

Table of Contents

CMS MessageLogger Service: File Destinations.....	1
Specifying an Explicit Extension.....	1

CMS MessageLogger Service: File Destinations

```
import FWCore.ParameterSet.Config as cms
process.load("FWCore.MessageLogger.MessageLogger_cfi")

process.MessageLogger = cms.Service("MessageLogger",
    destinations = cms.untracked.vstring (
        'detailedInfo'
        , 'critical'
        , 'cout'
        , 'cerr'
    )

    critical      = cms.untracked.PSet(
        threshold = cms.untracked.string('ERROR')
    ),
    detailedInfo  = cms.untracked.PSet(
        threshold = cms.untracked.string('INFO')
    ),
    cerr          = cms.untracked.PSet(
        threshold = cms.untracked.string('WARNING')
    )
)

# set the number of events
process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(5)
)

process.myAnalysisModule = cms.EDAnalyzer('ModuleThatIssuesMessages')
process.p = cms.Path(process.myAnalysisModule)
```

The strings listed in the `vstring` files list each represent the name of a file destination (a default extension `.log` is appended). Later, optional further configuration can be supplied for each listed destination. This further configuration typically establishes a threshold and/or limits of how often messages are reacted to, and is written as a **PSet** with name matching the file name.

There need not be a further configuration **PSet** for every listed destination. (In this example file, there is no further configuration for the `cout` destination.) Absent further configuration, the destination will react all messages issued.

Also, observe that the orders in which file names are listed and **PSet**'s are provided are arbitrary and need not match.

Specifying an Explicit Extension

Since the use of the dot character in a **PSet** name is discouraged, the `MessageLogger` assigns a default extension `.log`. The extension used can, however, be specified to be something else, as in:

```
import FWCore.ParameterSet.Config as cms
process.load("FWCore.MessageLogger.MessageLogger_cfi")
process.MessageLogger = cms.Service("MessageLogger",
    destinations = cms.untracked.vstring (
        'detailedInfo'
    )
    ,
    detailedInfo  = cms.untracked.PSet(
        extension = cms.untracked.string('.txt')
    )
)
```

)

Here, the file produced will be detailedInfo.txt.

---++Specifying an Explicit File Name

Instead of specifying just the extension, one can specify explicitly the filename to be used (either including the extension or with a default extension of [.log](#)), as in:

```
import FWCore.ParameterSet.Config as cms
process.load("FWCore.MessageLogger.MessageLogger_cfi")
process.MessageLogger = cms.Service("MessageLogger",
    destinations = cms.untracked.vstring (
        'detailedInfo'
    ),
    detailedInfo = cms.untracked.PSet (
        filename = cms.untracked.string('minutia')
    )
)
```

This is also a technique you could use to specify that a destination write to a file in a directory other than the working directory, as in:

```
import FWCore.ParameterSet.Config as cms
process.load("FWCore.MessageLogger.MessageLogger_cfi")
process.MessageLogger = cms.Service("MessageLogger",
    destinations = cms.untracked.vstring (
        'info'
    ),
    info = cms.untracked.PSet (
        filename = cms.untracked.string('logfiles/minutia.out')
    )
)
```

(When doing this, you should, of course, make certain that the specified path will exist. In the example above, if the directory [logfile](#) did not exist, the logger would not create one automatically, so the attempt to setup this destination would fail.) -- Main.SudhirMalik - 30-Aug-2011 --- This topic:

CMSPublic > SWGuideMessageLoggerFileName

Topic revision: r5 - 2012-05-13 - NitishDhingra



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback