

ESP-r Developers' Code Contribution Checklist

Last updated: February 6, 2009

1. Synopsis

This document outlines the documentation to be sent to the archivist when a code commit is made. When you have completed the steps outlined in the QA checklist, send a note to the archivist using the template below.

Ideally, code submissions should be related to one concept. For example changes which implement a new interface to an ideal zone control should be made separately from code that extends a statistical report in the results analysis module. A change in a data structure should be made to all associated code blocks in one commit if possible so that the ESP-r distribution is consistent. If there are multiple issues in the commit then each should be noted separately.

The documentation sent to the archivist is a summary of the contribution which will form part of the release note which will be published when ESP-r versions change. Readers will be looking for the significance and applicability of your changes. Avoid detail, as the archivist can look at the detailed documentation in your branch. Be sure to include the following in your summary:

- * the nature of the change
- * whether it is new functionality for the user, repairs a bug, tidies code, or is a work in progress
- * if new functionality is provided for the user, then indicate whether this affects the Simulator, Project Manager, Results Analyzer, etc.
- * what users might notice
- * what developers might notice
- * if a new method has been implemented provide a citation to a thesis or a paper that describes its theoretical basis
- * if a bug has been repaired, then indicate which application this affects and the nature of the change
- * if a bug fix alters simulation results describe what the changes are and what types of models would be impacted
- * if this change tidies code or alters a data structure but does not alter functionality indicate this
- * if the submission is work in progress indicate whether it is safe for others to use
- * IMPORTANT: Use proper grammar and spelling!

2. Example

The listing below is a sample of the documentation. Note that the revision numbers follow the following syntax. If you made two commits 101 and 103 this is represented by r100:104. If you made a single commit 345 then this is represented by r344:345.

Revisions to merge into development_branch:

- r100:103 and r345 of sub-branch Joe_Developer.

Summary

- Added scripts to convert models between native windows and Linux.
Users working across platforms will find these useful.
- Added site related data to flatfile format. No impact on users.
The flatfile continues to evolve. What is included is functional and should not be considered work in progress.

- Increase text buffer length for X11 version. Interface only, no impact on predictions.
- Further work on UK NCM notional model creation (databases, and automation of tasks, thermal bridges). Fewer interactions when generating notional model. The work has a likely future impact in terms of code consolidation and extension of the ESP-r data model and should be considered work in progress.

Testing

- Pre-commit and post-commit syntax checks on full source of all modules
- Scripts used to convert an exemplar model for use on another machine
- Interactive test of the X11 text buffer by listing and scrolling back through a QA report.
- Added site data to a flatfile and checked that the ESP-r model created included this information.
- Interactive creation of a notional model noting altered dialogues.
- Compile Solaris F90 X11
- Compile Linux GCC 3.4 X11 and GTK and text-only
- Compile OSX GCC 3.4 X11 and GTK
- Test merge into development branch with no conflicts reported
- Compile test merge with Linux GCC 3.4 X11

Revisions to take from Joe_Developer:

r100:103

r344:345
