

CCPP Training

College Park, MD, March 12-13, 2019

# The Suite Definition File (SDF)

Dom Heinzeller  
Global Model Test Bed



# Outline of Talk

- Purpose
- Syntax
- Examples
- Stumbling blocks

Location of suite definition files in NEMSFv3gfs codebase:

```
NEMSFv3gfs/ccpp/suites/suite_*.xml
```

# Why do we need this?

Why an additional suite definition file (SDF)?

- there is already a model namelist input.nml
- namelist input.nml requires additional information, for example from a physics driver (order of schemes etc.)
- CCPP replaces the physics driver, i.e the SDF must provide that information

Additional information for the CCPP to run the physics:

- name of the physics schemes to run
- order in which to run the physics
- groups of physics that run together
- subcycling (run at higher frequency, surface iteration)

# Basic syntax

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<suite name="MyFirstSDF" lib="ccppphys" ver="2">  
  <init>global_init_scheme</init>  
  <group name="group_name">  
    <subcycle loop="1">  
      <scheme>scheme_name</scheme>  
    </subcycle>  
  </group>  
  <finalize>global_finalize_scheme</finalize>  
</suite>
```

# Basic syntax

name of the suite,  
for example  
suite\_FV3\_GFS.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<suite name="MyFirstSDF" lib="ccppphys" ver="2">  
  <init>global_init_scheme</init>  
  <group name="group_name">  
    <subcycle loop="1">  
      <scheme>scheme_name</scheme>  
    </subcycle>  
  </group>  
  <finalize>global_finalize_scheme</finalize>  
</suite>
```

don't change

# Basic syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<suite name="MyFirstSDF" lib="com.dtc.sdf" ?>
  <init>global_init_scheme</init>
  <group name="group_name" ?>
    <subcycle loop="1">
      <scheme>scheme_name</scheme>
    </subcycle>
  </group>
  <finalize>global_finalize_scheme</finalize>
</suite>
```

suite-level initialization  
routine, not used at this time  
(use scheme inits only)

suite-level finalization  
routine, not used at this time  
(use scheme final. only)

# Basic syntax

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<suite name="MyFirstSDF" lib="ccppphys" ver="2">  
  <!-- <init></init> -->  
  <group name="group_name">  
    <subcycle loop="1">  
      <scheme>scheme_name</scheme>  
    </subcycle>  
  </group>  
  <!-- <finalize></finalize> -->  
</suite>
```

# Basic syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

```
<suite name="MyFirstSDF" lib="ccppphys" ver="2">
```

```
  <!-- <init></init> -->
```

```
  <group name="group_name">
```

```
    <subcycle loop="1">
```

```
      <scheme>scheme_name</scheme>
```

```
    </subcycle>
```

```
  </group>
```

```
  <!-- <finalize></finalize> -->
```

```
</suite>
```

name of the group of schemes to run, e.g. physics, radiation, time\_vary



# Basic syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<suite name="MyFirstSDF" lib="SDF" >
  <!-- <init></init> -->
  <group name="group_name">
    <subcycle loop="1">
      <scheme>scheme_name</scheme>
    </subcycle>
  </group>
  <!-- <finalize></finalize> -->
</suite>
```

how often to run the group  
per physics time step (can  
use for surface iteration)

# Basic syntax

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<suite name="MyFirstSDF" >
  <!-- <init></init> -->
  <group name="group_name">
    <subcycle loop="1">
      <scheme>scheme_name</scheme>
    </subcycle>
  </group>
  <!-- <finalize></finalize> -->
</suite>
```

name of the scheme to run,  
e.g. gfdl\_cloud\_microphys

# Example GFS suite, radiation group

```
<group name="radiation">  
  <subcycle loop="1">  
    <scheme>GFS_suite_interstitial_rad_reset</scheme>  
    <scheme>GFS_rrtmg_pre</scheme>  
    <scheme>rrtmg_sw_pre</scheme>  
    <scheme>rrtmg_sw</scheme>  
    <scheme>rrtmg_sw_post</scheme>  
    <scheme>rrtmg_lw_pre</scheme>  
    <scheme>rrtmg_lw</scheme>  
    <scheme>rrtmg_lw_post</scheme>  
    <scheme>GFS_rrtmg_post</scheme>  
  </subcycle>  
</group>
```



suite interstitial



scheme interstitial



scheme

# Example GFS suite, physics group

```
<group name="physics">  
  <subcycle loop="1">  
    <scheme>GFS_suite_interstitial_phys_reset</scheme>  
    <scheme>GFS_suite_stateout_reset</scheme>  
    <scheme>get_prs_fv3</scheme>  
    <scheme>GFS_suite_interstitial_1</scheme>  
    <scheme>dcyc2t3</scheme>  
    <scheme>GFS_surface_generic_pre</scheme>  
    <scheme>GFS_suite_interstitial_2</scheme>  
  </subcycle>  
  <!-- Surface iteration loop -->  
  ...
```



# Example GFS suite, physics group

```
<!-- Surface iteration loop -->  
<subcycle loop="2">  
  <scheme>sfc_ex_coef</scheme>  
  <scheme>GFS_surface_loop_control_part1</scheme>  
  <scheme>sfc_nst_pre</scheme>  
  <scheme>sfc_nst</scheme>  
  <scheme>sfc_nst_post</scheme>  
  <scheme>lsm_noah</scheme>  
  <scheme>sfc_sice</scheme>  
  <scheme>GFS_surface_loop_control_part2</scheme>  
</subcycle>  
<!-- End of surface iteration loop -->  
...
```

# Example – replace Noah with RUC

```
<!-- Surface iteration loop -->  
<subcycle loop="2">  
  <scheme>sfc_ex_coef</scheme>  
  <scheme>GFS_surface_loop_control_part1</scheme>  
  <scheme>sfc_nst_pre</scheme>  
  <scheme>sfc_nst</scheme>  
  <scheme>sfc_nst_post</scheme>  
  <scheme>lsm_ruc</scheme> <scheme>lsm_noah</scheme>  
  <scheme>sfc_sice</scheme>  
  <scheme>GFS_surface_loop_control_part2</scheme>  
</subcycle>  
<!-- End of surface iteration loop -->  
...
```

# Things to keep in mind

The SDF does not contain all the information of the namelist

- still need to have a namelist file input.nml
- at present, some information is redundant and must be consistent, or bad things can happen
- example: namelist value of imp\_physics must match the choice of microphysics in SDF

Work is underway to remove this redundancy

- scheme-specific namelist parameters (e.g. tuning)
- host-model specific control parameters (e.g. choice of MP)
- user overrides, error checking