Deltares

Introducing the new whatif-functionality

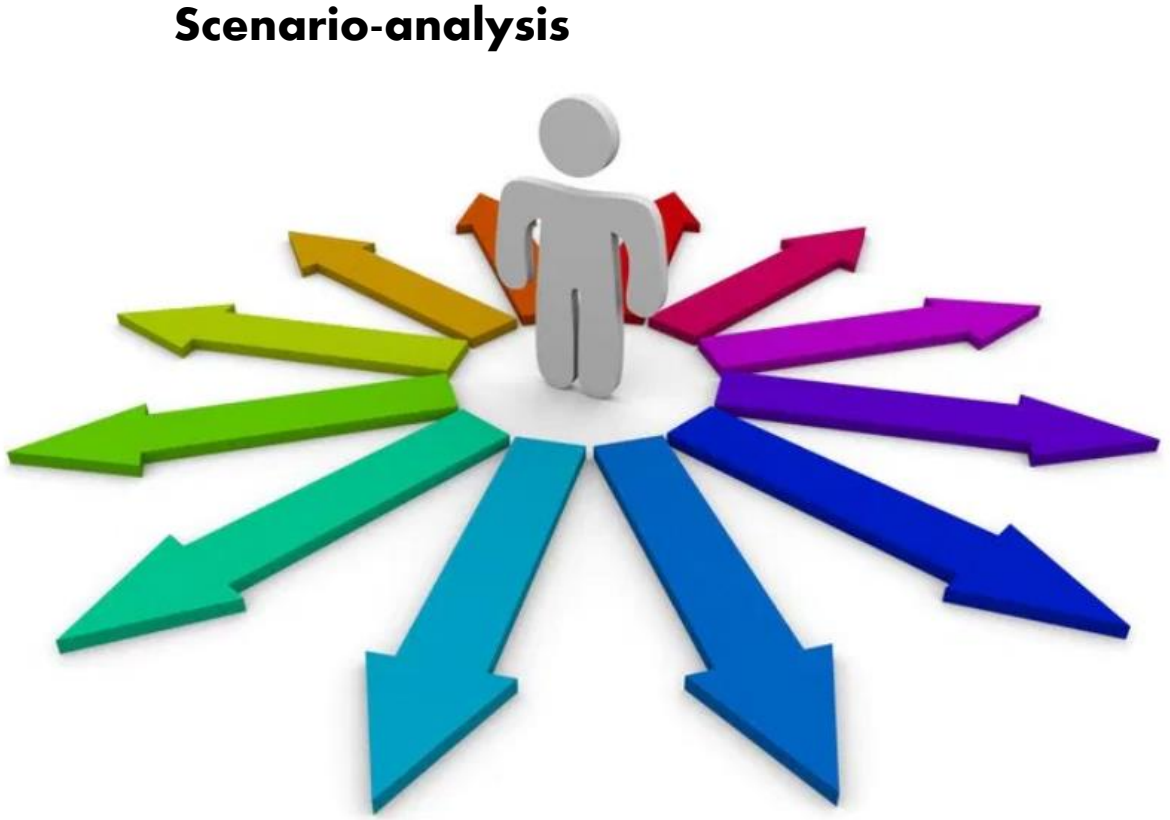
and its application modes

Peter Gijsbers

November 10, 2022: Delft-FEWS International User Days – break out

Outline

- Why a new what-if functionality?
  - What is the Computational Framework ?
- How does it look and work ?
- How to configure ?
- Let’s try on your config

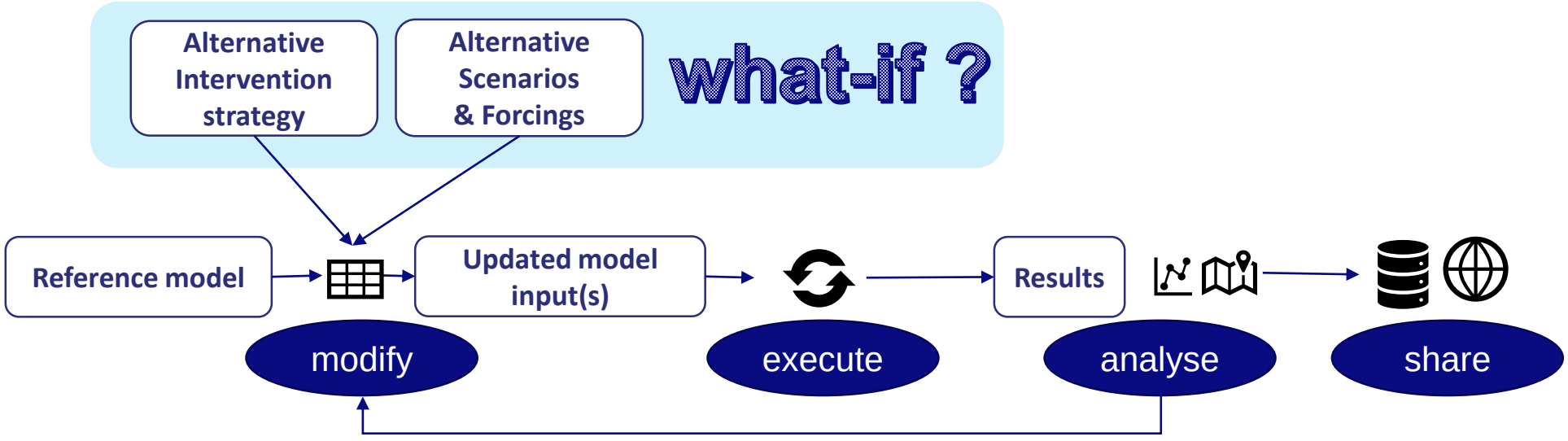


Why ?

Customers are interested in what-if scenario analysis:

- during real-time operations
- in model studies

Delft-FEWS is used for those purposes



Deltares

Why ? Application characteristics:

Realtime Operations versus Model Studies

Realtime operations:

- multi-user system (OC/MC/DB/FSS)
- system time (and start time of runs) continuously updated
- automated data removal based on expiry times

Model studies:

- typically single user (Stand-alone)
- fixed system time/start time of the run
- desire for manual control over data removal
- sometimes wish to incorporate non-Fews displays

Computational Framework mode

Why ?

What-if scenario

only applied when what-if **scenario** is **selected** while dispatching a specific workflow

ability to change input data (legacy)

- typical profile timeseries
- module dataset selection
- module parameter selections/adjustments
- properties

Modifiers

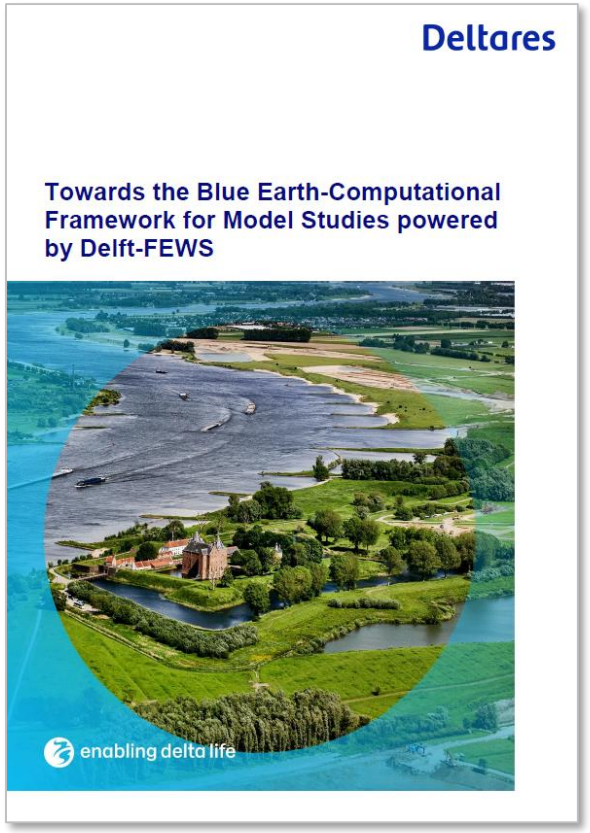
always applied when modifier exists

ability to change input data

- timeseries
- location attributes
- spatial
- model parameters

Why?

investigation study 2019



- Improvement wishes
  - Application of modifiers in what-if run only
  - What-if management and access
  - On-the-fly differences
  - Data management
- Renewal of legacy what-if functionality suites the Roadmap Code clean up

Decision

- End of 2019, Deltares decides to improve the existing Delft-FEWS capabilities to facilitate those needs.
- Introduce term ‘Computational Framework’ as an application mode (and architecture choice) for model studies

Computational Framework mode

What has been achieved ?

Software enhancements implemented:

new what-if functionality

- editor allows combination of modifiers, properties, module datasets
- what-if access integrated in IFD
  - what-if editor
  - what-if visualization in TimeSeriesDisplay
  - what-if visualization in SpatialDisplay
- by default executed as non-approved runs

on-the-fly difference plots between selected runs:

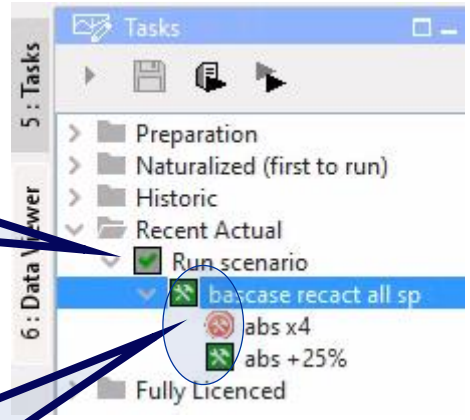
- Spatial Display
- TimeSeriesDisplay after selecting function: difference plot

What-If access
integrated in IFD

- What-if access via IFD (task/forecast tree)
  - Whatif Editor used for composition, comparing inputs, deleting

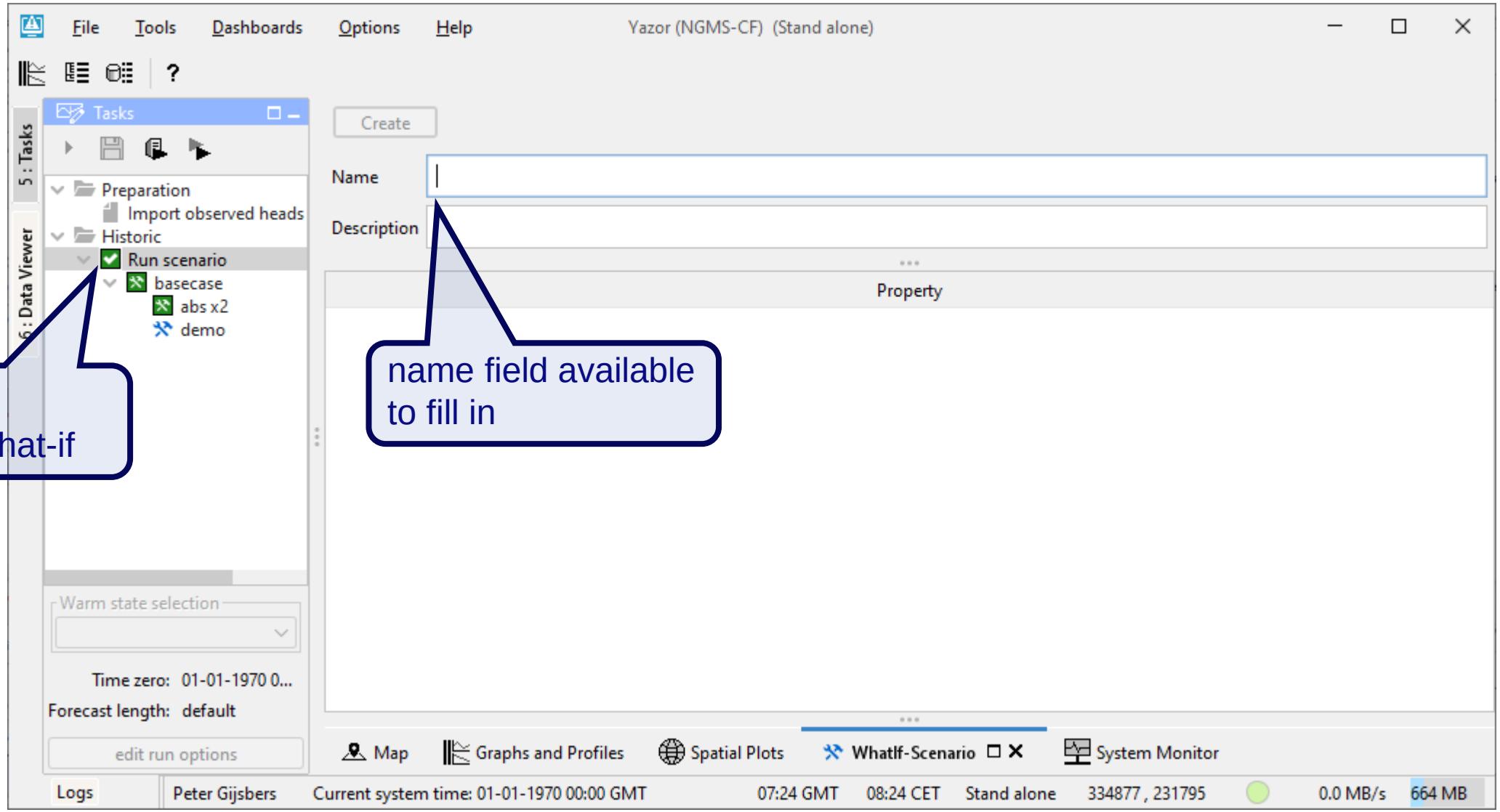
IFD-segment node

IFD-what-if nodes

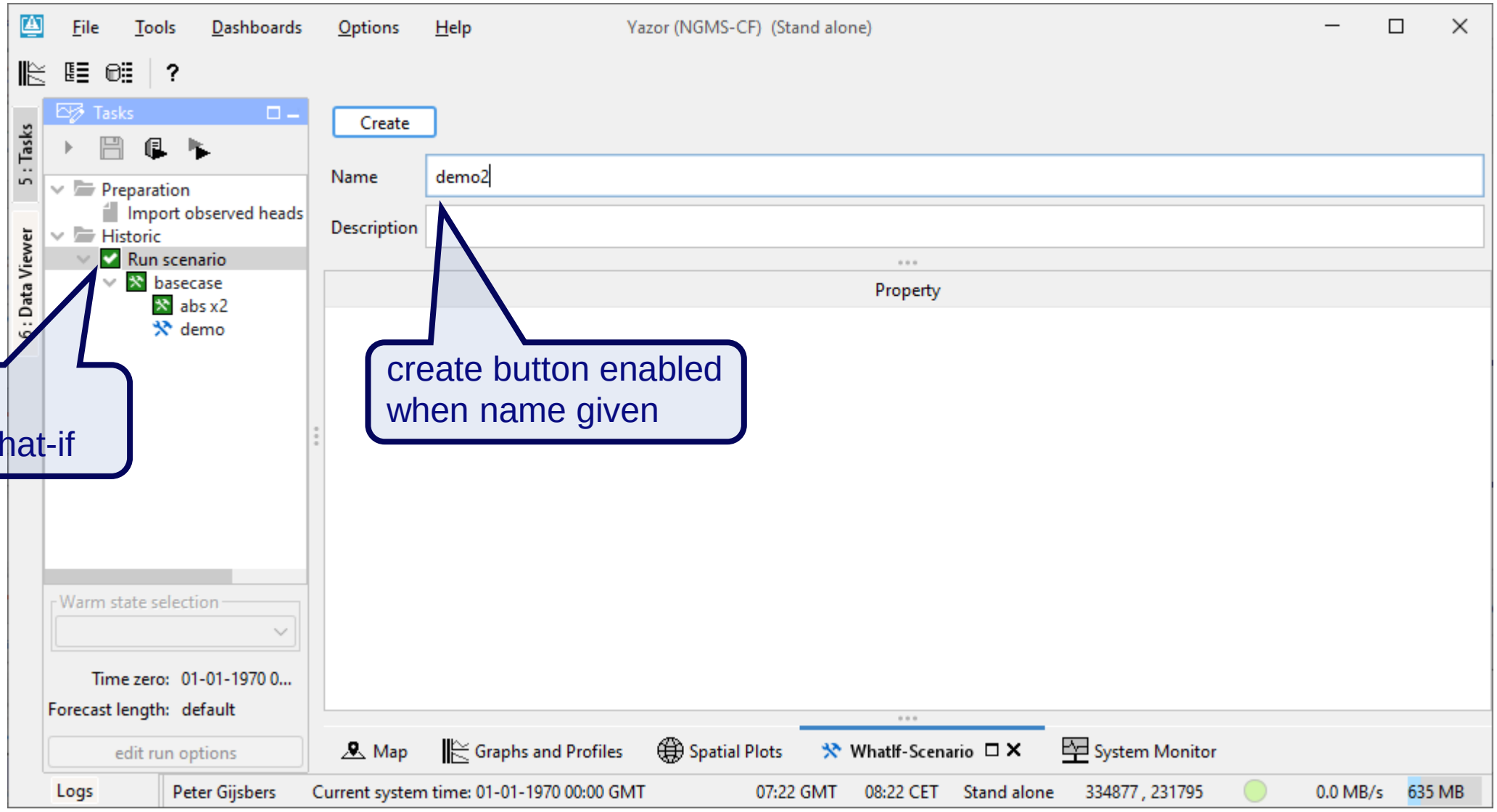


Icon	Meaning	Status
	whatif defined (not run yet)	Can be changed
	whatif run locally	Can be changed
	whatif run on server	Can not be changed
	whatif run failed	

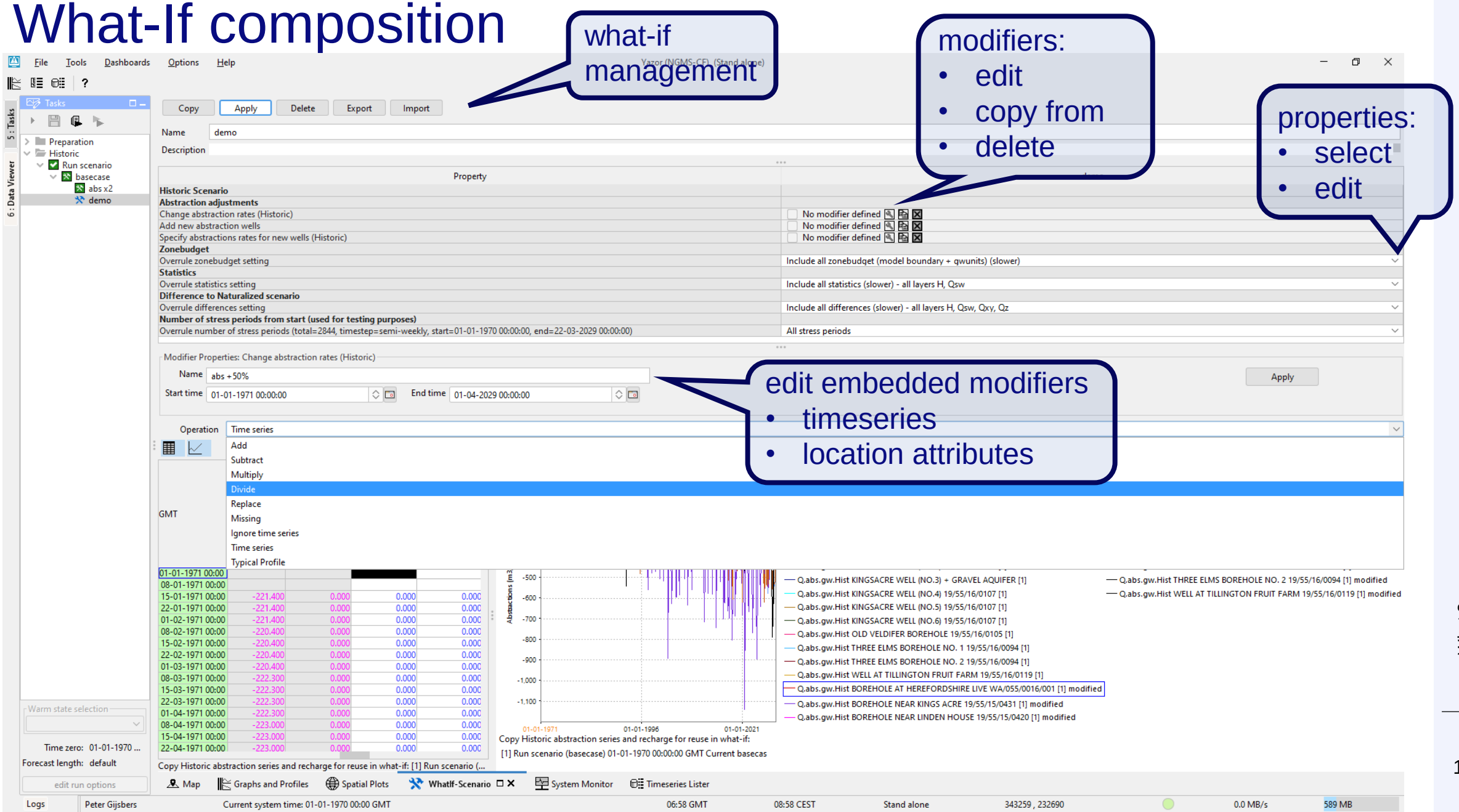
What-If creation



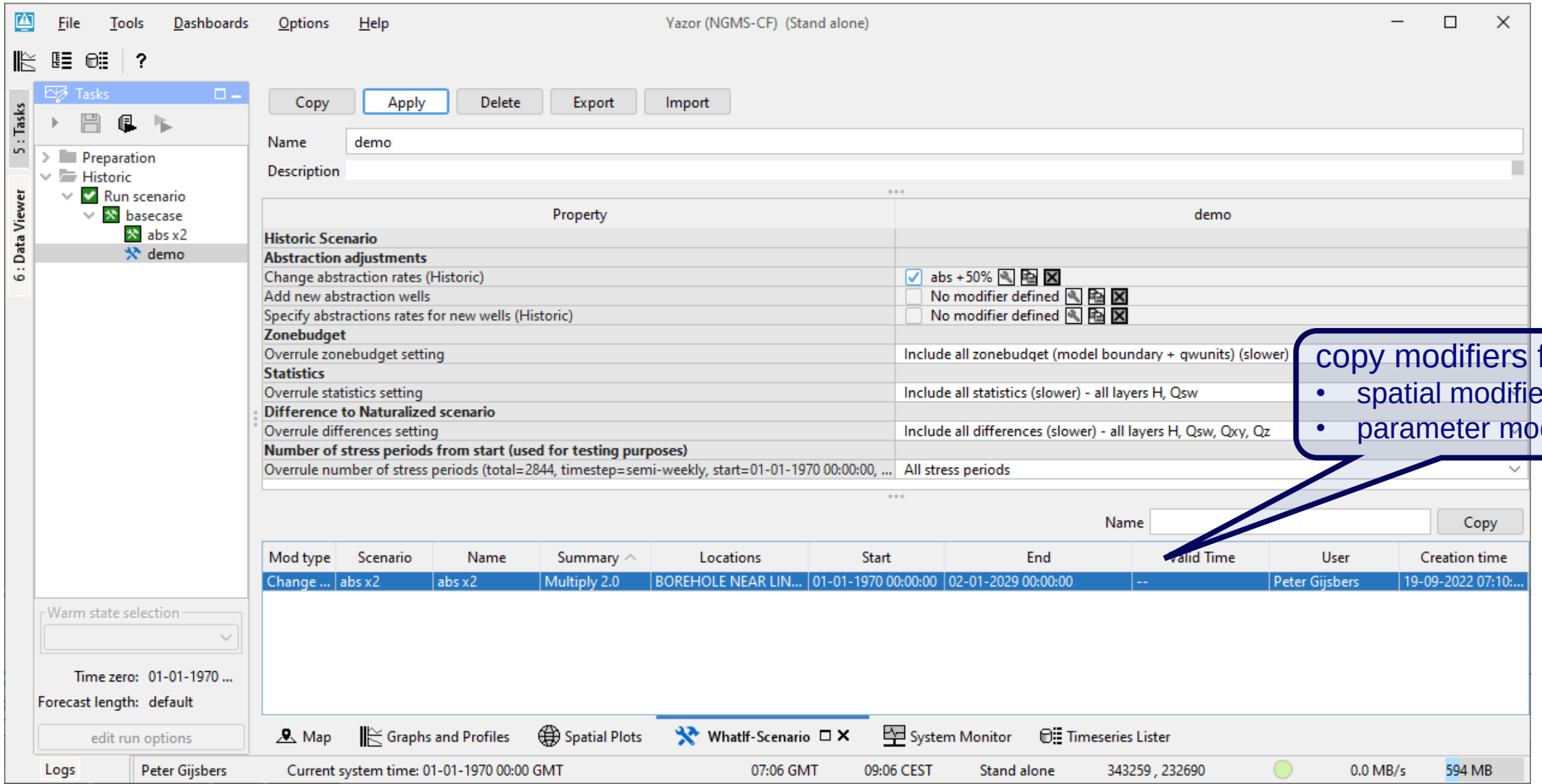
What-If creation



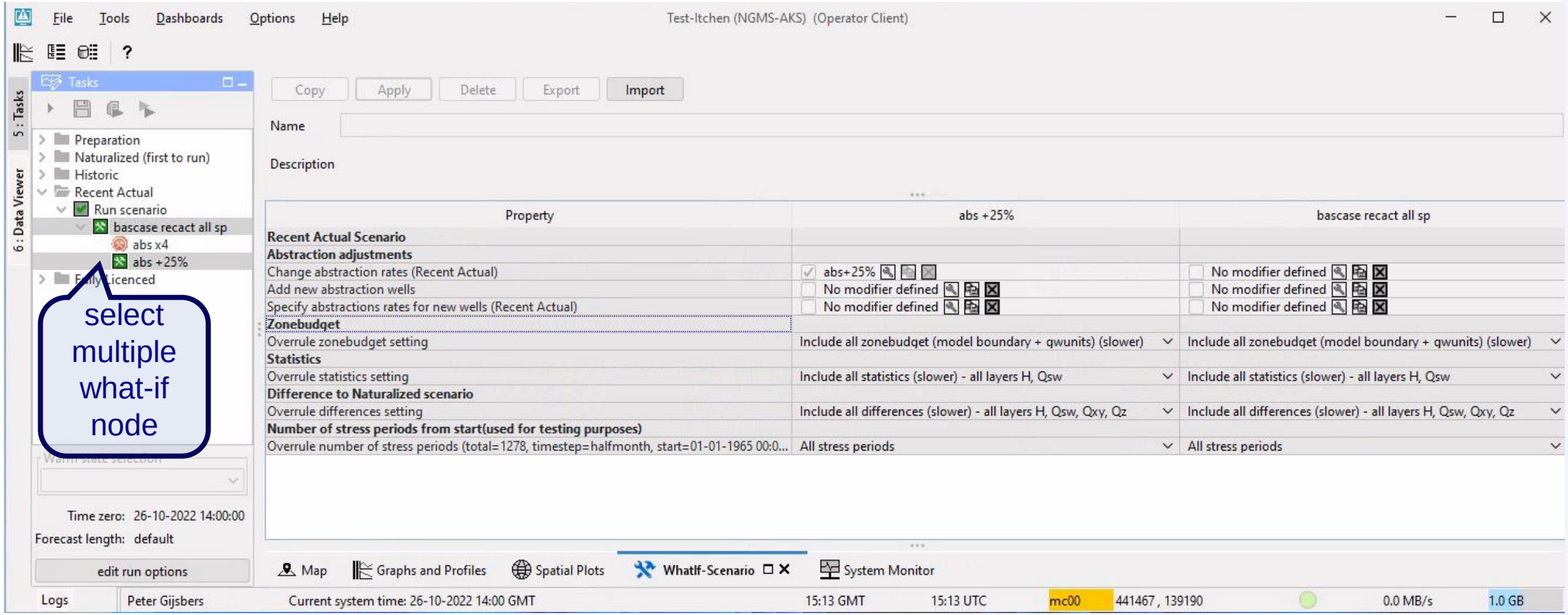
What-If composition



What-If composition



What-If composition and comparison

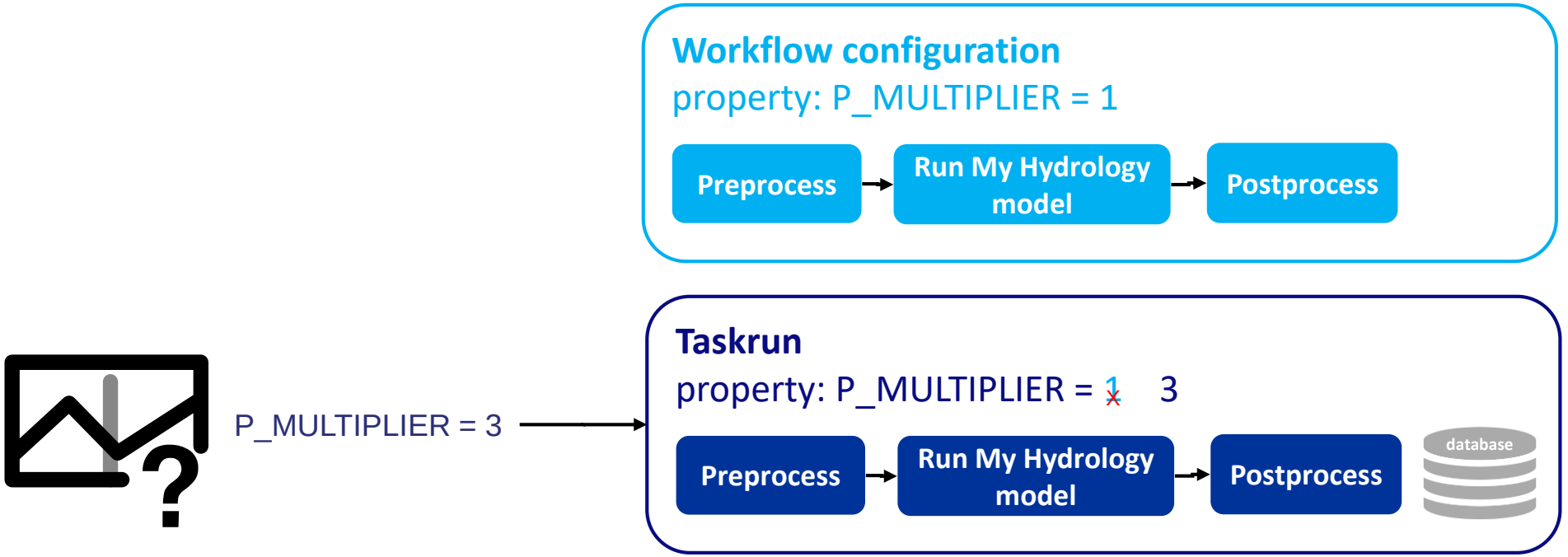


Accessibility in multi-user context

- What-if definitions are only visible in the IFD for all OCs if the what-if has been submitted to the server
- Permission control has not (yet) been implemented for what-if management (creation, deletion)

- Once created, what-ifs can also be selected from the Manual Forecast Display

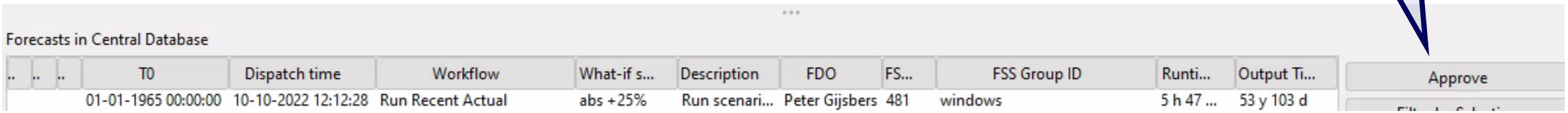
What-if execution
How does it work ?



What-if runs are *unapproved* by default

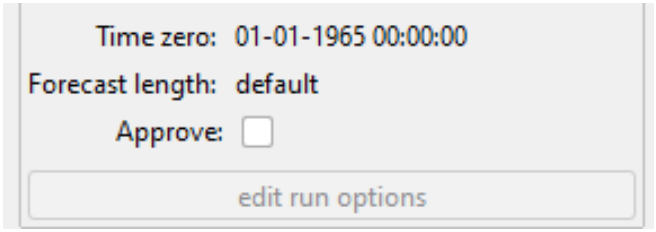
What-if execution
Want to approve the what-if run ?

- Use the Approve option in the Forecast Management dialog



- Auto-approve before submitting the run
  - configure the visibility of the ‘Approve checkbox’ of the IFD (Topology.xml):

    <enableAutoApprove>true</enableAutoApprove>
  - Put a check in the box before submitting the what-if run



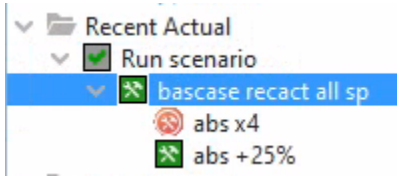
Inspecting results
Access via IFD

Inspecting a what-if run

- Select the whatif-node in the IFD

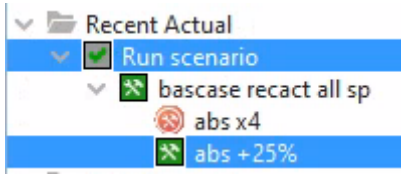
  → The associated timeseries are visualized in the TimeSeries and SpatialDisplay

  → The Current Run is not shown



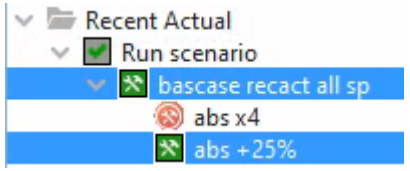
Inspecting a what-if run and the Current Run

- Select the segment node of the IFD as well as a what-if node



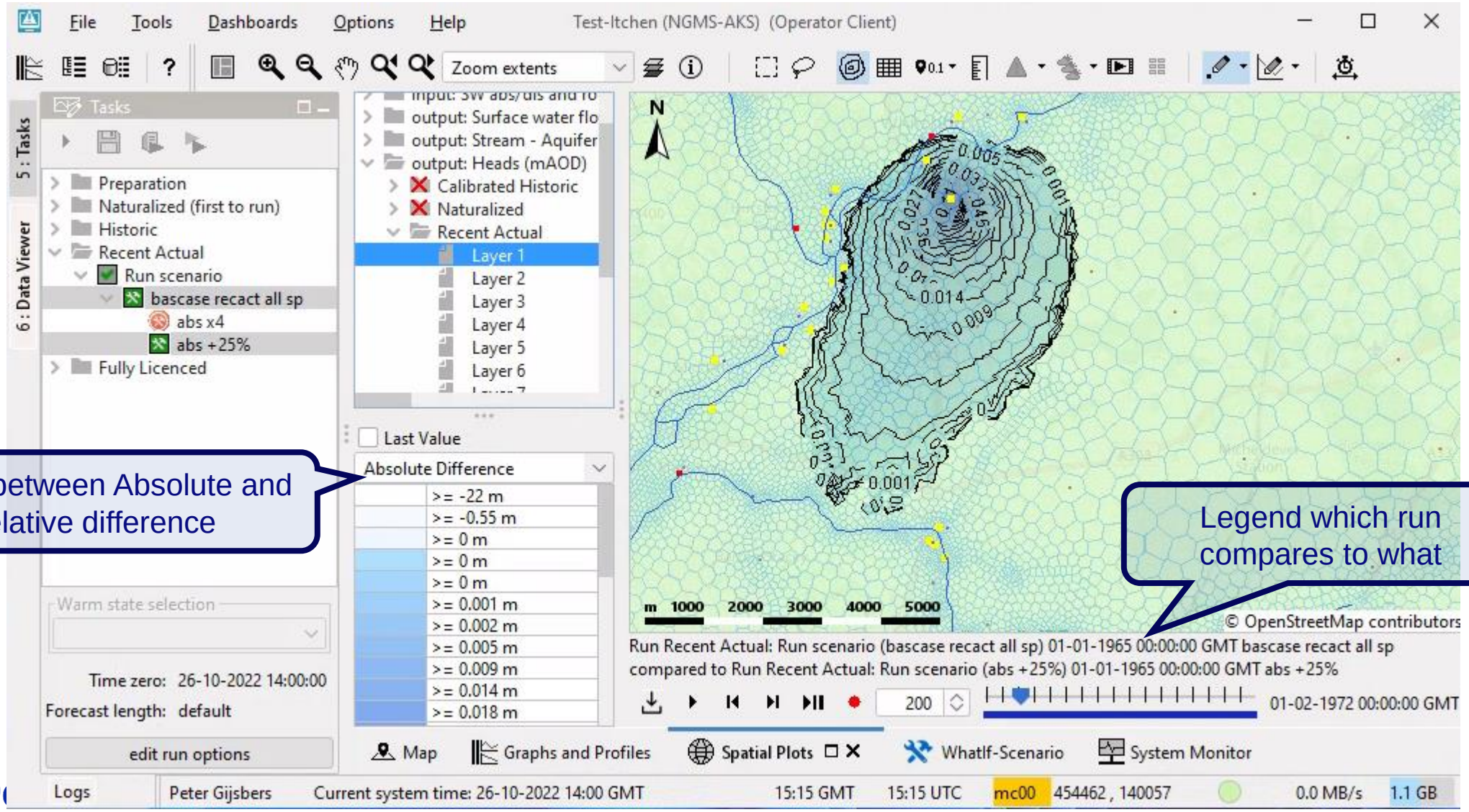
Inspecting multiple what-if runs

- Select multi what-if nodes



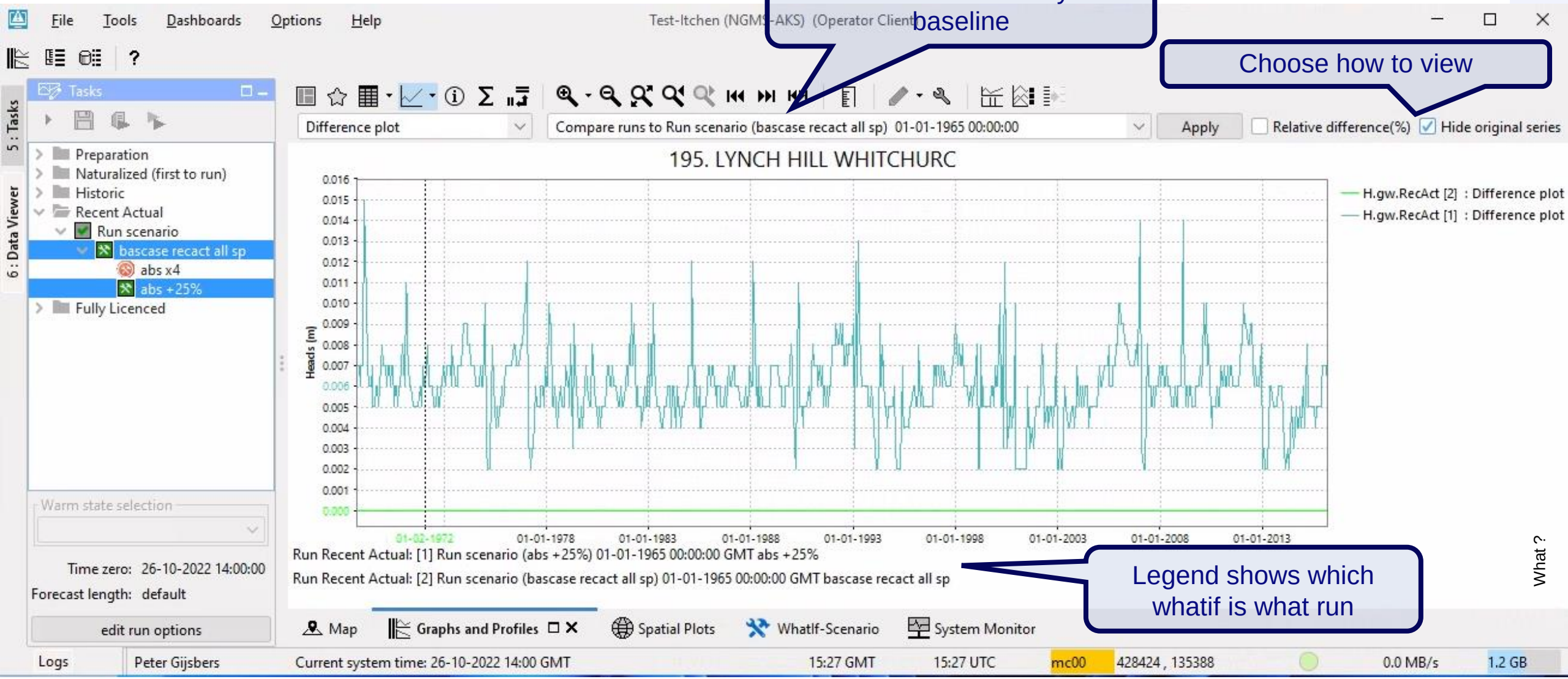
What-If comparison: Spatial Display

access via IFD



What-If comparison: Time Series Display

access via IFD

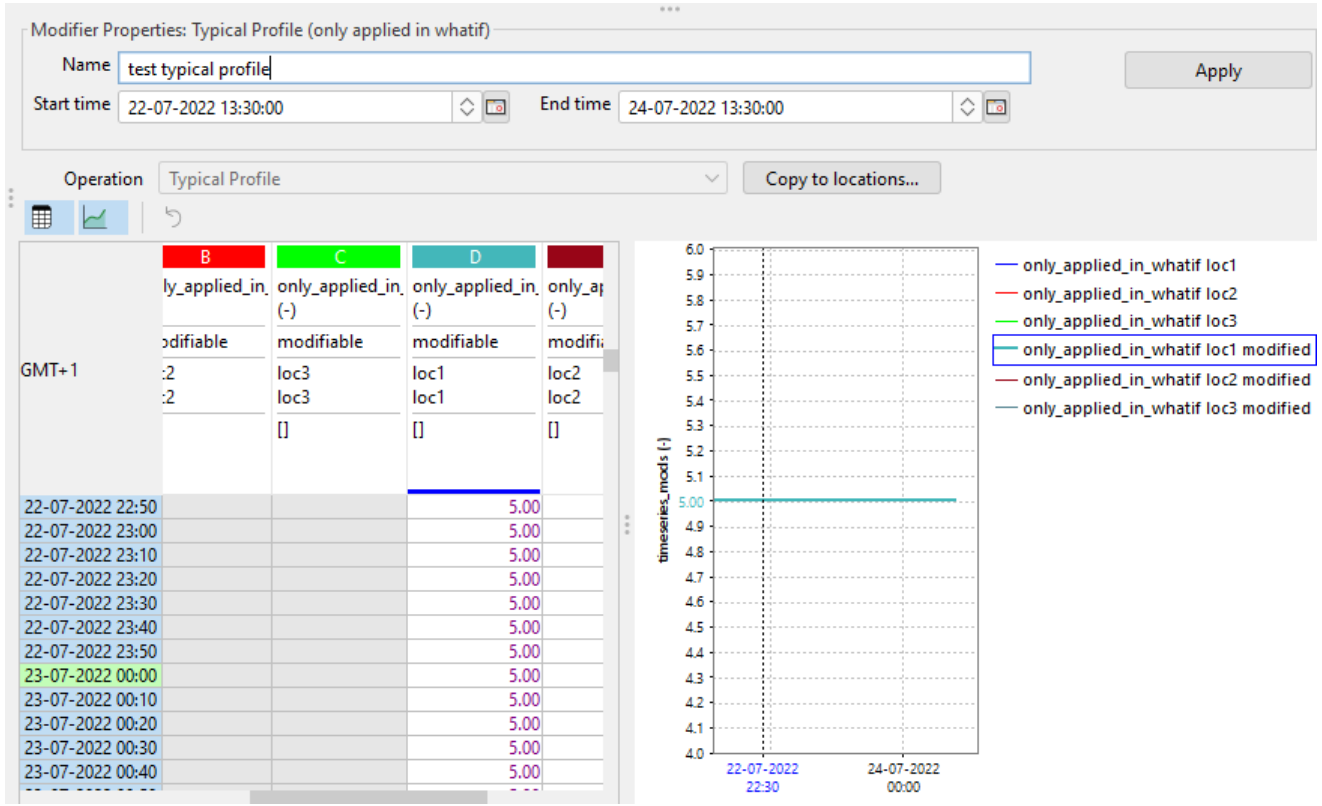


typicalProfile Modifier

- A typicalProfile modifier applies a timeseries array relative to T0 (‘fixed’ when creating the modifier)
- Timeshift applied to the array when applying at a different T0 (as is common in live systems)

‘fixed’ version of a timeseriesModifier
(also available as operation choice in normal timeseriesModifier)

‘T0’ timestamp of the typical profile



Configuration

- The **structure** of a WhatIf is specified in a WhatIfTemplate:
  - RegionConfigFiles/**WhatIfTemplates**.xml

- A WhatIfTemplate is workflow specific
  - **connection** defined in **WorkflowDescriptors**.xml

- Modifiers can be made whatif-specific:
  - **modifier behavior** defined in **ModifierTypes**.xml

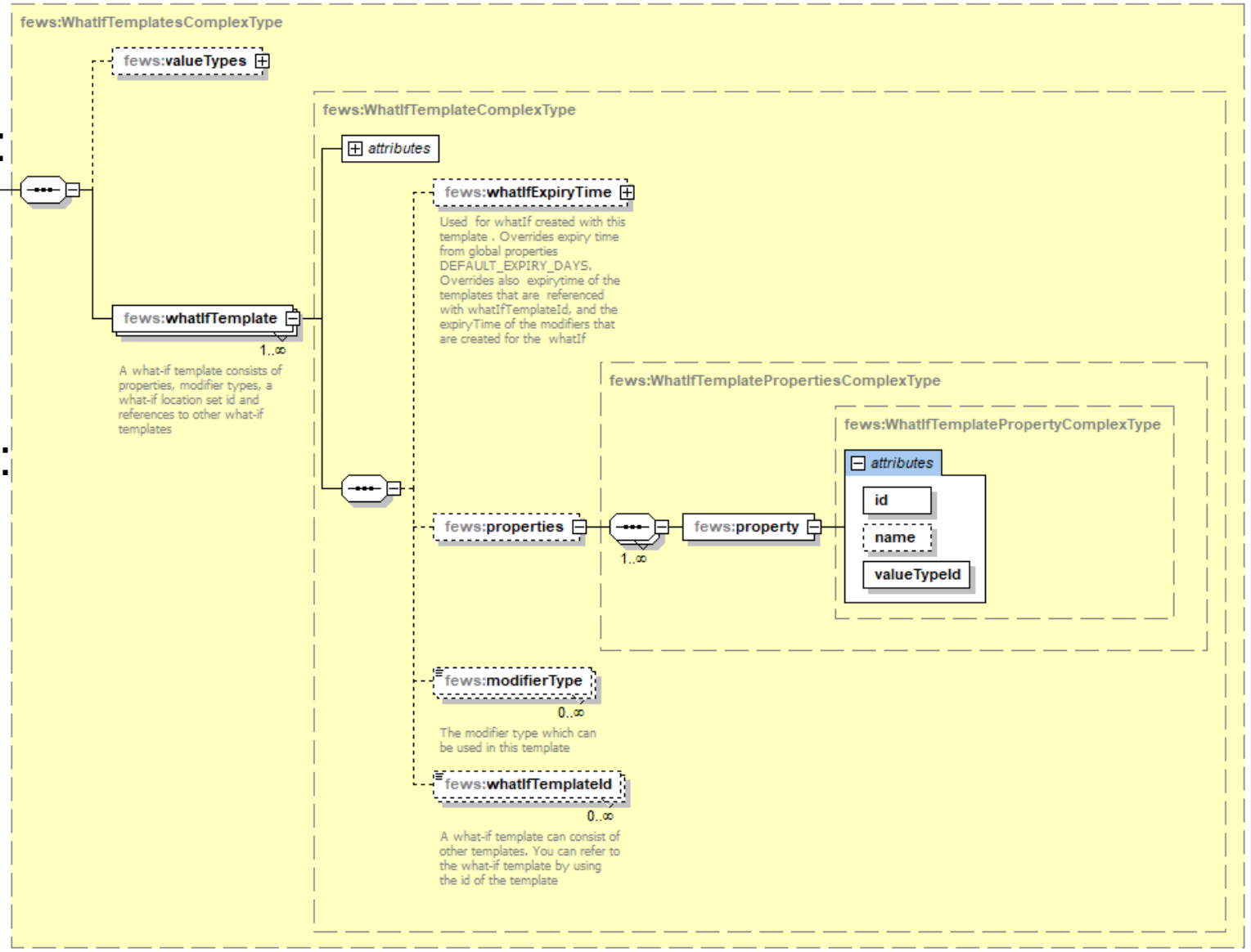
Configuration: WhatIfTemplates.xsd

A **whatif-instance** can hold:

- property values
- references to modifiers included

A **WhatifTemplate** can hold:

- property keys
- Modifier types
- whatif-template nesting



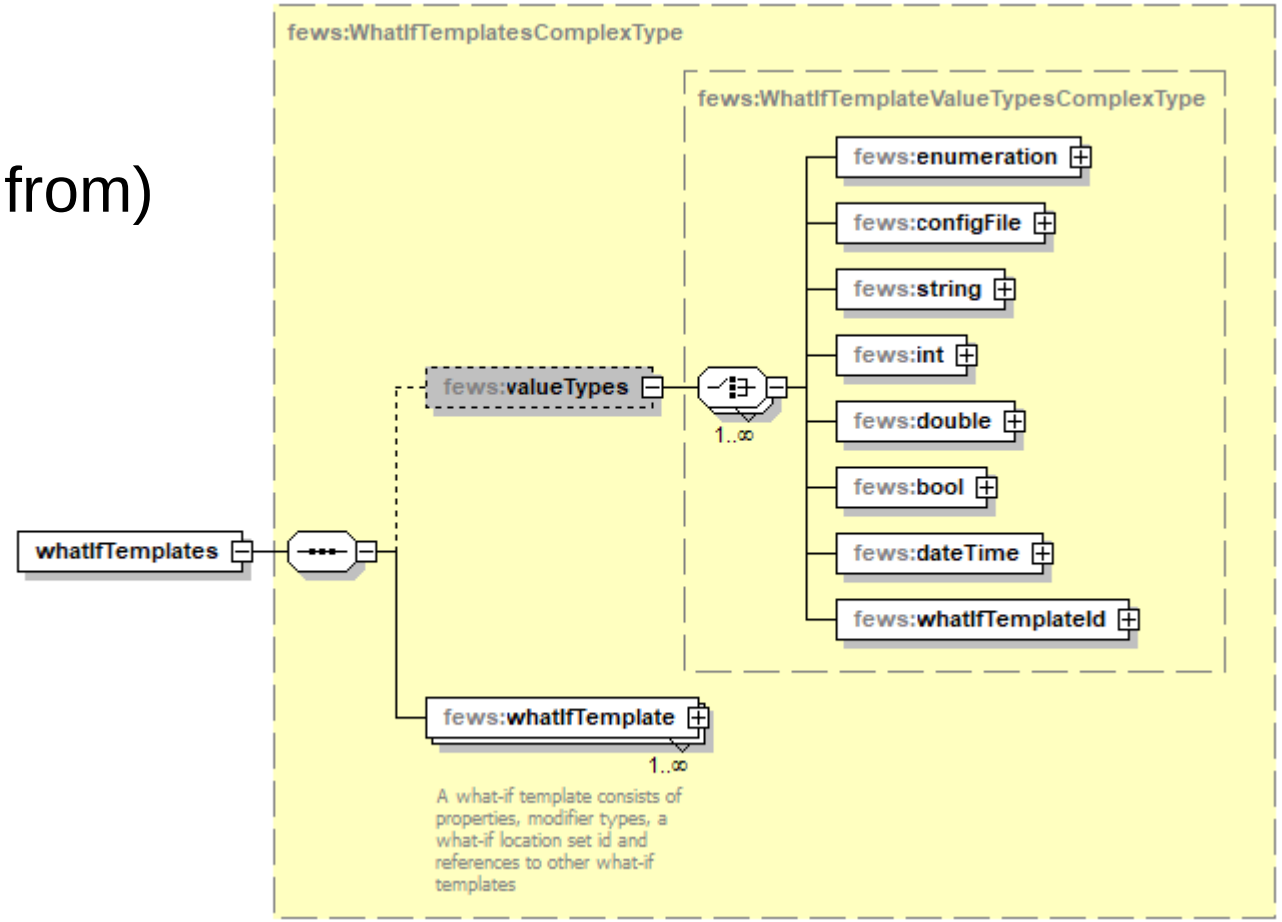
Deltares

Configuration: WhatIfTemplates.xsd

valueTypes define property keys
(and possibly its values to select from)

Available configuration options:

- string (entry)
- enumeration (selection)
- int (entry)
- double (entry)
- configfile (selection)
  - parameter file
  - moduledataset
  - cold state
- whatIfTemplateId
  - can be used to reference an existing what-if instance to pass on associated timeseries



Configuration: WhatIfTemplates.xml example

- Example property

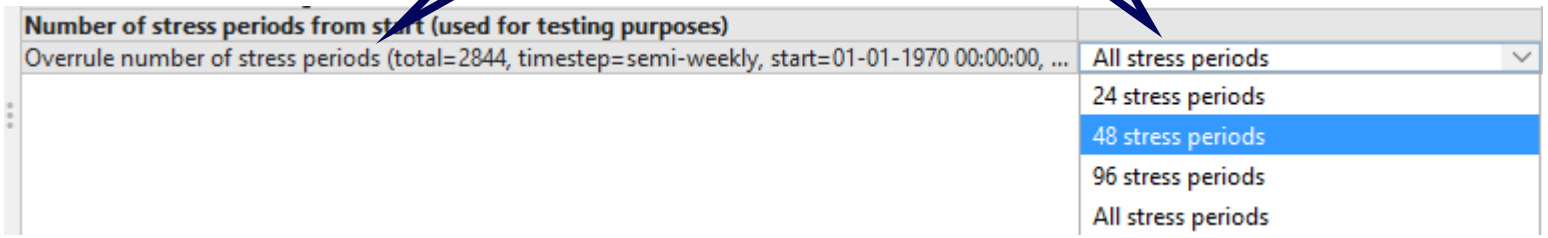
```
<whatIfTemplate id="$MODELID$_n_stress_hist" name="Number of stress periods from start (used for testing purposes)">
  <properties>
    <property id="N_STRESS_PERIOD" valueTypeId="n_stress_period_selection" name="Overrule number of stress periods"/>
  </properties>
</whatIfTemplate>
```

```
<valueTypes>
  <enumeration id="n_stress_period_selection" default="1000000">
    <value code="24" label="24 stress periods"/>
    <value code="48" label="48 stress periods"/>
    <value code="96" label="96 stress periods"/>
    <value code="1000000" label="All stress periods"/>
  </enumeration>
```

Workflow property key

Property based on ValueType

ValueType definition



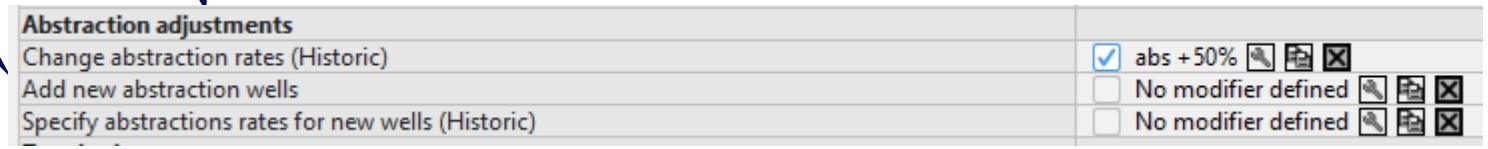
Configuration: WhatIfTemplates.xml example

- Example modifier

```
<whatIfTemplate id="$MODELID$_change_abstractions_hist" name="Abstraction adjustments">
  <modifierType>change_abstraction_rates_hist</modifierType>
  <modifierType>add_new_wells</modifierType>
  <modifierType>new_wells_abstraction_rates_hist</modifierType>
</whatIfTemplate>
```

Property based on ValueType

WhatIfItemplate header



Configuration: WhatIfTemplates.xml example

- Example whatif-template nesting

```
<whatIfTemplate id="$MODELID$_whatif_hist" name="Historic Scenario">
    <whatIfTemplateId>$MODELID$_change_abstractions_hist</whatIfTemplateId>
    <whatIfTemplateId>$MODELID$_zb_hist</whatIfTemplateId>
    <whatIfTemplateId>$MODELID$_n_stress_hist</whatIfTemplateId>
</whatIfTemplate>
```

Each whatIfTemplate in bold

Historic Scenario
Abstraction adjustments
Change abstraction rates (Historic)
Add new abstraction wells
Specify abstractions rates for new wells (Historic)
Zonebudget
Overrule zonebudget setting
Statistics
Overrule statistics setting
Difference to Naturalized scenario
Overrule differences setting
Number of stress periods from start (used for testing purposes)
Overrule number of stress periods (total=2844, timestep=semi-weekly, sta

Configuration: connecting to a workflow

A WhatIfTemplate is connected to a workflow in the WorkflowDescriptors.xml

```
<workflowDescriptor id="RunHistoric" name="Run Scenario" forecast="true" visible="true" autoApprove="false">
  <workflowFileName>RunModelWorkflow</workflowFileName>
  <runExpiryTime unit="hour" multiplier="6"/>
  <properties>
    <string key="REFTYPE" value="hist"/>
    <string key="ZBLEVEL" value="all"/>
    <string key="STATPROCLEVEL" value="All"/>
    <string key="DIFPROCLEVEL" value="All"/>
    <string key="N_STRESS_PERIOD" value="1000000"/>
  </properties>
  <timeOut unit="hour" multiplier="12"/>
  <whatIfTemplateId>$MODELID$_whatif_hist</whatIfTemplateId>
</workflowDescriptor>
```

reference to what-if template

Deltares

Configuration: modifierTypes.xsd

Modifier behavior:

- always apply in workflow (=default)
- only apply if referenced in what-if workflow
- never apply in what-if workflow

Specified using:

- whatIfModifierType (optional)

```
<timeSeriesModifier id="change_abstraction_rates_hist" name="Change abstraction rates (Historic)">
  <expiryTime unit="day" multiplier="3650"/>
  <expiryTimeDeletedModifiers unit="day" multiplier="365"/>
  <whatIfModifierType>a</whatIfModifierType>
  <timeSeries>
    <moduleInstanceId>apply if referenced in what-if workflow</moduleInstanceId>
    <parameterId>QS
    <locationSetId>$MODELID$_wells_all_shape</locationSetId>
    <timeSeriesType>simulated forecasting</timeSeriesType>
  </timeSeries>
  <defaultStartTime>time zero</defaultStartTime>
  <defaultEndTime>end run</defaultEndTime>
  <resolveInWorkflow>true</resolveInWorkflow>
  <resolveInPlots>false</resolveInPlots>
</timeSeriesModifier>
```

always apply in workflow
apply if referenced in what-if workflow
never apply in what-if workflows

Configuration: Explorer.xml

- Add the what if editor to your displays in the Explorer.xml

```
<explorerTask name="Whatlf">
    <predefinedDisplay>what if editor</predefinedDisplay>
    <toolbarTask>true</toolbarTask>
    <menubarTask>true</menubarTask>
    <loadAtStartup>true</loadAtStartup>
</explorerTask>
```

Data administration

What-if holds:

- id
- name
- property values

Modifier tables holds:

- modifier-whatif relation

Mod type	Name	Summary	...	Start	End	...	User	...	Cre...	Act...	Priority	De...	Copy	Scenario
Change abstraction rates...	abs +50%	Multiply 1.5	...	01-01-19...	01-04-20...	...	Peter...	...	28-...	☑		☒		demo
Change abstraction rates...	abs x2	Multiply 2.0	...	01-01-19...	02-01-20...	...	Peter...	...	19-...	☑		☒		abs x2

Thank you

- Available in Release 2022.01
  - We are finetuning the some GUI items

- Questions ?
- Opportunities ?
- Concerns ?

- Let’s try on your configuration

Let’s try it on your configuration

Make your own plan:

- Choose a **workflow** in your configuration for which you would like to make a scenario variant
  - e.g. a multiplier on precipitation
- Choose which choices you want to offer to the user:
  - as a fixed multiplier using a property
  - as a timeseries multiplier using a (typical profile) modifier

Let's try it on your configuration

Adjust your configuration:

- WhatIfTemplates.xml:
  - create a valueType for your property (most likely an double)
  - create a WhatIfTemplate where you include this valueType
- Workflowdescriptors.xml
  - add the default value of the property to your workflow
  - add the whatifTemplate reference
- Explorer.xml
  - Add the whatif editor as a display
- adjust your processing steps (Module Config Files)
  - use the property in your preprocessing of the precipitation

WhatIfTemplates.xml

```
<valueTypes>
  <double id="p_multiplier" min="0" default="1"/>
</valueTypes>
<whatIfTemplate id="whatif_p_multiplier" name="WhatIf Precipitation multiplier">
  <properties>
    <property id="P_MULTIPLIER" name="Precipitation multiplier" valueTypeId="p_multiplier"/>
  </properties>
</whatIfTemplate>
```

WorkflowDescriptors.xml

```
<workflowDescriptor id="Run_Hydro_model" name="Run my Hydrology model" forecast="true" visible="true" autoApprove="true">
  <properties>
    <double key="P_MULTIPLIER" value="1"/>
  </properties>
  <whatIfTemplateId>whatif_p_multiplier</whatIfTemplateId>
</workflowDescriptor>
```

Explorer.xml

```
<explorerTask name="WhatIf">
  <predefinedDisplay>what if editor</predefinedDisplay>
  <toolbarTask>true</toolbarTask>
  <menubarTask>true</menubarTask>
  <loadAtStartup>true</loadAtStartup>
</explorerTask>
```

Let's try it on your configuration

Adjust your configuration:

- Modifiertypes.xml
  - create a typicalProfileModifier (or SpatialModifier)
- WhatIfTemplates.xml:
  - create a WhatIfTemplate where you include this modifierType
- Workflowdescriptors.xml
  - add the whatifTemplate reference

ModifierTypes.xml

```
<typicalProfileModifier id="typical_profile_p_multiplier" name="P multiplier profile">
  <whatIfModifierType>apply if referenced in what-if workflow</whatIfModifierType>
  <timeSeries>
    <moduleInstanceId>preprocess_Hydrology_model</moduleInstanceId>
    <parameterId>P</parameterId>
    <qualifierId>multiplier</qualifierId>
  </timeSeries>
  <defaultStartTime>start run</defaultStartTime>
  <offsetDefaultStartTime unit="hour" multiplier="-1"/>
  <defaultEndTime>end run</defaultEndTime>
  <resolveInWorkflow>true</resolveInWorkflow>
  <resolveInPlots>false</resolveInPlots>
</typicalProfileModifier>
```

WhatIfTemplates.xml

```
<whatIfTemplate id="whatif_p_multiplier" name="WhatIf Precipitation multiplier">
  <modifierType>typical_profile_p_multiplier</modifierType>
</whatIfTemplate>
```

WorkflowDescriptors.xml

```
<workflowDescriptor id="Run_Hydro_model" name="Run my Hydrology model" forecast="true" visible="true" autoApprove="true">
  <whatIfTemplateId>whatif_p_multiplier</whatIfTemplateId>
</workflowDescriptor>
```

- Explorer.xml
  - Add the whatif editor as a display
- adjust your processing steps (Module Config Files)
  - use the timeseries in your preprocessing of the precipitation

Explorer.xml

```
<explorerTask name="WhatIf">
  <predefinedDisplay>what if editor</predefinedDisplay>
  <toolbarTask>true</toolbarTask>
  <menubarTask>true</menubarTask>
  <loadAtStartup>true</loadAtStartup>
</explorerTask>
```

Contact

www.deltares.nl

@deltares

linkedin.com/company/deltares

info@deltares.nl

@deltares

facebook.com/deltaresNL