Simulation File Tree
(DBB 8-21-06)

There are three classes of files involved in a SWIM simulation

- Input files
- Results files
- Component working files

It is not necessary for all of these to be under the same directory but for simplicity we assume that to be the case here. The framework figures out what to name the top level Simulation Directory (e.g. /…/`IPS_run_XYZ`), where to put it and how to make the path to this directory known to the component scripts (if in fact they need to know it). Furthermore we assume that the "computer environment initialization" step in the Controller Script (see Section II) creates the directory tree.

`IPS_run_XYZ`
  |-- `simulation_setup` *(input files, this directory is never overwritten)*
  |-- `simulation_results` *(output files)*
  |-- `work` *(working files with a subdirectory for each component)*

To do a simulation the user will have to assemble somewhere in his user area a directory containing all of the necessary control files and initial input files for the run (described in Setup and Input sub-directory of Section I.1). There has to be a mechanism for the SWIM framework to know where this is (e.g. launching from the top level of this directory). The framework initially copies the `/simulation_setup` directory from the user area (or other simulation input staging area) into the Simulation Directory.

The current Plasma State file resides in the `/work` subdirectory. It is put there initially by the controller script and is updated after a successful time step by the Plasma State function PS_COMMIT_PLASMA_STATE. The controller script also copies the initial input data for each component from the appropriate component subdirectory of `simulation_setup` into the component subdirectory of work. After all of the components have successfully completed the time step the controller script creates a new subdirectory for that time step (identified by the time in milliseconds) in the `/history` and `/restart` subdirectories. It then gathers up the appropriate files in each component working directory and copies them to history and restart for the time step. It is the job of the component scripts to leave the working directories in condition to allow the controller script to effect that transfer. In particular the component leaves a list of files be transferred in `component_outfile_list`. In this design code developers and component developers need only be concerned about the structure of the /`work` subdirectory.

An initial layout is shown below:

simulation_setup subdirectory

simulation_setup

- system_config
- simulation_wide_data
  - simulation_control_file (this is a file not a directory)
  - initial_plasma_state (t_0) (this is a file not a directory)
  - machine_definition (may be empty for now)
  - simulation_events_waveforms (scheduled events, source and control waveforms)
- component_inputs
  - RF
    - RF_component (the component may need generic files)
      - RF_config
      - RF_required_list
      - RF_outfile_list
    - code_inputs (code specific files e.g. AORSA)
      - AORSA_required_list
      - AORSA_outfile_list
      - Standard AORSA input files …
  - FokkerPlanck
    - FP_component
      - FP_config
      - FP_required_list
      - FP_outfile_list
    - code_inputs (code specific input files e.g. CQL3D)
      - CQL3D_required_list
      - CQL3D_outfile_list
      - Standard CQL3D input files…
  - …
  - <other components>…

simulation_results subdirectory

```
simulation_results
  |-- history
    |-- t0 (identifier in milliseconds)
      |-- plasma_state
      |-- components
        |-- RF
        |-- Fokker Planck
        …
        |-- <other components>
    |-- t1 (identifier in milliseconds)
      |-- plasma_state
      |-- components
        |-- RF
        |-- FokkerPlanck
        …
        |-- <other components>
    …
    |-- tN…<other time steps>

  |-- restart
    |-- t0 (identifier in milliseconds)
      |-- RF
        |-- input_files
        |-- internal_state
      |-- Fokker Planck
        |-- input_files
        |-- internal_state
      …
      |-- <other components>
    |-- t1 (identifier in milliseconds)
      |-- RF
      |-- Fokker Planck
      …
      |-- <other components>…
    …
    |-- tN…<other time steps>
```

work subdirectory

work

```
  |-- plasma_state (current plasma state t_n)
  |-- RF
    |-- RF_component (the component may need generic files)
        |-- RF_config
        |-- RF_required_list
        |-- RF_outfile_list
        |-- RF_log
    |-- code_inputs (code specific input files e.g. AORSA)
        |-- AORSA_required
        |-- AORSA_outfile_list
        |-- AORSA standard input files …
    |-- RF_component_script (?)
    |-- executable (?)
    |-- code_outputs
        |-- AORSA standard input files …
        |-- AORSA_log
    |-- scratch
  …
  |-- <other components>…
```