



usp\_DBA\_LogError.txt

```

--Print input params
if @iDebug = 1
    begin
        print @piAppID print @piUspID print @piErrNum
    print @piStep print @piRowCnt
    end
-----
--Check input values & remove NULLS
if @piAppID is null set @piAppID = 0    --Not in app tbl
if @piUspID is null set @piUspID = 0    --Set uspname later
if @piErrNum is null set @piErrNum = 999999    --Dummy value to
signify NULL
if @piStep is null set @piStep = 0
if @piRowCnt is null set @piRowCnt = 0

set @dtErrTime = getdate()

set @vcDBUser = (select user)
if @vcDBUser = null set @vcDBUser = 'Unknown'

set @vcLogin = (select SYSTEM_USER)
if @vcLogin is null set @vcLogin = 'Unknown'

set @vcSvrName = left(@@SERVERNAME ,30) --which SQL svr
if @vcSvrName is null set @vcSvrName = 'Unknown'

set @vcDBName = left(db_name(),50)    --which DB
if @vcDBName is null set @vcDBName = 'Unknown'

--Current sys process ID
set @iSPID =(SELECT @@SPID)
if @iSPID is null set @iSPID = 0

--Find exe / program name
set @vcExeName = (select program_name from master.dbo.sysprocesses
where SPID=@iSPID)
if @vcExeName is null set @vcExeName = 'Unknown'

--PC executed from
set @vcHostName = (select hostname from master.dbo.sysprocesses
where SPID=@iSPID)
if @vcHostName is null set @vcHostName = 'Unknown'

--Find sp name
IF @piUspID = 0
    set @vcUspName = 'UNKNOWN'
ELSE
    set @vcUspName = (select name from sysobjects where id =
@piUspID)
-----
IF @@error > 0 GOTO ERR_HANDLER -- ie setting the var values
failed somewhere, so don't log.
BEGIN

```

```

                                usp_DBA_LogError.txt
--Find err msg
select @vcErrMsg=
      case
ID'                                when @piErrNum = 99999 then 'NULL error'
                                    when @piErrNum = 0 then 'Success'
                                    when @piErrNum >0 and @piErrNum < 50001
then (select description from master.dbo.sysmessages where error =
@piErrNum)
                                    when @piErrNum >50001 then (select descr
from BusinessLogicMessages where BLM_ID = @piErrNum)
      else 'none'
      end
END
-----
if @iDebug =1
    BEGIN
        print @dtErrTime print @vcDBUser print @vcLogin
print @vcSvrName print @vcDBName
        print @iSPID print @vcExeName print @vcHostName
print @vcUspName print @vcErrMsg
    END
-----
IF @@error > 0 GOTO ERR_HANDLER -- ie setting the errmsg or debug
failed somewhere, so don't log.
BEGIN
    insert into SQLErrorLog
    (
    AppID,
    ExeName,
    HostName,
    dbuser,
    Login,
    usp,
    ErrTime,
    ErrNum,
    ErrMsg,
    Step,
    Rowcnt,
    SvrName,
    DBname
    )
    values
    (
    @piAppID,
    @vcExeName,
    @vcHostName,
    @vcDBUser,
    @vcLogin,
    @vcUspName,
    @dtErrTime,
    @piErrNum,
    @vcErrMsg,
    @piStep,

```

usp\_DBA\_LogError.txt

```
@piRowCnt,  
@vcSvrName,  
@vcDBname  
)
```

END

```
-----  
-----  
IF @@error > 0 GOTO ERR_HANDLER    --Insert into SQL error log  
failed, thus no errlogID to rtn.
```

BEGIN

```
--Rtn err log ID to calling proc.  
--No NULLS or breaks comparisons.  
select @iErrLogID = ErrorId  from SQLErrorLog  
where  
AppID=@piAppID and  
ExeName=@vcExeName and  
HostName=@vcHostName and  
dbuser=@vcDBuser and  
usp=@vcUspName and  
ErrTime=@dtErrTime and  
ErrNum=@piErrNum and  
Step=@piStep and  
Rowcnt=@piRowCnt
```

END

```
-----  
-----  
IF @@error > 0 OR @@rowcount <> 1 GOTO ERR_HANDLER    --Could not  
find SQL error log ID.
```

BEGIN

```
Return @iErrLogID
```

END

```
-----  
-----  
ERR_HANDLER:
```

```
--Failed  
return -1
```

GO