

Extending Analysis Descriptions

An analysis itself can also be extended by use of the `nicsa.analysis.analysisDefinitionExtension` extension point. This extension point lets plugins add additional analysis description xml which will be added at runtime to existing analyses. For this to be useful, the base analysis java code must have been written to support extensibility. This `analysisDefinitionExtension` point provides no support for the analysis java code extensibility -- it only provides a mechanism for these analysis extension plugins to contribute to the base analysis' UI.

To use this, extend the `analysisDefinitionExtension` point, specifying the id of the base analysis, and pointing to a file that contains xml which will be dynamically added to the analysis description of the base analysis.

Notes and Tips

Visibility

Often these extensions are used in a case where the user can select one way of calculating something. The best way to handle this is to use a parameter to select which of these extensions to use, and then make the UI parameters for that extension only appear when that particular extension is selected.

As Map

Because the base analysis has no way of knowing what parameters all the extensions will provide, it is best to use the `as-map` attribute in the parameters so that they will be passed in as extra mapped parameters, not requiring setter methods on the analysis task.

Example

An example of how to use this mechanism is in `nicsa.maeviz.hazard`, the Scenario Earthquake analysis. The analysis can support multiple types of liquefaction, and defines an extension point and interface for new types of liquefaction to implement. In the UI, it provides a selector to choose one type:

```
<parameter group="Liquefaction" optional="true" phylum="string" format="extension-point" key="liquefaction-type" friendly-name="Liquefaction type" default="hazus-liquefaction" as-map="true">
  <description>Select method for computing liquefaction</description>
  <extension point="nicsa.maeviz.hazard.liquefaction"/>
  <autofill/>
</parameter>
```

For each type of liquefaction, the analysis description extension adds parameter fields based on what is necessary for that liquefaction type. Note that they use the `as-map` attribute, as well as specifying their visibility to only be shown when their own type of liquefaction has been selected:

For hazus-style liquefaction:

```
<analysis-description-extension>

  <parameter group="Liquefaction" optional="true" phylum="dataset" key="hazus-liquefaction.groundwatermap"
friendly-name="Groundwater map" as-map="true">
    <show-when parameter="liquefaction-type" value="hazus-liquefaction"/>
    <description>Ground water map for hazus liquefaction</description>
    <types>
      <type>groundWaterDepthMap</type>
    </types>
  </parameter>

  <parameter group="Liquefaction" optional="true" phylum="dataset" key="hazus-liquefaction.
liquefactionSusceptibilityMap" friendly-name="Liquefaction Susceptibility Map" as-map="true">
    <show-when parameter="liquefaction-type" value="hazus-liquefaction"/>
    <types>
      <type>liqSusceptibility</type>
    </types>
  </parameter>

</analysis-description-extension>
```