

03 Analysis of a Configuration



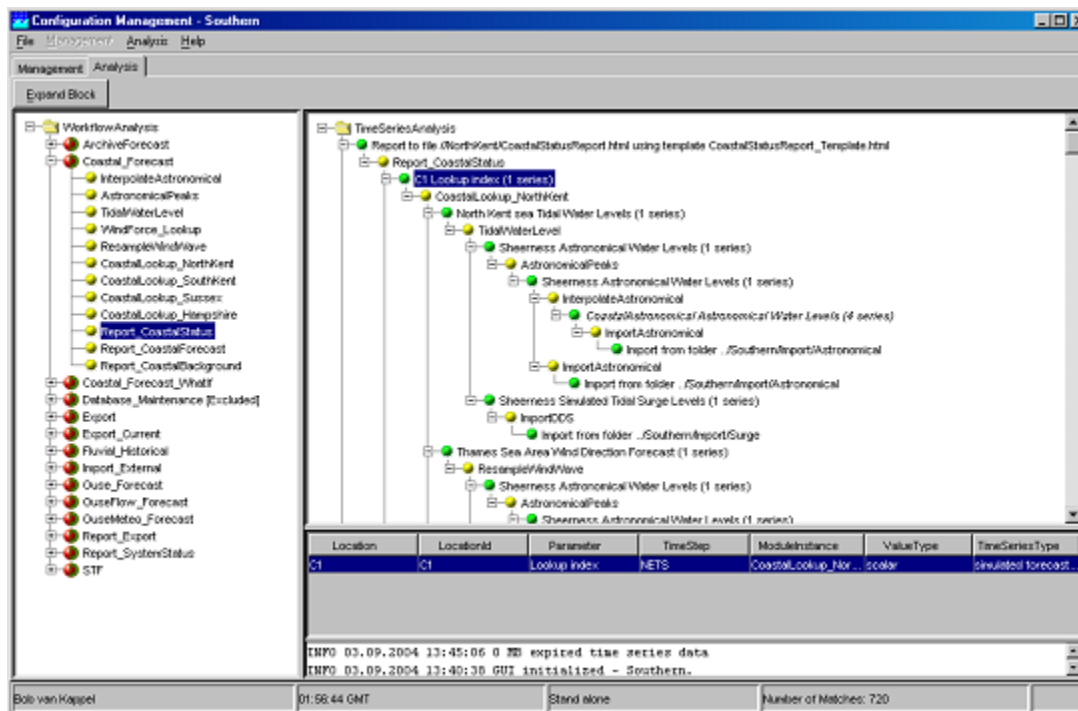
As of release 2007/01 the Analysis function is no longer available.

- Analysis of a Configuration
 - Approach
 - Implementation
 - Matching of timeseries
 - Handling of cyclic references
 - Handling of special cases
 - Using the Configuration Manager Analysis Tool

Analysis of a Configuration

Configuration analysis provides the means to in detail analyze a configuration. The analysis uses the dependencies between three configuration objects: workflows, Module Instances and timeSeriesSets to provide a visual overview of the configuration.

To analyze a configuration, activate the Analysis tab of the Configuration Manager.



Approach

The principle of the configuration analysis is quite simple. For each workflow in the configuration, all module instances that are used are shown in the order in which they are used. For a selected module instance then all timeseries that are created by that module instance are displayed as the top level timeseries.

For each of the timeseries in cascading order the following questions are answered:

- which module instance created this timeseries?
- which timeseries are required by that module instance?
- etc

This procedure is recursively followed through until for example an import module instance is encountered. Following this through allows broken links, unexpected starts or ends to be easily found in a configuration.

Implementation

When the Analysis mode is selected, a database of all timeseries is created. This involves that all module instances are analyzed for all the timeseries that are created and that are required within each module instance. For a selected module instance the input and output timeseries are matched in order to build the analysis tree.

The configuration files have been made such that a large amount of freedom is given to the user in setting up a configuration.

Matching of timeseries

Timeseries are matched on the following keys:

Key	Comment
locationId	A locationId is unique. A timeseries may also be identified using a locationSetId, which is a collection of locationId's. A locationSetId may consist of a large number of locationId's. When a locationSetId is used, the locationSetId is shown in the tree and the underlying timeseries are shown in a table.
parameterId	the parameterId is unique
timeseriesType	the timeSeriesType is unique
timeStep	the time step is unique
moduleInstanceId	a timeseries will in most cases be assigned a moduleInstanceId that is equal to the moduleInstance that creates it. However, this is not a rule. It may equally be possible that a time series is assigned a different moduleInstanceId. For each timeseries it is therefore required that both moduleInstanceId's are used to identify the timeseries: <ul style="list-style-type: none">- the moduleInstanceId of the module that creates the timeseries. This moduleInstanceId is the same as the configuration filename- the moduleInstanceId that is assigned to the timeseries when it is created

Handling of cyclic references

The configuration files allow a certain degree of cyclic referencing. A cyclic reference is caused when a module creates a timeseries that matches the input timeseries. In this case an infinite loop would be caused in the analysis processing. *Note that for the actual configuration a cyclic reference does not pose any problem whatsoever.*

For certain modules a cyclic reference is expected and must be handled explicitly. This is the case for the interpolation module and for the transformation module. In other cases when a cyclic reference is found no absolute inference can be made regarding the nature of the cyclic reference, and further analysis must be stopped.

Transformation module

A typical example of apparent cyclic referencing in the transformation module configuration is given with the following example, in which a time series is created through merging a number of time series where after in the same configuration the merged time series is checked to ensure that no values below a given value are found.

The handling of the time series is carried out in two transformation steps:

- Step 1: merge the inputs using data hierarchy. This results in a merged time series.
- Step 2: the merged time series is transformed using an arithmetic function, applied in two segments. The result is exactly the same as the time series in the step 1.

The above example illustrates that using the exact same timeseries as the resulting output timeseries in two separate transformation sets is allowed. The same could also be achieved through two different module instances, but would obviously lead to additional moduleInstanceId's.

Handling of this case is straightforward. When in a single transformation module multiple transformationSets are configured having the exact same timeseries as a result, it may be safely assumed that the last timeseries is the principal result timeseries. The other timeseries may be skipped in the analysis.

Interpolation module

A typical example of apparent cyclic references in the interpolation module is given by the following example, in which interpolation of a single timeseries is applied three times. Each interpolation however is configured in a different manner for a special reason. The input and output timeseries for each interpolation are however exactly the same.

- Step 1: Interpolation of the burn-in period (required to create a smooth start-up for a hydrodynamic model). This is a simple linear interpolation, over a fixed length in a fixed period. This period is usually longer than the period allowed in step 1
- Step 2: Ensure that any small gaps up to e.g. 2 hours, are interpolated using linear interpolation
- Step 3: Ensure that any remaining gaps are filled in with a default value. This interpolation is required to prevent unexpected crashing of the hydrodynamic model.

Handling of this case is straightforward. When in a single interpolation module multiple interpolationSets are configured having the exact same timeseries as a result, it may be safely assumed that the last timeseries is the principal result timeseries. The other timeseries may be skipped in the analysis.

Handling of special cases

There are a number of special cases, exceptions to a general rule, that must be handled correctly by the analysis.

General rule: each module requires an input and an output timeseries.

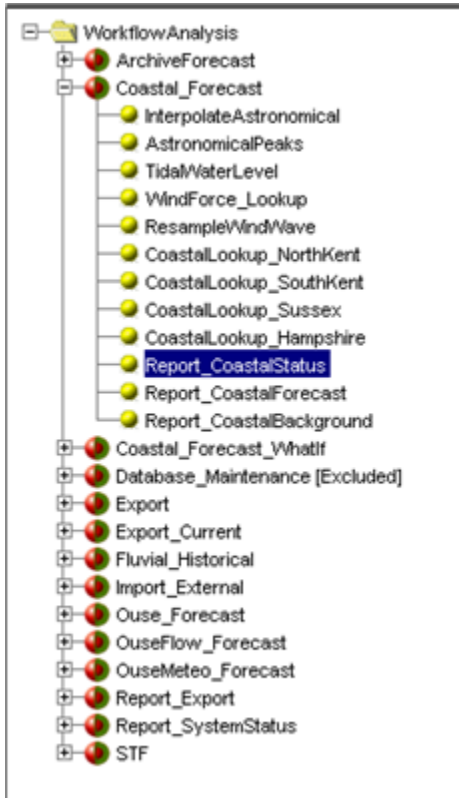
Special case: some module may obtain data from other sources or may not produce a timeseries. The special cases are given in the table below.

Module	Input to the module	Output from the module
importRun	import xml files from filesystem	

transformation	Optional: timeseries or profile data	
export		xml file to filesystem
general adapter	Optional: timeseries or import from filesystem in PI format	optional: timeseries or export to filesystem
interpolation	Input and output timeseries are always the same	

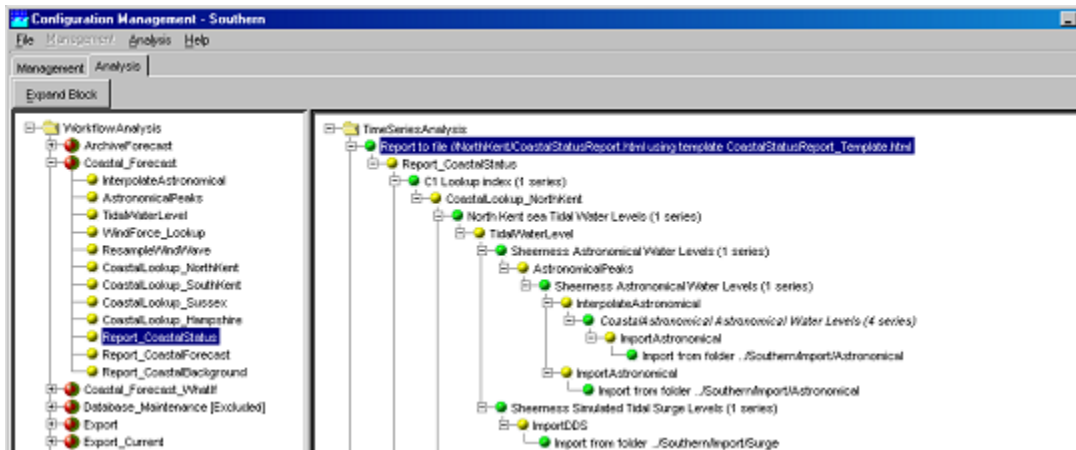
Using the Configuration Manager Analysis Tool

For each workflow in the configuration, all module instances that are used are shown in the order in which they are used. These are displayed in the left pane of the window:

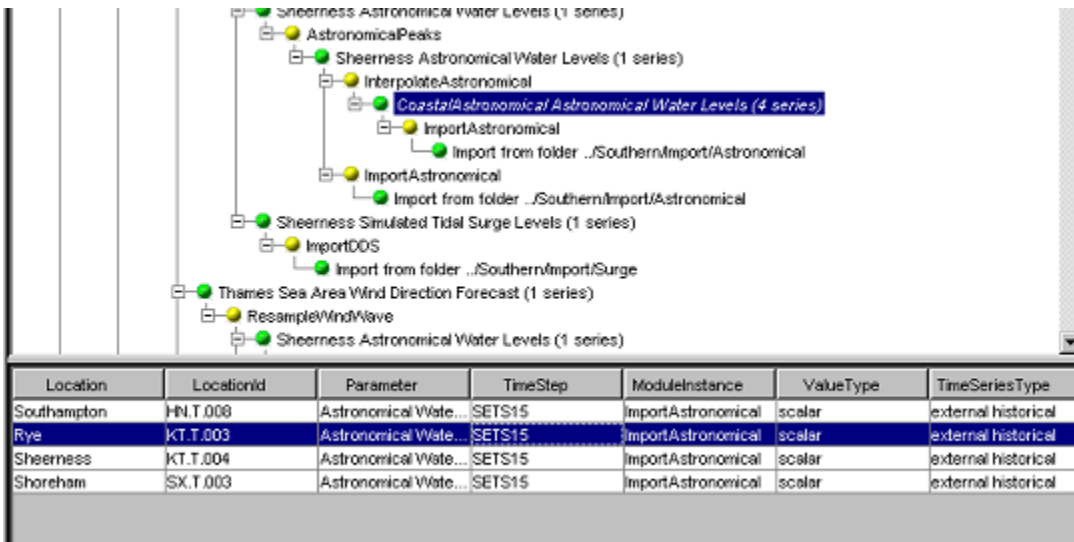


The Configuration Manager configuration allows certain workflows to be specifically excluded from the analysis. In some cases this is required if the workflow would create cyclic references that cannot be resolved internally. In the above example, this is the case for the database_Maintenance workflow.

For a selected module instance then all timeseries that are created by that module instance are displayed as the top level timeseries:



In the right pane, Module Instances are shown with a green ball icon, while time series sets are shown with a yellow icon. A time series set may consist of a single time series or possibly multiple time series. The details of the time series are shown in the table below the right hand pane:



In the above example the selected time series set consists of 4 time series.

Finding blind starts

Each result time series will be created through a module instance. This module instance in turn requires input data, either imported or created in the system. For each result time series, all module instances that are used to create it can be followed through, analyzing the input data that is used. The configuration manager will flag a blind start with an explicit error icon if a module instance input time series cannot be found.