdstap: Integer with the original timestep in minutes, NaN if orginal timestep should not be changed, is shown in GUI.</p> <p>tijdstapkeuzes: Structure with fields:  <ul style="list-style-type: none"> <li>- "tijdstapstring": Cellstring with text to appear in combobox with timechoices.</li> <li>- "tijdstapvalue": Vector with timestep in Matlab datum format, default values: 10 min, 1 en 3 hour and one day.</li> </ul> </p>
OUTPUT	<p>begintijd: Datum with the new startingtime.</p> <p>eindtijd: Datum with the new endingtime.</p> <p>tijdstap: Integer with new timestep in minutes, NaN if original timestep should be used.</p> <p>aggregatie: Boolean, TRUE → aggregate timeseries if possible.</p>
SEE ALSO	<b>interp_blok</b>

## 2.25 setValidatieStatus.m

<b>SUMMARY</b>	Set status for timeseries in Waddenzee.
<b>CALL</b>	<code>dia = setValidatieStatus(dia, fname_instr, fname_drempels)</code>
<b>INPUT</b>	<p><b>dia:</b> DIA structure, use <code>readdia_R14</code> to read a .dia file.</p> <p><b>fname_instr:</b> String with name of file with instruments.</p> <p><b>fname_drempels:</b> (Optional) string with name of file with validation rules.</p>
<b>OUTPUT</b>	<p><b>dia:</b> DIA structure, use <code>readdia_R14</code> to read a .dia file.</p>
<b>EXAMPLE</b>	<p>The following statuscodes are assigned to the timeseries. The assignment of the codes is per buoy (sLoccod-sVatcod). Codes 87 en 88 are only used for Directional Waveriders.</p> <p>81 Hm0 &lt; 0.3 m 82 Hm0/(di+3) &gt; 0.5 83 Hmax &gt; 2.0 * Hm0 84 Hm0 / H1/3 &lt; 0.875 of Hm0 / H1/3 &gt; 1.125 85 E(0.03-0.05) &gt; E(0.05-0.07) 86 E(0.40-0.50) &gt; E(0.30-0.40) 87 Wind Richting - Golfrichting (0.350-0.5 Hz)  &gt; 30 88 Golfrichting (&gt; 0.03 - 0.1 Hz) landrichtingssectoren 45-225gr</p> <p>Format of file with instruments:</p> <pre>Location instrument location_wind instrument_wind datel depth1 ... AMLZGTB32 WAVERDDR DENHDR FASTRCRDR 10-11-2007 103 AMLZGTB42 WAVERDDR DENHDR FASTRCRDR 10-11-2007 104 31-12-2007 105 AMLZGTB52 WAVERDDR DENHDR FASTRCRDR 10-11-2007 104 31-12-2007 105 AMLZGTB62 WAVERDR 10-11-2007 104 31-12-2007 105</pre>
<b>SEE ALSO</b>	<b><code>readdia_R14</code></b>

## 2.26 set\_taxis.m

<b>SUMMARY</b>	Make RKS or TPS block by specifying <b>begintime, endtime, timeunit and timestep.</b>
<b>CALL</b>	<code>S = set_taxis(S, tbegin, teind, tijdstapeenheid, tijdstap)</code>
<b>INPUT</b>	<p><b>S:</b> Existing RKS or TPS administrationbuffer, may be empty.</p> <p><b>tbegin:</b> Datumum with begin time.</p> <p><b>teind:</b> Datumum with end time.</p> <p><b>tijdstapeenheid:</b> (Optional) String with timeunit, see DONAR Manual Part 7, section 2.9.3</p> <p><b>tijdstap:</b> (Optional) timestep in tijdstapeenheid units.</p>
<b>OUTPUT</b>	<p><b>S:</b> Structure with RKS or TPS (<b>reeksadministratiebuffer</b>) with new values.</p>
<b>EXAMPLE</b>	<pre>blok(k).RKS=set_taxis(blok(k).RKS,...                     min(taxis_totaal),max(taxis_totaal)); blok(k).TPS=set_taxis(blok(k).TPS,...                     min(taxis_totaal),max(taxis_totaal));</pre>

SEE ALSO	<b>combineRKS, combineTPS, cmp_taxis</b>
----------	--

## 2.27 splitlongdate.m

SUMMARY	Split one or more dates of the form YYYYMMDDHHMM into two numbers: date: YYYYMMDD and time: HHMM.
CALL	[datum, time] = splitlongdate(longdate)
INPUT	longdate: Vector of integers of format YYYYMMDDHHMM.
OUTPUT	datum: Vector of integers of format YYYYMMDD. time: Vector of integers of format HHMM.
EXAMPLE	[datum, time] = splitlongdate(200808141200)
SEE ALSO	<b>long2datenum, datenum, datestr, datenum2long</b>

## 2.28 writedia\_R14.m

SUMMARY	Write DIA structure to file.
CALL	suces = writedia_R14(S, fname)
INPUT	S: Dia structure to save, fields: Dia Structure, with fields: +----IDT   +----sFiltyp (char)   +----sSyscod (char)   +----lCredat (double)   +----sCmtrgl (char) +----blok +----W3H (struct): see emptyW3H +----MUX (struct): empty, see emptyMUX +----TYP (struct): empty +----RGH (struct): empty, see emptyRGH +----RKS (struct): see emptyRKS +----TPS (struct): empty, see emptyTPS +----WRD (struct): see emptyWRD fname: String with the name of the file to create.
OUTPUT	rc: Integer returncode: rc == 0 operation successful. rc ~= 0 error, rc contains the DIM errorcode.
SEE ALSO	<b>readdia_R14</b>