

A. CONVERTING FROM BUTTRESS RULES

The primary change to rules was the removal of the **VarLength** mixin as a super class of **RuleSetSource**. Rules are now stored on the IV **ruleList** of **RuleSetSource**. If any user had added properties to a rule with a construct similar to the following expression, these will need to be reimplemented in some fashion:

```
(PutNthValue ($ some-Instance-Of-A-RuleSetSource) ruleNumber prop)
```

A number of other changes were made to account for changes from Buttress LOOPS to the product release. These should not be noticeable to the general user.

A utility was built to allow rules that were saved in Buttress to be loaded into the product release of LOOPS. In order to convert from Buttress, follow this procedure.

- Load the necessary files for Rules as described in the Section 3.12, "Loading Rules."
- Load the file `LOOPSRULESBACKWARDS`. This redefines `ConvertLoopsFiles` to add functionality for converting rules.
- Do not load the Buttress files that need to be converted. Instead call:

```
(ConvertLoopsFiles files-to-convert T T)
```

This will load, convert, remake, and recompile the files specified by *files-to-convert*.

Any rules that were saved with auditing turned on will need to run through the rule compiler. The old rule compiler output a form that was `(_ ($ StandardAuditRecord) NewTemp)`. The message **NewTemp** no longer exists. The rule compiler now puts out the form `(_ ($ StandardAuditRecord) New)`.

If rules were saved with auditing turned on, then call `(ConvertLoopsFiles files-to-convert T)`. Deleting the final "T", will not enable the cleanup. After recompiling the rules to eliminate the message **NewTemp**, remake your files.

[This page intentionally left blank]