

# Workflow XML Schema

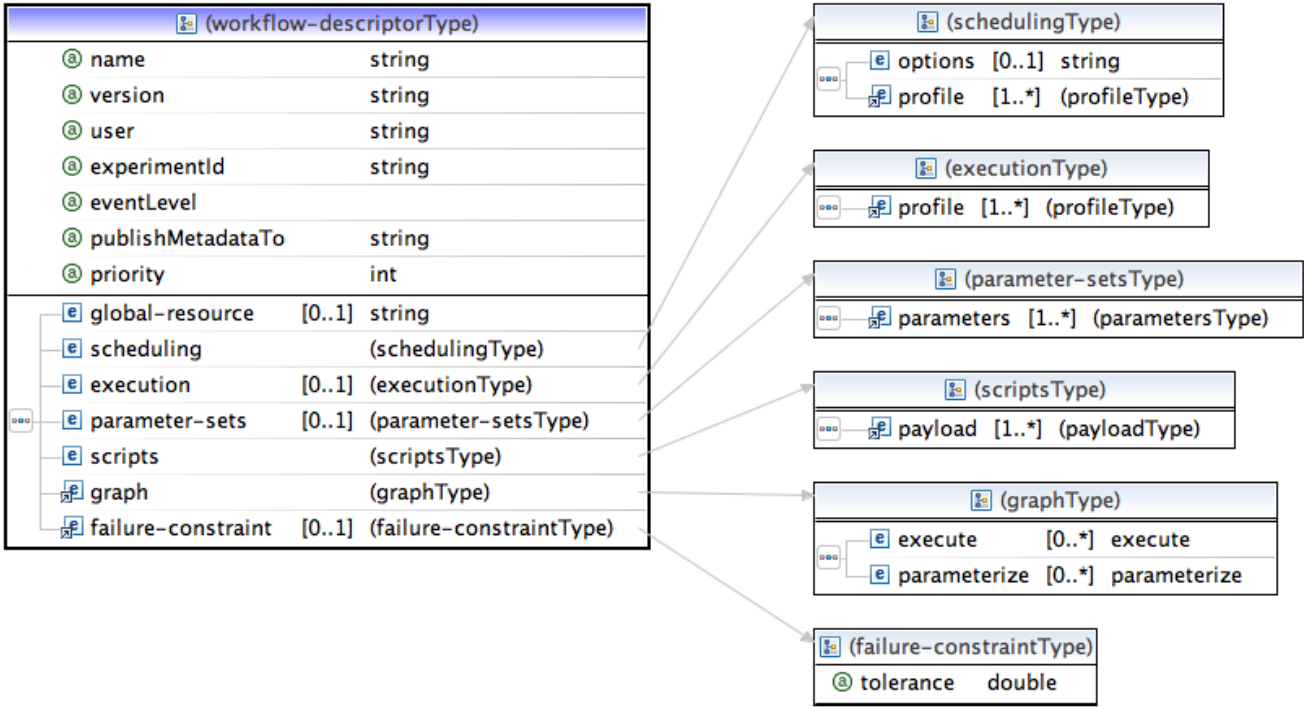
This page contains diagrammatic representations of the Workflow Description XML Schema, along with some examples.

The Annotated XSD is available for download from an attachment to this page: [workflow-descriptor.xsd](#)

NOTE: The actual object which is submitted to the broker is the [BrokerSubmission](#), a simple wrapper which tunnels the Workflow XML in serialized form.

## Schema Diagrams

### Workflow-Descriptor Type



Essentially the same as the Workflow object in the IBroker interface. Siege displays this XML to the user via an intermediate object, called the WorkflowBuilder, which allows the entire description, including the script payloads, to be represented as XML ([Elf/Ogrescript](#)), if appropriate, inside an Eclipse editor. When launching to the Broker, the builder wraps the payload (XML or otherwise) as CDATA and sets this string as the contents of the payload element.

eventLevel	Tells the event-sending components, such as services or <a href="#">Elf</a> , involved in handling the workflow, which level of information to publish as remote events on the bus (rather than or in addition to logging). The following list of options is in ascending order, such that successive levels include the preceding ones: ERROR, INFO, STATUS, PROGRESS, DEBUG.
publishMetadataTo	Indicates to a potential metadata agent that the events for this workflow should be processed for cataloguing. The value of the attribute should indicate the type of agent. If the attribute is missing, no metadata handling will take place.
priority	Expresses a global priority for this workflow. (Currently unused.)
global-resource	Scheduling can be overridden by setting this element. The named resource is applied to all nodes in the expanded graph, unless the individual node carries a different resource indication. If no resources are set, the resource is determined by a call to the scheduler (not yet fully implemented, but coming soon).
failure-constraint	See further below. These are global constraints applying to the entire workflow graph; each node can also have its own set of constraints. Default behavior is that a single node failure makes the entire workflow fail.

### Scheduling Type



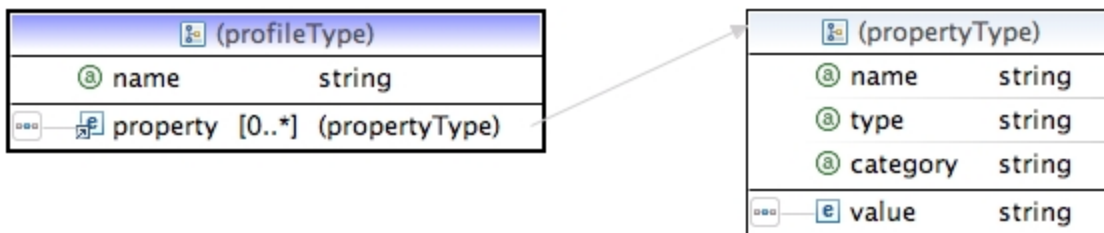
<b>options</b>	A comma-delimited list of possible strategy names for the scheduler to follow (e.g., "next-to-run" or "increase-priority") in the user's order of preference.
<b>profile</b>	Each of these elements represents a set of scheduling constraints expressed as properties (for instance, the typical parts of an RSL for those familiar with Globus submissions). Every executable node in the graph is associated with one or more of these sets.

## Execution Type



Optional profiles which group together runtime properties which may be required by one or more nodes in the execution graph.

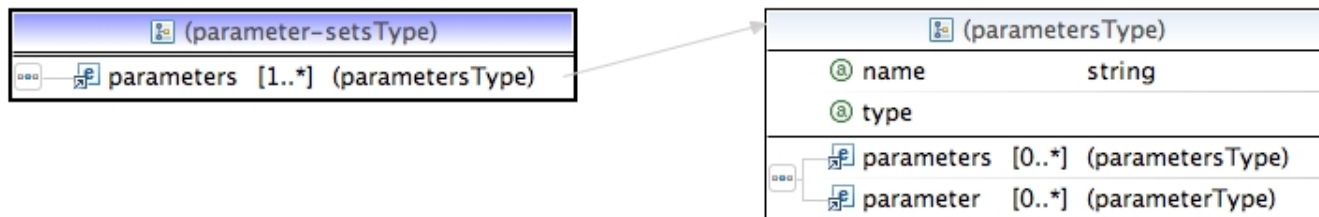
## Profile Type



## Property Type

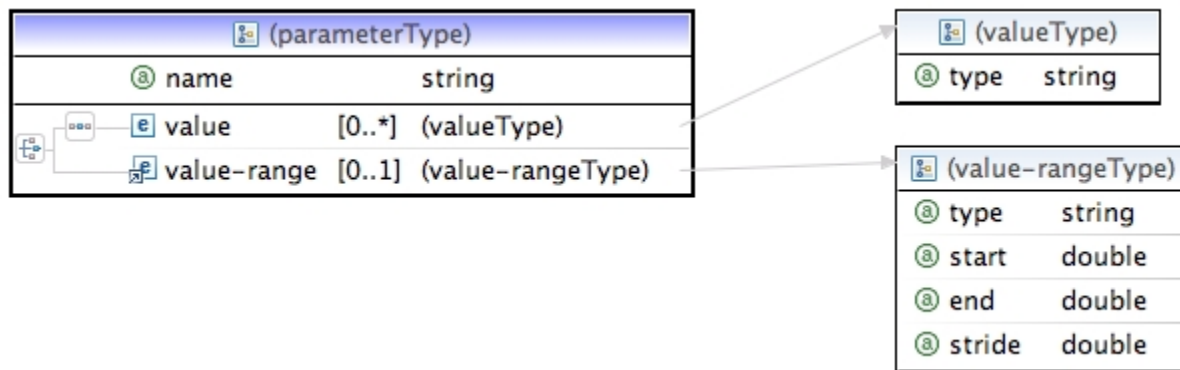
This object is used widely in our Java-based code; type denotes any of the serializable types which the Broker knows about, but in the current context, type is usually limited to strings or primitives. The value element can also be expressed as an attribute of the property element. Category is a catch-all attribute which allows us to tag the property for special purposes.

## Parameter-sets, Parameters and Parameter Type



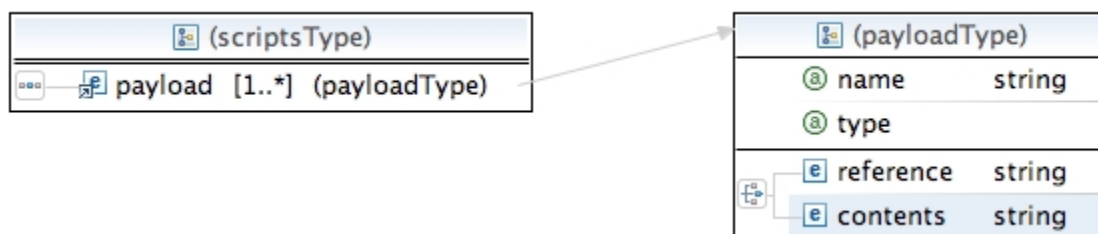
Optional named descriptions of how to parameterize nodes in the graph.

<b>parameters</b>	For this recursive object to be valid, it must bottom out in at least one simple (wfb:parameter) element; for an example, see below.
<b>type</b>	1. <i>product</i> signifies the Cartesian product obtained by crossing each of its members with the others. 2. <i>covariant</i> signifies the selection of one element from each of its members, in the order given. The "bin" represented by each member must have matching cardinality, or the parameterization will fail with an exception.



Simple parameter sets can be of two types: a list of typed values, or a single value-range description.

### Scripts and Payload Type

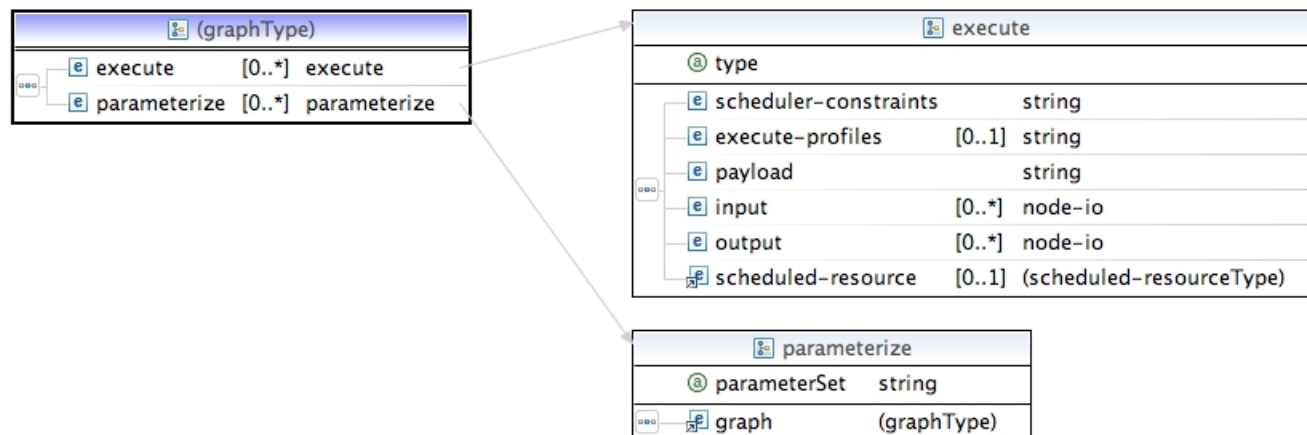


The list of actual executable scripts associated with this workflow, normalized out from the graph representation, since the relation between executable nodes and scripts can be N:1.

<b>type</b>	<i>elf</i> or <i>csh</i> ; but see below.
<b>reference</b>	A URI for a file to fetch and read in as the script content.
<b>contents</b>	The actual script content. As noted above, this is habitually <a href="#">Elf</a> container XML (holding <a href="#">Ogrescript</a> executable XML) - the only script for which there is full support in the Broker infrastructure; but in theory this could be any executable script (pending the addition of the appropriate modules to the Broker and/or ELF itself).

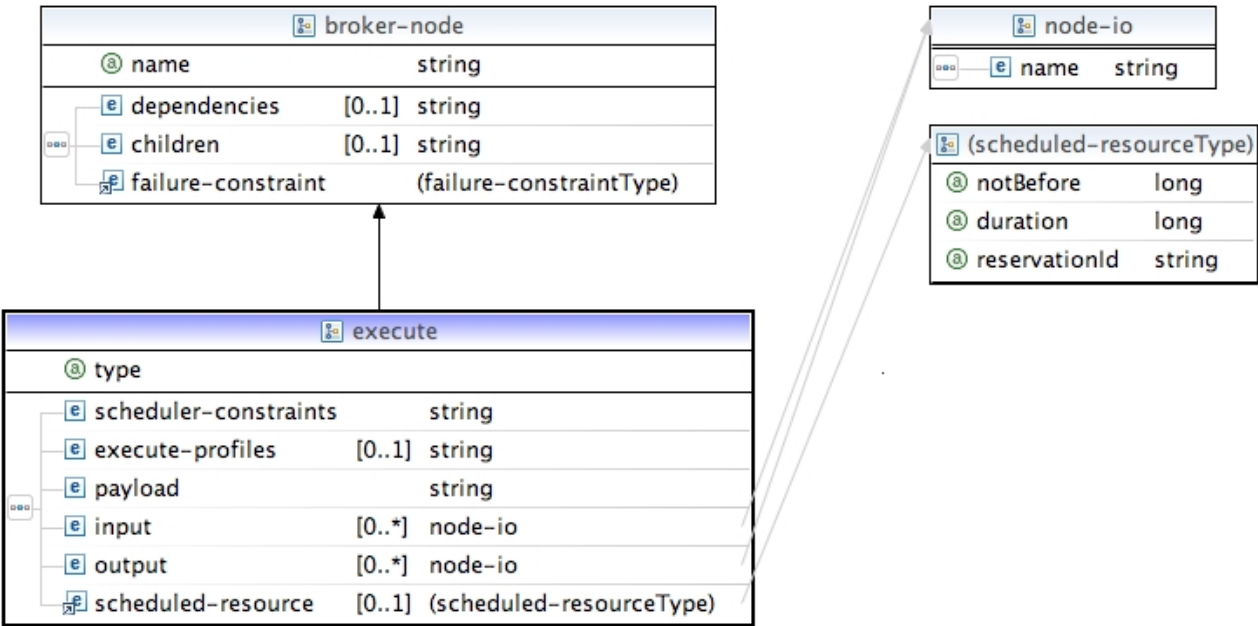
An implementation note: currently the NCSA group uses ELF/Ogrescript as the payload type, but the Broker does contain a module for launching raw C-shell scripts, with the proviso that this only be done using a host mapped to the GRAM protocol; otherwise, the Broker will have no way of knowing if the script completed successfully (unlike when running in the ELF container). In the future, we may provide for running such shell scripts through ELF (since the latter is easily extended).

### Graph Type



A directed, acyclical graph whose edges are untyped. In essence, this is a control-flow, not a data-flow, graph, but the (optional) use of the input/output elements on the execute nodes allows one to propagate data values (usually small values, not entire Fortran arrays!) downstream, establishing an implicit data-flow; see further [Node Input and Output](#). There should be at least one node in the graph.

Execute Type

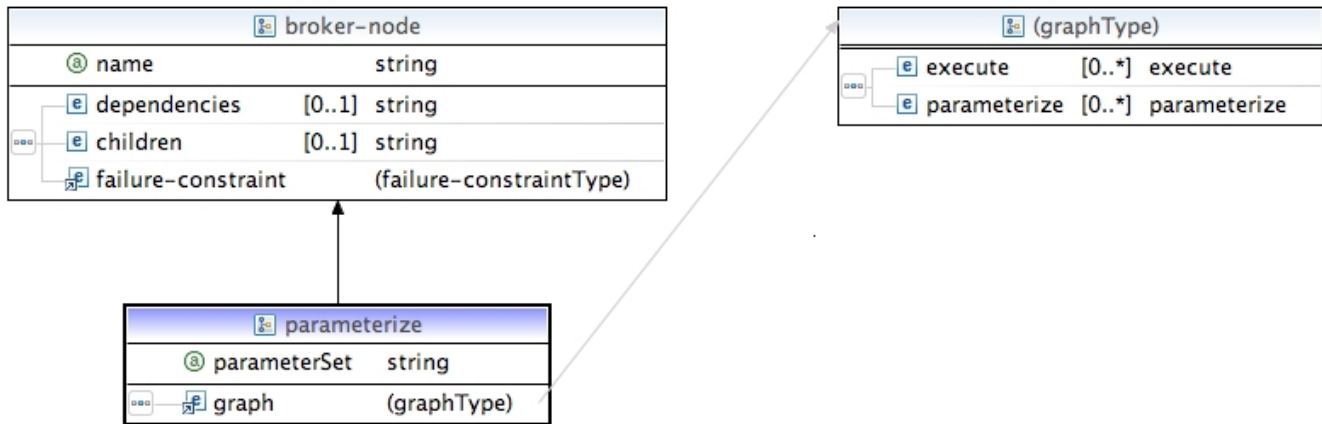


The principal type of node. It consists of a profile plus an executable script. Scheduler-constraints and execute-profiles are comma-delimited (ordered) lists of references to the workflow's respective map entries; these are merged (with successive properties overwriting previous identical ones) into a single profile for this particular node. Payload similarly references the workflow map.

Input and output also indicate properties that belong to a node's profile. If input elements are present, dependency nodes will be searched for corresponding properties, and these will also be merged into this node's execution profile. If output elements are indicated, the job status message which returns via the event bus from the ELF container will be searched for corresponding properties, and these will be added to the node's persistent profile so as to make them available to child nodes should they be required. In the case of parameterization, output values are suffixed with the parameterization index for the given member producing them; input values can specifically look for such suffixed values, or they can simply reference the prefix of the property's name, in which case all such values will be made available to the node as an array. Note that the return of output values is a feature of the ELF container; returned output is not possible when running raw shell scripts. The output name must correspond to a variable which remains in the global scope of the [Ogrescript environment](#) upon completion of its execution.

broker-node	Parent type for the two concrete node types. Dependencies and children are comma-delimited lists of names referencing other nodes in the graph.
type	1. <i>remote</i> means that the protocols appropriate to the remote resource will be used to stage in and execute the script. 2. <i>local</i> means that the script will be run on the resource local to the Broker in temporary space. In either case, the job is always run as the user; hence if the user is not in the gridmap file on the Broker's resource, the job cannot run locally (on the other hand, the local gridmap file is usually kept up to date with the rest of the NCSA resources).
failure-constraint	See below.
scheduled-resource	Contents is the resource name on which to run (note that this name must be mapped in one of our supporting services: the <a href="#">Host Information Service</a> ). If not explicitly set, the resource is obtained through scheduling algorithms, or from the "global resource", if the latter is set on the workflow.

Parameterize Type



The indicated parameter set (referencing the workflow's parameter-set map) will be used to expand the node by applying the parameters to the entire sub-graph. The latter is identical in type to the principal graph of the workflow; hence parameterization can be to an arbitrary depth.

An implementation note: parameterized nodes are only expanded when they are ready to run. Thus if a graph contained an execute node (e0) followed by a parameterized node (p0), and the latter also had a graph containing an execute node (e1) followed by another parameterized node (p1), (p0) would only get expanded after (e0) ran, and for each member j produced by (p0), each (p1j) would get expanded only after its corresponding (e1j) had run.

### Failure Constraint Type

The contents of this element, if expressed, is a comma-delimited list of dependencies which must successfully complete for this node to run. The default behavior is that all of its dependencies must be satisfied. An alternative to expressing these constraints is to set the "tolerance" attribute; this indicates the percentage of dependencies for which failure will be allowed for this node to run. Default = 0.0. The latter is useful for programming a node which acts as a barrier on a large group of parameterized nodes, all results of which may not be necessary for the node to execute or for valid results to be obtained from the workflow. When expressed on a workflow, these constraints are applied to its leaf (that is, final, or childless) nodes.

Example: Simple single-node workflow

```

<workflow-builder name="test-singleton" experimentId="SingleNodeTest" user="arossi" eventLevel="DEBUG">
  <global-resource>tun.test</global-resource>
  <scheduling>
    <profile name="all">
      <property name="ELF_HOME" category="environment">
        <value>/u/ncsa/arossi/elf-latest</value>
      </property>
      <property name="count">
        <value>1</value>
      </property>
      <property name="jobType">
        <value>single</value>
      </property>
      <property name="maxWallTime">
        <value>2</value>
      </property>
      <property name="project">
        <value>adn</value>
      </property>
      <property name="queue">
        <value>debug</value>
      </property>
      <property name="submissionType">
        <value>batch</value>
      </property>
    </profile>
  </scheduling>
  <execution>
    <profile name="TOUCH">
      <property name="name">
        <value>>TOUCH</value>
      </property>
    </profile>
  </execution>
  <graph>
    <execute name="preprocess" type="remote">
      <scheduler-constraints>all</scheduler-constraints>
      <execute-profiles>TOUCH</execute-profiles>
      <payload>touch</payload>
    </execute>
  </graph>
  <scripts>
    <payload name="touch" type="elf">
      <elf>
        <workdir />
        <serial-scripts>
          <ogrescript name="touch-script">
            <touch uri="file:${runtime.dir}/${name}" />
          </ogrescript>
        </serial-scripts>
      </elf>
    </payload>
  </scripts>
</workflow-builder>

```

#### Example: Multiple node workflow

```

<workflow-builder name="Triggered-ARPS-WRF" user="arossi" eventLevel="DEBUG" experimentId="FULL_TEST">
  <global-resource>tund.ncsa.uiuc.edu</global-resource>
  <scheduling>
    <profile name="interactive">
      <property name="ELF_HOME" category="environment">
        <value>/u/ncsa/arossi/elf-0.2.8</value>
      </property>
      <property name="submissionType"><value>interactive</value></property>
    </profile>
  </scheduling>

```

```

</profile>
<profile name="batch-single">
  <property name="ELF_HOME" category="environment">
    <value>/u/ncsa/arossi/elf-0.2.8</value>
  </property>
  <property name="project"><value>adn</value></property>
  <property name="jobType"><value>single</value></property>
  <property name="submissionType"><value>batch</value></property>
  <property name="maxMemory"><value>1024</value></property>
  <property name="maxWallTime"><value>15</value></property>
  <property name="count"><value>1</value></property>
  <property name="stderr"><value>stderr_%J</value></property>
  <property name="stdout"><value>stdout_%J</value></property>
</profile>
<profile name="batch-mpi">
  <property name="ELF_HOME" category="environment">
    <value>/u/ncsa/arossi/elf-0.2.8</value>
  </property>
  <property name="project"><value>adn</value></property>
  <property name="jobType"><value>single</value></property>
  <property name="submissionType"><value>batch</value></property>
  <property name="maxMemory"><value>1024</value></property>
  <property name="maxWallTime"><value>30</value></property>
  <property name="count"><value>16</value></property>
  <property name="stderr"><value>stderr_%J</value></property>
  <property name="stdout"><value>stdout_%J</value></property>
</profile>
</scheduling>
<execution>
  <profile name="TRIGGER_DATA">
    <!-- ***** will be set by trigger script ***** -->
    <property name="CENTER_LAT"><value>38.0</value></property>
    <property name="CENTER_LON"><value>-98.0</value></property>
    <property name="PRODUCT_NUMBER"><value>0</value></property>
    <!-- ***** ***** -->
    <property name="FORECAST_LENGTH"><value>6</value></property>
  </profile>
  <profile name="SOURCE_URIS">
    <!-- mirror
    <property name="NAM_URI"><value>gridftp://lead1.unidata.ucar.edu/gridftp/LEAD/model/NCEP/LEADNAM<
/value></property>
    <property name="ADAS_URI"><value>gridftp://lead1.unidata.ucar.edu/data/pub/other/lead/ADAS/10km<
/value></property>
    -->
    <property name="NAM_URI"><value>gridftp://chinkapin.cs.indiana.edu/data/ldm/pub/native/grid/NCEP
/LEADNAM</value></property>
    <property name="ADAS_URI"><value>gridftp://chinkapin.cs.indiana.edu//data/ldm/pub/other/lead/ADAS
/10km</value></property>
    <property name="TCP_BUF"><value>1048576</value></property>
  </profile>
  <profile name="LOCAL_PATHS">
    <property name="RUN_ID"><value>CAPS-LEAD-${STRT_YMDH}-T${PRODUCT_NUMBER}</value></property>
    <property name="CAPS_HOME"><value>/cfs/projects/wrf</value></property>
    <property name="CAPS_BIN"><value>${CAPS_HOME}/bin</value></property>
    <property name="CAPS_DATA"><value>${CAPS_HOME}/data</value></property>
    <property name="CAPS_INPUT"><value>${CAPS_HOME}/input/2km</value></property>
    <property name="BASE_DIR"><value>/cfs/scratch/users/arossi/${RUN_ID}</value></property>
    <property name="DATA_DIR"><value>${BASE_DIR}/extm</value></property>
    <property name="OUTPUT_DIR"><value>${BASE_DIR}/output</value></property>
    <property name="MPI"><value>${CAPS_BIN}/cmpi.csh</value></property>
    <property name="BROKER_URI"><value>http://tbl.ncsa.uiuc.edu:8043/broker/services/Broker</value><
/property>
  </profile>
  <profile name="ARCHIVE_URIS">
    <property name="ARCHIVE_URI"><value>mssftp://mss.ncsa.uiuc.edu/u/ac/arossi/CAPS-LEAD/${STRT_YMDH}-
T${PRODUCT_NUMBER}</value></property>
    <property name="TCP_BUF"><value>2097152</value></property>
  </profile>
</execution>
<scripts>
  <payload name="STAGE_IN" type="elf">

```

```

<elf>
  <serial-scripts separate-script-dirs="false">
    <ogrescript name="STAGE_IN">
      <echo message="Computing latest GMT data cycle time"/>
      <declare name="STRT_Y" string="2007"/>
      <declare name="STRT_M" string="02"/>
      <declare name="STRT_D" string="26"/>
      <declare name="STRT_H" string="12"/>
      <future-time offset="0" hourInterval="${FORECAST_LENGTH}">
        <return-value assignedName="STRT_Y" defaultName="future-time-year"/>
        <return-value assignedName="STRT_M" defaultName="future-time-month"/>
        <return-value assignedName="STRT_D" defaultName="future-time-date"/>
        <return-value assignedName="STRT_H" defaultName="future-time-hour"/>
      </future-time>
      <declare name="STRT_YMDH" string="${STRT_Y}${STRT_M}${STRT_D}${STRT_H}"/>
      <declare name="STRT_DATE" string="${STRT_Y}-${STRT_M}-${STRT_D}_${STRT_H}:00:00"/>
      <declare name="INIT_TIME" string="${STRT_Y}-${STRT_M}-${STRT_D}.${STRT_H}:00:00"/>
      <echo stdout="true" message="Init time: ${STRT_YMDH}"/>
      <echo message="Setting up directory structure." />
      <mkdir uri="file:${BASE_DIR}"/>
      <mkdir uri="file:${DATA_DIR}"/>
      <echo message="Fetching NAM data for ${STRT_DATE}" />
      <copy target="file:${DATA_DIR}">
        <configuration>
          <property name="tcpBufferSize" value="${TCP_BUF}" type="int"/>
        </configuration>
        <source base="${NAM_URI}/${STRT_YMDH}">
          <include>eta40grb.*f00</include>
          <include>eta40grb.*f03</include>
          <include>eta40grb.*f06</include>
        </source>
      </copy>
      <echo message="Fetching ADAS data for ${STRT_DATE}" />
      <copy target="file:${DATA_DIR}">
        <configuration>
          <property name="tcpBufferSize" value="${TCP_BUF}" type="int"/>
        </configuration>
        <source base="${ADAS_URI}">
          <include>*${STRT_YMDH}.net000000</include>
          <include>*${STRT_YMDH}.netgrdbas</include>
        </source>
      </copy>
      <echo message="Validating data for ${STRT_DATE}"/>
      <if>
        <or>
          <not><is-file path="${DATA_DIR}/eta40grb.${STRT_YMDH}f06"/></not>
          <not><is-file path="${DATA_DIR}/ad${STRT_YMDH}.net000000"/></not>
          <not><is-file path="${DATA_DIR}/ad${STRT_YMDH}.netgrdbas"/></not>
        </or>
        <echo message="data for ${STRT_DATE} is incomplete; trying previous 6-hour interval"/>
        <delete dir="file:${BASE_DIR}"/>
        <future-time offset="-6" hourInterval="${FORECAST_LENGTH}">
          <return-value assignedName="STRT_Y" defaultName="future-time-year"/>
          <return-value assignedName="STRT_M" defaultName="future-time-month"/>
          <return-value assignedName="STRT_D" defaultName="future-time-date"/>
          <return-value assignedName="STRT_H" defaultName="future-time-hour"/>
        </future-time>
        <assign name="STRT_YMDH" string="${STRT_Y}${STRT_M}${STRT_D}${STRT_H}"/>
        <assign name="STRT_DATE" string="${STRT_Y}-${STRT_M}-${STRT_D}_${STRT_H}:00:00"/>
        <assign name="INIT_TIME" string="${STRT_Y}-${STRT_M}-${STRT_D}.${STRT_H}:00:00"/>
        <echo message="Fetching NAM data for ${STRT_DATE}" />
        <mkdir uri="file:${BASE_DIR}"/>
        <mkdir uri="file:${DATA_DIR}"/>
        <copy target="file:${DATA_DIR}">
          <configuration>
            <property name="tcpBufferSize" value="${TCP_BUF}" type="int"/>
          </configuration>
          <source base="${NAM_URI}/${STRT_YMDH}">
            <include>eta40grb.*f00</include>
            <include>eta40grb.*f03</include>
            <include>eta40grb.*f06</include>
          </source>
        </copy>
      </if>
    </ogrescript>
  </serial-scripts>

```



```

        </source>
    </copy>
    <echo message="Fetching ADAS data for ${STRT_DATE}" />
    <copy target="file:${DATA_DIR}">
        <configuration>
            <property name="tcpBufferSize" value="${TCP_BUF}" type="int"/>
        </configuration>
        <source base="${ADAS_URI}">
            <include>*${STRT_YMDH}.net000000</include>
            <include>*${STRT_YMDH}.netgrdbas</include>
        </source>
    </copy>
    <echo message="Validating data for ${STRT_DATE}" />
    <if>
        <or>
            <not><is-file path="${DATA_DIR}/eta40grb.${STRT_YMDH}f06"/></not>
            <not><is-file path="${DATA_DIR}/ad${STRT_YMDH}.net000000"/></not>
            <not><is-file path="${DATA_DIR}/ad${STRT_YMDH}.netgrdbas"/></not>
        </or>
        <throw>external data incomplete for start ${STRT_DATE}; cannot run analysis</throw>
    </if>
    </if>
    <mkdir uri="file:${OUTPUT_DIR}" />
    </ogrescript>
</serial-scripts>
</elf>
</payload>
<payload name="ARPSTRN" type="elf">
    <elf>
        <serial-scripts separate-script-dirs="false">
            <ogrescript name="ARPSTRN">
                <declare name="runname" string="lead20_${STRT_YMDH}" />
                <declare name="input-file" string="${BASE_DIR}/${runname}.trnin" />
                <echo message="Running arsptrn" />
                <simple-process execution-dir="${OUTPUT_DIR}"
                    inFile="${input-file}" outFile="${OUTPUT_DIR}/arpstrn.out">
                    <command-line>${CAPS_BIN}/arpstrn</command-line>
                </simple-process>
            </ogrescript>
        </serial-scripts>
    </elf>
</payload>
<payload name="WRFSTATIC" type="elf">
    <elf>
        <serial-scripts separate-script-dirs="false">
            <ogrescript name="WRFSTATIC">
                <declare name="input-file" string="${BASE_DIR}/ar${STRT_YMDH}.staticin" />
                <echo message="Running wrfstatic" />
                <simple-process execution-dir="${OUTPUT_DIR}"
                    inFile="${input-file}" outFile="${OUTPUT_DIR}/wrfstatic.out">
                    <command-line>${CAPS_BIN}/wrfstatic</command-line>
                </simple-process>
            </ogrescript>
        </serial-scripts>
    </elf>
</payload>
<payload name="EXT2ARPS_LBC" type="elf">
    <elf>
        <serial-scripts separate-script-dirs="false">
            <ogrescript name="EXT2ARPS_LBC">
                <declare name="runname" string="em${STRT_YMDH}" />
                <declare name="input-file" string="${BASE_DIR}/${runname}-lbc.extin" />
                <echo message="Running ext2arps-lbc" />
                <simple-process execution-dir="${OUTPUT_DIR}"
                    <command-line>${MPI} ${count} ${CAPS_BIN}/ext2arps_mpi ${input-file} ${OUTPUT_DIR}
                    /ext2arps-lbc.out</command-line>
                </simple-process>
            </ogrescript>
        </serial-scripts>
    </elf>
</payload>

```

```

<payload name="EXT2ARPS_IN" type="elf">
  <elf>
    <serial-scripts separate-script-dirs="false">
      <ogrescript name="EXT2ARPS_IN">
        <declare name="runname" string="ei${STRT_YMDH}"/>
        <declare name="input-file" string="${BASE_DIR}/${runname}-t0.extin"/>
        <echo message="Running ext2arps-t0"/>
        <simple-process execution-dir="${OUTPUT_DIR}">
          <command-line>${MPI} ${count} ${CAPS_BIN}/ext2arps_mpi ${input-file} ${OUTPUT_DIR}
/ext2arps-t0.out</command-line>
        </simple-process>
      </ogrescript>
    </serial-scripts>
  </elf>
</payload>
<payload name="ARPS2WRF" type="elf">
  <elf>
    <serial-scripts separate-script-dirs="false">
      <ogrescript name="ARPS2WRF">
        <declare name="input-file" string="${BASE_DIR}/ar${STRT_YMDH}.arps2wrfin"/>
        <simple-process execution-dir="${OUTPUT_DIR}">
          <command-line>${MPI} ${count} ${CAPS_BIN}/arps2wrf_mpi ${input-file} ${OUTPUT_DIR}
/arps2wrf.out</command-line>
        </simple-process>
      </ogrescript>
    </serial-scripts>
  </elf>
</payload>
<payload name="WRF" type="elf">
  <elf>
    <serial-scripts separate-script-dirs="false">
      <ogrescript name="WRF">
        <file-monitor-process execution-dir="${OUTPUT_DIR}" outFile="${OUTPUT_DIR}/wrf.out">
          <command-line>${MPI} ${count} ${CAPS_BIN}/wrf.exe ${OUTPUT_DIR}/wrf.out</command-line>
          <file-monitor>
            <pattern base="file:${OUTPUT_DIR}">
              <include>wrfout_d0*</include>
            </pattern>
            <new-file-trigger>
              <filter pattern="wrfout_d0.*"/>
              <actions>
                <file-progress-action topic="TROLL"/>
              </actions>
            </new-file-trigger>
          </file-monitor>
        </file-monitor-process>
      </ogrescript>
    </serial-scripts>
  </elf>
</payload>
<payload name="WRF_POST" type="elf">
  <elf>
    <serial-scripts separate-script-dirs="false">
      <ogrescript name="WRF_POST">
        <mkdir uri="file:${BASE_DIR}/plot"/>
        <declare name="wrf.input" />
        <read-in-namelist file="${OUTPUT_DIR}/namelist.input">
          <entry>time_control.run_hours</entry>
          <entry>time_control.history_interval</entry>
        </read-in-namelist>
        <declare name="start.date"/>
        <declare name="last_hour" long="$E{( ${time_control.run_hours} * 60 / ${time_control.
history_interval} )}" castTo="int"/>
        <declare name="total" long="$E{${last_hour} + 1}" castTo="int"/>
        <date-format inputString="${STRT_YMDH}" format="yyyyMMddHH">
          <return-value assignedName="start.date" defaultName="date" />
        </date-format>
        <declare name="workflowIds">
          <map/>
        </declare>
        <echo message="groupId ${groupId}"/>
      </ogrescript>
    </serial-scripts>
  </elf>
</payload>

```

```

<subscribe subscriberId="statusEvents">
  <filter topic="TROLL">
    <event-header-comparator>
      <property-comparator comparator="EQUALS">
        <property name="groupId" value="{groupId}" />
      </property-comparator>
      <property-comparator>
        <property name="workflow" value="NOT_NULL" />
      </property-comparator>
      <property-comparator>
        <property name="node" value="NULL" />
      </property-comparator>
      <property-comparator comparator="EQUALS">
        <property name="EVENT_CLASS"
          value="ncsa.tools.events.types.events.StatusEvent" />
      </property-comparator>
    </event-header-comparator>
  </filter>
</subscribe>
<for var="i" from="0" to="{last_hour}">
  <declare name="curr.date" />
  <date-math date="{start.date}" field="hour" add="{i}">
    <return-value assignedName="curr.date" defaultName="date" />
  </date-math>
  <parse-date name="curr.date" date="{curr.date}" declare="false" />
  <declare name="year" string="{curr.date$I{0}}" />
  <declare name="month" string="{curr.date$I{1}}" />
  <declare name="day" string="{curr.date$I{2}}" />
  <declare name="hour" string="{curr.date$I{3}}" />
  <declare name="hour-incr-secs" />
  <add-leading-zeros number="{E{$i} * 3600}" places="6">
    <return-value defaultName="zerosAdded" assignedName="hour-incr-secs" />
  </add-leading-zeros>
  <echo message="waiting for wrf output for {year}-{month}-{day}_{hour}" />
  <while>
    <not><is-file path="{OUTPUT_DIR}/wrfout_d01_{year}-{month}-{day}_{hour}:00:
00_ready" /></not>

    <sleep timeout="15000" />
  </while>
  <declare name="pid" string="{year}-{month}-{day}_{hour}" />
  <declare name="handleId" />
  <echo message="submitting arpsplt workflow for wrfout_d01_{year}-{month}-{day}_{hour}"
/>

  <submit assign="handleId"
    experimentId="{groupId}" serviceUri="{BROKER_URI}" workflowName="Triggered-
ARPSPLT.{pid}"

    workflowBuilder="{CAPS_INPUT}/arpsplt.xml">
    <properties>
      <property name="PRODUCT_NUMBER"><value>{PRODUCT_NUMBER}</value></property>
      <property name="STRT_YMDH"><value>{STRT_YMDH}</value></property>
      <property name="STRT_DATE"><value>{STRT_DATE}</value></property>
      <property name="PLOT_DIR"><value>{BASE_DIR}/plot/h_{i}</value></property>
      <property name="PLOT_0_DIR"><value>{BASE_DIR}/plot/h_0</value></property>
      <property name="PLOT_ID"><value>{pid}</value></property>
      <property name="SECS_INCR"><value>{hour-incr-secs}</value></property>
    </properties>
  </submit>
  <echo message="submitted {pid}, got handle: {handleId}" />
</for>
<echo message="waiting for all postprocessing workflows to complete ..." />
<while condition="{E{true}}">
  <declare name="nextEvent" />
  <next-event subscriberId="statusEvents">
    <return-value defaultName="nextEvent" />
  </next-event>
  <if>
    <is-null object="{nextEvent}" />
    <continue />
  </if>
  <declare name="header" />
  <declare name="wfId" />

```

```

        <declare name="status"/>
        <get bean="{nextEvent}" property="header-properties" assign="header"/>
        <get bean="{header$I{workflow}}" property="value" assign="wfId"/>
        <get bean="{nextEvent}" property="status" assign="status"/>
        <if>
            <or>
                <equals first="{status}" second="DONE"/>
                <equals first="{status}" second="FAILED"/>
                <equals first="{status}" second="CANCELLED"/>
            </or>
            <put key="{wfId}" map="{workflowIds}"/>
            <echo message="STATUS: {wfId} {status}; total received: {workflowIds$I*}"/>
            <if>
                <equals firstInt="{workflowIds$I*}" secondInt="{total}"/>
                <echo message="last event; exiting ..."/>
                <break/>
            </if>
        </if>
    </while>
    <unsubscribe subscriberId="statusEvents"/>
</ogrescript>
</serial-scripts>
</elf>
</payload>
<payload name="STAGE_OUT" type="elf">
    <elf>
        <serial-scripts separate-script-dirs="false">
            <ogrescript name="STAGE_OUT">
                <echo message="Archiving data to mss"/>
                <mkdir uri="{ARCHIVE_URI}"/>
                <copy target="{ARCHIVE_URI}">
                    <configuration>
                        <property name="list-recursive" value="true" type="boolean"/>
                        <property name="transferMode" value="gridftp-stream"/>
                        <property name="target-active" value="true" type="boolean"/>
                        <property name="tcpBufferSize" value="{TCP_BUF}" type="int"/>
                    </configuration>
                    <source base="file:{BASE_DIR}">
                        <include>output/**</include>
                        <include>plot/**</include>
                    </source>
                </copy>
                <ftp-chmod recursive="true" permissions="766"
                    dir="{ARCHIVE_URI}"/>
            </ogrescript>
        </serial-scripts>
    </elf>
</payload>
</scripts>
<graph>
    <execute name="STAGE_IN" type="remote">
        <children>ARPSTRN</children>
        <scheduler-constraints>interactive</scheduler-constraints>
        <execute-profiles>TRIGGER_DATA,SOURCE_URIS,LOCAL_PATHS</execute-profiles>
        <payload>STAGE_IN</payload>
        <output>
            <name>STRT_YMDH</name>
            <name>STRT_DATE</name>
            <name>INIT_TIME</name>
        </output>
    </execute>
    <execute name="ARPSTRN" type="remote">
        <dependencies>STAGE_IN</dependencies>
        <children>EXT2ARPS_IN,EXT2ARPS_LBC,WRFSTATIC</children>
        <scheduler-constraints>batch-single</scheduler-constraints>
        <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
        <payload>ARPSTRN</payload>
        <input>
            <name>STRT_YMDH</name>
            <name>STRT_DATE</name>
            <name>INIT_TIME</name>

```

```

    </input>
  <output>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </output>
</execute>
<execute name="EXT2ARPS_IN" type="remote">
  <dependencies>ARPSTRN</dependencies>
  <children>ARPS2WRF</children>
  <scheduler-constraints>batch-mpi</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
  <payload>EXT2ARPS_IN</payload>
  <input>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </input>
</execute>
<execute name="EXT2ARPS_LBC" type="remote">
  <dependencies>ARPSTRN</dependencies>
  <children>ARPS2WRF</children>
  <scheduler-constraints>batch-mpi</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
  <payload>EXT2ARPS_LBC</payload>
  <input>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </input>
</execute>
<execute name="WRFSTATIC" type="remote">
  <dependencies>ARPSTRN</dependencies>
  <children>ARPS2WRF</children>
  <scheduler-constraints>batch-single</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
  <payload>WRFSTATIC</payload>
  <input>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </input>
  <output>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </output>
</execute>
<execute name="ARPS2WRF" type="remote">
  <dependencies>EXT2ARPS_IN,EXT2ARPS_LBC,WRFSTATIC</dependencies>
  <children>WRF,WRF_POST</children>
  <scheduler-constraints>batch-mpi</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
  <payload>ARPS2WRF</payload>
  <input>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </input>
  <output>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </output>
</execute>
<execute name="WRF" type="remote">
  <dependencies>ARPS2WRF</dependencies>
  <children>STAGE_OUT</children>
  <scheduler-constraints>batch-mpi</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
  <payload>WRF</payload>

```

```

<input>
  <name>STRT_YMDH</name>
  <name>STRT_DATE</name>
  <name>INIT_TIME</name>
</input>
<output>
  <name>STRT_YMDH</name>
  <name>STRT_DATE</name>
  <name>INIT_TIME</name>
</output>
</execute>
<execute name="WRF_POST" type="remote">
  <dependencies>ARPS2WRF</dependencies>
  <children>STAGE_OUT</children>
  <scheduler-constraints>interactive</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS</execute-profiles>
  <payload>WRF_POST</payload>
  <input>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </input>
</execute>
<execute name="STAGE_OUT" type="remote">
  <dependencies>WRF,WRF_POST</dependencies>
  <scheduler-constraints>interactive</scheduler-constraints>
  <execute-profiles>TRIGGER_DATA,LOCAL_PATHS,ARCHIVE_URIS</execute-profiles>
  <payload>STAGE_OUT</payload>
  <input>
    <name>STRT_YMDH</name>
    <name>STRT_DATE</name>
    <name>INIT_TIME</name>
  </input>
</execute>
</graph>
</workflow-builder>

```

## Examples: Parameterization

The top level parameter set, named 'compute', ends up with 10 members, formed by picking one element from each of three 'bins' in order:

- bin 1: the Cartesian product formed by two parameter groups; (conditioning, physics) and t the first group is another covariant set of 2 values apiece the second group is a simple range which produces 5 values:
  - ((conditioning-0,physicsP),(conditioning-1, physicsQ)) X (-1.0, -0.5, 0, 0.5, 1.0);
- bin 2: the Cartesian product formed by two parameters: input and logfile input has 10 values, logfile a single value;
- bin 3: a value range with 10 integer values from 0 to 9;

For covariant types, the bins must be of the same cardinality, or an exception will be thrown.

The 10 cases explicitly:

conditioning- algorithm	physics	t	input	logfile	case
file:/conditioning-0	file:/physicsP	-1.0	file:/input-x4083	file:/log	0
file:/conditioning-0	file:/physicsP	-0.5	file:/input-x63	file:/log	1
file:/conditioning-0	file:/physicsP	0	file:/input-z762	file:/log	2
file:/conditioning-0	file:/physicsP	0.5	file:/input-x111	file:/log	3
file:/conditioning-0	file:/physicsP	1.0	file:/input-b059	file:/log	4
file:/conditioning-1	file:/physicsQ	-1.0	file:/input-z4985	file:/log	5
file:/conditioning-1	file:/physicsQ	-0.5	file:/input-a3118	file:/log	6
file:/conditioning-1	file:/physicsQ	0	file:/input-c5593	file:/log	7
file:/conditioning-1	file:/physicsQ	0.5	file:/input-x2067	file:/log	8
file:/conditioning-1	file:/physicsQ	1.0	file:/input-z4391	file:/log	9

```

<parameter-sets>
  <parameters name="compute" type="covariant">
    <parameters type="product">
      <parameters type="covariant">
        <parameter name="conditioning-algorithm">
          <value>file:/conditioning-0</value>
          <value>file:/conditioning-1</value>
        </parameter>
        <parameter name="physics">
          <value>file:/physicsP</value>
          <value>file:/physicsQ</value>
        </parameter>
      </parameters>
      <parameter name="t">
        <value-range type="double" start="-1.0" end="1.0"
          stride="0.5" />
      </parameter>
    </parameters>
    <parameters type="product">
      <parameter name="input">
        <value>file:/input-x4083</value>
        <value>file:/input-x63</value>
        <value>file:/input-z762</value>
        <value>file:/input-x111</value>
        <value>file:/input-b059</value>
        <value>file:/input-z4985</value>
        <value>file:/input-a3118</value>
        <value>file:/input-c5593</value>
        <value>file:/input-x2067</value>
        <value>file:/input-z4391</value>
      </parameter>
      <parameter name="logfile">
        <value>file:/log</value>
      </parameter>
    </parameters>
    <parameter name="case">
      <value-range type="int" start="0.0" end="9.0" />
    </parameter>
  </parameters>
</parameter-sets>

```

An example of a full workflow builder with parameterization follows.

```

<workflow-builder name="test-p-wrf" experimentId="test" user="arossi" eventLevel="DEBUG">
  <profile name="batch-mpi">
    <property name="ELF_HOME" category="environment">
      <value>/u/ncsa/arossi/elf-latest</value>
    </property>
    <property name="count">
      <value>16</value>
    </property>
    <property name="jobType">
      <valuex>single</value>
    </property>
    <property name="maxWallTime">
      <value>60</value>
    </property>
    <property name="project">
      <value>adn</value>
    </property>
    <property name="submissionType">
      <value>batch</value>
    </property>
  </profile>
  <parameter-sets>
    <parameters name="WRF" type="covariant">

```

```

    <parameter name="ctrlat_wrf">
      <value>37</value>
      <value>38</value>
      <value>39</value>
      <value>40</value>
    </parameter>
    <parameter name="ctrlon_wrf">
      <value>-97</value>
      <value>-96</value>
      <value>-95</value>
      <value>-94</value>
    </parameter>
  </parameters>
</parameter-sets>
<scripts>
  <payload name="WRF" type="elf">
    <elf>
      <workdir />
      <serial-scripts separate-script-dirs="false">
        <ogrescript name="run-WRF">
          <declare name="input-file" string="{runtime.dir}/namelist.input"/>
          <echo message="Substituting on input file for wrf"/>
          <modify-namelist namelistTemplate="/usr/var/wrf2.2/input/namelist.input" targetFile="{input-
file}">
            <replace variable="ctrlat_wrf" value="{ctrlat_wrf}"/>
            <replace variable="ctrlon_wrf" value="{ctrlon_wrf}"/>
          </modify-namelist>
          <echo message="Running wrf"/>
          <simple-process execution-dir="{runtime.dir}" outFile="{runtime.dir}/wrf.out">
            <command-line>mpirun -np 16 -lsf /usr/var/wrf2.2/main/wrf.exe</command-line>
          </simple-process>
        </ogrescript>
      </serial-scripts>
    </elf>
  </payload>
</scripts>
<graph>
  <parameterize name="WRF" parameterSet="WRF">
    <graph>
      <execute name="WRF" type="remote">
        <scheduler-constraints>batch-mpi</scheduler-constraints>
        <payload>WRF</payload>
      </execute>
    </graph>
  </parameterize>
</graph>
</workflow-builder>

```