



In opdracht van RIKZ

Wavix

Documentatie op source
code niveau

Modelit
Rotterdamse Rijweg 126
3042 AS Rotterdam
Telefoon +31 10 4623621



info@modelit.nl
www.modelit.nl

In opdracht van RIKZ

Wavix

Documentatie op source code niveau

Datum 15 November 2004

Modelit
KvK Rotterdam 24290229



Documentatiepagina

Opdrachtgever RIKZ

Titel **Wavix**
Documentatie op source code niveau

Datum 15 November 2004

Projectteam opdrachtgever

Projectteam Modelit

Projectomschrijving

Trefwoorden Help informatie voor Wavix M-modules

Inhoud

1 Inleiding.....	1
2 Afhankelijkheidschema's.....	2
2.1 Functies van de categorie: Hoofdscherm.....	2
2.2 Functies van de categorie: Databeheer.....	3
2.3 Functies van de categorie: Netwerkbeheer.....	4
2.4 Functies van de categorie: Regressiebeheer.....	4
2.5 Functies van de categorie: Conhop.....	5
3 Documentatie per functie.....	1

1 Inleiding

Iedere M-module die in het kader van WAVIX is vervaardigd bevat een header die een samenvatting geeft op van de code. In deze header kunnen de volgende elementen worden aangetroffen.

SUMMARY

Een eenregelige beschrijving die de functienaam bevat en met een paar steekwoorden beschrijft wat de functie doet.

CALL

Een regel die de aanroep wijze beschrijft

INPUT

Per invoer element: een beschrijving van de invoer

OUTPUT

Per uitvoer element: een beschrijving van de uitvoer

SEE ALSO

Een lijst van gerelateerde m-modules

CALL BACK

Een lijst van aangeroepen functies

Naast deze informatie is iedere functie nog in meer detail gedocumenteerd door middel van commentaarregels die als toelichting in de code staan.

De wavix functies zijn onderverdeeld in de volgende categorieën:

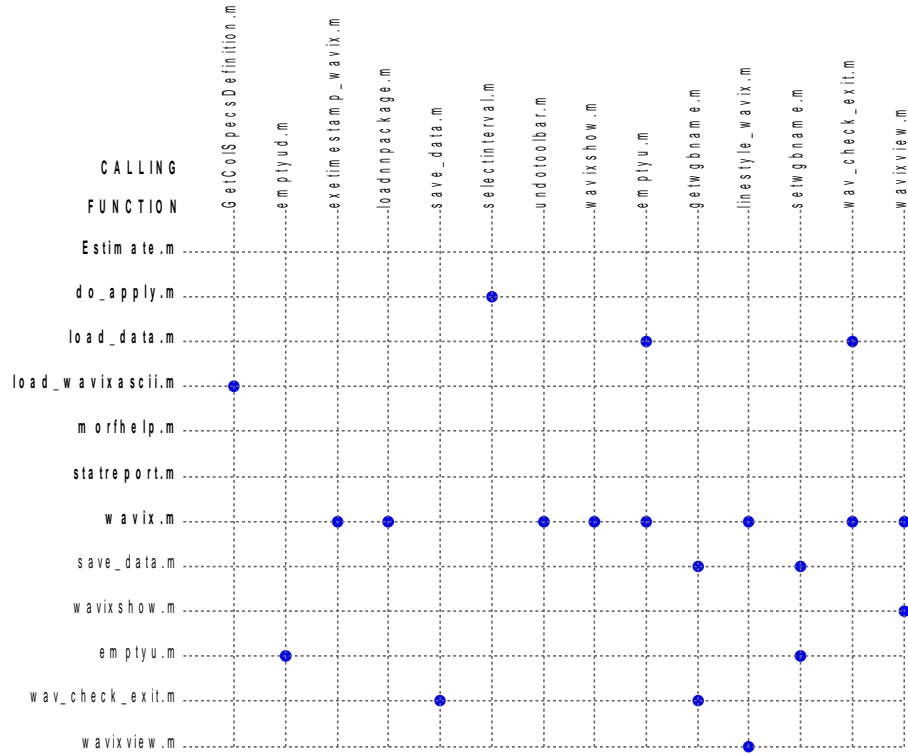
- Hoofdscherm. Dit zijn de functies die de GUI voor het hoofdscherm vormgeven, en de functies die direct uit deze GUI worden aangeroepen;
- Databeheer. Dit zijn de functies die de GUI voor databeheer vormgeven, en aanverwante functies;
- Netwerkbeheer. De functies die het Neurale Netwerk beheer vormgeven;
- Regressiebeheer. De functies die behoren tot de Regressiebeheer GUI;
- Conhop. De rekenfuncties voor het toepassen van Neurale Netwerk modellen en Conhop optimalisatie;
- Hulpfuncties. Functies voor algemene doeleinden die vanuit elke directory kunnen worden aangeroepen.

Binnen iedere functiecategorie (behalve voor de hoofdfuncties) is middels schema's aangegeven welke functie door welke functie wordt aangeroepen. Deze schema's zijn te vinden in het volgende hoofdstuk.

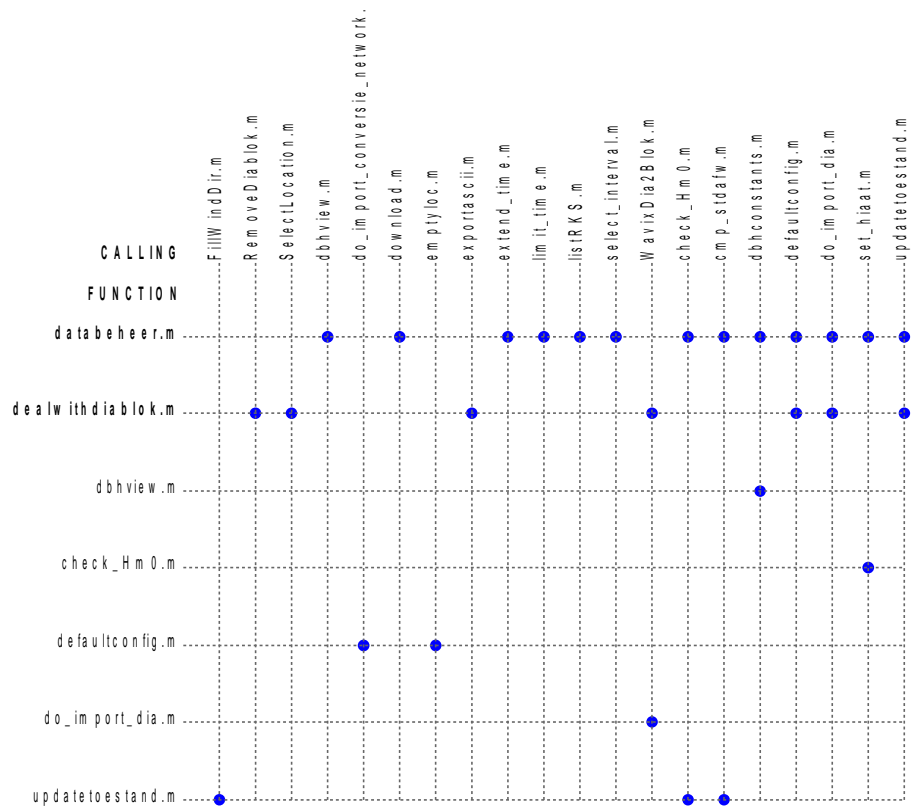
De documentatie per functie is opgenomen in hoofdstuk 3. Ook hier zijn de functies geclusterd per categorie conform de bovengenoemde indeling.

2 Afhankelijkheidschema's

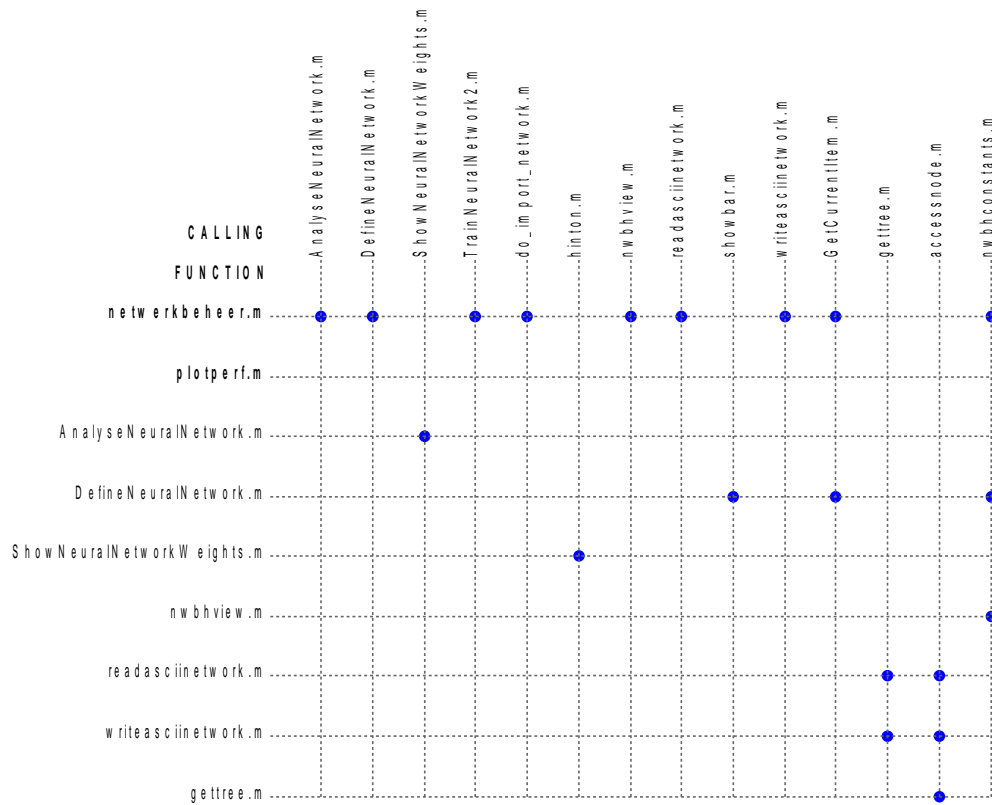
2.1 Functies van de categorie: Hoofdscherm



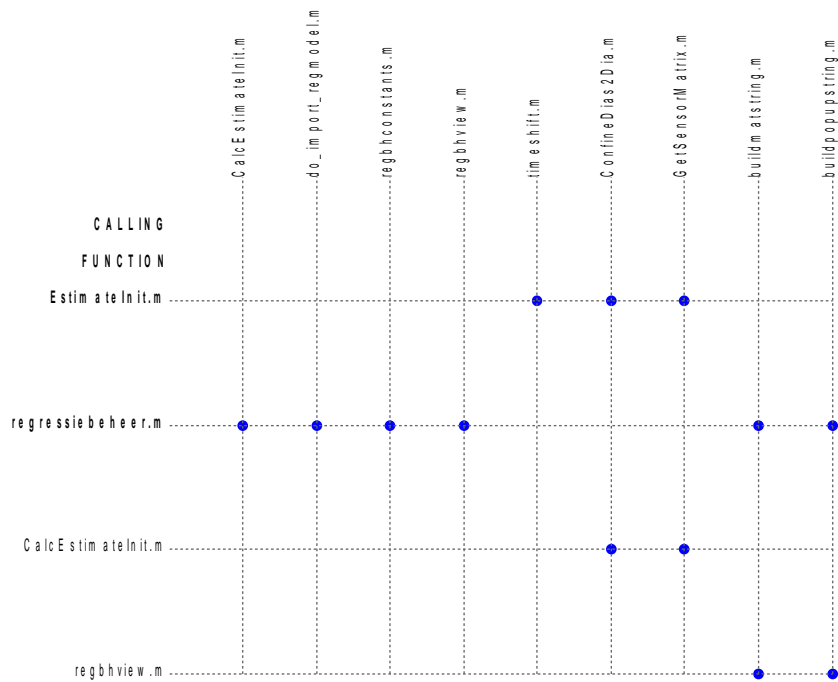
2.2 Functies van de categorie: Databekeer



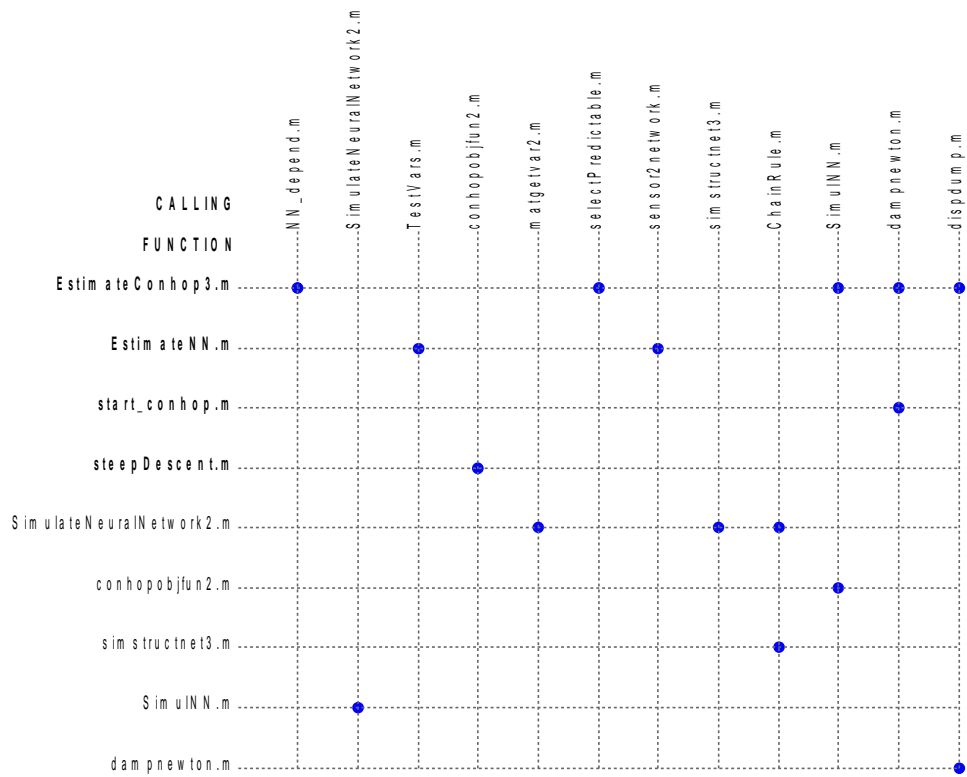
2.3 Functies van de categorie: Netwerkbeheer



2.4 Functies van de categorie: Regressiebeheer



2.5 Functies van de categorie: Conhop



3 Documentatie per functie

```
=====
DIRECTORY: C:\d\modelit\wavixIV\HOOFDSCHERM
=====
```

INHOUD:

```
Estimate.m
GetColSpecsDefinition.m
do_apply.m
emptyu.m
emptyud.m
exetimestamp_wavix.m
getwgbyname.m
linestyle_wavix.m
load_data.m
load_wavixascii.m
loadnnpackagem.m
morfhelph.m
save_data.m
selectinterval.m
setwgbyname.m
statreport.m
undotoolbar.m
wav_check_exit.m
wavix.m
wavixshow.m
wavixview.m
```

```
*****
```

FILE: Estimate.m

Estimate - Schat de reeksen van de hoofdsensoren

mode: uit te voeren actie

opt: meegegeven optie vanuit start_conhop

```
*****
```

FILE: GetColSpecsDefinition.m

GetColSpecsDefinition -Bouw sleutels voor het selecteren van de kolommen

CALL

[ColSpecHoofd, ColSpecNeven, groot2klein, Parameters]=GetColSpecsDefinition

INPUT

geen

OUTPUT

ColSpecHoofd: Een array van structures waarvan de lengte overeenkomt met het aantal WAVIX kolommen -1. Dit array bevat de sleutels en conversie instructies voor de WAVIX tabel met HOOFDsensoren.

De strucxtures van dit array hebben de volgende velden:

sLoccod: een CELL ARRAY met primaire, secundaire, etc sleutel voor locatie voor andere kolommen dan die voor windgegevens is momenteel alleen de primaire sleutel bepaald. Er zijn maximaal 6 sleutels.

De sleutel 'NB' wordt in dia2wavix als speciale waarde behandeld.

sParcod: een CHAR STRING met de sleutel voor paramter type.

sVatcod: een CELL ARRAY met primaire, secundaire, etc sleutel voor detectortype voor andere kolommen dan die voor windgegevens is momenteel alleen de primaire sleutel bepaald. Er zijn maximaal 6 sleutels.

De sleutel 'NB' wordt in dia2wavix als speciale waarde behandeld.

De lengte van dit cell array moet overeenkomen met het aantal sleutels voor locaties.

verplicht: code voor het type waarschuwing bij een niet gevonden blok

factor: Ophoogfactor voor Dia2Wavix (bijvoorbeeld voor het geval dat DONAR een andere eenheid gebruikt dan WAVIX)

verschil: Vaste ophoging

bewerking: Veld dat een bepaalde bewerking karakteriseert.

Momenteel zijn de volgende bewerkingen ondersteund:

1 ==> Conversie van TE3 naar TE10 (4*sqrt)

2 ==> Bijgissen van hiaten door middel van lineaire interpolatie

ColSpecNeven: Een array van structures waarvan de lengte overeenkomt met het aantal WAVIX kolommen -1. Dit array bevat de sleutels en conversie instructies voor de WAVIX tabel met NEVENsensoren.

groot2klein: Een lijst corresponderend met de lijst van parametersoorten, met voor elke parameter de locatie van de WAVIX kolom voor deze parameter in de repeterende blokken van de WAVIX tabel

Parameters: Een DIA structure met voor elke parameter een blok dat de Metagegevens bevat. De blokken staan in de structure in dezelfde volgorde als dat ze voorkomen in de WAVIX tabel.

WERKWIJZE

Een gedeelte van de informatie wordt gehaald uit de stuurfile 'wavixlocaties.txt'.

Deze file bevat informatie over de volgende locaties:

Locaties (in deze volgorde)

1	AUK	AUKFPFM	Aukfield platform
2	K13	K13APFM	K13a platform
3	SON	SCHIERMNOND	Schiermonnikoog noord
4	ELD	EIELSGT	Eierlandse Gat
5	YM6	IJMDMNTSPS	IJmuiden munitiestortplaats
6	MPN	NOORDWMPT	Noordwijk meetpost
7	EUR	EURPFM	Euro platform
8	LEG	LICHTELGRE	Lichteiland Goeree
9	SWB	???	Schouwenbank
10	SCW	???	Scheur West

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Lees de sleutels voor locatie, parameter en meetapparaat uit de stuurfile

Plot een overzicht van Golf en Water sensoren (alleen primaire sleutel)

Vervang niet gespecificeerde velden door de code 'NB'

Plot een overzicht van Wind sensoren (primaire, secundaire,.. 6e sleutel)

Definieer de parametercode met het array 'sParcod'

Definieer de bijbehorende conversie bewerkingen in de tijdelijke tabel 'ParcodOpties'

kolom 1: ophoogfactor

kolom 2: verschil

kolom 3: foutmelding bij afwezigheid

kolom 4: bewerking, 1==> 4*sqrt

2==> Hiaten lineair interpoleren

kolom 5: locatie van de corresponderende kolom in het groot WAVIX formaat

Leidt de locaties van de parameters in het grote WAVIX formaat af.

LET OP! Pas kolom 5 van de Tabel ParcodOpties aan indien deze locaties niet correct

blijken te zijn (in dit geval is er sprake van een documentatiefout in de oorspronkelijke WAVIX documentatie)
 Bouw nu de 90 structures die als sleutel gaan fungeren
 WAVIX kolom 1 tm 6: golven
 WAVIX kolom 7 tm 8: wind
 WAVIX kolom 9: water
 Indien meer dan drie uitvoerargumenten vereist (aanroep uit wavix2dia):
 Bepaal ook de dia met parameter gegevens
 Lees deze uit de file 'wavixparameters.dia'
 Maak een char array van de parameter coderingen van de blokken
 Bepaal de locatie van de gespecificeerde parameters in dit char array
 Presenteer een overzicht van de metagegevens
 breek af indien een of meer blokken met metagegevens ontbreken

```
FILE: do_apply.m
do-apply - bewerk geselecteerde items

CALL
  function do_apply(obj,event,mode)

INPUT
  obj,event: info over aanroepend object (button)
  mode: bewerking
    ok:      keur geselecteerde periode goed
    estimate: pas geschatte waarde toe
    hiaat:   zet de geselecteerde periode op hiaat
```

```
FILE: emptyu.m
emptyu - maak een lege userdata structure aan

CALL
  db=emptyu

INPUT
  C      : structure met constanten
  fname : te openen filenaam

OUTPUT
db: een 'leeg' werkgebied.

AANPAK
  Deze functie komt in de plaats van een klassieke declaratie en maakt het mogelijk om:
  1. structures te rangschikken in een array
  2. te testen op bepaalde veldwaarden, zonder een dergelijke test vooraf te laden gaan door een test met 'isfield'

SEE ALSO
  emptygrid, emptyud, emptyu, emptyraai
```

```
*****
FILE: emptyud.m
  emptyud - maak een lege userdata structure aan

CALL
  grid=emptyud

INPUT
  stamp: inititiele tijd stempel voor het veld timeofchange
        TIP: laat dit veld overeen komen met het veld 'timeofcommit'
        in een overkoepelende structure

OUTPUT
  grid: een 'lege' userdata structure.

AANPAK
  Deze functie komt in de plaats van een klassieke declaratie en maakt het
  mogelijk om:
  1. structures the rangschikken in een array
  2. te testen op bepaalde veldwaardes, zonder een dergelijke test vooraf
     te laden gaan door een test met 'isfield'

SEE ALSO
  emptyu
```

```
*****
FILE: exetimestamp_wavix.m
  This file has been generated automatically by function exetimestamp_create
```

```
*****
FILE: getwgbname.m
  retourneer naam van werkgebied

INPUT
  -

OUTPUT
  filename: volledige filenaam met pad
           Vb: C:\d\modelit\wavixIV\Unititled.wv4
  stamnaam: filenaam zonder pad en extensie
           Vb: Unititled

SEE ALSO
  getwgbname, setwgbname
```

```
*****
```

```

FILE: linestyle_wavix.m
linestyle - PROPERTIES VOOR LIJNEN BINNEN WAVIX
CALL
    linestyle=linestyle_wavix
INPUT
    none
OUTPUT
    linestyle: struct met:
                linestyle.<linetype>: properties van lijn
EXAMPLE
    h=line(lstyle.hiaat); %initialiseer lijn

```

```

*****
FILE: load_data.m
load_data - callback van menu 'laad werkgebied'
User interface voor het laden van een eerder bewaard werkgebied

```

```

CALL
    load_data(fname)

INPUT
    fname: naam van te laden file

OUTPUT
    geen
    De property 'userdata' wordt aangepast:

```

```

AANPAK
- Kijk of oude data bewaard moeten blijven (wav_check_exit)
- Haal de naam van de invoerfile op (getfile)
- Schakel interactie uit (mbd_suspend)
- Laad data
- Schakel interactie in (mbd_restore)
- Verwijder introtext (digivalwinresize)
- Schakel menu's in die van data afhangen (activatemenus)
- Schakel de save menus uit (de volgende aanroep van wav_check_exit activeert ze weer)
- Pas de naam van het window aan
- Update scherm (update)

```

```

*****
FILE: load_wavixascii.m
load_wavixascii - callback van menu 'laad werkgebied'
User interface voor het laden van een eerder bewaard werkgebied

```

```

CALL
    load_wavixascii(fname)

INPUT
    fname: naam van te laden file

OUTPUT
    geen
    De property 'userdata' wordt aangepast:

```

AANPAK

- Kijk of oude data bewaard moeten blijven (morf_check_exit)
- Haal de naam van de invoerfile op (getfile)
- Schakel interactie uit (mbd_suspend)
- Laad data
- Schakel interactie in (mbd_restore)
- Verwijder introtext (digivalwinresize)
- Schakel menu's in die van data afhangen (activatemenus)
- Schakel de save menus uit (de volgende aanroep van morf_check_exit activeert ze weer)
- Pas de naam van het window aan
- Update scherm (update)

FILE: loadnnpackagem

loadnnpackagem - refereer naar Neural Netwerk functies die binnen WAVIX worden gebruikt

Deze functie is nodig om Wavix te kunnen compileren

FILE: morfhelp.m

No help comments found in morfhelp.m.

Use the Help browser Search tab to [C:\d\modelit\wavixIV\HOOFDSCHERM\morfhelp.m](matlab:docsearch)>search the documentation, or type "[help help](matlab:help help)" for help command options, such as help for methods.

FILE: save_data.m

save_data - callback van menu 'bewaar werkgebied'
User interface voor het bewaren van een werkgebied

CALL

save_data(obj,event) %directlty from button
save_data(obj,event,fname) %from other function

INPUT

obj,event: niet gebruikt
fname: te bewaren file
ongedefinieerd (nargin<=2) ==> vraag gebruiker om filenaam
string ==> gebruik deze naam
empty string ==> gebruik werkgebied naam (tenzij nog niet gekozen)

OUTPUT

saved: 1 als daadwerkelijk gesaved
0 als cancel ingedrukt

AANPAK

- Haal constantes en userdata op
- Bepaal de filenaam van de te bewaren data
- Schakel GUI tijdelijk uit (mbd_suspend)
- Bewaar data
- Activeer GUI (mbd_restore)
- De activeer save buttons
Deze worden bij de eerste wijziging weer door check_exit geactiveerd

FILE: selectinterval.m

selectrange - selecteer meerdere periodes

CALL

selectrange(x1,x2)

INPUT

mode: meegegeven vlag bij aanroep
leftclick: met linkermuis in grafiek geklikt
L: locatie index
x1: periode indices, of begin selectie periode
x2: einde selectie periode (optioneel)

OUTPUT NAAR SCHERM

selecteer 1 of meerdere periodes

AANGEROEPEN UIT:

wavix bewerkingsscherm. geïnstalleerd met o.a. zoomtool (wavixview)

APPROACH

Het gedrag van de functie hangt af van de manier van aanroepen.

Als met 1 argument aangeroepen (via selectnew):

Aangeroepen als callback van lijst met alfanumerieke data.

x1 bevat een lijst met periode indices.

Teken de bijbehorende cirkels.

Als met 2 argumenten aangeroepen (via zoomtool):

x1 = begin selectie periode

x2 = eind selectie periode

Bepaal alle tussenliggende, te selecteren periodes.

Gebruik de functie FIND_SELECTABLE om na te gaan welke periodes in aanmerking komen.

Teken cirkels.

Selecteer Rijen uit lijst met alfanumerieke data.

SEE ALSO: SELECTNEW

FILE: setwgbname.m

setwgbname - wijzig naam van werkgebied

INPUT

```
filename: filenaam (te gebruiken in titel van hoofdscherm en op te slaan)
extra: achtervoegsel
```

SEE ALSO

```
wgbdirectory, getwgbname, setwgbname
```

```
*****
```

FILE: statreport.m

```
statreport - genereer een rapport
             met statistieken per reeks geaggregeerd op globaal, locatie,
             parameter en reeksniveau
```

```
*****
```

FILE: undotoolbar.m

```
Creeer standaard buttons voor toolbars in Wavix applicatie
```

INPUT

```
present: vlaggen voor het wel/niet opnemen van de volgende buttons:
present(1) : naar hoofdscherm
present(2) : naar raaibeheer
present(3) : naar neurale netwerk beheer
present(4) : naar regressie beheer
present(5) : presenteren logboek
present(6) : presenteren statistieken
present(7) : help
present(8) : reset redo/undo history
```

```
*****
```

FILE: wav_check_exit.m

```
wav_check_exit - Check of alle data bewaard zijn
```

CALL

```
status=wav_check_exit
```

INPUT

```
status: 'changed' er zijn nieuwe data om eventueel te bewaren
        'nochanges' er is niets te bewaren
```

OUTPUT

```
status = 0 ==> er waren geen onbewaarde data
status = 1 ==> er waren onbewaarde data, deze zijn bewaard
status = 1 ==> er waren onbewaarde data, deze zijn niet bewaard
status = -1 ==> er waren onbewaarde data, de gebruiker heeft CANCEL ingedrukt
```

EXAMPLE

```
wav_check_exit('changed'): % Een volgende aanroep van wav_check_exit zal gebruiker
                          % vragen of wijzigingen moeten worden opgeslagen
```

```
% Save menu's worden geactiveerd
```

APPROACH

```
Indien geen invoerargument:
  Check de status van het menu "save data"
```

```
Indien wel een invoerargument
  Verander de status van het menu "save data"
```

```
Deze functie wordt aangeroepen iedere keer nadat iets in
de dataset wordt gewijzigd.
```

```
*****
FILE: wavix.m
  wavix - installeer wavix scherm
```

```
*****
FILE: wavixshow.m
  show - Visualisatie van ALLE data in de morfologie applicatie
  aangeroepen DBOBJ object. Ook bij UNDO en REDO.
```

INPUT

```
udnew      : structure met data uit werkgebied
ind: een CELL array met
      struct arrays met velden
      'type'
      'subs'
      queued{k}(2).subs bevat naam van te updaten veld.
```

OUTPUT

```
geen
```

OUTPUT NAAR SCHERM

```
Update alle objecten die gerelateerd zijn aan
data in het werkgebied.
```

APPROACH

```
*****
FILE: wavixview.m
  INPUT
  udnew : structure met Morfologie WGB data
  opt   : structure met veld opt_S.opt, welke de settings database bevat
  upd   : structure met 0 (niet updaten) en 1( wel updaten)
         voor alle scherm elementen
```

```
=====
DIRECTORY: C:\d\modelit\wavixIV\DATABASEHEER
=====
```

INHOUD:

```
FillWindDir.m
RemoveDiablok.m
SelectLocation.m
WavixDia2Blok.m
check_Hm0.m
cmp_stdafw.m
databeheer.m
dbhconstants.m
dbhview.m
dealwithdiablok.m
defaultconfig.m
do_import_conversie_network.m
do_import_dia.m
download.m
emptyloc.m
exportascii.m
extend_time.m
limit_time.m
listRKS.m
select_interval.m
set_hiaat.m
updatetoestand.m
```

```
*****
```

FILE: FillWindDir.m

FillWindDir - Bepaal de windrichting bij windstilte en veranderlijke wind

INPUT:

<leeg>

OUTPUT:

de waarden 0 (windstilte) en 990 (veranderlijke wind) in de dias met de windrichting zijn nu vervangen door geïnterpoleerde waarden. en de centrale database is geactualiseerd. stdW van de aangepaste waarden wordt op 30 gezet en bit 5 van status wordt gezet (bit 5 is herkomst)

```
*****
```

FILE: RemoveDiablok.m

RemoveRaaien - verwijder raaien uit werkgebied

INPUT

rmvIDs : IDs van de te verwijderen raaien

OVERIGE INPUT

userdata worden opgehaald

OVERIGE OUTPUT

userdata worden weggeschreven

FILE: SelectLocation.m

DEFINIEER FRAMES

FILE: WavixDia2Blok.m

Converteer Wavix element "dia" naar Donwar "blok" element

INPUT

dia: wgb dia structure

OUTPUT

blok: blok structure die in dia wordt weggeschreven

AANGEROEPEN UIT:

Deze functie wordt aangeroepen in de volgende situaties:

- wanneer twee Dia's worden samengevoegd. De wavix datastructuur wordt dan tijdelijk omgezet in een Donar structuur. Hierdoor wordt het mogelijk om de algemene utility "dia_merge" te benutten. (procedure do_import_dia.m)
- wanneer een reeks wordt geconverteerd. Dit gebeurt door de een nieuwe reeks aan te maken en te importeren met do_import_dia)
- alle overige plaatsen waar do_import_dia vanuit Wavix wordt aangeroepen.

FILE: check_Hm0.m

Voer een consistentie check uit: vergelijk hiaten in reeks met hiaten in corresponderende Hm0 reeksen

FILE: cmp_stdafw.m

cmp_stdafw

FILE: databeheer.m

databeheer - installeer de databeheer GUI

INPUT

obj,event: standaard callback argumenten (worden niet gebruikt)

OUTPUT

geen uitvoervariabelen

WERKWIJZE

Deze functie kijkt of het databeheer scherm al is geïnstalleerd en maakt het in dat geval current. Zo niet, dan wordt het databeheer scherm geïntialiseerd. Deze functie module bevat alle define- functies waarmee het scherm wordt opgebouwd, en de meeste van de callback functies die vanuit het scherm kunnen worden aangeroepen.

ZIE OOK:

dbhview

FILE: dbhconstants.m

definieer een aantal constantes die specifiek zijn voor het databeheer scherm

AANGEROEPEN VANUIT:

 DATABASEER.M
 DBHVIEW.M

FILE: dbhview.m

INPUT

udnew : databse
opt : GUI settings
upd : te updaten scherm elementen
C : constantes
HWIN : handle van hoofdscherm

FILE: dealwithdiablok.m

Verwijder, Selecteer of Deselecteerde gemarkeerde raaien
Wordt aangeroepen uit contextmenu van lijst of button
Conventie 1: de lijst "listobj" heeft als userdata de raai ID's
Conventie 2: de gemarkeerde items in lijst "listobj" dienen gewijzigd te worden

```

FILE: defaultconfig.m
defaultconfig - selecteer reeksen als hoofdsensor

CALL
    [db,reportstr]=defaultconfig(db,C,mode,dia,WavixLoc)
INPUT
    db      : undoredo object
    C      : constanten
    actie   : uit te voeren actie
            remove: verwijder de locaties uit array dia uit locatietabel
            add    : voeg de locaties uit array dia toe aan de locatietabel
            addascii: voeg de locaties uit array dia toe aan de
                    locatietabel voor zover deze locaties voorkomen in
                    een stuurfile
    dia     : array met dia array
    WavixLoc: struct array dat correspondeert met dia. (optioneel). bevat
            het veld wavixloc: wavix locatie waarvoor de reeks geldt.
OUTPUT
    db: aangepaste wgb structure
    reportstr: weg te schrijven logboek aantekening

```

```

*****

```

```

FILE: do_import_conversie_network.m
do_import_conversie_network - Import NetworkArray

CALL
    [u] = do_import_conversie_network(C,fname,NetworkArray,u)

INPUT
    C          : Structure with constants
    fname      : String with the filename
    NetworkArray : Array with network structures
    u          : The database (stored)

OUTPUT
    u          : Updated database

```

```

*****

```

```

FILE: do_import_dia.m
do_import_dia - voer de import actie vaoo dia uit

CALL
    db=do_import_dia(C,fname,blok,db)

INPUT
    C      : Structure met constanten
    fname  : Cell array met filenamen
    blok   : structure array met dia blokken (optioneel); velden:
            W3H
            MUX
            TYP
            RKS

```

```

                TPS
                WRD <===Volgens DONAR datastructuur
db              : werkgebied structure

```

```

OUTPUT
  db              : bijgewerkte wgb structure

```

```

AANGEROEPEN UIT
  databeheer
  load_wavixascii

```

```

*****
FILE: download.m
  open ftp

```

```

*****
FILE: emptyloc.m

```

No help comments found in emptyloc.m.

Use the Help browser Search tab to [C:\d\modelit\wavixIV\DATABASEEER\emptyloc.m](matlab:docsearch)>search the documentation, or type "[help help](matlab:help help)" for help command options, such as help for methods.

```

*****
FILE: exportascii.m
  exportascii - GUI voor het exporteren van reeksen in kaal ascii formaat

```

```

INPUT
  C: constantes
  db: handle naar wgb
  indx: index van de te eporteren reeksen

```

```

CALLBACK:
  do_export

```

```

*****
FILE: extend_time.m
  extend_time - breid tijdsinterval van de reeksen in de database uit

```

```

INPUT:
  db              : wavix werkgebied
  indx            : reeks index (correspondeert met kolom indx in M)
  M               : <struct> de database in matrixvorm (zie db2mat)

```

```

OUTPUT:

```



```
db      : bijgewerkt wavix werkgebied
```

```
*****
```

```
FILE: limit_time.m
```

```
limit_time - perk tijdsinterval van reeks in
```

```
INPUT
```

```
db      : wavix werkgebied
indx    : reeks index
taxis   : (datum) geselecteerde tijd as
```

```
INPUT
```

```
db      : bijgewerkt wavix werkgebied
reportstr : <string> commentaar voor het logboek
```

```
ZIE OOK
```

```
set_hiaat
limit_time
```

```
*****
```

```
FILE: listRKS.m
```

```
listRKS - Hulpfunctie. vul een struct array van RKS structures op basis
van een WAVIX dia array
```

```
INPUT
```

```
dia: Wavix dia array
Velden:
  ID
  blok
  stdW
  V
  stdV
  etc.
indices: te gebruiken indices (default: alle)
```

```
OUTPUT
```

```
RKSs: RKS array
```

```
SEE ALSO
```

```
listW3H
```

```
*****
```

```
FILE: select_interval.m
```

```
-----
```

```
*****
```

```
FILE: set_hiaat.m
```

```
set_hiaat - markeer bepaalde punten als hiaat
```

```
INPUT
  optie: bij te werken veld:
          W: waarde veld
          V: voorspelling
  db: wgb structure
  indx: index in wgb struct array dia
  f_hiaat: index van te markeren hiaten
  msg: message voor undo lijstweergave
OUTPUT
  db: bijgewerkte wgb structure
```

```
*****
FILE: updatetoestand.m
      updatetoestand
```

```
=====
DIRECTORY: C:\d\modelit\wavixIV\NETWERKBEHEER
=====
```

```
INHOUD:
AnalyseNeuralNetwork.m
DefineNeuralNetwork.m
GetCurrentItem.m
ShowNeuralNetworkWeights.m
TrainNeuralNetwork2.m
accessnode.m
do_import_network.m
gettree.m
hinton.m
netwerkbeheer.m
nwbhconstants.m
nwbhview.m
plotperf.m
readasciinetwork.m
showbar.m
writeasciinetwork.m
```

```
*****
FILE: AnalyseNeuralNetwork.m
      AnalyseNeuralNetwork - analysis tool for a trained neural network
```

```
CALL
      AnalyseNeuralNetwork(NeuralNetwork)
```

```
INPUT
      NeuralNetwork: Network to display in GUI
```

```
CALLED FROM
      netwerkbeheer
```

```
*****
FILE: DefineNeuralNetwork.m
DefineNeuralNetwork - Definieer een nieuw feed-forward neuraal netwerk

INPUT:
NeuralNetwork: <struct> van the type 'netwerk', zie emptystruct('netwerk')
                NeuralNetwork kan leeg zijn of reeds aangemaakt

OUTPUT:
NeuralNetwork: <struct> van the type 'netwerk', zie emptystruct('netwerk')
                NeuralNetwork is leeg als operatie afgebroken
```

```
*****
FILE: GetCurrentItem.m
GetCurrentItem - get current selected item(s) of a listbox

CALL:
    [Selection,SelectionIndex] = GetCurrentItem(listbox)

INPUT:
listbox      : handle of a listbox

OUTPUT:
Selection    : the selected items in the listbox
SelectionIndex : the corresponding values

DATE:    04-May-2004

STATUS:   Beta, kjh
```

```
*****
FILE: ShowNeuralNetworkWeights.m
ShowNeuralNetworkWeights - visualize the weights and biases of
                           every layer and every ensemble member
```

```
*****
FILE: TrainNeuralNetwork2.m
TrainNeuralNetwork - Train the neural network in NetworkStruct

INPUT:
W          : <matrix> gemeten waarden, aantal periodes bij aantal reeksen groot
stdW      : <matrix> standaarddeviaties, aantal periodes bij aantal reeksen groot
Wkey      : <struct> met velden met bijbehorende (loc,var,veldapp)
                combinatie per reeks
                - sLoccod
```

```

- sParcod
- sVatcod
NetworkStruct: <struct> met een neuraal netwerk (Zie emptystruct('netwerk'))

```

OUTPUT:

```

NetworkStruct: <struct> met een neuraal netwerk, de members
                worden in deze routine gevuld, d.w.z.
                de gewichten en bias worden gevuld
comment       : <string> commentaar voor het logboek wordt gebruikt in
                netwerkbeheer

```

```

*****

```

```

FILE: accessnode.m

```

```

accessnode - get the subscripts to access node 'node' of structure
            'structure'

```

```

*****

```

```

FILE: do_import_network.m

```

```

do_import_network - Import NetworkArray

```

CALL

```

[u] = do_import_network(C, fname, NetworkArray, u)

```

INPUT

```

C           : Structure with constants
fname      : String with the filename
NetworkArray : Array with network structures
u          : The database (stored)

```

OUTPUT

```

u          : Updated database

```

```

*****

```

```

FILE: gettree.m

```

```

gettree - get the treestructure of a structure

```

INPUT:

```

structure: a structure

```

OUTPUT:

```

tree:      vector with the tree structure, element i contains the
            index of node i's predecessor, zero being the root node
labels:    the names of the nodes

```

```

DATE: 10-Jun-2004 22:57:15

```

FILE: hinton.m

HINTONWB Hinton graph of weight matrix and bias vector.

Syntax

```
hinton(W,b,maxw,minw)
```

Description

HINTONWB(W,B,M1,M2)

W - SxR weight matrix

B - Sx1 bias vector.

MAXW - Maximum weight, default = max(max(abs(W))).

MINW - Minimum weight, default = M1/100.

and displays a weight matrix and a bias vector represented as a grid of squares.

Each square's AREA represents a weight's magnitude.

Each square's COLOR represents a weight's sign.

RED for negative weights, GREEN for positive.

Examples

```
W = rands(4,5);
```

```
b = rands(4,1);
```

```
hintonwb(W,b)
```

See also [hintonw](matlab:help hintonw).

FILE: netwerkbeheer.m

netwerkbeheer - import and export neural networks

FILE: nwbhconstants.m

definieer een aantal constantes die specifiek zijn voor het netwerkbeheer scherm

FILE: nwbhview.m

INPUT

udnew : database

opt : GUI settings

upd : te updaten scherm elementen

C : constantes

HWIN : handle van hoofdscherm

FILE: plotperf.m

PLOTPERF Plot network performance.

Syntax

```
plotperf(tr,goal,name,epoch)
```

Description

PLOTPERF(TR,GOAL,NAME,EPOCH) takes these inputs,
 TR - Training record returned by train.
 GOAL - Performance goal, default = NaN.
 NAME - Training function name, default = ''.
 EPOCH - Number of epochs, default = length of training record.
 and plots the training performance, and if available, the performance goal, validation performance, and test performance.

Example

Here are 8 input values P and associated targets T, plus a like number of validation inputs VV.P and targets VV.T.

```
P = 1:8; T = sin(P);
VV.P = P; VV.T = T+rand(1,8)*0.1;
```

The code below creates a network and trains it on this problem.

```
net = newff(minmax(P),[4 1],{'tansig','tansig'});
[net,tr] = train(net,P,T,[],[],VV);
```

During training PLOTPERF was called to display the training record. You can also call PLOTPERF directly with the final training record TR, as shown below.

```
plotperf(tr)
```

Reference page in Help browser

```
<a href="matlab:doc plotperf">doc plotperf</a>
```

FILE: readasciinetwork.m

readasciinetwork - read networks in ascii format

INPUT:

```
filename:    string with the name of the file
db:         wgb structure. Wordt alleen voor bijwerken logboek
           gebruikt. Mag [] zijn als functie voor Preview doelen
           wordt gebruikt.
```

OUTPUT:

```
networkArray: array of networks
OPM: wanneer aanroep niet succesvol is wordt een leeg
```

db: [0x1] structure array gererourneerd. In de aanroepende routine kan dus evt. getest worden met isempty(networkArray) evt bijgewerkte wgb structure.

APPROACH:

- lees de file in en verwijder commentaar (% regels)
- bepaal de indices van de blokken (sjabloon en netwerk)
- lees de sjablonen in en construeer tmpnetworkStruct voor elk sjabloon met daarin de opgegeven velden
- lees de netwerken in en construeer voor elk netwerk een tmpnetworkStruct met daarin de gedefinieerde velden
- combineer de tmpnetworkStruct van de sjablonen met de tmpnetworkStruct van de netwerken tot een netwerkstruct

FILE: showbar.m
 showbar - bar met labels en selectiemogelijkheid

INPUT:

- alpha : waarden van de elementen
- labelx : label voor x-as
- labeledy : label voor y-as
- selected : index van geselecteerde elementen (worden groen gekleurd), niet-geselecteerde worden rood gekleurd
- threshold : drempel waarboven de elementen groen gekleurd worden
- value : 'value' -> de waarde wordt zichtbaar als op de bar wordt geklikt
 'index' -> de index wordt zichtbaar als op de bar wordt geklikt

FILE: writeasciinetwork.m
 writeasciinetwork - schrijf de netwerken in het werkgebied weg als .asc bestanden

=====
 DIRECTORY: C:\d\modelit\wavixIV\REGRESSIEBEHEER
 =====

INHOUD:

- CalcEstimateInit.m
- ConfineDias2Dia.m
- EstimateInit.m
- GetSensorMatrix.m
- buildmatstring.m
- buildpopupstring.m
- do_import_regmodel.m
- regbhconstants.m

regbhview.m
 regressiebeheer.m
 timeshift.m

FILE: CalcEstimateInit.m

CalcEstimateInit - bepaal de verhoudingen tussen de variabelen op een locatie met dezelfde variabele op de andere locaties. Daarbij wordt onderscheid gemaakt tussen verschillende windsnelheid en windrichtingsklassen op de locatie zelf voor het geval dat er geen waarnemingen zijn voor een bepaalde variabele op een locatie wordt er ook een regressie uitgevoerd van de variabele met de windsnelheid op de betreffende locatie

INPUT:
 geen input

OUTPUT:

vhg: <vhg-struct> met de velden:

- richting :windrichtingsklassen
- snelheid :windsnelheidsklassen
- locs :locaties
- factor :de verhoudingsgetallen per parameter aantal windsnelheidsklassen bij aantal windsnelheidsklassen cellmat van aantal locaties bij aantal locaties matrices
- sigma :de spreidingen (zie factor)

msgstr: <string> met eventueel gegevens over de combinaties die niet genoeg informatie bevatten om de regressie mee uit te voeren

AANPAK:

stap 1: Bepaal de hoofdsensoren voor de te schatten variabelen en voor de windrichting en windsnelheid

stap 2: Selecteer een locatie en haal de windrichting en windsnelheid reeksen van deze locatie op deze bepalen de klassen voor de reeksen op deze locatie

stap 3: Selecteer een variabele (y) voor de locatie die in stap 2 geselecteerd is

stap 4: Selecteer dezelfde variabele als in stap 3 op een andere locatie (x) en selecteer ook de windsnelheid op de locatie van stap 2

stap 5: Maak de tijdsas van reeks x gelijk aan de tijdsas van reeks y

stap 6: Voer voor elke windrichtingsklasse en windsnelheidsklasse paar een regressie uit van het type $y = a*x$ en bereken tevens de spreiding van het residue

stap 7: Sla de resultaten op in een vhg-structure en save deze

FILE: ConfineDias2Dia.m

ConfineDias2Dia - Voeg de waarden van verschillende dia's samen en hou daarbij de tijdsas van de eerste dia aan

INPUT:

varargin: <cell array> met tenminste 2 dia's

Output:

W: <matrix> van afmeting: lengte van waarden eerste dia bij
aantal dias gespecificeerd in varargin (length(varargin))
S: (optioneel) <matrix> zelfde afmeting als W, met de stdW
(spreidingen)

Aanpak:

Stap 1a: Maak de matrix W en zet de waarden van de eerste dia
(varargin{1}) in de eerste kolom, de andere kolommen bevatten
NaN's
Stap 1b: Als nargin == 2, Maak de matrix S en zet de stdW van de eerste dia
(varargin{1}) in de eerste kolom, de andere kolommen bevatten
NaN's
Stap 2a: Pak een voor een de waarden van de dias in varargin{2:end}
Stap 2b: Pak een voor een de stdW van de dias in varargin{2:end}
Stap 3: Maak de tijdsas van de vector gelijk aan de tijdsas van de
eerste dia, vul hiaten op met NaN's, maak gebruik van het feit
dat de dia's dezelfde tijdstap hebben

FILE: EstimateInit.m

EstimateInit - Schat de reeksen van de hoofdsensoren bij m.b.v.
verhoudingsgetallen

INPUT:

obj: handle van de calling pushbutton
event: leeg
db: de database
- de hoofdsensoren gedefinieerd in db.data.loc
- de W-velden van de hoofdreeksen

OUTPUT:

INDIRECT: de velden V en stdV van de reeksen van de hoofdsensoren in
de database worden geupdate

AANPAK:

stap 1: Bepaal de hoofdsensoren voor de te schatten variabelen
en voor de windrichting en windsnelheid
stap 2: Selecteer een locatie en haal de windrichting en
windsnelheid reeksen van deze locatie op deze bepalen de
klassen voor de reeksen op deze locatie
stap 3: Selecteer een variabele (y) voor de locatie die in stap 2
geselecteerd is
stap 4: Selecteer dezelfde variabele als in stap 3 op alle andere
locaties (x) en selecteer ook de windsnelheid op de locatie
van stap 2
stap 5: Maak de tijdsas van de reeksen x gelijk aan de tijdsas van reeks
y
stap 6: Haal voor elke windrichtingsklasse en windsnelheidsklasse
de matrices op die met CalcEsti mateInit zijn geschat en
bereken het gewogen gemiddelde van de aanwezige waarden,
(weging is omgekeerd evenredig met de spreidingsmatrix)
stap 7: bereken Th0 als een schatting tussen de heersende
windrichting en de golfrichting in de vorige periode, stdV
wordt vast gekozen op 30

FILE: GetSensorMatrix.m

SensorMatrix - zet de hoofdsensoren in een matrix

Input:

LocStructure: <structure> (u.data.loc) met de nummers van de reeksen van de hoofdsensoren met tenminste de velden:

- 1) WINDRTG
- 2) WINDSHD
- 3) De velden gespecificeerd in Vars

en met tenminste de locatienamen gespecificeer in Locs aanwezig in LocStructure.sLoccod

Locs: <cell array> met de locaties waarvoor de hoofdsensoren moeten worden gebruikt (moeten aanwezig zijn in LocStructure.sLoccod)

Vars: <cell array> met de variabelen waarvoor de hoofdsensoren moeten worden gebruikt (moeten velden zijn van LocStructure)

Wind: <cell array> met de windvariabelen waarvoor de hoofdsensoren moeten worden gebruikt (moeten velden zijn van LocStructure)

Output:

VarMat: <matrix length(Locs) bij length(Vars)> met hoofdsensoren per locatie voor de variabelen gespecificeerd in Vars

WindMat: <matrix length(Locs) bij 2> met hoofdsensoren per locatie voor WINDRTG en WINDSHD

warning: <cellstring> eventuele foutmelding of waarschuwing die in de 'calling' functie zal worden gemeld

Locs: <cell array> met de locaties waarvoor de hoofdsensoren moeten worden gebruikt (zijn aanwezig in LocStructure.sLoccod)

Vars: <cell array> met de variabelen waarvoor de hoofdsensoren moeten worden gebruikt (zijn aanwezig in LocStructure)

Wind: <cell array> met de windvariabelen waarvoor de hoofdsensoren moeten worden gebruikt (zijn aanwezig in LocStructure)

FILE: buildmatstring.m

buildmatstring - maak de matrix behorend bij de factor en sigma velden van het vhg veld in de database

FILE: buildpopupstring.m

buildpopupstring - maak de matrix behorend bij de factor en sigma velden van het vhg veld in de database

FILE: do_import_regmodel.m

do_import_regmodel - Import regression model to the database

CALL:

u = do_import_regmodel(vhg,u,fname)

INPUT:

vhg : Structure with regression model data
met relevante velden

- richting

- snelheid

- locs

- factor

- sigma

u : De database

fname : (Optional) string with the filename

OUTPUT:

u : Updated database

FILE: regbhconstants.m

definieer een aantal constantes die specifiek zijn voor het regressiebeheer scherm

FILE: regbhview.m

INPUT

udnew : database

opt : GUI settings

upd : te updaten scherm elementen

C : constantes

HWIN : handle van hoofdscherm

FILE: regressiebeheer.m

netwerkbeheer - import and export neural networks

FILE: timeshift.m

shiftvector - verschuif de vector in de tijd

INPUT:

```
vector      - <array> de vector de verschoven moet worden in de tijd
deltaT      - <array of int> een vector met tijdverschuivingen b.v.
              [-2 -1 0 1]
```

OUTPUT:

```
shiftvector - <array> lengte vector bij lengte deltaT elke kolom van
              shiftvector is vector verschoven in de richting van een
              element van deltaT en aangevuld met NaN's aan de randen
```

```
=====
DIRECTORY: C:\d\modelit\wavixIV\CONHOP
=====
```

INHOUD:

```
ChainRule.m
EstimateConhop3.m
EstimateNN.m
NN_depend.m
SimulNN.m
SimulateNeuralNetwork2.m
TestVars.m
conhopobjfun2.m
dampnewton.m
dispdump.m
matgetvar2.m
selectPredictable.m
sensor2network.m
simstructnet3.m
start_conhop.m
steepDescent.m
```

```
*****
```

```
FILE: ChainRule.m
ChainRule -
```

INPUT:

```
transform1
+----type
| +----linear:   time invariant linear transform: y=A.x
| +----diagonal: time dependent transform only gradient of transform specified
| |             number of inputs is equal to number of outputs
| |             transformation matrix is diagonal
| +----full:    time dependent transform. all arguments specified
+----M:         transform matrix. shape depends on type
+----Ninput:   number of inputs (height)
+----Noutput:  number of outputs (height)
```

```
*****
```

```
FILE: EstimateConhop3.m
EstimateConhop - Schat de reeksen van de hoofdsensoren bij m.b.v. de
                  Conhop operator, callback van het databeheer scherm
```

INPUT

```

opt: berekeningsopties
+---- improveInit:   (1/0) verbeter initiele schatting
+---- fastRepair:    (1/0) toepassen fast repair
+---- fastRepairVal: (N ) voer N iteraties uit
+---- optim:         (1/0) pas conhop optimalisatie toe
+---- estimateNeven  (1/0) schatten op basis van Neven sensoren
+---- estimateall    (1/0) schatten alles

```

```
*****
```

```
FILE: EstimateNN.m
```

```
EstimateNN - Schat de reeksen van de hoofdsensoren bij d.m.v. een neuraal
           netwerk
```

```
*****
```

```
FILE: NN_depend.m
```

```

IDsDepend: de IDs van de benodigde invoer reeksen (voorzover
           gevonden)
IDsMissing: De IDs van de te voorspellen reeksen waarbij geen Neural
           netwerk kon worden gevonden
WkeyMissing: de sleutels van benodigde invoerreeksen die niet
           aanwezig zijn in het werkgebied
doubleIDS: sleutels behorend bij IDs die op meer dan 1 manier
           berekend kunnen worden.
NN_indx: de indices van de te gebruiken netwerken die corresponen met
           IDs (0 op de plaatsen van niet gevonden reeksen)

```

```
*****
```

```
FILE: SimulNN.m
```

```
simultane toepassing neurale netwerken
```

```
INPUT
```

```

W           : <matrix> met data (aantal perioden bij aantal reeksen)
stdW        : <matrix> met stdafw (zelfde grootte als W)
f_required  : <lineaire index> de elementen die herberekend moeten worden
f_3         : de gevraagde KOLOMMEN uit de Jacobiaan, dat zijn
           de vrij te variëren variabelen
NN_data     : read only structure structure met gegevens voor Neurale Netwerk
+---- LocVarVeldapp uit: M.LocVarVeldapp
+---- diatijd      uit: M.LocVarVeldapp
+---- NetworkStructObj
+---- NetworkDepend: structure die onderlinge afhankelijkheden
           |           | weergeeft bij NetworkStructObj
           | +----indxUsed: vector met indices die verwijzen naar LocVarVeldapp en
           |           | weergeven welke reeks gebruikt wordt als input
           | +----tShift:  vector die de tijdverschuiving weergeeft
           | +----indxResult: index van de geschatte reeks
+---- SensorIndx: voorspelde reeks (Opm. In toekomst zal deze
           verdwijnen. Deze is redundant met indxResult

```

```
OUTPUT
```

```

W_est: vector met geschatte waarden (hoogte=lengthe(f_required))
stdW : vector met bijbehorende standaard afwijkingen
      (hoogte=lengthe(f_required))
JacW: Matrix met gedeelte van de Jacobiaan
      (hoogte=lengthe(f_required) breedte=lengthe(f_3))

```

```
*****
```

```
FILE: SimulateNeuralNetwork2.m
```

```
SimulateNeuralNetwork - Simuleer het neurale netwerk in NetworkStruct
```

```
INPUT:
```

```

W
stdW
LocVarVeldapp

```

```

NetworkStruct - <struct> structure met o.a. een veld netwerk
                met een netwerk structuur

```

```

periodeIndex  - <optioneel> index van de te selecteren kolommen van de
                data, wordt gebruikt met conhop

```

```
OUTPUT:
```

```

output        - <vector> van lengte periodeIndex met voorspelde waarden
sigma         - <vector> van lengte periodeIndex met voorspelde standaarddeviaties
message       - <string> met eventuele boodschap (nog niet gebruikt)
jaco          - <sparse matrix> hoogte=#periodeIndex;
                breedte=prod(size(W))

```

```
*****
```

```
FILE: TestVars.m
```

```
TestVars - Test if the variables in NetworkStruct.data are present in the database
```

```
INPUT:
```

```

M              - <struct> de kopie van de centrale database (via db2mat)
NetworkStruct - <struct> met relevante velden
                NetworkStruct.data.invoer en
                NetworkStruct.data.uitvoer
mode          - <string> mogelijke waarden
                'training'
                'simulation'

```

```
*****
```

```
FILE: conhopobjfun2.m
```

```
conhopobjfun - de doelfunctie voor de Consistency Measure
```

```
INPUT
```

```

x              : de te variëren variabele
funpars       : cell array met parameters, in de volgende volgorde:
                M              : een kopie van het werkgebied
                SensorIndx     : SensorIndx(i) hoort bij het netwerk met index i

```

```

* geeft aan welke kolom in matrix M voorspeld wordt
* correspondeert met NetworkStructObj
NetworkStruct : struct array met neurale netwerken
I_hiaat       : Lineaire index naar de hiaten (De index van x!!)
I_affected    : Indices van de elementen die door I_hiaat worden
               beïnvloed. Cell-array correspondeert met NetworkStruct
I_jacaffected : geeft aan welke hiaten verantwoordelijk
               zijn voor beïnvloeding, correspondeert met I_affected,
uit
               I_affected kan de reeks en tijdstip gehaald worden
waarop
               I_jacaffected van toepassing is

INPUT
  x      : <vector> punt waarop de doelfunctie geevalueerd moet worden
  funpars: <cell array> met inhoud:
          W      : vector met waarnemingen
          stdW   : vector met waarnemingsfouten.
                  Let op!! de elementen f_3 zijn hierin al op nul gezet.
          NN_data: de neurale netwerk gegevens
          f_3    : indices van vrij te variëren waarden (lineaire index in
W)==I_wederzijds
          f_4    : indices van door f_3 beïnvloede waarden (lineaire index in W)
          E4tE3  : Het resultaat van een vermenigvuldiging van E4'*E3
          E4tE1W : Het resultaat van een vermenigvuldiging van E4'*E1*W

OUTPUT
  f: doelfunctiewaarde
  g: gradient
  H: hessiaan (J'*J)

```

```
*****
```

```
FILE: dampnewton.m
```

```
dampnewton - Levenberg-Marquardt type damped Newton method for nonlinear
             optimization
```

```
CALL:
```

```
[running_time,x,f,g,H] = dampnewton(fun,par,x,options)
```

```
INPUT:
```

```

fun      - <function handle> moet gedefinieerd zijn als [f,g,H] = fun(x,par)
par      - eventuele parameters voor fun, mag leeg zijn
x0       - <vector> startpunt
options  - <struct> met de volgende velden:
          mu           : starting value for the Marquardt parameter.
          epsilon1    : ||g||_inf <= epsilon1 (formula 5.12)
          epsilon2    : ||dx||2 <= epsilon2*(epsilon2 + ||x||2)
          maxiter     : maximum number of iterations

```

```
OUTPUT:
```

```

x        - optimale waarde voor de variabelen
f        - functiewaarde
g        - gradient
H        - Hessiaan

```

```
METHODE:
```

```

- Section 5.2 in P.E. Frandsen, K. Jonasson, H.B. Nielsen,
  O. Tingleff: "Unconstrained Optimization", IMM, DTU. 1999.

```

- "damping parameter in marquardt's method"
 Hans Bruun Nielsen, IMM, DTU. 99.08.10 / 08.12

FILE: dispdump.m
 display message and store it for report

FILE: matgetvar2.m
 matgetvar - genereer de reeks(en) (W en stdW) voor een opgegeven locatie
 variabele veldapparaat tijdstip(verschuivingen) combinatie
 vanuit de matrix die gemaakt is met db2mat

INPUT:

W(P,N) : meetdata volledige verzameling
 stdW(P,N): meetfouten
 Wkey(N) : bijbehorende sleutels (ZIE sleutel2struct)
 +----sLoccod: char str
 +----sParcod: char str
 +----sVatcod: char str
 WTkey(M) : sleutels van op t ehalen reeksen (ZIE parseNNInvoer)
 +----sLoccod: char str
 +----sParcod: char str
 +----sVatcod: char str
 +----tShift: integer

OUTPUT:

W - <array> de waarden voor de loc var veldapp tijd combinatie
 stdW - <array> de deviaties voor de loc var veldapp tijd combinatie
 index - <int> de index van de dia die hoort bij de loc var veldapp
 tijd combinatie

N.B. als var == 'WINDRTG' of 'Th0' dan wordt var opgesplitst
 in een x- en y-richting

FILE: selectPredictable.m
 Filter IDsPredict van reeksen die mogen worden bijgeschat

INPUT

db: wgb structure
 NN_name: naam van het struct array dat de NN herbergt
 IDsPredict: te voorspellen reeksen

OUTPUT

IDsPredict: IDs van reeksen waarvoor geldt:
 - Alleen reeksen waarvoor een NN aanwezig is worden voorspeld
 - Alleen Neurale netwerken waarvoor alle invoer reeksen aanwezig zijn
 mogen worden gebruikt.
 report: bijdrage aan logboek

SEE ALSO:
 NN_depend
 initialiseer de uitvoer

```
*****
FILE: sensor2network.m
  sensor2network - bepaal de index van het netwerk dat bij elke hoofdsensor hoort

INPUT
  C: constants
OUTPUT
  SensorIndx: de index in het WGB van de Hoofdsensor reeksen
  NetIndx    : de index van het Neural Network (bijbehorende NN)
  comment    : bevat eventueel foutmelding
```

```
*****
FILE: simstructnet3.m
  simstructnet - simuleer met een netwerk in structuurformaat

INPUT:
  netstruct      - <struct> zie emptystruct('netwerk')
  inputdata(M,P) - <matrix> aantal inputs bij aantal patronen

OUTPUT:
  result(H,P) - <matrix> (aantal outputs maal aantal patronen) bij aantal members
  T(MxH,P)    - <matrix> met invloed van (1->1,2->1,... 1->2,2->2,...etc)
                (doorgaans is er maar 1 output en geldt H=1)
```

AANPAK:
 deze functie is in principe gelijk aan sim van de neural network toolbox, met het verschil dat

- 1) alleen het resultaat van de simulatie wordt teruggegeven
- 2) er gewerkt wordt met een structuur en niet met een netwerk object
 alle gegevens zijn te vinden in
 netstruct.ensemble.member: met de bias en gewichten
 netstruct.netwerk: met de netwerkstructuur: aantalneuronen
 transferfuncties aantal lagen etc.
- 3) deze routine werkt alleen voor feedforward netwerken

```
*****
FILE: start_conhop.m
  GUI voor opstarten CONHOP
  EstimateConhop
  is the database empty?
```

FILE: steepDescent.m

No help comments found in steepDescent.m.

Use the Help browser Search tab to [search the documentation](matlab:docsearch C:\d\modelit\wavixIV\CONHOP\steepDescent.m), or type "[help help](matlab:help help)" for help command options, such as help for methods.

=====
 DIRECTORY: C:\d\modelit\wavixIV\HULPFUNCTIES
 =====

INHOUD:

ComputeStd.m
 binstatus2donstat.m
 binstatus2type.m
 classify.m
 constantes_wavix.m
 db2mat.m
 dbtools.m
 donstat2binstatus.m
 emptystruct.m
 eval_outliers.m
 fieldnameprint.m
 get_C.m
 get_db.m
 get_opt_main.m
 getbinstatus.m
 listW3H.m
 mattools.m
 parseNNInvoer.m
 reeksaanduiding.m
 separatestr.m
 setbinstatus.m
 sleutel2struct.m
 wavicon.m
 wavixhelp.m

FILE: ComputeStd.m

ComputeStd - compute the measurement error of series given by dia

INPUT:

db: database structure with relevant fields: db.data.dia, db.data.loc
 dia: dia for which the measurement error has to be determined

OUTPUT

sigma: the measurement error of the series
 sigma = NaN if not all data needed for determination of sigma are present in the database

DOCUMENTATION: WAVIX RKZ-471 All Fours NeuralTech december 1997 page 7

(Het groene boekje)

De nauwkeurigheid voor de windrichtingsklassen wordt gezet op 30 als

- richting(in graden) gelijk is aan 0 (windstilte)

(richting 0 wordt weergegeven als 360 !)
 - richting(in graden) gelijk is aan 990
 (veranderlijke wind)

DATE: 05-Sep-2004

FILE: binstatus2donstat.m

transformeer Wavix binaire status naar Donar codering

INPUT

status: (uint8 vector) binnen wavix gebruikte binaire status
 bit 1: Hiaat
 bit 2: Controle
 bit 3: Outlier
 bit 4: Validatie status
 bit 5: Herkomst

OUTPUT

Donar status: vector met waardes:
 0 : gewone waarneming
 25: geïnterpoleerde waarde
 99: hiaat

ZIE OOK

donstat2binstatus
 binstatus2donstat
 setbinstatus
 getbinstatus
 binstatus2type

FILE: binstatus2type.m

binstatus2type - stel geaggregeerd statutype in (combinatie van alle status bits)

INPUT

status: uint8 bits met status

OUTPUT

statustype: bepaald weld symbool geplot wordt, mogelijke waarden
 C.ALLES
 C.HIAAT
 C.OUTLIER
 C.ANDERS

bvalide : 1: valide 0: nog niet valide
 bherkomst : 1: geïnterpoleerd 0: niet geïnterpoleerd
 aggregstatus:
 aggregstatus.numhiaat
 aggregstatus.numoutlier
 aggregstatus.numanders
 aggregstatus.numvalide
 aggregstatus.numtotal

WERKWIJZE

deze procedure roept eerst "getbinstatus" aan om de uint8 statuscodes

te ontcijferen. Daarna wordt op basis van een aantal beslisregels een statutstype bepaald. Om te voorkomen dat een tweede aanroep van getbinstatus noodzakelijk is worden ook enige andere attributen geretourneerd.

ZIE OOK
 donstat2binstatus
 binstatus2donstat
 setbinstatus
 getbinstatus
 binstatus2type

FILE: classify.m

classify - deel de vector W in in klassen

Input:

W: <nx1 matrix> met de waarden die geclassificeerd moeten worden

Klassen: <rowvector> met de klassegrenzen voor W

Output:

Classificatie: <nx1 matrix> met de klassenummers (nul of length(Klassen) als een element niet ingedeeld kan worden in een van de opgegeven klassen

Voorbeeld: klassificeer de vector [1 m.b.v. de klassen [0 5 10 15]
 12
 9]
 stap 1: maak van [0 5 10 15] A:=[0 5 10 15
 0 5 10 15
 0 5 10 15]
 stap 2: maak van [1 B:=[1 1 1 1
 12 12 12 12
 9] 9 9 9 9]
 stap 3: B > A --> C:=[1 0 0 0
 1 1 1 0
 1 1 0 0]
 stap 4: bereken C.*[1 2 3 4 D:=[1 0 0 0
 1 2 3 4 --> 1 2 3 0
 1 2 3 4] 1 2 0 0]
 stap 5: de klassen zijn nu het max(D, [],2)
 oftewel [1
 3
 2]

FILE: constantes_wavix.m

constantes_wavix - definieer constantes voor de WAVIX applicatie

CALL

constantes_wavix(dummy_arg)
 C=constantes_wavix(dummy_arg)

INPUT

dummy_arg: het definieren van tenminste 1 invoer argument heeft tot gevolg dat een hulpscherm wordt gestart voor het instellen van de opties

OUTPUT

C: een structure met een groot aantal velden
 ieder veld bevat een constante die in de applicatie gebruikt wordt

AANPAK

Door gebruik te maken van constanten wordt vermeden dat door kleine spelfouten fouten in de applicatie sluipen die niet gedetecteerd worden met een foutmelding.
 Bovendien kunnen opties op deze wijze centraal gewijzigd worden

FILE: db2mat.m

db2mat - zet de centrale database om in matrices

INPUT:

db - <struct> centrale wavix database
 IDs: IDs van op te halen reeksen
 starttime - (optional) <datenum> het tijdstip van het begin van het tijdsinterval
 waarvoor de data geselecteerd moet worden
 endtime - (optional) <datenum> het tijdstip van het eind van het tijdsinterval
 waarvoor de data geselecteerd moet worden
 N.B. als starttime en endtime niet gespecificeerd zijn dan
 worden voor deze tijden de vroegste en laatste tijd van
 alle dias gebruikt

OUTPUT:

Mat - <struct> de database omgezet in een structure met de velden
 - DiaIndx reeks index ivm terugschrijven data
 - Wkey <struct> reekssleutel voor zoeken in NN definitie
 +---- sLoccod
 +---- sParcod
 +---- sVatcod
 - tijdsas gemeenschappelijke tijdsas van de vroegste
 tot laatste waarneming van alle dias
 - diatijd <matrix> dimensies: 2 bij aantal dias met in:
 rij1 de startindex voor de tijdsas van de dia
 rij2 de eindindex voor de tijdsas van de dia
 - W In elke kolom van de velden W, stdW, V, stdV en
 status
 - stdW de waarden van de dia op de correcte
 - V plek t.o.v. de tijdsas
 - stdV
 - status

FILE: dbtools.m

dbtools -

```
*****
```

```
FILE: donstat2binstatus.m
```

```
INPUT
```

```
Donar status: vector met waarden:
0 : gewone waarneming
25: geiniterpoleerde waarde
99: hiaat
```

```
OUTPUT
```

```
status: (uint8 vector) binnen wavix gebruikte binaire status
bit 1: Hiaat
bit 2: Controle
bit 3: Outlier
bit 4: Validatie status
bit 5: Herkomst
```

```
ZIE OOK
```

```
donstat2binstatus
binstatus2donstat
setbinstatus
getbinstatus
binstatus2type
```

```
*****
```

```
FILE: emptystruct.m
```

```
emptystruct - maak structures aan die al het goede formaat hebben
```

```
INPUT:
```

```
type: <string>
```

```
OUTPUT:
```

```
'convnetwerk' -> S.naam = ''
                  S.status = 0          %0 niet getrained, 1 getrained, 2 getrained
zonder uitvoer informatie
                  S.netwerk = emptystruct('objnetwork')
                  S.data = emptystruct('data')
                  S.output = []
                  S.target = []
                  S.preprocess = emptystruct('preprocess')
                  S.ensemble = emptystruct('ensemble')
                  S.output = emptystruct('output')
                  S.Delta = []
'netwerk' ->      S.naam = ''
                  S.status = 0          %0 niet getrained, 1 getrained, 2 getrained
zonder uitvoer informatie
                  S.netwerk = emptystruct('objnetwork')
                  S.data = emptystruct('data')
                  S.output = []
                  S.target = []
                  S.preprocess = emptystruct('preprocess')
                  S.ensemble = emptystruct('ensemble')
                  S.output = emptystruct('output')
                  S.Delta = 0
```

```

'ensemble'  ->  S.herhalingen = []
                S.trainingset = []
                S.validatieset = []
                S.testset = []
                S.member = emptystruct('member')
'member'    ->  S.IW = []
                S.LW = []
                S.b = []
                S.output = []
                S.testindex = []
'preprocess'-> S.meanp = []
                S.stdp = []
                S.meant = []
                S.stdt = []
                S.transmat = []
                S.pca = []
'tmpnetwork'-> S.naam = []
                S.invoer = ''
                S.uitvoer = ''
                S.neuronen = []
                S.transferfunctie = ''
                S.trainfunctie = ''
                S.doelfunctie = ''
                S.herhalingen = []
                S.pca = []
                S.trainingset = []
                S.validatieset = []
                S.testset = []
'parameters' S.epochs = 100
                S.goal = 0
                S.lr = 0.0100
                S.lr_dec = 0.7000
                S.lr_inc = 1.0500
                S.max_fail = 5
                S.mem_reduc = 1
                S.min_grad = 1.0000e-06
                S.mu = 0.0010
                S.mu_dec = 0.1000
                S.mu_inc = 10
                S.mu_max = 1.0000e+010
                S.max_perf_inc = 1.0400
                S.mc = 0.9000
                S.deltamax = 50
                S.delta_inc = 1.2000
                S.delta_dec = 0.5000
                S.delta0 = 0.0700
                S.sigma = 5.0000e-005
                S.lambda = 5.0000e-007
                S.searchFcn = 'srchbac'
                S.scale_tol = 20
                S.alpha = 0.0010
                S.beta = 0.1000
                S.delta = 0.0100
                S.gama = 0.1000
                S.low_lim = 0.1000
                S.up_lim = 0.5000
                S.maxstep = 100
                S.minstep = 1.0000e-006
                S.bmax = 26
                S.show = 25
                S.time = Inf

```

```

'dia'->          S.ID = 1
                  S.blok = []
                  S.stdW = []
                  S.V = []
                  S.stdV = []
                  S.status = [] %status toegevoegd 21 aug 2004
'vhg'->          S.richting = []
                  S.snelheid = []
                  S.locs = []
                  S.factor = []
                  S.sigma = []
'model'->        S.stuurfile = ''
                  S.netwerkfile = ''
                  S.vhgfile = ''
'objnetwork' -> S.neuronen = []
                  S.transferfunctie = ''
                  S.trainfunctie = ''
                  S.doelfunctie = ''
                  S.parameters = emptystruct('parameters')
'toestand' ->    S.main = ''
                  S.sub = ''

```

```

*****
FILE: eval_outliers.m
evaluate outliers
Flushes and stores db

```

```

*****
FILE: fieldnameprint.m
remove illegal symbols from fieldname

```

```

*****
FILE: get_C.m
get_C : haal constantes structure op

CALL
    C=get_C

INPUT
    none

OUTPUT
    C: constantes structure

SEE ALSO
    mbdstore
    get_db

```

```
FILE: get_db.m
  get_db : haal db structure op uit userdata van hoofdscherm

CALL
  [db,C]=get_db

INPUT
  none

OUTPUT
  db: centraal undoredo object

SEE ALSO
  mbdstore
  get_C
```

```
FILE: get_opt_main.m
  Macro voor het ophalen van de options structure van het Wavix hoofdscherm
```

```
FILE: getbinstatus.m
  INPUT
    status: haal statusbits op voor alle statutypes of alleen voor
    hiaat status

  OUTPUT
    Binary status informatie (0 of 1) (uint8 formaat)
      bit 1: Hiaat (FACTOR 1)
      bit 2: Controle (FACTOR 2)
      bit 3: Outlier (FACTOR 4)
      bit 4: Validatie status (FACTOR 8)
      bit 5: Herkomst (FACTOR 16)

  ZIE OOK
    bitget
    donstat2binstatus
    binstatus2donstat
    setbinstatus
    getbinstatus
    binstatus2type
```

```
FILE: listW3H.m
  listW3H - Hulpfunctie. vul een struct array van W3H structures op basis
```

van een WAVIX dia array

INPUT

dia: Wavix dia array

Velden:

ID

blok

stdW

V

stdV

etc.

indices: te gebruiken indices (default: alle)

OUTPUT

W3Hs: W3H aarray

SEE ALSO

listRKS

FILE: mattools.m

mattools - voer operaties uit op Mat, Mat is verkregen uit de database door db2mat toe te passen

FILE: parseNNInvoer.m

Utility of string op bepaalde manier te interpreteren en als struct weg te schrijven en naar struct te converteren

INPUT

invoer: cellarray met specificatie inputs (totaal #INPUTS stuks). Op de volgende manier opgeslagen

VOORBEELD

'IJMDMNTSPS H1/3 WAVERDDR -1 -2 -4'

'IJMDMNTSPS Hm0 WAVERDDR 0 -1 -2'

OUTPUT

S(AANTAL INPUTS): struct array met sleutels

+----sLoccod: char str

+----sParcod: char str

+----sVatcod: char str

+----tShift: integer

SEE ALSO

sleutel2struct

FILE: reeksaanduiding.m

reeksaanduiding - maak een standaard regel en header aan

INPUT

dia: element(en) uit Wavix dia reeks

FILE: separatestr.m

separatestr - separates the string in parts delimited by a space ' '

FILE: setbinstatus.m

INPUT

Binary status informatie (0 of 1)

bit 1: Hiaat

bit 2: Controle

bit 3: Outlier

bit 4: Validatie status

bit 5: Herkomst

OUTPUT

status: statusbits voor alle data

ZIE OOK

donstat2binstatus

binstatus2donstat

setbinstatus

getbinstatus

binstatus2type

FILE: sleutel2struct.m

Utility om parameters uit regel te lezen en weg te schrijven in structure

INPUT

LocVarVeldapp: cell array met reeks-sleutels in formaat %s %s %s

Voorbeeld:

'EURPFM TH1/3 SAAB' 'EURPFM Tm02 SAAB' 'EURPFM TH1/3 WAVEC'

OUTPUT

S(AANTAL NETWERKEN): struct array met sleutels

+----sLoccod: char str

+----sParcod: char str

+----sVatcod: char str

SEE ALSO

parseNNInvoer

FILE: wavicon.m

Stel de icon in voor wavix applicatie

```
*****  
FILE: wavixhelp.m  
wavixhelp -
```