DCS–M04–GDDR Customisation Guide

Customisation Guide for GDDR

Abstract	GDDR Customisation Guide
Document Reference	
Document Type	Customisation Guide
Project/Application	DCS-GDDR
Version	01.01
Status	Draft
Date of Issue	20 May 2005
File Location	DCS:\M00-Mainframe\DCS M04 GDDR Customisation Guide.doc
# Pages	21

Produced by	Phil Davies
Reviewed by	Didier Lemaitre, Renaud Colin, Michael Quigley
Authorised by	Didier Lemaitre

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Scope

This document describes the post-installation tasks/actions required inorder to complete the GDDR configuration process once the GDDR installation process has been completed.

1.2 Objectives

It is the intention of this document to describe in detail how to Customise GDDR from the z/OS System Programmers perspective.

1.3 Intended audience

z/OS System Programmers responsible for the Installation and Maintenance of GDDR.

1.4 References

Ref. nbr.	Reference	Title
[1]		Project Definition Report
[2]		Project High Level Technical Design
[3]		

1.5 Change history

Version	Nature of change	Date
00.00	Create Document (draft)	20-05-2005
01.01		
01.02		
01.03		

1.6 Forecast changes

Version	Nature of change	Date

1.7 Abbreviations

Abbreviation	Full text
DCS	Data Center Strategy
HLTLS	High Level Test & Launch Strategy
GDDR	Geographically Dispersed Disaster Recovery
ConGroup	EMC Software Product Consistency Group
K1	GDDR K-System at the DC1 Site
K2	GDDR K-System at the DC2 Site
K3	GDDR K-System at the DC3 Site

1.8 Definitions

Terminology	Definition
Systems	The word system or systems used in this document refers to an z/OS Image and all systems tasks and applications running in it.
K-System	A controlling LPAR, one in each of DC1, DC2 and DC3. Monitors the status of the DASD and DASD mirroring to DC2 and DC3.
K-System Master Function	The K-System that is currently responsible for monitoring the status of DASD and DASD mirroring. Is also responsible for taking action when problems with DASD and DASD mirroring are detected.
Primary Site	The live site where the Production z/OS Images run. Usually DC1.
Secondary Site	The backup site that has a synchronous copy of the Production data. Usually DC2.
Tertiary Site	A backup site that has an asynchronous copy of the Production data. Always DC3.
CA-OPS/MVS MSF	CA-OPS/MVS Multi System Facility
Trip	The action ConGroup takes when it detects that one or more R1 devices in a consistency group cannot propagate data to their corresponding secondary (R2) devices. During a trip, ConGroup suspends all the primary (R1) devices in the consistency group. This suspension ensures that the data flow to the secondary (R2) side is halted and the data on the remote side of the configuration is consistent.
J0	Journal-0 is EMC's term for SRDF/S
JA	Journal-A is EMC's term for SRDF/A
SNOW	Symmetric Native Ordered Writes (SNOW) is EMC's internal code name for SRDF/A

2. CUSTOMISATION OF GDDR

2.1 Overview of Customisation

The following chapter details the customisation of GDDR inorder to complete the installation of GDDR.

These changes must be made on the GDDR K-Systems ONLY

2.1.1 System Changes

The following documents the system changes/updates required inorder to complete the installation of GDDR.

2.1.1.1 SYS1.PARMLIB(BPXPRMxx)

Add the following mount to the BPXPRMxx member of SYS1.PARMLIB:

```
MOUNT FILESYSTEM(' gddr_hfs_dataset_name ')
        MOUNTPOINT('/gddr')
        TYPE(HFS)
        MODE(RDWR)
```

Where '**gddr_hfs_dataset_name**' is the name of the GDDR HFS dataset allocated and filled during the Installation process.

Activate this change via an IPL or perform an initial manual mount.

2.1.1.2 SYS1.PARMLIB(IKJTSOxx)

Add the following entries to the IKJTSOxx member of SYS1.PARMLIB:

```
To AUTHCMD add entries:
GDDR0SMF          /* GDDR SMF WRITER        */  +
SCFRDFME          /* EMC ME Utility         */  +
SCFRDFM6          /* EMC M6 Utility         */  +


To AUTHPGM add entries:
GDDR0SMF          /* GDDR SMF WRITER        */  +
SCFRDFME          /* EMC ME Utility         */  +
SCFRDFM6          /* EMC M6 Utility         */  +
```

EMCTF   /* EMC Timefinder Mirror   */ +

Activate this change via an IPL or dynamically change by using the TSO PARMLIB UPDATE(xx) command.

2.1.1.3 BSOFT.OPSvrm.CNTL(USSENVxx)

Add the following directories to the PATH and LIBPATH statements to the OPSUSS STC parameter file:

- /gddr/

- /gddr/source (temporary home for HMC API)

Activate this change by re-cycling the OPSUSS Started Procedures.

2.2 Protecting GDDR Resources

The following details how to define the security environment required by GDDR.

2.2.1 Define Owner for GDDR Resources

The following RACF group is required to be defined, GDDR$, it is used for the following:

- As an owning group for GDDR specific resources

ADDGROUP GDDR$ OWNER(SYS1) SUBGROUP(SYS1)

2.2.2 Define Group for GDDR Users

The following RACF group is required to be defined, GDDR$USR, it is used for the following:

- As a permit group to hold GDDR Users

ADDGROUP GDDR$USR OWNER(GDDR$) SUBGROUP(GDDR$)

2.2.3 Define the GDDR STC Userid

Define the RACF userid used by the RACF STC's by entering the following command:

adduser GDDR nooidcard nopassword name('GDDR STCs') owner(stc) dfltgrp(stc)

Give the GDDR userid a TSO Segment, this is required inorder to give the userid OPER authority by way of the TSOAUTH class:

Altuser GDDR tso(userdata(0000))

2.2.4 Associate STC Userid to GDDR Started Procedures

Enter the following commands to associate the GDDR STC RACF userid with the GDDR Started Procedures.

2.2.4.1 STC GDDRHBM – Heartbeat Monitor

Associate a RACF userid of GDDR to the GDDR Heartbeat Monitor started procedure GDDRHBM:

rdefine started GDDRHBM.* owner(stc) stdata(user(gddr),group(stc))

2.2.4.2 STC GDDREVM – Event Monitor

Associate a RACF userid of GDDR to the GDDR Event Monitor started procedure GDDREVM:

rdefine started GDDREVM.* owner(stc) stdata(user(gddr),group(stc))

2.2.4.3 Refresh RACF Instorage Profiles

After creating and/or updating the above RACF profiles, don't forget to refresh the instorage profiles:

- setr refresh raclist(started)

2.2.5 Grant RACF Access to Required Resources

The following RACF access is required by the GDDR RACF userid GDDR.

2.2.5.1 Permit access to Host Component Commands

- permit 'emc.validate.access' generic access(update) id(gddr)

The following RACF group and dataset profile are required before the above permission can be granted:

- ADDGROUP EMC OWNER(SYS1) SUPGROUP(SYS1)
- ADDSD 'EMC.VALIDATE.ACCESS' GENERIC UACC(NONE) OWNER(SYS1)

2.2.5.2 Permit Access to OPS/MVS Facilities

TSOAUTH access is required so that GDDR functions running with the GDDR RACF userid can modify CA-OPS/MVS Global variables.

The GDDR RACF userid requires a RACF TSO segment inorder to permit this access.

- Grant ACCESS(READ) to PROFILE OPER in class TSOAUTH to user GDDR
- Pe oper class(tsoauth) access(read) id(gddr)
- Setr refresh raclist(tsoauth)

2.2.5.3 Permit Access to GDDR Datasets

The GDDR RACF userid requires READ access to the GDDR SMP/E target datasets:

- Permit 'GDDR.*' access(read) id(gddr)

The GDDR RACF userid requires ALTER access to the GDDR global variables dataset inorder to make backup copies of its CA-OPS/MVS global variables:

- Permit 'gddr.global.vars' generic access(alter) id(gddr)

2.2.5.4 Permit Access to the ESS Interface

If GDDR is to trigger the ESS Interface, the GDDR RACF userid requires Update access to the dataset specified on the **CA7.XPSJCL** GDDR parameter.

PE 'ca7_xpsjcl_dataset' GENERIC ACCESS(UPDATE) ID(GDDR)

2.2.6 User Access to GDDR Resources

The following RACF permissions are required to authorise GDDR Administrators and GDDR Operators access to GDDR Resources.

Specifically access is required to:

- Access the GDDR Application
- Submit GDDR Scripts that run with the GDDR RACF userid.

2.2.6.1 GDDR Scripts

The RACF SURROGAT class profile GDDR.SUBMIT must be defined before GDDR Users can be permitted access:

```
RDEFINE SURROGAT GDDR.SUBMIT UACC(NONE) OWNER(GDDR$)
```

To allow GDDR Users to submit GDDR scripts that will run with the GDDR RACF userid the following permission is required:

```
PE GDDR.SUBMIT CLASS(SURROGAT) ACCESS(READ) ID(GDDR$USR)
```

2.2.6.2 GDDR SMP/E Target Libraries

The following RACF DATASET class profiles must be defined before GDDR users can be permitted access:

```
ADDSD 'hlq.GDDR.MSGS' GENERIC UACC(NONE) OWNER(GDDR$)
ADDSD 'hlq.GDDR.PANELS' GENERIC UACC(NONE) OWNER(GDDR$)
```

To allow GDDR Users access to the GDDR Application the following permission is required:

```
PE 'hlq.GDDR.MSGS' GENERIC ACCESS(READ) ID(GDDR$USR)
PE 'hlq.GDDR.PANELS' GENERIC ACCESS(READ) ID(GDDR$USR)
```

Where '**hlq**' is the chosen High Level Qualifier for the GDDR SMP/E Target libraries.

2.2.6.3 Grant Users Access To GDDR

To grant users access to the GDDR Application, connect the relevant RACF userids to RACF group GDDR$USR:

CONNECT tso_racf_userid GROUP(GDDR$USR)

2.3   GDDR as a CA-OPS/MVS User Application

The following updates are required inorder to make GDDR available from CA-OPS/MVS as an OPS/MVS User Application.

2.3.1   CA-OPS/MVS Invocation REXX - ZOPSMVS

Update the CA-OPS/MVS Invocation REXX to add the following GDDR datasets.

2.3.1.1   OPSPLIB

Add the following GDDR Panel library to the top of the OPSPLIB concatenation:

- GDDR.PANELS

2.3.1.2   OPSMLIB

Add the following GDDR Message library to the top of the OPSMLIB concatenation:

- GDDR.MSGS

2.4   GDDR AOF Rules

The following details how to make GDDR's AOF Rules available to CA-OPS/MVS.

2.4.1   CA-OPS/MVS GDDR AOF Rules

The GDDR supplied CA-OPS/MVS AOF rules must be copied to the appropriate PMSG.RULES dataset and enabled on every Production system that GDDR will manage.

They must also be copied and enabled on every GDDR K-System.

All GDDR AOF rules must also be Auto-Enabled as well as Enabled.

The GDDR supplied CA-OPS/MVS AOF rules can be found in GDDR SMP/E target library:

- GDDR.AOF.RULES

2.5 CA-OPS/MVS Changes

The following lists the changes required to CA-OPS/MVS.

2.5.1 CA-OPS/MVS System Changes

The following details the changes required to CA-OPS/MVS System Parameters.

2.5.1.1 Suppress Message OPS1370H

Message OPS1370H is produced as a result of OPSWTO and results in untidy displays of GDDR messages in the SYSLOG. This message should be suppressed inoder to make the SYSLOG and OPSLOG more readable as far as GDDR messages go.

Messages with the "H" security level cannot be suppressed using AOF Rules, so you have to do either of the following to change the security level of the message inorder for an AOF rule to take affect:

- In SYS1.PARMLIB(OPSSPAxx) add T=OPSPRM_SET("OPS1370", "O")

- To a OPS/MVS startup REXX add "OPSPARM SET(OPS1370) VALUE('O')"

2.5.1.2 Create GDDR_BUSINESS_APPL

On each Production system, a new RDF table will be defined in OPS/MVS, it will be named GDDR_BUSINESS_APPL, it will have one column only, that being JOBNAME. Create using OPS/MVS option 2.6.I (insert new table).

RDF table GDDR_BUSINESS_APPL will contain an entry for each Production application that GDDR must close down when performing planned action ALLSITEn. The JOBNAME must match that defined in the STCTBL.

Having a separate table on each system allows maximum flexibility, in that, the list of tasks to be managed by GDDR can be different on each system and don't have to be hard coded in GDDR.

At present, the only entry required is for a dummy application ACTEOC which is used as a pre-requisite for all business applications.

2.5.2 CA-OPS/MVS Rule Changes

The following changes to CA-OPS/MVS Rules are required inorder to allow GDDR to update Global variables and use the OPSCMD function.

2.5.2.1 PSEC.RULES(GLOBAL)

To allow GDDR authority to update GDDR CA-OPS.MVS global variables, add the following line into the PSEC rule named GLOBAL:

```
IF userid = "GDDR"   THEN RETURN "ACCEPT"
```

2.5.2.2 PSEC.RULES(OPSCMD)

To allow GDDR facilities running with the GDDR RACF userid to use the CA-OPS/MVS function OPSCMD add the following line to the PSEC rule named OPSCMD:

```
IF userid = "GDDR"   THEN RETURN "ACCEPT"
```

2.5.2.3 PSEC.RULES(OPSWTO)

To allow GDDR facilities running with the GDDR RACF userid to use the CA-OPS/MVS function OPSWTO add the following line to the PSEC rule named OPSWTO:

```
IF userid = "GDDR"   THEN RETURN "ACCEPT"
```

2.5.3 CA-OPS/MVS Started Procedures

The following CA-OPS/MVS started procedures are required to up updated to include required GDDR resources in their allocations.

These changes must only be made to CA-OPS/MVS started procedures on the GDDR K-Systems.

2.5.3.1 OPSMVS

Exact changes required are yet to be determined.

2.5.3.2 OPSOSF

To the OPSOSF CA-OPS/MVS started procedure, add the OPSEXEC DDNAME if not already used, specifying the following GDDR SMP/E target library:

- GDDR.OPSEXEC

If there is already an OPSEXEC DDNAME, add the following GDDR SMP/E target library to the top of the concatenation:

- GDDR.OPSEXEC

2.6   GDDR Started Procedures

The following describes how to customise and make available the following GDDR Started procedures:

- GDDREVM – GDDR Event Monitor

- GDDRHBM – GDDR Heart Beat Monitor

2.6.1   Make Started Procedures Available

Make the started procedures available by copying the following procedures from GDDR.PROCLIB to SYS1.PROCLIB or equivalent:

- GDDREVM

- GDDRHBM

2.6.1.1 Update SYSTSIN DDname

Update each started procedure so that the dataset name pointed at by the SYSTSIN DDname is pointing at the relevant input member in GDDR.PARMLIB:

- Member GDDREVM for the GDDR Event Monitor

- Member GDDRHBM for the GDDR Heartbeat Monitor

2.6.2   Define Started Procedures to OPS/MVS SSM

To place the stop and start of the GDDR Monitor started procedures under the control of OPS/MVS, make the following updates to OPS/MVS SSM.

2.6.2.1 STCTBL

In the RDF table STCTBL add or modify existing entries, such that only the following entries exist for the GDDR started procedures:

Name/Jobname	Desired_State	Mode	Type	PREREQ
GDDREVM	UP	A	GDDREVM	JES2 OPSMVS
GDDRHBM	UP	A	GDDRHBM	JES2 OPSMVS

2.6.2.2 STCTBL_ACT

In the RDF table STCTBL_ACT add or modify exisiting entries, such that only the following entries exist for the GDDR started procedures:

ACTION_CURRENT	ACTION_DESIRED	ACTION_RES_TYPE	A C T I O N _ T E X T
DOWN	UP	GDDREVM	…START &JOBNAME
DOWN	UP	GDDRHBM	…START &JOBNAME
UP	DOWN	GDDREVM	…CANCEL &JOBNAME

UP	DOWN	GDDRHBM	...CANCEL &JOBNAME

3. APPENDIX

3.1 OPS/MVS SSM

The following describes how to setup the EMC started tasks in OPS/MVS SSM.

3.1.1 Update SSM Tables

On each Production system and K-System the following must be added or updated in the indicated RDF Tables:

3.1.1.1 STCTBL

In the RDF table STCTBL add or modify existing entries, such that only the following entries exist for the EMC software products:

Name/Jobname	Desired_State	Mode	Type	PREREQ
EMCCGRP	UP	A	EMCCGRP	JES2 EMCRSPAK
EMCINIT	UP	A	EMCINIT	JES2 EMCRSPAK
EMCRSPAK	UP	A	EMCRSPAK	JES2

Note: On the K3 system, the mode for EMCCGRP must be set to "I" as there is no need to run Consistency Group at DC3.

3.1.1.2 STCTBL_ACT

In the RDF table STCTBL_ACT add or modify exisiting entries, such that only the following entries exist for the EMC software products:

ACTION_CURRENT	ACTION_DESIRED	ACTION_RES_TYPE	A C T I O N _ T E X T
DOWN	UP	EMCCGRP	…START &JOBNAME,PRM=DCn,SUB=MSTR
DOWN	UP	EMCINIT	…START &JOBNAME
DOWN	UP	EMCRSPAK	…START &JOBNAME,SUB=MSTR
UP	DOWN	EMCCGRP	…STOP &JOBNAME
UP	DOWN	EMCINIT	…STOP &JOBNAME
UP	DOWN	EMCRSPAK	…F &JOBNAME,INI,SHUTDOWN