ECS Configuration Change Request

1. Originator	2. Log Date:	3. CCR #:	4. Rev:	5. Tel:	6. Rm #:	7. Dept.
Gary Lake	5/4/00	00-0491		301-614-5751	N126-6	Sys Eng.

8. CCR Title: DUE to Collect Daily DPR Statistics Data

9. Originator Signature/Date	10. Class	11. Type:	12. Need Date:
Gary Lake 5/2/2000	II	CCR	May 12, 2000

13. Office Manager Signature/Date	14. Category of Change:	15. Priority: (If "Emergency" fill in Block 28).
[signature] 2 May 2000	Initial ECS Baseline Doc.	Routine

16. Documentation/Drawings Impacted:	17. Schedule Impact:	18. CI(s) Affected:
None		PLS

19. Release Affected by this Change:	20. Date due to Customer:	21. Estimated Cost:
5A		None - Under 100K

22. Source Reference: ☐NCR (attach) ☐Action Item ☐Tech Ref. ☒GSFC ☐Other:
DUE-007

23. Problem: (use additional Sheets if necessary)
We do not have a storage location for recording daily DPR event logging information.

24. Proposed Solution: (use additional sheets if necessary)
Two new tables will be added to the pdps database to store data on the total number of DPRs completed during one day time periods, and the totals for the corresponding CPU usage parameters. Also, a table will be added to store descriptions of daily production events. A stored procedure will be written in Sybase's Transact SQL to select sums of values from the PIDprMetrics table in the pdps database from the preceding day. The results of this query will be stored in a new pdps database table. This operation will be repeated once every day by cron.

25. Alternate Solution: (use additional sheets if necessary)
A similar solution can be implemented to that which is proposed, however, instead of storing the data in a table in the pdps database, the data may be stored in a newly created database. The advantage to this approach is in the complete separation of the new code from existing code.

26. Consequences if Change(s) are not approved: (use additional sheets if necessary)
We will either record the DPR production information in paper logs, Excel spreadsheets, or not record the information. Recorded in this way, the information is not easily interpreted, and the cost of information analyis in person-hours is significantly increased.

27. Justification for Emergency (If Block 15 is "Emergency"):

28. Site(s) Affected: ☐EDF ☐PVC ☐VATC ☐EDC ☒GSFC ☐LaRC ☐NSIDC ☐SMC ☐AK ☐JPL ☐EOC ☐IDG Test Cell ☐Other

29. Board Comments:	30. Work Assigned To:	31. CCR Closed Date:
Approved as DUE - No impacts		

32. EDF/SCDV CCB Chair (Sign/Date):	Disposition: Approved App/Com. Disapproved Withdraw Fwd/ESDIS ERB Fwd/ECS
33. M&O CCB Chair (Sign/Date): [signature] 12 May 00	Disposition: Approved App/Com. Disapproved Withdraw Fwd/ESDIS ERB Fwd/ECS
34. ECS CCB Chair (Sign/Date):	Disposition: Approved App/Com. Disapproved Withdraw Fwd/ESDIS ERB Fwd/ESDIS

ECS/EDF/SCDV/M&O

CM01JA00

ORIGINAL

ADDITIONAL SHEET

CCR #: CO-0191 **Rev:** **Originator:** Gary Lake

Telephone: 301-614-5751 **Office:** N126-6

Title of Change: DUE to collect DPR Statistics Data

Proposed solution (cont.) The procedure to be executed will be contained in a file that is listed as a new entry in the crontab with the appropriate specifications for the execution times.

The following SQL code may be used to create the required table in the pdps database, and to store data on a periodic basis. The 3MB per year sizing estimate is based on a sampling rate of 1 execution per hour of the stored procedure that contains the record insert commands. If the number of PGE's doubles, then the number of MB per year will double.

Component Description

```
PIDprMetrics
create table PIDprMetrics (pgeName varchar(17) not null,
 HourlySumPgeCpuTime float null,
  HourlySumSystemCpuTime float null,
   HourlySumPgeElapsedTime float null,
    dprIdCount int null,
     SampleStartTime  datetime not null,
      SampleEndTime datetime not null,
 primary key (SampleStartTime, pgeName))

GetDPRStatistics

# BE SURE THAT EXEC PRIVILEGES HAVE BEEN PROPERLY
# GRANTED TO $DBUSERNAME FOR PROCEDURE GetLatestDPRStatistics
DBUSERNAME=<user name>
DBPASSWD= <password>

SYBASE= <file path>/sybase
DSQUERY= g0pls01_srvr
$SYBASE/bin/isql -U$DBUSERNAME -P$DBPASSWD
go
exec GetLatestDPRStatistics
go
EOF
3.4.3 GetLatestDPRStatistics
Create Proc GetLatestDPRStatistics
as
insert PIDprMetrics
 select "DPREP",sum(pgeCpuTime),sum(systemCpuTime),
sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -1,getdate()), getdate()
  from PIDprCompletion
   where dprId like '%Att%' or dprId like '%Eph%'
    and completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()

insert PIDprMetrics
 select "PGE01",sum(pgeCpuTime),sum(systemCpuTime), sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -1,getdate()),
getdate()
  from PIDprCompletion
   where dprId like '%PGE01%'
    and completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()
```

ORIGINAL

```
insert PIDprMetrics
 select "PGE02",sum(pgeCpuTime),sum(systemCpuTime), sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -1,getdate()),
getdate()
  from PIDprCompletion
   where dprId like '%PGE02#%'
    and completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()

insert PIDprMetrics
 select "PGE02A",sum(pgeCpuTime),sum(systemCpuTime), sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -
1,getdate()), getdate()
  from PIDprCompletion
   where dprId like '%PGE02A%'
    and completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()

insert PIDprMetrics
 select "PGE03",sum(pgeCpuTime),sum(systemCpuTime), sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -1,
getdate()), getdate()
  from PIDprCompletion
   where dprId like '%PGE03%'
    and completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()

insert PIDprMetrics
 select "PGE057",sum(pgeCpuTime),sum(systemCpuTime), sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -1,
getdate()), getdate()
  from PIDprCompletion
   where dprId like '%PGE057%'
    and completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()

insert PIDprMetrics
 select "PGE Hourly Totals",sum(pgeCpuTime),sum(systemCpuTime), sum(pgeElapsedTime),count(distinct dprId), dateadd(hh, -
1, getdate()), getdate()
  from PIDprCompletion
   where completionDate > dateadd(hh, -1,getdate())
     and completionDate < getdate()
return
```

CM01AJA00