Table of Contents

How to Configure Output Modules

Complete:

Goal of this page

This document describes how to configure output modules. In CMSSW_0_4_0_pre2, configuration of output modules has undergone a significant change; this document describes only the new system.

Example of Configuration

As are all other framework modules, output modules are configured using a `edm::ParameterSet` instance. In addition to whatever other parameters a specific output module class may require, *all* output module classes allow the parameter **outputCommands**. This parameter is of type `vector` of `string`, and carries zero or more "commands" that determine which branches in the `Event` are to be written out by that output module.

An example configuration is:

```
cms.untracked.vstring(
              'keep *',
              'drop *_*_*_HLT',
              'keep FEDRawDataCollection_*_*_*'
)
```

This configuration tells the system to:

1. Write out (keep) all branches, except ...
2. Do not write out (drop) all branches from the HLT process, except ...
3. Write out all products of type FEDRawDataCollection.

Commands are applied in the order of presentation in the vector. Each command carries an *action* (keep or drop) and a *specification* (*e.g.* *_*_*_HLT). The specification is compared to each branch name. The specification matches the branch name if all four fields of the branch name match the corresponding fields of the specification. The four fields are separated by underscores. NOTE: implicitly the first command is always `'drop *'` . Therefore if you want to only specify `drop` commands you should use a `'keep *'` as the first command.

A special case is when the entire specification is the one character "*". This is interpreted as "*_*_*_*". In all other cases, exactly 3 underscores must appear in the specification. The fields must contain only alphanumeric characters or one of two available wildcards. The wildcard "*" will match zero or more characters and these characters can be anything. The wildcard "?" will match exactly one character and that character can be anything. One restriction is that these wildcards will only match a sequence of characters contained inside one of the four fields. They can match an entire field or some smaller part of a field, but nothing larger. All 3 underscores must explicitly appear in the specification. (Prior to release 1_3_0_pre1, the ? wildcard was not available and the * wildcard would only match an entire field and nothing smaller)

If no parameter named "outputCommands" is found, then a default value is used. The default value is:

```
cms.untracked.vstring(
        'keep *'
)
```

and so by default all objects are written by a module so configured.

Note: If a class has been declared non-persistent, any branch consisting of EDProduct of that exact class will never be output. For more information, see the appropriate section in SWGuideCreatingNewProducts

Real life examples

See Configuration/EventContent/python/EventContentCosmics_cff.py or EventContent/python/EventContent_cff.py

The simple output writing algorithm.

For purposes of understanding the behavior of the output, the algorithm used behaves as if it did the following:

1. All branches are assigned a 'write bit' set to `false`.
2. Each command is applied to each branch.
    1. if the specification in the command matches the branch name, the branch's 'write bit' is set to the value determined by the 'action' of the rule (keep sets it to `true`, drop sets it to `false`) b. if the specification in the command does not match the branch name, the 'write bit' for the branch is not changed.
3. Commands are considered in the order of their appearance in the vector, so that later commands can override the actions of earlier commands.

Note on branch names

Branch names have a four-part form with the format: **PT_ML_IN_PN**

1. PT is the *product type*; this is also called the "friendly class name".
2. ML is the *module label*; this is the label of the module that created the product.
3. IN is the *instance name*; this may be (and often is) a zero-length string.
4. PN is the *process name*; this is the name of the process in which this object was created.

Most people will probably never see this or need to know this, but in some places the same branch name will occur with a period appended to the end. This is required for some purposes by ROOT. This period should not be used in "keep" or "drop" statements.

Review Status

Reviewer/Editor and Date (copy from screen)	Comments
Main.William Tanenbaum - 14 Jan 2007	page last content editor
JennyWilliams - 31 Jan 2007	editing to include in SWGuide

Responsible: WilliamTanenbaum
Last reviewed by: Sudhir Malik- 24 January 2009

This topic: CMSPublic > SWGuideSelectingBranchesForOutput
Topic revision: r17 - 2011-07-28 - WilliamTanenbaum