

## **1.0 Windows Eudora Perforce Process**

### **1.1 The Players**

#### **1.1.1 Branches**

- 1.1.1.1 Sandboxes - branches of the depot where code not yet intended for use may be submitted.
- 1.1.1.2 Trunk - the main line of Eudora source, where new features are added and Eudora is in general made to progress.
- 1.1.1.3 Release Branch - a branch of the depot made for a specific release, when that release is imminent.

#### **1.1.2 People**

- 1.1.2.1 The Master Integrator - this person is solely responsible for and is given sole permission to make changes to the Release Branch (save for changes that occur as part of the build process itself).
- 1.1.2.2 QA - Eudora's QA and release build team.
- 1.1.2.3 Pond Scum - Engineers working on Eudora, including the Master Integrator when he doesn't have his little Master Integrator cap on.

### **1.2 Textual Conventions**

#### **1.2.1 Change Descriptions**

- 1.2.1.1 All perforce change descriptions **MUST** begin with a status in square brackets. The status will be drawn from a set of statuses to be discussed later. The status **MUST** be the first item in the description, and a brief textual description **MUST** follow the status on the same line.
- 1.2.1.2 If the change fixes a bug entered in Visual Intercept, the second (and following, if more than one bug is fixed) lines of the change description **MUST** be the Subject of the incident exactly as appearing in Visual Intercept.
- 1.2.1.3 Free-form elaboration and comments **MAY** follow the incident subject(s).
- 1.2.1.4 For example:
  - 1.2.1.4.1 [INTEGRATE] Fixes father-stabbing problem.  
4.2 (b32) (5555): Bleeding from male parents.  
Switched ::StabFather() to ::StabMother().

#### **1.2.2 Visual Intercept Descriptions**

- 1.2.2.1 When submitting or integrating a change that fixes a specific incident, the developer **MUST** add the branch name and change number to the Description field in VI, using the syntax:  
[branchname:Pchangenumber]
- 1.2.2.2 The developer **SHOULD** add explanatory text if needed.
- 1.2.2.3 For example:
  - 1.2.2.3.1 [Sandbox:P1999] Now properly stabbing mothers, not fathers.

### **1.3 The Process**

#### **1.3.1 Step 1 (optional) - Using a Sandbox**

- 1.3.1.1 Developers may submit changes to a sandbox at any time. They may have their own sandboxes all to themselves, or share sandboxes with other developers as appropriate. Code checked into sandboxes needn't compile, work, or avoid painful bowel movements.
- 1.3.1.2 Changes submitted to sandboxes may use any of the following statuses:
  - 1.3.1.2.1 [HOLD] - this change is not to be integrated at any time
  - 1.3.1.2.2 [REVIEW] - this change needs to be reviewed by someone. The developer will have to arrange for this review to occur.

- 1.3.1.2.3 [INTEGRATE] - the change is ready to be integrated into the trunk. The developer will have to integrate it or arrange for the integration to occur.
- 1.3.1.3 Subsequent actions may alter the change to any of these statuses, in addition to the ones already mentioned:
  - 1.3.1.3.1 [REJECTED] - the change was reviewed by another developer who thought it was a terrible idea. The original developer should back it out (after possible appeal to the Master Integrator).
  - 1.3.1.3.2 [DEAD] - the change was backed out by the developer, using another change that MUST also be marked [DEAD].
  - 1.3.1.3.3 [DONE] - the change was integrated into the trunk.
- 1.3.2 Step 2 - Using the Trunk.**
  - 1.3.2.1 Developers satisfied that their changes are ready for use may integrate them or submit them directly to the trunk.
    - 1.3.2.1.1 Any developer submitting changes to the trunk that do not compile will be sent to Singapore with a can of spray paint.
    - 1.3.2.1.2 Changes should not be submitted to the trunk unless the developer has a high degree of confidence that the changes are appropriate and working. If this confidence is lacking, the developer should use a sandbox, as in Step 1, and obtain review from someone else.
  - 1.3.2.2 Changes submitted to the trunk MUST use of these statuses:
    - 1.3.2.2.1 [REVIEW] - the change should be reviewed, but developer is not ready for it to be integrated.
    - 1.3.2.2.2 [INTEGRATE] - developer believes change is ready for integration into release branch.
    - 1.3.2.2.3 [DEFER-REVIEW] - developer wishes the change to be reviewed, but does not intend the change to go into the current release branch. It will instead be going into a future release, assuming it passes review.
  - 1.3.2.3 Once a change is submitted to the trunk, any VI incidents it fixes should be given a state of "Pending".
  - 1.3.2.4 Once the change has been submitted to the trunk, it will be reviewed by the Master Integrator. The developer does not need to ask the Master Integrator for this high honor--the Master Integrator will scan the trunk for such changes.
- 1.3.3 Step 3 - The Master Integrator and the Release Branch**
  - 1.3.3.1 The Master Integrator MUST scan the trunk for changes marked [REVIEW] and [INTEGRATE], and either review or integrate them himself or arrange for someone else to do so.
  - 1.3.3.2 The Master Integrator may change the status of the change to [REVIEW], [REVIEW-DEFER], [INTEGRATE], [REJECTED], [DEAD], or [DONE] with the same implications as in a Sandbox. If the Master Integrator sets the status to [REVIEW], [REVIEW-DEFER] or [INTEGRATE], he must arrange with another developer for this review or integration to occur.
  - 1.3.3.3 An additional status is available to the Master Integrator:
    - 1.3.3.3.1 [DEFER] - parks the change; it will not be integrated into the release branch, but will be included a future release.

1.3.3.4 When a change is integrated into the Release Branch, the Master Integrator MUST mark any VI incidents that are fixed by the change as Fixed, and MUST add an appropriate branch and change identifier to the Description section of the incident(s).

**1.3.4 Step 4 - QA and Builds**

1.3.4.1 Before doing a build, QA MUST scan the trunk for changes marked [REVIEW] or [INTEGRATE]. These changes should be brought to the attention of the Master Integrator, who will decide if the build should proceed without the changes.