



# **DLF Logging for the new Stager**

**Castor Dev Team  
(October 2007)**

# Contents:

- 1. Objectives:
  - Logging in the C Stager.
  - Goals for the new Cpp Stager.
- 2. Introduction to DLF:
  - Components.
  - Useful examples.
- 3. New Cpp Stager Overview:
  - Architecture.
- 4. Logging in the new Cpp Stager:
  - Used DLF Levels.
  - Types of DLF Messages.
  - Information on the DLF Parameters.
  - Examples:
    - Get
- 5. Suggestions?
- 6. Appendix:
  - DBService DLF Messages.

# 1. OBJECTIVES:

- **Logging C Stager:**



- Different levels:

- {VERBOSE, ERROR, DB\_ERROR, SYSCALL, WARNING, LOG}

- Using as identifier:

- struct Cns\_fileid: {fileid, server}

- Printing information:

- Representative of the Objects' **attributes**.
    - Not so representative of the Requests' **flow**.
      - Triggered action: Disk2DiskCopy, TapeRecall...
      -
  - ->**Useless for Debugging/Monitoring purposes.**

# 1. OBJECTIVES:

## ■ Goals for the new Cpp Stager:



- To indicate what has been done for each Subrequest:
  - Triggered action: Disk2DiskCopy, TapeRecall...
  -
- To identify properly the Subrequest.
  - {Subrequest, Request, User, type, SvcClass}
  -
- To include Relevant Information for debugging/Monitoring.
- 
- To identify the failure point (just in case...)

# 2. INTRODUCTION TO DLF:

## ■ Components:

- DLF Severity Level:
  - Error, Monitoring, Debug...
- DLF Messages:
  - {intMessageNb, stringMessage}
- DLF Param:
  - {Cuuid\_t subrequestUuid}:
  - {constStringGeneral, parameter}
    - where parameter can be: string, number...
- dlf\_writep:
  - (Cuuid\_t requestUuid, int severity, int messageNb, int nbParam, Cns\_fileid\*)
  -

## ■ Useful Examples:

- rmMasterDaemon, jobManager.

# 3. NEW CPP STAGER OVERVIEW:

- **Arquitecture:**



- Main Daemon.



- Threads Pools.



- Threads: Services.



- Helpers.



- RequestHandlers.

## 4. LOGGING IN THE NEW CPP STAGER:

- > Detailed list {messageNb, stringMessage} in:
  - “StagerDlfMessages.hpp”

- 

- **Used DLF Levels:**

- 

- DLF\_LVL\_ERROR.**

- 

- DLF\_USER\_ERROR: Brand new one.**

- 

- DLF\_LVL\_MONITORING.**

- 

- DLF\_LVL\_DEBUG**

- 

-

## 4. LOGGING IN THE NEW CPP STAGER:

- **Types of DLF Messages:** Mainly
  - Related with Stager's components:
    - To DLF\_LVL\_DEBUG.
      - To show:
        - Subrequest Details.
        - Type of Subrequest.
  - Related with the result of a Subrequest:
    - To DLF\_LVL\_MONITORING.
      - To show:
        - Triggered action (= result of the subrequest)
  - Related with possible errors (Exceptions):
    - To DLF\_LVL\_ERROR/USER\_ERROR.
      - To show:
        - Error/Exception.
        - Where it has happened.



# 4. LOGGING IN THE NEW CPP STAGER:

## ■ Information on the DLF Parameters:

- Since in `dlf_writep` the parameters are:
  - `Cuuid_t requestUuid.`
  - `Cns_fileid: {fileId, server}`
- 
- The missed information on its call is:
  - `Cuuid_t subrequestUuid.`
  - Request type.
  - User: {`eid, egid`}
  - `SvcClassName.`
  - Function.
  - Others:
    - `setGCWeightRequest.`
    - `rmRequest.`

# 4. LOGGING IN THE NEW CPP STAGER:

## ■ **Examples : Get**

- In case or early Error:
  - DLF\_LVL\_ERROR/USER\_ERROR:
    - “Impossible to get the Services”,
    - “Impossible to get the SvcClass”,
    - “User asking for a non existing file”,
    - “Invalid user permission”...
- Normal flow:
  - DLF\_LVL\_DEBUG:
    - “Get execution”
  - DLF\_LVL\_MONITORING: Triggered option:
    - “Disk2DiskCopy”
    - “Tape Recall”
-



## 5. Suggestions???

## 6. Appendix: DBService DLF Messages:

- **DLF\_LVL\_MONITORING:**
  - “Archiving subrequest”
  - “Nothing to be done (STAGED)”
  - “Starting Repack Migration”
  - “Triggering Disk2DiskCopy”
  - “Triggering Tape Recall”
  - “Recreating CastorFile”
  - “SetGC details”
  - “Rm details”

## 6. Appendix: DBService DLF Messages:

- **DLF\_LVL\_ERROR/USER\_ERROR:**
- 
- - “Impossible to get the Service”**
  - “Impossible to get the SvcClass”**
  - “User asking for a Non existing file”**
  - “Wrong user permission”**
  - “Recreation impossible”**
  - “Invalid File System”**
  -