

Functionele omschrijving PreSRM v1.3

Alex Vermeulen – ECN

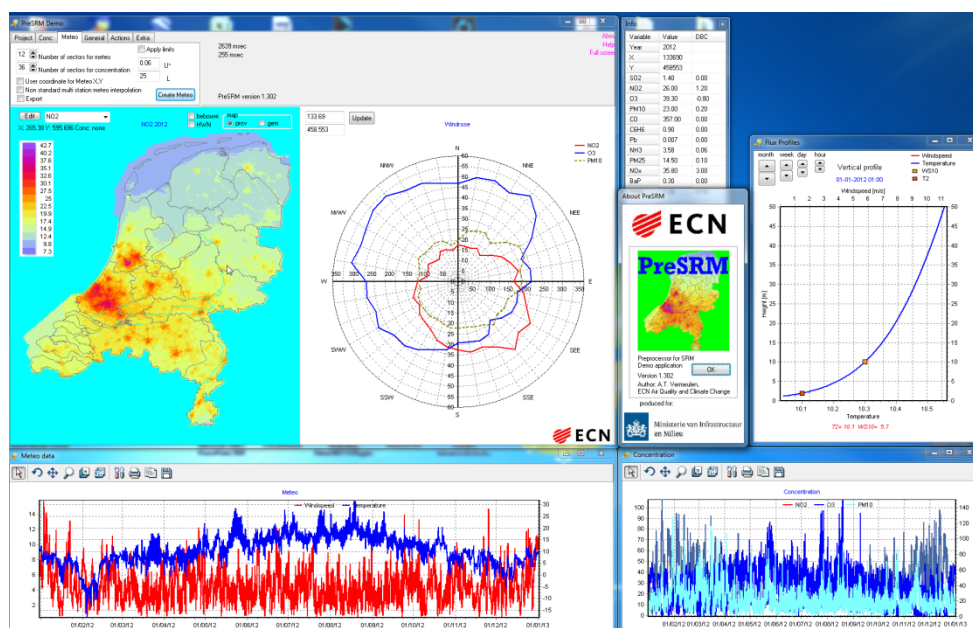
Mei 2013 - versie 1.303

Inleiding

PreSRM is een software bibliotheek voor rekenmodellen gebruikt bij de beoordeling van luchtkwaliteit. In Nederland schrijft de Regeling Beoordeling Luchtkwaliteit (RBL) voor onder welke condities welke rekenmethodes gebruikt moeten worden om aan de luchtkwaliteitsnormen te toetsen. De RBL is onderdeel van de Wet Luchtkwaliteit. In de RBL worden drie Standaard RekenMethodes (SRM's) beschreven. Tevens wordt daarin vastgesteld welke invoergegevens gebruikt moeten worden voor emissies, achtergrondconcentraties, meteorologie en terreinruwheid. Deze gegevens worden jaarlijks geüpdate en gepubliceerd via de website van het ministerie van IenM. Deze gegevens zijn niet zonder meer in de modellen te gebruiken. Een aanzienlijk aantal bewerkingen, per SRM verschillend, is noodzakelijk. Zo moeten vaak de meteorologiegegevens geïnterpoleerd naar de locatie van de berekening, moeten de concentratie gegevens geclassificeerd naar windrichting, gecorrigeerd voor dubbeltelling of verfijning van het grid, etc. PreSRM neemt veel van deze bewerkingen uit handen en biedt de ontwikkelaar van een SRM een transparante en eenvoudige interface waarmee de benodigde gegevens direct en efficiënt verkregen worden.

Omvang en levering van preSRM

PreSRM wordt geleverd inclusief de laatste officiële basisgegevens van VROM voor meteorologie, ruwheid en grootschalige achtergrondconcentraties. PreSRM wordt eenvoudig geïnstalleerd via een installatieprogramma dat op de achtergrond geïnstalleerd kan worden tegelijk met de eigenlijke rekenmethode applicatie. In deze versie wordt PreSRM geleverd als 32 en 64 bits bibliotheek voor de MS-Windows omgeving en 64 bits voor MacOS-X. Een interface unit voor de Delphi ontwikkelomgeving wordt meegeleverd, maar PreSRM is eenvoudig ook in andere ontwikkelomgevingen te gebruiken. Een simpele console mode demo-applicatie en een uitgebreide GUI demo-applicatie wordt meegeleverd inclusief bronbestanden.



Na installatie neemt PreSRM ongeveer 395 MB in op de harde schijf. Dit komt voornamelijk door de omvang van de GCN bestanden voor de periode 1995-2030. Bij het uitvoeren van een prognostische berekening voor een gebied ter grootte van heel Nederland en alle mogelijke stoffen kan PreSRM een aanzienlijk beslag op het werkgeheugen leggen van maximaal 350 MB. Aangezien volgens RBL het maximale gebied voor een berekening slechts 25 bij 25 kilometer kan bedragen blijft in de praktijk het gebruik van werkgeheugen beperkt tot enkele tientallen MB.

Belangrijkste wijzigingen in PreSRM V1.3

Versie 1.3 van PreSRM is aangepast aan de RBL gegevens van 2012. Deze versie werkt alleen met de bijgeleverde dataset met bestanden voor meteorologie, ruwheden en GCN.

- Het pad voor de PreSRM gegevens in de call naar PS_Proj_Init dient in deze versie gegeven te worden als pAnsiChar parameter en de string waar deze naar pAnsiChar verwijst dient een AnsiString te zijn. Dit expliciete type is nodig omdat in nieuwere versies van Delphi het default String type uit double-byte characters bestaat, die niet compatibel zijn met de single byte AnsiString, het oude default String type.
- PreSRM wordt geleverd als een 32- en een 64-bits bibliotheek (dll) voor geschikt MS Windows XP, Vista, 7 and 8
- De ruwheidsgegevens zijn gebaseerd op de iets nieuwere LGN6 landgebruikskaart (nog niet in versie 1.302).
- De ruwheidsgegevens kunnen ook rekening houden met de invloed van de lokale orografie op de ruwheid. Zie de extra parameter in PS_Roughness (pag. 6) en PS_Roughness_Area (pag. 7). Vanwege de compatibiliteit met eerdere versie van PreSRM is het mogelijk het meetellen van deze extra ruwheid aan en uit te zetten. In het algemeen leidt het meewegen van de topografie op veel plaatsen tot redelijk grote toename van de lokale ruwheid, hetgeen in het algemeen leidt tot een sterkere verdunning van emissies. Lokale depositie wordt tevens sterker, maar dit effect kan geheel of gedeeltelijk (over)gecompenseerd worden door lagere concentraties door diezelfde grotere ruwheid, geheel afhankelijk van de lokale omstandigheden en broneigenschappen.
- Om bij iedere aanroep van PS_ClassData_Init een textbestand met de uitkomst van de berekeningen te genereren dient een environment variabele %PRESRM_CLASSDATA% gedefinieerd te zijn. Dit tekstbestand wordt in de map voor tijdelijke bestanden geschreven. Deze map wordt gedefinieerd door de environment variabele %TEMP%. De bestandsnaam is classdata_XXXX.txt, waarin XXXX een random nummer van vier hexadecimale getallen is. De demo applicatie schrijft na iedere berekening van de meteorologie een bestand met de naam classdata.txt in de werk map.
- De in PreSRM opgenomen GCN bibliotheek schreef in eerder versies voor iedere aanroep informatie aan het bestand GCN.LOG toegevoegd. Bij gelijktijdig gebruik van meerdere berekeningen met PreSRM is het daarom verstandig om via de environment variabele GCNLOGDIR het uitvoerpad voor dit bestand per rekentaak te variëren. Bestaat deze variabele niet dan wordt geen informatie in GCN.LOG geschreven.
- Enkele bugfixes in de demo-applicatie

Belangrijkste wijzigingen in PreSRM V1.2

Versie 1.2 van PreSRM is aangepast aan de RBL gegevens van 2011. Deze versie werkt alleen met de bijgeleverde dataset met bestanden voor meteorologie, ruwheden en GCN.

- PreSRM levert nu ook GCN achtergronden voor roet (EC)
- Zeezout concentraties zijn nu gebaseerd op de nieuwe gegevens volgens Hoogerbrugge et al (2012). PreSRM lever alleen een indicatieve waarde voor de zeezoutcorrectie in dagen (2-4 afhankelijk van de locatie) en een indicatie voor de correctie in $\mu\text{g}/\text{m}^3$.
- Naar keuze kan PreSRM nu ook meteorologische gegevens leveren gebaseerd op geïnterpoleerde gegevens gebruikmakende van 17 weerstations i.p.v. de standaard die is gebaseerd op de stations Eindhoven en Schiphol. Default is nog de RBL methode op basis van twee stations.

- PreSRM bevat nu een routine **PS_MeteoVerticalProfile** voor de berekening van de verticale gradient van temperatuur en windsnelheid in de surface layer (ruwweg onderste 60 meter van de grenslaag)
- Ruwheden klassen voor SRM2 zijn nu conform RBL en hebben waarden van 0 tot en met 3.
- De PreSRM bibliotheek is nu ook beschikbaar in een versie voor MacOS (dynlib)

Belangrijkste wijzigingen in PreSRM V1.1

Versie 1.1 van PreSRM is aangepast aan de RBL gegevens van 2010. Deze versie werkt alleen met de bijgeleverde dataset met bestanden voor meteorologie, ruwheden en GCN.

- De ruwheidsgegevens zijn geactualiseerd op basis van LGN5+ kaarten, een dataset is aanwezig met resoluties van 250, 1000 en 4000 meter. Standaard wordt de 1000m resolutie benut.
- De gegevens voor 2010 op basis van actuele meteorologie en concentraties zijn toegevoegd. Prognoseberekeningen worden nu uitgevoerd voor 2011 en verder
- In deze versie worden de GCN waarden voor prognosejaren liggende tussen 2011 en 2015, 2015 en 2020, 2020 en 2030 afgeleid door interpolatie van de jaargemiddelde waarden van de omliggende jaren. Hierdoor is de omvang van de PreSRM data behoorlijk ingeperkt en legt minder beslag op de diskcapaciteit.
- De verfijning van de 'kale' GCN waarden voor Schiphol zijn nu correct doorgevoerd voor de periode met beschikbare gegevens: 2008-2030. Let op: de GCN aps-bestanden meegeleverd met PreSRM zijn gecorrigeerd voor de invloed van Schiphol.
- De correcties voor fijn stof concentraties uit de landbouw in deze GCN set zijn meegenomen voor de beschikbare periode: 2010-2030
- De correcties voor de invloed van de tweede Maasvlakte in de periode 2015-2030 zijn meegenomen
- Negatieve uurwaarden gerapporteerd door GCN worden gecorrigeerd tot een concentratie met waarde nul.
- Berekeningen voor de periode 2021-2029 zijn nu ook mogelijk
- Meteorologie is afkomstig van de ruwe data voor stations zoals verstrekt op de web-site van KNMI. PreSRM is voorbereid op gebruik van de meteorologische gegevens van 15 stations en interpolatie van de waarden van de meest dichtbijzijnde stations voor ieder projectmidden. Deze versie maakt nog gebruik van de door RBL voorgeschreven interpolatie van gegevens van Schiphol en Eindhoven.
- Het grenslaagschema voor meteorologie levert geen records meer met ontbrekende stabiliteitsgegevens. In het geval dat het iteratieschema geen oplossing geeft worden de laatst berekende waarden van het uur daarvoor toegepast.
- PreSRM hanteert een minimum windsnelheid van 0.5 m/s. Deze waarde is instelbaar van 0.1-10 m/s door het aanroepende programma.
- De melding van fouten in aanroep of gegevens aan het aanroepende programma is verbeterd, PreSRM geeft nu in meer gevallen en duidelijker foutmeldingen. Naar keuze kan PreSRM zelf de foutmeldingen aan de gebruiker verzorgen en eventueel via e-mail rapporteren aan ECN.
- Vanaf versie PreSRM v1.11 bevat PreSRM de mogelijkheid om de meteorologie aan te passen voor stabiele omstandigheden en lage wrijvingssnelheden.
- PreSRM bevat diverse kleinere bugfixes

Overzicht van de werking van PreSRM

PreSRM is zoveel mogelijk geoptimaliseerd voor snelheid. Door op een slimme manier zoveel mogelijk gegevens van te voren klaar te zetten kan het herhaald opvragen van gegevens heel snel verlopen. Om dat mogelijk te maken werkt PreSRM met enkele centrale objecten die, na initialisatie met de benodigde gegevens, aangeroepen kunnen worden door de applicatie. Deze objecten worden vanwege de portabiliteit naar andere ontwikkelgegevens steeds doorgegeven als pointers.

Een applicatie kan van ieder type object meerdere exemplaren aanmaken maar moet deze natuurlijk zelf bijhouden en zorgdragen voor het opruimen na gebruik. Voor dat opruimen is de functie **PS_Cleanup** aanwezig in PreSRM.

Het Project object

Het object Project definieert de grenzen van het rekendomein, het rekenjaar en het pad naar de benodigde gegevens. Voor alle gridpunten binnen het opgegeven domein zal PreSRM de bijbehorende objecten initialiseren. Volgens de RBL is een rekengebied maximaal 25 bij 25 km.

Het Project object heeft verder geen functies voor de gebruikers maar moet wel worden doorgegeven als referentie aan de overige objecttypes van PreSRM.

Het kan voorkomen dat het gridpunt midden in het domein geen geldige waarden oplevert in GCN. Dan wordt binnen een beperkt gebied rond dit punt gezocht naar het dichtstbijzijnde gridpunt met geldige gegevens. In het uiterste geval dat geen geldig punt gevonden kan worden, wordt een exceptie gegenereerd.

Via het project object kunnen tevens gegevens voor terreinruwheid en een suggestie voor de te hanteren zeezoutcorrectie voor PM_{10} worden opgevraagd.

Het PS Meteo object

Het **PS_Meteo** object berekent voor een punt binnen het **PS_Project** object de uurlijkse meteorologie, volgens de interpolatiemethode voorgeschreven door de RBL, uit de meteorologie voor de stations Eindhoven en Schiphol voor een op te geven terreinruwheid (standaard 0.08 meter). De opschaling van de wind naar blending height kan naar keuze met gebruik van de lokale en momentane stabiliteit of uitgaande van een neutrale stabiliteit plaatsvinden.

Als het punt wordt opgegeven met coördinaten X en Y van -1, dan wordt het middelpunt van het PS_Project domein aangehouden.

Met de functie **PS_Meteo_GetData** kan men een meteo record opvragen. Een meteo record is van het type **tMeteoRec** en bevat alle relevante gegevens voor SRM2 en SRM3 berekeningen.

Het PS_Conc object

Het PS_Conc project berekent voor een projectdomein en een op te geven set stoffen de jaargemiddelde en uurlijkse concentraties (indien beschikbaar in GCN). Bij initialisatie worden alle benodigde uurlijkse GCN gegevens geladen voor het midden van het domein zoals bepaald in het PS_Proj object. De jaargemiddelde gegevens worden opgevraagd voor alle gridpunten binnen het domein.

Met behulp van de opvraag functies **PS_MeanConc** en **PS_ConcGetData** wordt steeds voor een actueel punt de concentraties geleverd, in het geval van uurlijkse gegevens worden deze geschaald zodat het jaargemiddelde van de uurlijkse gegevens gelijk is aan het GCN gemiddelde voor dat punt. Tevens levert PreSRM indien van toepassing de gegevens voor de dubbeltellingscorrectie voor de invloed van het hoofdwegennet op de concentraties en houdt rekening met op- en overslag in Rijnmond en de verfiende concentratiegrids rond Schiphol.

Door beperkingen in de GCN concentratiegegevens voor bepaalde componenten gelden de volgende bijzonderheden:

- PM_{25} en NO_x concentraties zijn alleen beschikbaar vanaf 2009 resp. 2008
- Dubbeltellingscorrecties voor HWN zijn alleen beschikbaar vanaf 2007 (NO_x : 2008)
- CO en Benzeen concentraties voor prognose jaren (vanaf 2010) worden gelijkgesteld aan de concentraties van 2009, wegens ontbreken van prognose gegevens
- Pb concentraties zijn op dit moment alleen aanwezig voor de jaren 2001-2003 en 2006-2007. Voor overige jaren en prognoses worden de gegevens identiek aan die van 2007 geleverd.
- Uurlijkse variaties van de concentraties binnen een gerealiseerd jaar kunnen alleen berekend worden voor CO, NO_2 , NO_x (2009-2012), O_3 , PM_{10} en SO_2 . In de overige gevallen zijn de uurlijkse concentraties gelijk aan de jaargemiddelden.
- NH_3 gegevens zijn op dit moment alleen beschikbaar voor 2008-2011, 2015-2030

In incidentele gevallen kan het in prognoseberekeningen en langs de landgrenzen voorkomen dat GCN geen geldige concentratiewaarden kan leveren. Dit wordt teruggemeld door GCN en PreSRM via een exceptie. De volgende Delphi code is een voorbeeld om vóór de uitvoer van een berekening te testen of geldige GCN waarden voor het rekendomein gegenereerd zijn:

```

PS_HandleExceptions(false);

// Init conc...
conc := nil;
try
    conc:=PS_Conc_Init(proj,usecomp,xx,yy);
except
    ps_Cleanup(dummy,dummy,conc,dummy);
    conc:= nil;
end;
if ps_errorcode<>0 then
    conc:= nil;

if conc <> nil then
begin
    try
        // Hier uw berekeningen
        .....

    finally
        ps_Cleanup(dummy,dummy,conc,dummy);
        conc:=nil;
    end;
end
else //if conc1<>nil
begin
    MessageBox('Geen concentraties!');
end;

```

Het PS_ClassData object

Het object **PS_ClassData** levert ten behoeve van SRM2 implementaties voor een op te geven locatie de windrozen voor meteorologie en concentraties. Daarom moet aan dit object bij initialisatie doorgegeven worden voor welke **PS_Proj**, **PS_Meteo** en **PS_Conc** objecten de resultaten gegenereerd moeten worden. Bij initialisatie kan separaat bepaald worden hoeveel segmenten voor concentratie (default 36) en meteorologie (default 12) moeten worden toegepast. Het punt waarvoor een windroos teruggegeven moet worden wordt eerst met **PS_Class_SetCoord** aangegeven. Vervolgens kan met de functies **PS_Class_Meteo** en **PS_Class_Conc** de windroosinformatie worden opgevraagd per windrichtingssector. In de versie 2.0 die half mei 2010 geleverd zal worden is tevens voorzien in de berekening van effectieve jaargemiddelde droge depositiesnelheden voor alle relevante componenten voor een locatie, op basis van berekeningen op uurbasis en een hoge resolutie landgebruikskaart. Voor de berekeningen wordt gebruik gemaakt van de recent vernieuwde DEPAC module van RIVM.

Beschrijving van de datatypen en functies

type

```

TComponent = (SO2,NO2,O3,PM10,CO,C6H6,Pb,NH3,PM25,NOx,BaP,Roet);
CompSet = SET OF TComponent;

```

TComponent is de definitie van de stoffen waarvoor in deze versie van PreSRM concentraties kunnen worden opgevraagd. Bij initialisatie van **PS_Conc** objecten wordt deze met een parameter van type **TCompSet** ingesteld.

```

tCompArray = Array [tcomponent] of double;

```

Array type waarmee functie **PS_DryDepEffective** de resultaten van de berekening per stof teruggeeft.

```

TMeteoRec = record
    Time      : TDateTime;      // Datetime in days since 1-1-1900
    OWS,WD,    // Original windspeed [m/s], Winddir [deg] at z=10m
    TT,RH,     // Temperature [Celcius], Relative Humidity [%]
    NN,GR,     // Clouds [octecs], Shortw. incoming radiation [W/m2]
    PP,        // Precipitation [mm/hr]
    WS60,     // Windspeed [m/s] at blending height z=60m
    WS,        // Interpolated windspeed [m/s] at z=10m
    Sv,        // Std dev of cross wind

    Monin,     // Monin Obukhov length [m]
    Sens,     // Sensible heatflux [W/m2]
    Ustar,    // Friction Velocity [m/s]
    K1,K3,    // rate constants for O3/NOx [s-1]
    Tl,       // []
    Sw,       // Std dev vertical wind [m/s]
    MixH      : double;        // PBL height [m]
    Status    : byte;
end;

```

Record met meteorologische gegevens voor een rekenpunt. De variabelen vanaf K1 zijn alleen relevant voor SRM3 berekeningen.

Function PS_Version:double;stdcall;

Geeft het versienummer van de gebruikte PreSRM bibliotheek terug. Het hoofdversienummer staat in het gedeelte voor de decimale punt, het subversienummer staat in het gedeelte achter de decimale punt.

procedure PS_Cleanup(var proj,meteo,conc,classdata:pointer);stdcall;

Opruimfunctie voor alle objecten van PreSRM. Niet alle objecten hoeven geïnitieerd te zijn. Dit kan door een dummy pointer met waarde nil door te geven. Bijvoorbeeld:

```

Var
    Dum : pointer
Begin
    Dum:=nil;
    PS_Cleanup(proj,dum,dum,dum);

```

Function PS_Proj_Init(year:Integer;
X,Y,WX,WY:Integer; // RD in meters
Path:pAnsiChar;
Prognose:Boolean):Pointer;stdcall;

Initialisatie van een **PS_Proj** object.

Parameters:

Year	rekenjaar (2005-2020,2030)
X,Y	midden coördinaat van het domein in Amersfoortse coördinaten [meter]
WX,WY	breedte en Hoogte van het domein [meter]
Path	basis map (directory) met de PreSRM gegevens (inclusief trailing slash)
Prognose	True bij gebruik van 10-jarige meteo (Verplicht voor jaren 2010-2030)
Result	pointer naar nieuwe PS_Proj object

Procedure PS_nnmtime(time:tDateTime;
Year,Month,Day,Hour:word);stdcall;

Ontbinding van een standaard variabele van type **tDateTime** (zoals gebruikt in **tMeteoRec**) in waarden voor kalenderjaar, -maand en dag en uur van de dag. Bijzonder aan deze ontbinding is dat de uren lopen van 1 tot en met 24, in plaats van de standaard 0 tot en met 23. Als de variabele **Hour** de waarde 24 heeft, dan is de datum in **Year, Month, Day** die van de vorige dag. Dus bijvoorbeeld een **tDateTime** die normaal gesproken de tijd 23 december 2009 0:00 aanduidt levert volgens deze functie 22 December 24:00 op. Deze functie is toegevoegd voor compatibiliteit met de wijze waarop SRM3 modellen met tijd rekenen. In andere gevallen wordt aanbevolen met de tijd ontbindingsfuncties uit de unit **SysUtils** of **DateUtils** te rekenen (zoals **decodedate**, **decodetime**, **hourof** etc).

Parameters:

Time	Te ontbinden variabele voor de tijd
Year	Kalenderjaar
Month	Kalendermaand
Day	Dag van de maand
Hour	Uur van de dag. Let op: krijgt waarden van 1 tot en met 24

```
function PS_SeaSaltCorrection(proj:pointer;
                             x,y:Integer;
                             var corrconc,
                             corrdays : integer):boolean;stdcall;
```

Opvragen van een suggestie voor de te hanteren zeezoutcorrectie in dagen voor PM₁₀ voor een gridpunt. De waarden zijn gebaseerd op een naar een 1 bij 1 km grid geconverteerde gemeente- en provinciekaart voor Nederland met als attribuut de via de RBL voorgeschreven zeezoutcorrectie in µg/m³ c.q. dagen overschrijding van PM₁₀>50 µg/m³. Let wel, officieel geldt dat de enig correcte waarde die is voor de betreffende gemeente c.q. provincie waar het punt toe behoort. Door de vergridding naar 1 bij 1 km kan het voorkomen dat een punt een waarde voor de zeezoutcorrectie toegewezen krijgt van een naburig punt

Parameters:

Proj	pointer naar PS_Proj object
X,Y	coördinaat van het gridpunt in Amersfoortse coördinaten [meter]
Corrconc	Suggestie voor de waarde van de zeezout correctie [µg/m ³], -999 indien geen geldige waarde gevonden (bijvoorbeeld boven water of buiten Nederland)
Corrdays	Suggestie voor de waarde van de zeezout correctie [dagen >50], -999 indien geen geldige waarde gevonden (bijvoorbeeld boven water of buiten Nederland)
Result	Geeft met false aan dat voor tenminste een van beide waarden geen geldige waarde gevonden kon worden

```
function PS_Roughness(proj:pointer;
                     x,y:Integer;
                     var z0class:integer;
                     const Oro : boolean = false
                     ):double;stdcall;
```

Opvragen van de oppervlakteruwheid voor een gridpunt

Parameters:

Proj	pointer naar PS_Proj object
X,Y	coördinaat van het gridpunt in Amersfoortse coördinaten [meter]
Z0class	resultaat: SRM2 ruwheidsklasse (0=0.03m 1=0.1m 2=0.3m 3=1.0m)
Oro	TRUE: pas extra ruwheid door hoogteverschillen toe; FALSE: geen extra

Result ruwheid door hoogteverschillen
 ruwheidsklasse [meter]

```
function PS_Roughness_Area(proj:pointer;  
                           x,y,wx,wy:Integer;  
                           UseLogAveraging:boolean;  
                           var z0class:integer;  
                           const Oro : boolean = false):double;stdcall;
```

Opvragen van de oppervlakteruwheid representatief voor een groter oppervlak.

Parameters:

Proj	pointer naar PS_Proj object
X,Y	midden coördinaat van het gebied in Amersfoortse coördinaten [meter]
WX,WY	breedte en hoogte van het opvraagpunt in Amersfoortse coördinaten [meter]
UseLogAveraging	TRUE: gebruik logaritmische middeling; FALSE: normale lineaire middeling
Z0class	resultaat: SRM2 ruwheidsklasse (0=0.03m 1=0.1m 2=0.3m 3=1.0m)
Oro	TRUE: pas extra ruwheid door hoogteverschillen toe; FALSE: geen extra ruwheid door hoogteverschillen
Result	ruwheidsklasse [meter]

```
function PS_Meteo_Init(proj:pointer;  
                      UseStability:boolean;  
                      UserRoughness,  
                      Ustar_limit,  
                      MoninObu_limit :Double;  
                      MeteoX,MeteoY:Integer;  
                      const MultiStation:boolean=false):Pointer;stdcall;
```

Initialisatie van een **PS_Meteo** object.

Parameters:

Proj	pointer naar PS_Proj object
UseStability	False: veronderstel neutrale atmosfeer bij schaling wind naar 60 meter True: verreken lokale en huidige stabiliteit bij opschaling
UserRoughness	te gebruiken oppervlakteruwheid bij downscalen, SRM2 default =0.08 [meter]
MeteoX,MeteoY	coördinaat van het meteo gridpunt in Amersfoortse coördinaten [meter]
Result	pointer naar nieuwe PS_Meteo object
UStar_Limit	Ondergrens voor wrijvingsnelheid U* [m/s] , indien <0 wordt de waarde niet gebruikt. Indien groter dan nul worden berekende waarden voor U* kleiner dan deze grens afgekapt naar de limietwaarde en de moninobuhkov lengte daarop aangepast
MoninObu_Limit	Bovengrens voor de Monin-Obukhov lengte L [m]. Indien <0 wordt de waarde niet gebruikt. Indien groter dan nul worden berekende waarden voor L kleiner dan deze grens afgekapt naar de limietwaarde en de U* daarop aangepast
MultiStation	Indien TRUE wordt de meteorologie berekend op basis van de beschikbare gegevens van 17 meetstations. Indien FALSE wordt de RBL standaard toegepast waarbij geïnterpoleerd wordt tussen Eindhoven en Schiphol.

```
function PS_MeteoGetData(meteo:pointer;nr:integer;  
                        var data:tmeteorec):boolean;stdcall;
```


Ophalen van een meteorecord met een rangnummer. Het aantal records kan opgevraagd worden met **PS_MeteoNrRecords**. Bij terugkeer levert deze functie een record met alle benodigde uurlijkse waarden voor de meteo

Parameters:

Meteo	pointer naar een PS_Meteo object
Nr	volgnummer van het meteorecord (0..PS_MeteoNrRecords-1)
Data	resultaat: record met de gewenste meteogegevens
Result	True indien succesvol

```
function PS_MeteoGetByTime(meteo:pointer;Time:tDateTime;var
                           data:tmeteorec):boolean;stdcall;
```

Ophalen van een meteorecord voor een bepaalde tijd . Bij terugkeer levert deze functie een record met alle benodigde uurlijkse waarden voor de meteo. Als de opgevraagde tijd niet beschikbaar is levert de functie de waarde false

Parameters:

Meteo	pointer naar een PS_Meteo object
Time	Tijd waarvoor de meteo opgezocht moet worden
Data	resultaat: record met de gewenste meteogegevens
Result	True indien succesvol

```
function PS_MeteoNrRecords(meteo:pointer):integer;stdcall;
```

Geeft het aantal records (uren) in de meteodata voor het rekenjaar weer. In prognose modus is dit aantal ruim 87600, in andere jaren is dit 8760 of 8784.

Parameters:

Meteo	pointer naar een PS_Meteo object
Result	aantal records (uren) in de meteodata

```
procedure PS_GetMeteoLongPeriod
(RDX,RDY,YearStart,YearEnd:Integer;
 z0,
 Ustar_limit,
 MoninObu_limit:double;
 DataDir:pChar;
 var data: tMeteoArray
 const MultiStation:boolean=false):Pointer;stdcall;
```

Retourneert de uurlijkse meteorologische gegevens voor een meerjarige periode in het interval 1995 tot en met 2009. Bijzonder aan deze procedure is dat er geen project of meteo objecten voor te hoeven geïnitieerd en doorgegeven. Deze objecten worden intern aangemaakt. De array met de meteogegevens **data** is van het type **tMeteoArray** zoals gedefinieerd in de unit **modeldefs.pas**. De data array moet door de aanroepende applicatie gealloceerd zijn met de juiste lengte. Deze lengte kan als volgt berekend worden:

```
uses SysUtils, DateUtils, modeldefs, pre_srm;

var
  StartYear,
  EndYear,
```

```

    datalen : integer;
    data : tMeteoRecArray;
begin
    StartYear:=1997;
    EndYear:=2007;
    datalen:=round(encodedate(endyear+1,1,1)-
encodedate(startyear,1,1))*24;
    setlength(data,datalen);
    GetMeteoLongPeriod(180500,400500,StartYear,EndYear,-1,-1,-
1, '.\data\' ,data);

```

Parameters:

RDX, RDY	De rijksdriehoekcoördinaten [meter] van het punt waarvoor de meteorologische gegevens moeten worden berekend
YearStart	Beginjaar van het tijdsinterval waarvoor gegevens opgevraagd worden (1995-2009)
YearEnd	Eindjaar van het tijdsinterval waarvoor gegevens opgevraagd worden (1995-2009)
z0	Te gebruiken ruwheidslengte bij de interpolatie, indien de opgegeven z0 kleiner dan nul is, wordt door de routine de ruwheidslengte van het opgegeven punt RDX, RDY opgezocht en gebruikt.
UStar_Limit	Ondergrens voor wrijvingssnelheid U^* [m/s] , indien <0 wordt de waarde niet gebruikt. Indien groter dan nul worden berekende waarden voor U^* kleiner dan deze grens afgekapt naar de limietwaarde en de moninobuhkov lengte daarop aangepast
MoninObu_Limit	Bovengrens voor de Monin-Obukhov lengte L [m]. Indien <0 wordt de waarde niet gebruikt. Indien groter dan nul worden berekende waarden voor L groter dan deze grens afgekapt naar de limietwaarde en de U^* daarop aangepast
DataDir	basis map (directory) met de PreSRM gegevens (inclusief trailing slash)
Data	Dynamische array te vullen met de meteorologische gegevens, moet voor de aanroep gealloceerd zijn door de aanroepende applicatie
MultiStation	Indien TRUE wordt de meteorologie berekend op basis van de beschikbare gegevens van 17 meetstations. Indien FALSE wordt de RBL standaard toegepast waarbij geïnterpoleerd wordt tussen Eindhoven en Schiphol.

```

function PS_Conc_Init(proj:pointer;
                      comp:compset
                      ConcX,ConcY:Integer):Pointer;stdcall;

```

Initialisatie van een **PS_Conc** object. Door aan het coördinatenpaar **ConcX, ConcY** de waarde -1,-1 door te geven wordt het projectmidden van het doorgegeven project (**proj**) gebruikt om de uurlijkse waarden uit GCN op te vragen.

Parameters:

Proj	pointer naar PS_Proj object
Comp	set van in het PS_Conc object weer te geven componenten
ConcX	RD X coördinaat van het opvraagpunt voor uurlijkse waarden [m]
ConcY	RD Y coördinaat van het opvraagpunt voor uurlijkse waarden [m]
Result	pointer naar het nieuwe resulterende PS_Conc object

```

function PS_MeanConc(conc:pointer;
                    c:tcomponent;
                    x,y:integer;

```

```
var dbc:single):single;stdcall;
```

Levert de jaargemiddelde concentratie voor een component op een bepaalde locatie. De resulterende waarde is exclusief de dubbeltellingscorrectie voor de invloed van het HWN. In het geval dat deze correctie toegepast moet worden dient de waarde van deze correctie gegeven door variabele dbc van het resultaat te worden afgetrokken. In de buurt van de luchthaven Schiphol wordt de bijdrage van deze luchthaven op een resolutie van 100 meter in de achtergrondwaarde meegenomen.

Parameters:

Conc	pointer naar PS_Conc object
C	de component waarvoor de waarde gegeven moet worden
X,Y	de coördinaat van het punt waarvoor de waarde gegeven moet worden
Dbc	waarde voor de dubbeltellingscorrectie voor de bijdrage van HWN
Result	resultaat voor de jaargemiddelde concentratie (inclusief HWN invloed)

```
procedure PS_ConcGetData(conc:pointer;  
                          c:tcomponent;  
                          x,y:integer;  
                          var time:tTimArray;  
                          var data:tSingleArray;  
                          var dbc:Single);stdcall;
```

Levert de uurgemiddelde waarden voor de concentratie van de gewenste component op de gewenste plek exclusief de dubbeltellingscorrectie voor invloed van HWN. De toe te passen waarde voor de HWN dubbeltellingscorrectie wordt teruggegeven in de parameter dbc. Het resultaat zijn twee gevulde dynamische arrays met de tijd in tDateTime formaat en de concentraties in $\mu\text{g}/\text{m}^3$. De initialisatie op de juiste lengte dient door de aanroepende applicatie verzorgd te worden. De benodigde lengte is op te vragen via de **PS_MeteoNrRecords** call uit een **PS_Meteo** object voor hetzelfde **PS_Project** object.

Parameters:

Conc	pointer naar PS_Conc object
C	de component waarvoor de waarde gegeven moet worden
X,Y	de coördinaat van het punt waarvoor de waarden gegeven moet worden
time	dynamische array met de uurlijkse tijde
data	dynamische array met de uurlijkse concentratiewaarden (inclusief HWN invloed)
dbc	waarde voor de dubbeltellingscorrectie voor de invloed van het HWN

```
procedure PS_ConcGetDataByYear(conc:pointer;  
                               c:tcomponent;  
                               x,y,Year:integer;  
                               var time:tTimArray;  
                               var data:tSingleArray;  
                               var nr:Integer;  
                               var dbc:Single);stdcall;
```

Levert de uurgemiddelde waarden voor de concentratie van de gewenste component op de gewenste plek exclusief de dubbeltellingscorrectie voor invloed van HWN. De toe te passen waarde voor de HWN dubbeltellingscorrectie wordt teruggegeven in de parameter dbc. Het resultaat zijn twee gevulde dynamische arrays met de tijd in tDateTime formaat en de concentraties in $\mu\text{g}/\text{m}^3$. De initialisatie op de juiste lengte dient door de aanroepende applicatie verzorgd te worden. De benodigde lengte is op te vragen via de **PS_MeteoNrRecords** call uit een **PS_Meteo** object voor hetzelfde **PS_Project** object.

Parameters:

Conc	pointer naar PS_Conc object
C	de component waarvoor de waarde gegeven moet worden
X,Y	de coördinaat van het punt waarvoor de waardes gegeven moet worden
Year	Het jaar waarvoor de concentratiegegevens moeten worden opgehaald, altijd gelijk aan het projectjaar in een verleden jaar, in geval van een prognoseberekening een jaar van 1995-2004.
time	dynamische array met de uurlijkse tijden
data	dynamische array met de uurlijkse concentratiewaarden (inclusief HWN invloed)
Nr	Aantal waarden in de data en time array (8760 of 8784)
dbc	waarde voor de dubbeltellingscorrectie voor de invloed van het HWN

```
function PS_ClassData_Init(proj,meteo,conc:pointer;  
                           nrsecwind:integer;  
                           nrsecconc:integer;  
                           dbc:boolean):Pointer;stdcall;
```

Initialisatie van een **PS_ClassData** object

Parameters:

proj	pointer naar PS_Project object
meteo	pointer naar PS_Meteo object
conc	pointer naar PS_Conc object
nrsecwind	aantal windroos sectoren voor de meteorologie (12)
nrsecconc	aantal windroos sectoren voor de concentraties (36)
bbc	true: pas dubbeltellingscorrectie voor HWN toe
Result	pointer naar het resulterende PS_Classdata object

```
procedure PS_Class_SetCoord(ClassData:Pointer;X,Y: Integer);stdcall;
```

Instellen van de huidige coördinaten van het **PS_Classdata** object

Parameters:

ClassData	pointer naar het PS_Classdata object
X,Y	de coördinaat van het punt waarvoor de waardes gegeven moet gaan worden

```
function PS_Class_Meteo(ClassData : Pointer;  
                        USec,  
                        UClass : Integer;  
                        Var Weight: Double):boolean;stdcall;
```

Geeft de waarden van de fractie van voorkomen van de windsnelheidsklasse UClass (0..2) in windsector USec (0..NrSecWind-1). Windroos is opgebouwd uit segmenten van hoek $360/\text{NrSecWind}$. Sector 0 loopt van $360\text{-hoek}/2\text{..hoek}/2$ graden, sector 1 loopt van $\text{hoek}/2\text{..}3*\text{hoek}/2$ (bv 355-5, resp 5..15 graden bij NrSecWind=36, dus 10 graden sectoren). Windsnelheidsklasse 0 is van $u < 2.75$ m/s, klasse 1 is van $u = 2.75\text{-}4.75$ m/s en klasse 2 is $u > 4.75$ m/s.

Parameters:

ClassData	pointer naar het PS_Classdata object
UClass	windsnelheidsklasse (0..2)
Usec	windsector (0..NrSecWind-1).
Weight	fractie van voorkomen van de gevraagde klasse van windsnelheid en –richting

Result True indien geslaagd

```
Function PS_Class_Conc(ClassData  : Pointer;  
                        Sec          : Integer;  
                        Comp         : TComponent;  
                        Var Conc    : Double;  
                        Var Weight : Double):boolean;stdcall;
```

Geeft de waarden van de fractie van voorkomen in windsector USec (0..NrSecWind-1) en de bijbehorende concentraties van de gevraagde component.

Parameters:

ClassData	pointer naar het PS_Classdata object
Usec	windsector (0..NrSecWind-1).
comp	de component waarvoor de waarde gegeven moet worden
Conc	concentratie van de gewenste component in de windsector
Weight	fractie van voorkomen van de gevraagde klasse van windsector
Result	True indien geslaagd

```
procedure PS_DryDepEffective(Proj,Meteo,Conc:pointer;  
                             c          : CompSet;  
                             x,y       : Integer;  
                             var Result : TCompArray);stdcall;
```

TODO

Parameters:

Nieuw in PreSRM V1.1

```
procedure PS_HandleExceptions(ByDLL:boolean);stdcall;
```

PreSRM maakt gebruik van de MadExcept exception handler om runtime fouten op te sporen en de gegevens daarvan weer te geven. Standaard vangt de DLL zelf de fouten af in MadExcept en presenteert de gebruiker in dat geval een Dialoog met informatie over de fout. De melding kan opgeslagen worden op disk, ge-emaild naar de ontwikkelaar of genegeerd. De gebruiker kan daarna het programma beëindigen of proberen voort te zetten. Deze afvangst van exceptions door PreSRM zelf kan ook uitgeschakeld worden, waarna het aanroepende programma zelf eventuele excepties in PreSRM moet afhandelen. Naast de gebruikelijke Delphi excepties genereert PreSRM de volgende eigen excepties: EPreSRMError en EGCNError. PreSRM is standaard gecompileerd met Range en Stack checking opties aan.

Parameters:

ByDLL	Logische waarde die aangeeft of PreSRM de excepties zelf moet afvangen (TRUE) of dat de aanroepende applicatie dat elf doet (FALSE). Standaard is TRUE.
--------------	---

```
function PS_ErrorCode:Integer;stdcall;
```

In het geval dat de eigen EPreSRM exceptie handler niet gebruikt wordt een foutcode aangemaakt. Via deze functie kan deze foutcode worden opgevraagd. Normaal geeft deze functie de waarde nul terug.

```
procedure PS_MeteoSetMinimum Windspeed(MinimumWS:Double);stdcall;
```

Door het aanroepen van deze procedure kan het aanroepende programma de waarde van de minimale windsnelheid aanpassen. Standaard staat deze ingesteld op 0.5 m/s. Alle windsnelheden beneden deze waarde worden op deze minimum waarde ingesteld. Deze instelling werkt globaal voor alle projecten aangemaakt na het wijzigen van deze waarden.

Parameters:

MinimumWS Waarde voor de minimum windsnelheid (0.5-10 m/s)

Voorbeeldprogramma (console mode)

```
program PreSRM2txt;

{$APPTYPE CONSOLE}

uses
  SysUtils, modeldefs, Pre_SRM;

var
  t,
  tnr,
  notused,
  Year,
  xx,yy      : integer;
  z0         : double;
  proj,
  conc,
  meteo,
  dummy      : Pointer;
  data       : tMeteoRec;
  dbc_no2    : Single;
  dbc_pm10   : Single;
  no2data,
  pm10data   : tSingleArray;
  time       : tTimArray;
  f          : System.text;
  datadir    : string;
  usecomp    : CompSet;
begin
  writeln('PreSRM Demo program');
  writeln('PreSRM version: ',ps_version:8:3);
  writeln;

  xx:=140000;
  yy:=460000;
  Year:=2008;
  datadir:='.\data\';
  usecomp:=[NO2,PM10];

  Writeln('Init project...');
  proj:=PS_Proj_Init(Year,
                    xx,yy,20000,20000,
                    datadir,Year>2009);
  Writeln('Init conc...');
  conc:=PS_Conc_Init(proj,usecomp,-1,-1);

  z0:=PS_Roughness(proj,xx,yy,notused);
  Writeln('Init meteo...');
  meteo:=PS_meteo_Init(proj,true,z0,xx,yy);

  tnr:=ps_meteoNrRecords(meteo);
  setlength(time,tnr);
  setlength(pm10data,tnr);
  setlength(no2data,tnr);
```

```

Writeln('Getting hourly concentrations...');
ps_concgetdata(conc,PM10,xx,yy,time,pml0data,dbc_pm10);
ps_concgetdata(conc,NO2,xx,yy,time,no2data,dbc_no2);
// repeat ps_concgetdata for other components (O3 etc)...

Writeln('Writing data to txt file...');
assignfile(f,'PreSRMout.txt');
rewrite(f);
writeln(f,'Datum/Tijd      WindSnel WindRich  Wind60 Temperat
Uster MoninObukhov      NO2');
for t:=0 to tnr-1 do
begin
  ps_meteogetdata(meteo,t,data);

  writeln(f,formatdatetime('DD-MM-YYYY HH:MM',data.time),' ',
        data.ws:8:2,' ',
        data.wd:8:1,' ',
        data.ws60:8:1,' ',
        data.TT:8:1,' ',
        data.Ustar:8:2,' ',
        data.Monin:12,' ',
        no2data[t]:8:2
        );
end;
closefile(f);

finalize(time);
finalize(pml0data);
finalize(no2data);

dummy:=nil;
ps_Cleanup(proj,meteo,conc,dummy);
end.

```