

## **Procedures for implementing developments into a new version of JULES**

The following procedures should be followed in order for any developments to be included into a future version of JULES:

1. The developer should notify the module leader (or leaders if the development involves more than one module) of the plans for development. This should be done at the earliest possible opportunity so that the module leaders have a good appreciation of the development work being undertaken within the community.
2. The module leaders should maintain a document that includes all current development activities within their module, which will be available at each of the science steering committee meetings and discussed as appropriate
3. Once nearing completion of the development, the developer should notify the module leader/s of an intended completion date
4. When a development is complete and complies with the coding standards, the code should be put through the benchmarking system to ensure that; (a) the model is still backwards compatible (i.e. the new developments have been put in on a switch that does not affect the previous version of the code), and (b) that the new development does not give an unacceptable level of performance for any of the modules.
5. The developer should also ensure that the technical and user documentation have been updated as appropriate to cover the new development.
6. Once the development has passed the benchmarking tests, the developer should hand the code and the documentation to the module leader so that they can arrange for a code review. This can be done either by the module leader themselves, or by another person who is familiar with the science. The main purpose of the code review is to ensure that the development abides by the coding standards, has correctly passed the benchmarking tests and has been documented correctly.
7. After passing the code review, the module leader should present the development to the management committee for approval to include in a future version of JULES.
8. The management committee will be responsible for deciding which developments go into the planned future releases of JULES and will decide on the timescale for release versions based upon the amount of work required for consolidation.
9. For all developments that have been approved for the next version of JULES, the developers will then pass their completed code and updated documentation to the JULES coordinator, who will consolidate all code changes into a single version.
10. The JULES coordinator will run the consolidated JULES code through the benchmarking system to ensure that; (a) the model is still backwards compatible, and (b) that the new developments do not give an unacceptable level of performance for any of the modules.
11. The JULES coordinator will undertake a series of tests with JULES coupled to the UM to ensure that the code developments meet the UM standards, including bit comparison over PE configurations
12. Once the code has passed all of these tests, the JULES coordinator will freeze the code and release the new version to the community.

## **Exceptions:**

There are a couple of exceptions to the above procedures for which not all of the elements have to be completed. There are two types of exceptions, and their changes to the above procedures are as follows:

- I. Developments that impact only on the control code.
  - Procedure (10): The JULES coordinator will only need to complete part (a), i.e. there will be no need to check for unacceptable levels of performance from the science code.
  - Procedure (11): This will not be required as the control code is not included in the UM implementation
  
- II. Large developments that require an interim upgrade to consolidate some of the code development
  - Procedure (4): The developer will only need to complete part (a), i.e. there will be no need to check for unacceptable levels of performance
  - Procedure (6): The code review will only be required to check coding standards, appropriate documentation and the backwards compatibility of the benchmarking tests, i.e. the development will not have to pass the benchmarking tests for unacceptable levels of performance.
  - Procedure (10): The JULES coordinator will only need to complete part (a), i.e. there will be no need to check for unacceptable levels of performance.

The interim upgrades under exception II will not be common practise and will only apply to very significant coding changes. Approval for any interim upgrade will need to be agreed by the management committee, and should be presented to them through the module leader, as with other developments and detailed in procedure (7).