

RELEASENOTES SGWM

Aan: Gebruikers SGWM
Van: HKV
Datum: 20 december 2023
Projectnummer: PR3597
Onderwerp: Releasenotes

What's New in Version 2.1.0

What's New

In deze versie zijn de volgende aanpassingen/verbeteringen doorgevoerd:

- Introductie van een *optionele* uitbreiding van de definitie van de somID, parameters, variabelen en calculationsettings met een indicator voor het aantal karakters dat gebruikt wordt bij het wegschrijven van een somID of een SGWM key in foldernamen en template bestanden.

Als in de somID gebruik wordt gemaakt van een parameter waarvan het aantal karakters van de verschillende waarden van deze parameter fluctueert, is het soms wenselijk om deze aan te vullen met zogenoemde voorloophnullen. Dit resulteert dan in een somID die altijd dezelfde lengte heeft. Stel je maakt gebruik van een parameter Q met de waarden 10 (2 karakters lang) en 250 (3 karakters lang) en een somID gelijk aan 'Q{Q}'. Dit resulteert dan in de sommen 'Q10' en 'Q250', respectievelijk 3 en 4 karakters lang. Het is vanaf nu mogelijk om de somID te definiëren met een notatie van de lengte (het aantal te gebruiken karakters), bijvoorbeeld 'Q{Q:3}'. De toevoeging ':3' geeft aan dat er nu altijd 3 karakters worden gebruikt bij het invullen van de parameters in de somID en als er minder karakters beschikbaar zijn, dan wordt gebruik gemaakt van voorloophnullen om de lege plekken op te vullen. Deze definitie resulteert dan in de sommen 'Q010' en 'Q250', waarbij de '0' voor '10' in de eerste somID een voorloophnul betreft omdat het aantal karakters van deze waarde (2) kleiner is dan wat in de somID is gedefinieerd (3).

Deze definitie kan ook gebruikt worden in een SGWM Key bij het gebruik van parameters, variabelen en calculationsettings in de template bestanden. Elke SGWM key kan dan *optioneel* aangevuld worden met een definitie van het aantal te gebruiken karakters. Dit op dezelfde manier door de toevoeging van een dubbelepunt en een aantal karakters in de definitie. Binnen één template is het toegestaan om met verschillende notaties te werken dus {Q:3} en {Q:4} kunnen beide in één template gebruikt worden.

Let op: bij het gebruik van waarden van het type *float* kan het zijn dat deze met een specifiek aantal decimalen is gedefinieerd. Het decimaalscheidingsteken inclusief de decimalen telt in dat geval mee voor het aantal karakters! Bijvoorbeeld een variabele 'h' met de waarden '1.23' en '2.44' kan als volgt gebruikt worden in een template '{h:6}'. Dit resulteert dan in de waarden '001.23' en '002.44' (twee voorloophnullen)!

Voor alle typen geldt dat wanneer het aantal karakters kleiner wordt gedefinieerd dan de waarde zoals deze gedefinieerd is in het project-definitie-bestanden of het module-definitie-bestand, altijd de waarde wordt gebruikt zoals gedefinieerd in het project-definitie-bestanden of het module-definitie-bestand. Bijvoorbeeld een variabele 'h' met de waarden '1.2' en '2.44' kan als volgt gebruikt worden in een template '{h:2}'. Dit resulteert dan in de waarden '1.2' en '2.44', waarbij in het laatste geval alsnog 3 karakters wordt toegepast ondanks de definitie ':2' omdat de waarde van de variabele zelf 3 karakters bevat!

- De definitie van een variabele is eenvoudiger gemaakt, zodat deze in lijn is met de definitie van een parameter in het project-definitie-bestand. Dit houdt dat in plaats van een zeer uitgebreid geneste definitie van <ParameterValues><ParameterValue><Parameters><Parameter> is overgestapt naar een eenvoudiger en minder geneste definitie <Values><Value>, zoals deze ook gebruikt wordt in de definitie van een parameter. Om dit mogelijk te maken hebben we vervolgens ook een aanpassing moeten doorvoeren in de definitie van de voorwaarden van een variabele (zie volgende bullet). Hieronder is een fictieve definitie van een variabele 'Wind' afhankelijk van een parameter 'Afvoer' weergegeven volgens de definitie van *voor deze aanpassing*.

```

<Computation>
  <Folder>computations</Folder>
  <Id>Q{Q:3}</Id>
  <Parameters>
    <Parameter>
      <Name>Afvoer</Name>
      <Key>Q</Key>
      <Type>float</Type>
      <Values>
        <Value>10</Value>
        <Value>250</Value>
      </Values>
    </Parameter>
  </Parameters>
  <Variables>
    <Variable>
      <Name>Wind</Name>
      <Key>U</Key>
      <Type>float</Type>
      <ParameterValues>
        <ParameterValue>
          <Parameters>
            <Parameter>
              <Key>Q</Key>
              <Value>10</Value>
              <Operator>EQ</Operator>
            </Parameter>
          </Parameters>
          <Value>2</Value>
        </ParameterValue>
        <ParameterValue>
          <Parameters>
            <Parameter>
              <Key>U</Key>
              <Value>250</Value>
              <Operator>EQ</Operator>
            </Parameter>
          </Parameters>
          <Value>4</Value>
        </ParameterValue>
      </ParameterValues>
    </Variable>
  </Variables>
</Computation>

```

Merk op dat de definitie van de variabele resulteert in een uitgebreide en complexe definitie.

Na de aanpassing in versie 2.1.0 reduceert deze zelfde definitie tot (een winst van 18 regels!):

```
<Computation>
  <Folder>computations</Folder>
  <Id>Q{Q:3}</Id>
  <Parameters>
    <Parameter>
      <Name>Afvoer</Name>
      <Key>Q</Key>
      <Type>float</Type>
      <Values>
        <Value>10</Value>
        <Value>250</Value>
      </Values>
    </Parameter>
  </Parameters>
  <Variables>
    <Variable>
      <Name>Wind</Name>
      <Key>U</Key>
      <Type>float</Type>
      <Values>
        <Value>Q == 10 ; 2</Value>
        <Value>Q == 250 ; 4</Value>
      </Values>
    </Variable>
  </Variables>
</Computation>
```

Merk op dat de structuur van de definitie van een variabele nu identiek is aan die van een parameter.

- In combinatie met voorgaande aanpassing is ook de definitie van de voorwaarden van een variabele aangepast. Behalve dat ook deze aanpassing bijdraagt aan een beknoptere definitie, maakt deze ook complexere definities mogelijk. Hier toe zijn zogenoemde functievoorschriften geïntroduceerd, bestaande uit een voorwaarde, gevolgd door één van onderstaande operators:
 - **==** gelijk aan (was EQ)
 - **<** kleiner dan (was LT)
 - **>** groter dan (was GT)
 - **<=** kleiner dan of gelijk aan (was LE)
 - **>=** groter dan of gelijk aan (was GE)
 , de parameterwaarde en de waarde van de variabele, gescheiden door een puntkomma teken.

Dit resulteert dan in de volgende definitie:

functievoorschrift met operator en parameterwaarde ; waarde variabele

Tevens kan gebruik gemaakt worden van combinaties van voorwaarden door de toevoeging van 'and', 'or' of 'not' in het voorschrift. Hieronder een aantal voorbeelden:

```
<Variables>
  <Variable>
    <Name>Wind</Name>
    <Key>U</Key>
    <Type>float</Type>
    <Values>
      <Value>Q == 10 ; 2</Value>
      <Value>Q == 250 ; 4</Value>
    </Values>
  </Variable>
</Variables>
```

Wanneer de parameter Q gelijk is aan waarde 10, dan krijgt de variabele U de waarde 2.
Wanneer de parameter Q gelijk is aan waarde 250, dan krijgt de variabele U de waarde 4.

Stel dat er twee parameters worden gedefinieerd:

- 'Q' met waarden 10 en 250 (float)
- 'K' (van kering) met waarden 'open' en 'dicht' (string)

Dan zijn onderstaande complexere definities mogelijk:

```
<Variable>
  <Name>Wind</Name>
  <Key>U</Key>
  <Type>float</Type>
  <Values>
    <Value>Q &lt;= 100 and K == 'open' ; 1</Value>
    <Value>Q &gt; 100 and K == 'open' ; 2</Value>
    <Value>Q &lt;= 100 and K == 'dicht' ; 3</Value>
    <Value>Q &gt; 100 and K == 'dicht' ; 4</Value>
  </Values>
</Variable>
```

Als Q kleiner dan of gelijk is aan 100 en de kering staat open dan krijgt U een waarde 1.

Als Q groter is dan 100 en de kering staat open dan krijgt U een waarde 2.

Als Q kleiner dan of gelijk is aan 100 en de kering is dicht dan krijgt U een waarde 3.

Als Q groter is dan 100 en de kering is dicht dan krijgt U een waarde 4.

Let op: waarden die gebruikt worden in het functievoorschrift en van het type string zijn, moeten altijd tussen enkel quotes opgenomen worden. De waarde van een variabele (als deze van het type string is) zelf (na de puntkomma dus) niet!

- Omdat bovengenoemde twee aanpassingen (in de definitie van een variabele) redelijk ingrijpend zijn, hebben wij in versie 2.1.0 een knop toegevoegd aan het menu 'Help', waarmee project-definitie-bestanden van voor versie 2.1.0 automatisch omgezet kunnen worden naar een versie die voldoet aan de voorwaarden van versie 2.1.0.

What's New in Version 2.0.1

What's New

In deze versie zijn twee minor bugs opgelost:

- Bij het vullen van de template bestanden wordt gecontroleerd of relatieve pad-verwijzingen naar andere bestanden binnen de modelschematisatie, correct zijn. Gebleken is dat deze controle in versie 2.0.0 soms tot 'false positives' kan leiden. In de log verschijnt dan de melding dat een bestand niet bestaat, terwijl dit wel het geval is. Deze bug is opgelost in versie 2.0.1.
- Het bleek niet mogelijk te zijn om een filter in de tabellen toe te passen op de kolom met som status. Dit is aangepast. Let wel: komt in een kolom slechts 1 waarde voor, dan is het filter niet beschikbaar.

What's New in Version 2.0.0

What's New

'SGWM' en 'SGWM voor MHWp5' (een specifieke versie van SGWM die ontwikkeld is voor de Maatgevend HoogWater processor versie 5) zijn geïntegreerd tot één versie, vanaf nu aangeduid met SGWM. Als gevolg van deze integratie zijn de volgende wijzigingen doorgevoerd:

- Splitsing van het project-definitie-bestand van SGWM in een project-definitie-bestand en een module-definitie-bestand. Het project-definitie-bestand bevat alle informatie die niet specifiek is voor de rekenmodule (rekenmodel/software) en het module-definitie-bestand bevat juist wel alle rekenmodule specifieke informatie. Hierbij zijn de volgende onderdelen uit het project-definitie-bestand en een module-definitie-bestand gewijzigd:
 - De blokken 'Templates', 'Input', 'States' en 'CalculationSettings' zijn verwijderd uit het project-definitie-bestand.
 - Het (voor SGWM nieuwe) module-definitie-bestand begint (en wordt afgesloten) met een tag 'MDB'. Daarbinnen zijn opgenomen tags voor de 'Name', 'Computation' (met een blok 'Templates', 'Input' en 'States'), en 'CalculationSettings'.
 - Tussen de tags 'Folder' in het blok 'Computation/Input' in het module-definitie-bestand, wordt vanaf deze versie niet meer de naam van de folder gedefinieerd waarin de invoerbestanden van een som worden weggeschreven door SGWM. De naam van deze folders is vanaf deze versie altijd identiek aan de somID zoals gedefinieerd in het project-definitie-bestand (zie verderop). Wel kan tussen deze tags een subfolder worden gedefinieerd die aangemaakt wordt binnen map van de somID. Let op, meerdere subfolders is ook mogelijk (gescheiden door een '/', maar begin de eerste subfolder nooit met een '/'. Dit mag dus 'input/bestanden', maar dit niet '/input/bestanden'.
 - Bij elke definitie van een 'State' in het blok 'Computations/States' in het module-definitie-bestand, is een extra verplichte tag nodig, namelijk 'Progress'. De waarde hiervan is true of false. De eerst 'State'-definitie met een waarde voor 'Progress' gelijk aan true, wordt door SGWM gebruikt om de voortgang van de sommen in de voortgangsbalk in UI weer te geven. Het is dus mogelijk om zelf te kiezen of de voortgang gemeten moeten worden op basis van het aanmaken van de invoer van sommen of het aantal succesvol uitgevoerde sommen.
- Introductie van een zelf te definiëren SGWM somID (verplicht veld). Onder 'Computation' in het project-definitie-bestand moet tussen de tags 'Id' de definitie van de somID opgegeven worden. Deze bestaat verplicht uit alle parameters (verplicht!) eventueel aangevuld met variabelen of een willekeurige tekst. Bijvoorbeeld; 'Modelrun_Q{Q}_H{H}'. Hierbij worden {Q} en {H} per som ingevuld door SGWM. Hiermee vervalt de functionaliteit uit voorgaande versie waarbij de somID automatisch werd bepaald uit een samentrekking van de Key van een parameter gevolgd door de waarde van de parameter van de som en waarbij de volgorde van de definitie van de parameters in het project-definitie-bestand werd aangehouden. **Het gebruik van de SGWM key {ID} (in de templates bestanden) is hiermee komen te vervallen!**

Hieronder een voorbeeld waarbij de somID wordt samengesteld uit alle parameters ({Q} en {H}), aangevuld met de variabele {U} en willekeurige tekst voorafgaand aan de parameters en variabele om de waarden te duiden en een tekst aan te einde van de somID.

```

<Computation>
  <Folder>computations</Folder>
  <Id>Q{Q}_H{H}_U{U}_D1</Id>
  <Parameters>
    <Parameter>
      <Name>Afvoer Rijn en Maas id</Name>
      <Key>Q</Key>
      <Type>float</Type>
      <Values>
        <Value>1</Value>
        <Value>2</Value>
        <Value>3</Value>
        <Value>4</Value>
        <Value>5</Value>
        <Value>6</Value>
        <Value>7</Value>
        <Value>8</Value>
        <Value>9</Value>
      </Values>
    </Parameter>
    <Parameter>
      <Name>Zeewaterstand id</Name>
      <Key>H</Key>
      <Type>float</Type>
      <Values>
        <Value>1</Value>
        <Value>2</Value>
        <Value>3</Value>
        <Value>4</Value>
        <Value>5</Value>
        <Value>6</Value>
      </Values>
    </Parameter>
  </Parameters>
  <Variables>
    <Variable>
      <Name>Windsnelheid id</Name>
      <Key>U</Key>
      <Type>string</Type>
      <ParameterValues>
        <ParameterValue>
          <Parameters>

```

- Optimalisatie van het bepalen van de status van sommen. De status van een som wordt bepaald door (de inhoud van) in- of uitbestanden in de som mappen te scannen. Bij heel veel sommen kan de scantijd behoorlijk oplopen. Om dit optimaliseren zijn er twee wijzigingen doorgevoerd:
 - Een scan van de status van de sommen wordt niet meer automatisch gestart door een handeling van SGWM, maar louter door een knop die door de gebruiker moet worden gactiveerd.
 - Zodra een definitieve status van som is bepaald¹ wordt deze niet opnieuw gescand.
- Het is nu mogelijk om naar een specifiek som te browsen, vanuit de sommenoverzicht tabel. Dit kan door met de rechtermuisknop, op een willekeurige rij in de tabel met sommen, te klikken. Er verschijnt dat een menu 'Open map'. Hiermee wordt een Windows verkennen venster gestart, waarin de folder van de bewust som is geselecteerd.
- Het is nu vanaf deze versie mogelijk om 'CalculationSettings' te gebruiken als SGWM key in templates. Bovendien is mogelijk om conditionele 'CalculationSettings' te definiëren. Dat wil zeggen dat een 'CalculationSetting' afhankelijk is van de waarde van een andere 'CalculationSettings'. Hieronder is voorbeeld weergegeven waarin de CalculationSetting 'Zeespiegelstijging' afhankelijk is van de waarde van de CalculationSetting 'Scenario' met key 'SCENARIO':

¹ Deze is gelijk aan status in de State definitie in het MDB, waarvan de waarde van de attribuut 'Progress' gelijk is aan 'true'.

```

<CalculationSettings>
  <CalculationSetting>
    <Name>Scenario</Name>
    <Key>{SCENARIO}</Key>
    <Type>combo</Type>
    <Items>prod_ZSS000;prod_ZSS008;prod_ZSS050;prod_ZSS100;prod_ZSS200;prod_ZSS300</Items>
    <Value>prod_ZSS000</Value>
    <Edit>true</Edit>
  </CalculationSetting>
  <CalculationSetting>
    <Name>Zeespiegelstijging</Name>
    <Key>{ZSS}</Key>
    <Type>float</Type>
    <Conditions>
      <Condition>
        <CalculationSettings>
          <CalculationSetting>
            <Key>SCENARIO</Key>
            <Value>prod_ZSS000</Value>
            <Operator>EQ</Operator>
          </CalculationSetting>
        </CalculationSettings>
        <Value>0.00</Value>
      </Condition>
      <Condition>
        <CalculationSettings>
          <CalculationSetting>
            <Key>SCENARIO</Key>
            <Value>prod_ZSS008</Value>
            <Operator>EQ</Operator>
          </CalculationSetting>
        </CalculationSettings>
        <Value>0.08</Value>
      </Condition>
    </Conditions>
  </CalculationSetting>

```

Als CalculationSetting 'Scenario' bijvoorbeeld gelijk is aan 'prod_ZSS000', dan wordt de waarde van de CalculationSetting 'Zeespiegelstijging' gelijk aan 0.00. Bij een waarde van 'Scenario' gelijk aan 'prod_ZSS008' is de waarde van de CalculationSetting 'Zeespiegelstijging' gelijk aan 0.08.

- Het scherm voor het aanmaken van een nieuw project is aangepast. Voorheen moesten er achtereenvolgens in verschillende stappen (via verschillende dialoogvensters) bestanden geselecteerd worden en een SGWM-projectnaam gedefinieerd worden. Dat is nu ondergebracht in één popup. Hierin moet ook de keuze gemaakt worden voor een type project: 1) een traditioneel SGWM-project voor een willekeurig rekenmodel of 2) een SGWM voor MHWp5 project.
- Er is een extra tabblad 'Project' toegevoegd aan de SGWM-gebruikersschil, waarin de voortgang van de sommen kan worden gemonitord.
- Het tabblad 'Sommen' in de SGWM-gebruikersschil is hernoemd naar 'SGWM'.
- Het tabblad 'Sommenset' in de SGWM-gebruikersschil is hernoemd naar 'WGWM' (Willekeurige Generator WaterModellen).
- Naast een overzicht van alle parameters worden nu in de SGWM-gebruikersschil ook de waarden van de variabelen getoond in het tabblad 'SGWM' (voorheen tabblad 'Sommenset'). In dit tabblad is ook een tabel met alle sommen opgenomen.
- In het tabblad 'WGWM' (voorheen 'Sommen') in de SGWM-gebruikersschil is een knop toegevoegd om het module-definitie-bestand te openen.
- Bij het filteren van de sommen in het tabblad 'SGWM' (voorheen 'Sommen') in de SGWM-gebruikersschil worden nu standaard allen sommen geselecteerd na het toepassen van een filter. Bij het opheffen van een filter worden alle sommen gedeselecteerd.
- Integratie van de handleidingen van 'SGWM' en 'SGWM voor MHWp5'.

Consequenties (voor gebruik reeds bestaande SGWM-configuraties)

Voor alle bestaande SGWM-configuraties geldt dat het project-definitie-bestand opgedeeld moet worden in een project-definitie-bestand en een module-definitie-bestand (conform eerder beschreven wijzigingen). Voor meer informatie zie ook de nieuwe handleiding van SGWM (versie 2.0.0). Opgesomd zijn dit de wijzigingen voor een bestaand PBD:

- Toevoegen van de tag <Id> onder <Modelschematisation><Computation> voor het definiëren van de somID.
- Verwijderen van <Templates> uit het blok <Computation>.
- Verwijderen van <Input> uit het blok <Computation>.
- Verwijderen van <States> uit het blok <Computation>.
- Verwijderen van <CalculationSettings>.

En het nieuwe MDB moet bevatten:

- Een naam <Name>.
- Een blok <Templates> met alle templates (conform de definitie uit het 'oude' PDB).
- Een blok <Input> met de folder naam en filenaam (conform de definitie uit het 'oude' PDB).
- Een blok <States> met status definities (conform de definitie uit het 'oude' PDB) waaraan toegevoegd moet worden het attribuut <Progress>.
- Een blok <CalculationSettings> met alle rekeninstellingen (conform de definitie uit het 'oude' PDB).

Als gevolg van de integratie van SGWM voor MHWp5 met SGWM zijn er ook consequenties voor bestaande MWHp5 configuraties. Voor het PDB zijn dit:

- Het toevoegen van het verplichte item <Modelsoftwarepackage>.
- Het verwijderen van de tags <Modelschematisations>, <BoundaryConditions> en <Computations> (zie ook [SGWM-19] bij SGWM versie 1.2.0 release).
- Het vervangen van <Code> door <Key>.
- Het vervangen van <Parameter> door <Key> in het gebruik conditions.
- De XML-tags 'Geometry', 'BoundaryCondition' en 'InitialCondition' zijn niet meer verplicht.
- SGWM keys worden niet meer met \$-tekens gemarkeerd, maar met een begin en eind accolade, zoals '{Parameters}'.
- Het verplicht gebruik van <Operator> in de definitie van de condities bij variabelen.

Voor het MDB zijn dit:

- Het verwijderen van het blok <Initial Conditions>.
- De definitie van de Key in <CalculationSettings> zonder brackets (voorheen \$-tekens).
- Het vervangen van het blok <Output> door een generieke definitie van <States>.
- Introductie van een type templatebestand, waarbij er vier type zijn gedefinieerd: 'inputfile' (het templatebestand is het primaire model invoerbestand van het rekenmodel), 'lsf' (het templatebestand bevat het lsf jobscrip waarmee een som wordt gestart), 'bsub' (het templatebestand bevat een bsub commando scrip waarmee een som kan worden toegevoegd aan de wachtrij van een rekenomgeving) en 'general' (een willekeurig ander template bestand, dat niet voldoet aan de voorgaande definities).

What's New in Version 1.2.0

What's New

- [SGWM-19] Het gebruik van verschillende modelsoftwarepackages in een PDB is vervallen. In een PDB wordt één modelsoftwarepackage beschreven, waardoor de XML-tags 'ModelSoftware-Packages' zijn komen te vervallen. Dit moet de bruikbaarheid (en eenvoud) van het PDB vergroten (gebruikers feedback).
- [SGWM-19] Het gebruik van meerdere modelschematisaties in een PDB is vervallen. In een PDB wordt nu maximaal één modelschematisatie beschreven, waardoor de XML-tags 'ModelSchematisations' zijn komen te vervallen. Dit moet de bruikbaarheid (en eenvoud) van het PDB vergroten (gebruikers feedback).
- [SGWM-19] De verwijzing naar de modelsoftwarepackage in een modelschematisatie is komen te vervallen om dat er nog maar één modelsoftwarepackage kan worden gedefinieerd in een PDB.
- [SGWM-19] Het gebruik van meerdere randvoorwaarden sets in een PDB is vervallen. In een PDB wordt nu maximaal één set randvoorwaarden beschreven, waardoor de XML-tags 'BoundaryConditions' zijn komen te vervallen. Dit moet de bruikbaarheid (en eenvoud) van het PDB vergroten (gebruikers feedback).
- [SGWM-19] Het gebruik van meerdere set berekeningen in een PDB is vervallen. In een PDB wordt nu maximaal één set berekeningen beschreven, waardoor de XML-tags 'Computations' zijn komen te vervallen. Dit moet de bruikbaarheid (en eenvoud) van het PDB vergroten (gebruikers feedback).
- [SGWM-19] De volgorde van de onderdelen van een modelschematisatie in een PDB is aangepast van 'Geometry-Computations-BoundaryConditions' naar 'Geometry-BoundaryCondition-InitialCondition-Computation'. De definitie van 'InitialCondition(s)' is verplaatst van 'BoundaryCondition' naar 'Modelschematisation'. Dit moet de bruikbaarheid (en eenvoud) van het PDB vergroten (gebruikers feedback).
- [SGWM-19] De definitie van 'Parameters' en 'Variabelen' is verplaatst van 'BoundaryCondition' naar 'Computation'. Dit moet de bruikbaarheid (en eenvoud) van het PDB vergroten (gebruikers feedback).
- [SGWM-3] introductie van generieke definitie van som statussen. Waar in voorgaande versie alleen de status 'som succesvol uitgevoerd' en 'som uitgevoerd met fout' kon worden gebruikt, is het nu mogelijk om zelf (meerdere) status definities aan te maken.
- [SGWM-19] De XML-tags 'Geometry', 'BoundaryCondition' en 'InitialCondition' zijn niet meer verplicht. In deze tags wordt een relatieve verwijzing gemaakt naar de naam van een (sub)folder met de betreffende bestanden van de betreffende modelschematisatie. Voor sommige (water)modellen zijn deze (sub)folders niet van toepassing en kunnen dus weggelaten worden uit het PDB.
- [SGWM-19] Introductie van een type template bestand, waarbij er vier type zijn gedefinieerd: 'inputfile' (het template bestand is het primaire model invoerbestand van het rekenmodel), 'lsf' (het template bestand bevat het lsf jobsript waarmee een som wordt gestart), 'bsub' (het template bestand bevat een bsub commando script waarmee een som kan worden toegevoegd aan de wachtrij van een rekenomgeving) en 'general' (een willekeurig ander template bestand, dat niet voldoet aan de voorgaande definities).

- [SGWM-19] SGWM keys worden niet meer met \$-tekens gemarkeerd, maar met een begin en eind accolade, zoals '{ID}'. Hiermee wordt voorkomen dat bij het gebruik van opeenvolgende SGWM keys verkeerd vervangingen plaatsvinden. Elke SGWM key heeft nu een expliciet begin en eind.
- [SGWM-19] Introductie van het gebruik van een XSD validatie schema om een project-definitie-bestand op juistheid en volledigheid te controleren.
- [SGWM-18] Met de aanpassingen uit [SGWM-19] is het tabblad 'Instellingen' komen te vervallen en verwijderd uit de gebruikersschil.
- [SGWM-44] Het is nu toegestaan om negatieve getallen toe passen in de waarden van parameters en variabelen in het project-definitie-bestand.
- [SGWM-37] Het uitbreiden van de testbank en het introduceren van user examples.
- [SGWM-17] Het herstructureren van de gebruikershandleiding en het meer focus geven de verschillende mogelijkheden bij het opstellen van een project-definitie-bestand, met diverse voorbeelden en user examples.

Bugs

- [SGWM-24] Het gebruik van hard-coded SGWM Keys behalve {ID} is opgelost.
- [SGWM-13] Het gebruik van Ascii-tekens groter dan 128 (zoals é, ö, ë, ã en °) is nu toegestaan bij het controleren van een status van een som.
- [SGWM-43] Het probleem van een verkeerde selectie bij een combinatie van sorteren en filteren is opgelost.
- [SGWM-23] Foutmeldingen bij het controleren van de aanwezigheid van verwijzingen in template bestanden zijn opgelost.

Consequenties (voor gebruik reeds bestaande SGWM configuraties)

Met alle aanpassingen aan het project-definitie-bestand zijn bestaande SGWM configuraties niet zomaar meer bruikbaar in deze versie. Verwezen wordt naar de laatste versie van de SGWM handleiding waarin beschreven is wat de eisen zijn van het nieuwe project-definitie-bestand.

What's New in Version 1.1.3

What's New

- [SGWM-38] Introductie van een nieuwe 'type' template, waarmee eenduidig aangegeven kan worden dat een template gebruikt moet worden als invoerbestand voor het rekenmodel. Voordeel van deze aanpassing het hierdoor ook mogelijk is om meerdere verschillende templates op te nemen die afhankelijk van een conditie (dit kunnen er ook meerdere zijn) gebruikt worden als invoerbestand.
- [SGWM-39] Meerdere parameters (een combinatie van parameters) gebruiken in de definitie van een variabele. Zie het voorbeeld hier onder waarin 3 parameters gebruikt worden voor definitie van de variabele 'HR2006CODE'.

```

<Variables>
  <Variable>
    <Name>HR2006 code tbv boundary_conditions</Name>
    <Code>HR2006CODE</Code>
    <Type>string</Type>
    <ParameterValues>
      <ParameterValue>
        <Parameters>
          <Parameter>
            <Code>U</Code>
            <Value>11</Value>
            <Operator>EQ</Operator>
          </Parameter>
          <Parameter>
            <Code>D</Code>
            <Value>202</Value>
            <Operator>EQ</Operator>
          </Parameter>
          <Parameter>
            <Code>L</Code>
            <Value>m100</Value>
            <Operator>EQ</Operator>
          </Parameter>
        </Parameters>
        <Value>K15D210L010</Value>
      </ParameterValue>
    </ParameterValues>
  </Variable>
</Variables>

```

Bugs

- [SGWM-40] Bug opgelost waarbij het vervangen van \$ID\$ in de naam van het invoerbestand niet goed werkte (aanpassing uit versie 1.1.2).
- [SGWM-41] Bug opgelost waarbij variabelen in template bestanden niet correct werden vervangen in het geval dat er meerdere parameter-waarden werden gebruikt in de definitie van een variabele.
- [SGWM-42] Verwijderen van de sortering van kolommen in de tabel met berekeningen omdat in combinatie met filters de selectie van sommen niet goed functioneert

Consequenties (voor gebruik reeds bestaande SGWM configuraties)

[SGWM-38] bij de definitie van templates MOET nu altijd minimaal 1 template opgenomen worden van het type 'input'. Dit is het bestand dat gebruikt wordt als invoerbestand voor het rekenmodel, bijv. bij SWAN het bestand *.swn. De overige templates zijn van het type 'lsf' of het rekenmodel; in onderstaand voorbeeld 'swan'. Met deze aanpassing is het ook mogelijk geworden om voor het invoerbestand verschillende templates te definiëren, die afhankelijk van een voorwaarden (condition) toegepast worden in een som.

```

<Computations>
  <Computation>
    <Name>LabelEUR01</Name>
    <Folder>computations</Folder>
    <Input>
      <Folder>$ID$</Folder>
      <File>eur$ID$.sw<del>n</del></File>
    </Input>
    <Output>
      <Name>*.o*</Name>
      <Value>end</Value>
    </Output>
    <Error>
      <Name>*.e*</Name>
      <Value>ERROR</Value>
    </Error>
    <Templates>
      <Folder>computations\templates\</Folder>
      <Template>
        <Type>input</Type>
        <File>eur_east.sw<del>n</del></File>
        <Conditions>
          <Condition>
            <Parameter>D</Parameter>
            <Value>202</Value>
            <Operator>LT</Operator>
          </Condition>
        </Conditions>
        <Update>true</Update>
      </Template>
      <Template>
        <Type>input</Type>
        <File>eur_west.sw<del>n</del></File>
        <Conditions>
          <Condition>
            <Parameter>D</Parameter>
            <Value>202</Value>
            <Operator>GE</Operator>
          </Condition>
        </Conditions>
        <Update>true</Update>
      </Template>

```



What's New in Version 1.1.2

What's New

- Mogelijkheid ingebouwd dat de naam van het invoerbestand van een rekenmodel (bijvoorbeeld *.swn bij een SWAN model) uitgebreid kan worden met de ID van de som, bijvoorbeeld `swaninvoer\$ID\$.swn`.
- Kleine tekstuele aanpassing van een logmelding in het logbestand (i.p.v. invoerbestand `aangepast`, nu invoerbestand `aangemaakt`).

Bugs

Bug opgelost bij het controleren of een parameter die gebruikt wordt in een variabele definitie, ook daadwerkelijk als parameter bestaat. Bij meerdere variabelen ging dit nog mis.

Consequenties (voor gebruik reeds bestaande SGWM configuraties)

Geen.

What's New in Version 1.1.1

What's New

- [SGWM-35] Verschillende folders die gebruikt worden in SGWM (geometry, boundary-conditions, etc) mogen nu elke willekeurige naam hebben. Er wordt wel (nog steeds) gecontroleerd of de mappen bestaan.
- [SGWM-36] Variabelen zijn niet meer verplicht in het project-definitie-bestand.

Bugs

Geen.

Consequenties (voor gebruik reeds bestaande SGWM configuraties)

Geen.

What's New in Version 1.1.0

What's New

- [SGWM-1] Gebruik ApplicationSettings als conditionele voorwaarden voor het selecteren van templates in het project-definitie-bestand is nu toegestaan.
- [SGWM-4] In de definitie van Variabelen in het project-definitie-bestand kan nu gebruik gemaakt worden van condities met een operator (groter dan, gelijk aan, etc.).
- [SGWM-5] Sorteermogelijkheid van kolommen in tabel met sommen uitgebreid.
- [SGWM-7] Parameter- en variabel codes hoeven niet meer noodzakelijk uit hoofdletters te bestaan.
- [SGWM-8] Naamgeving van condities items bij initiële condities in project-definitie-bestand is aangepast naar een logischer naamgeving.
- [SGWM-11] Nieuw icon specifiek voor SGWM (in plaats van een default icon).
- [SGWM-16] Gebruik Parameters en Variabelen in LSF templates.
- [SGWM-20] Condities zijn geen verplicht item meer bij de definitie van templates in het project-definitie-bestand.
- [SGWM-26] SGWM kan als executabel gebruikt worden. Tevens is er yml bestand beschikbaar gesteld waarmee een Anaconda (Python) omgeving voor SGWM ingericht kan worden.

Bugs

- [SGWM-8] Het aanmaken van een nieuw project in een bestaande folder is nu toegestaan.
- [SGWM-12] Vraagtekens in popup meldingen verwijderd.

- [SGWM-14] Kolombreedtes in tabel met sommen worden automatisch aangepast aan de breedte van de inhoud.
- [SGWM-15] Werking van selectie/filter mechanisme in tabel verbeterd.
- [SGWM-25] Een SGWM configuratie met slechts 1 parameter is nu toegestaan.
- [SGWM-28] Dubbelen entries van logmeldingen verwijderd.
- [SGWM-33] Verbeterde foutmelding bij configuratie fouten (in het project-definitie-bestand).

Consequenties (voor gebruik reeds bestaande SGWM configuraties)

- [SGWM-1] Gebruik ApplicationSettings als conditionele voorwaarden voor het selecteren van templates in het project-definitie-bestand.

Bij een conditie voor een template kan optioneel de eigenschap 'calculatiesetting' meegegeven worden. Als deze niet is gedefinieerd dan is de waarde default false. Hiermee zijn configuraties van SGWM 1.0.4 ook compatibel met versie 1.1.0. Als de eigenschap 'calculatiesetting' gelijk is aan true dan moet de parameternaam overeenkomen met de Key van een CalculationSetting (zie voorbeeld hieronder). Afhankelijk van het type calculationsetting (float, string of integer) kan gebruik gemaakt worden van de standaard operators (EQ (altijd bij een string), GT, LT, LE of GE).

```
<Template>
  <Type>software_name</Type>
  <File>template_6.txt</File>
  <Conditions>
    <Condition calculationsetting="true">
      <Parameter>CS_Float</Parameter>
      <Value>1</Value>
      <Operator>LE</Operator>
    </Condition>
  </Conditions>
  <Update>true</Update>
</Template>
```

```
<CalculationSetting>
  <Name>CS_Float</Name>
  <Key>$CS_Float$</Key>
  <Type>float</Type>
  <Value>1</Value>
  <Edit>true</Edit>
</CalculationSetting>
```

- [SGWM-8] Naamgeving van condities items bij initiële condities in project-definitie-bestand is aangepast naar een logischer naamgeving.

De naamgeving van de items bij initiële condities in project-definitie-bestand is aangepast van:

```
<InitialCondition>
  <File>Initial_3.txt</File>
  <InitialTime>14400</InitialTime>
  <Conditions>
    <Condition>
      <Parameter>Parameter_2</Code>
      <Value>value1</Value>
      <Operator>EQ</Operator>
    </Condition>
  </Conditions>
</InitialCondition>
```

naar:

```

<InitialCondition>
  <File>Initial_3.txt</File>
  <InitialTime>14400</InitialTime>
  <Conditions>
    <Condition>
      <ParameterCode>Parameter_2</ParameterCode>
      <ParameterValue>value1</ParameterValue>
      <Operator>EQ</Operator>
    </Condition>
  </Conditions>
</InitialCondition>

```

Oude SGWM configuraties moeten hierop bijgewerkt worden.

- [SGWM-4] In de definitie van Variabelen in het project-definitie-bestand kan nu gebruik gemaakt worden van condities met een operator (groter dan, gelijk aan, etc.).

In voorgaande versies van SGWM kon een variabele definitie alleen 1-op-1 voor elke waarde van een parameter gedefinieerd worden. Door nu de verplichte <Operator> tags op te nemen kan een variabele nu ook gedefinieerd worden onder een conditie groter dan/kleiner dan of gelijk aan een parameter waarde. Let wel oude SGWM configuraties moeten hierop bijgewerkt worden.

```

<Variable>
  <Name>Variabel_2</Name>
  <Code>Variabel_2</Code>
  <Type>string</Type>
  <ParameterValues>
    <ParameterValue>
      <Parameters>
        <Parameter>
          <Code>Parameter_2</Code>
          <Value>value1</Value>
          <Operator>EQ</Operator>
        </Parameter>
      </Parameters>
      <Value>var2_value1</Value>
    </ParameterValue>
  </ParameterValues>
</Variable>

```

N.B. in versie 1.0.3 is de tag <Update> geïntroduceerd bij de templates. Met deze eigenschap kan aangegeven worden of een template SGWM keys bevat die bijgewerkt moeten worden (waarde is true) of dat de template as-is alleen gekopieerd hoeft te worden (waarde is false). Let op de tags <Update> zijn verplicht bij de template definitie in het project-definitie-bestand. Mogelijk werken oude configuraties van SGWM niet meer.

```

<Template>
  <Type>software_name</Type>
  <File>template_3.txt</File>
  <Conditions>
    <Condition calculationsetting="false">
      <Parameter>Parameter_1</Parameter>
      <Value>100</Value>
      <Operator>EQ</Operator>
    </Condition>
    <Condition calculationsetting="false">
      <Parameter>Parameter_2</Parameter>
      <Value>value2</Value>
      <Operator>EQ</Operator>
    </Condition>
  </Conditions>
  <Update>true</Update>

```