Variable Array Templates: Requirements and Design

MPAS Development Team

October 18, 2013

Contents

Chapter 1

Summary

This document introduces variable array templates. Variable array templates are an easy way for a developer to define a variable array container with constituent variables once, in a template format, and later reference that template multiple times.

The benefit of doing this is a single variable array can be maintained and multiple instances of it can be used, thus keeping the total number of constituent variables consistent across multiple variable arrays.

Chapter 2

Requirements

2.1 Requirement: Templated variable arrays

Date last modified: 10/08/2013
Contributors: (Doug Jacobsen)

Templates are allowed to be defined for variable arrays, and variable arrays are allowed to use these templates.

2.1.1 Motivation and Use Example

As an example, the ocean core has a variable array called tracers with constituents consisting of temperature, salinity, and a tracer with value of 1 for testing purposes. This variable array is duplicated in multiple places as several quantities are computed using each of these tracers, including their state information, tendency information, and surface fluxes as some examples.

Currently, in each var_struct, the tracers variable array is defined as follows:

```
<var_array name="tracers" type="real" dimensions="nVertLevels nCells Time">
       <var name="temperature" units="deg C" description="Potential temperature"/>
       <var name="salinity" units="PSU" description="Salinity"/>
       <var name="tracer1" units="percent" description="Tracer of 1 for testing"/>
</var_array>
```

One of the difficulties in maintaining this, is if a new tracer needs to be added (for example, CO2 for biogeochemistry purposes), it has to be added to each of these variable arrays to keep the constituent dimension size consistent across all of the variable arrays.

When using templated variable arrays, these definitions would change to:

```
<var_array_template name="tracers" type="real" dimensions="nVertLevels nCells Time">
       <var name="temperature" units="deg C" description="Potential temperature"/>
       <var name="salinity" units="PSU" description="Salinity"/>
       <var name="tracer1" units="PSU" description="Salinity"/>
</var_array_template>

<var_struct name="state" time_levs="2">
       <var_array name="tracers" template="tracers" streams="io"/>
       <var_array name="tracersSurfaceValues" template="tracers" suffix="SurfaceValues"
                        description_mod="extrapolated to the ocean surface" streams="o"/>
</var_struct>
<var_struct name="tend" time_levs="0">
       <var_array name="tracers" template="tracers" suffix="Tendencies"
                        units_mod="m s^{-1}" description_mod="tendencies"/>
</var_struct>
```

Chapter 3

Design and Implementation

3.1 Implementation: Variable array templates

Date last modified: 10/08/2013 Contributors: (Doug Jacobsen)

Registry needs to be modified to allow templates to be defined within a Registry.xml file. The new element type will be:

```
<var_array_template name="name" type="real" dimensions="dims">
       <var name="name" name_in_code="name_in_code" units="units"
              description="details"/>
</var_array_template>
```

The variable array templates live at the same level as the variable structures.

This allows a template to define multiple variables as constituent variables. The name of the template will be used in variable arrays to reference this template. A reference to a template will look like:

```
<var_array name="name" template="template" suffix="Suffix"
               units_mod="unit modifer" description_mod="description modifier"
               streams="streams" persistence="persistence"/>
```

In this case, name will be the name of the variable array in the code, template is the name of the template to expand within the variable array, suffix will be appended to all of the constituent variables in the name field (not the name_in_code field if it's specified), units_mod will be appended to the units field, description_mod will be appended to the description field, streams sets the default stresm for all constituents (in the case of a templated variable array these cannot be overridden), and persistence sets the persistence of this particular instance of the variable array.

Following the developer guide lines listed in our developers guide, suffix should always be mixed case, and start with a capital letter. Also, only one

template	Defines the template to expand for this variable array. N
suffix	Defines the suffix for the constituents contained in this instance of the var_array. T
units_mod	Defines the modifer for the units defined in the template. These a
description_mod	Defines the modifer for the description defined in the template. These a

Table 3.1: New attributes for var_array

name	Defines the name of the template. Used in the ter
type	Defines the type of all instanc
dimensions	Defines the default dimensions for all instances of this template. Dimensions can be ov

Table 3.2: Attributes for var_array_template

instance of a templated var_array with a particular suffix can exist within each var_struct. This should

When registry is parsing the Registry.xml if the template attribute is specified on a variable array instance, registry searches for a matching variable array template. If a matching template is found, the constituents are expanded within the variable array. If a template is not found, registry attemps to use the variables defined within the variable array in Registry.xml if there are any. Note: if a template is not found, the resulting variable array might not have any constituent variables. This should be checked in a validator at some point.

3.1.1 Attributes

This project adds new attributes to the var_array constructs in Registry.xml in addition to defining the new var_array_template construct. Below are tables to help describe these:

3.1.2 Constituent and Array group indices

When using a templated variable array, only one instance of the "index_constituent_name" indices are written to each var_struct that the templated is used within. Similarly, only one instance of the "array_group_start" and "array_group_end" indices are written to each var_stuct.

This is the "constituent_name" and "array_group" portions of these indices are defined by the template. Meaning, the suffix for a particular instance of the var_array is not used when referencing a constituent, or an array_group.

Chapter 4

Testing

4.1 Testing and Validation: Templated variable arrays

Date last modified: 10/08/2013
Contributors: (Doug Jacobsen)

After templated variable arrays are implemented in registry, a variable array in one of the components will be replaced with a templated variable array. Simulation results should be bit-identical to the case where the variable array is explicitly defined.