ProcGetAltitudeResolution**Procedure ProcGetAltitudeResolution**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetAltitudeResolution            */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                              */
/*  Created on      :  3/25/97                              */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                  data from DsMdAltitudeResolution        */
/*                  table                                   */
/*  Calls           :  None                                 */
/*==============================================================*/

CREATE PROCEDURE ProcGetAltitudeResolution(@collectionId  ID = 0)
AS
BEGIN
IF (@collectionId = 0)
   BEGIN
     RAISERROR 40025
     RETURN -999
   END
ELSE
  BEGIN
  SELECT collectionId,
        AltitudeResolution
  FROM   DsMdAltitudeResolution
  WHERE  collectionId = @collectionId
  END
  IF @@error!=0
  BEGIN
     RAISERROR 45026
     RETURN -999
  END
END
go
```

ProcGetAltSysDef**Procedure ProcGetAltSysDef**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetAltSysDef                     */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                              */
/*  Created on      :  3/25/97                              */
/*  Purpose         :  Create Stored Procedure for retriving */
/*                  data from DsMdSpatial table             */
/*                                                          */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcGetAltSysDef(@collectionId ID = 0)
AS
BEGIN
IF (@collectionId = 0)
   BEGIN
     RAISERROR 40024
     RETURN -999
   END
ELSE
  BEGIN
   SELECT AltitudeDatumName,
        AltitudeDistanceUnits,
        AltitudeEncodingMethod
   FROM   DsMdSpatial
   WHERE  collectionId = @collectionId

   IF (@@error!= 0)
   BEGIN
         RAISERROR 33000 "ProcGetAltSysDef: Failed select for Altitude info"
         RETURN -999
   END

   SELECT AltitudeResolution
   FROM   DsMdAltitudeResolution
   WHERE  collectionId = @collectionId
  END
IF (@@error!= 0)
   BEGIN
         RAISERROR 45025
         RETURN -999
   END
END
go
```

ProcGetAnalysisSource**Procedure ProcGetAnalysisSource**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetAnalysisSource                */
/*  Release         :  B0                                   */
/*  Module          :  DataOriginator                       */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for selecting */
/*                  data from DsMdAnalysisSource            */
/*                  table                                   */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/

CREATE PROC ProcGetAnalysisSource(@dbID  ID = 0)
AS
```

```
BEGIN
IF (@dbID = 0)
   BEGIN
     RAISERROR 40026
     RETURN -999
   END
ELSE
  BEGIN
  SELECT
        AnalysisShortName,
        AnalysisLongName,
        AnalysisTechnique,
        AnalysisType
  FROM   DsMdAnalysisSource
  WHERE  dbID = @dbID
  END
IF @@error!=0
   BEGIN
         RAISERROR 45027
         RETURN -999
   END
END
go
```

ProcGetAncillaryInput**Procedure ProcGetAncillaryInput**

Code

```
/*================================================================*/
/*  Name          :  ProcGetAncillaryInput                        */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                ancillary input data associated with            */
/*                Granules.                                       */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetAncillaryInput(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
   BEGIN
     RAISERROR 40027
     RETURN -999
   END
 ELSE
   BEGIN
         SELECT AncillaryInputType,
       AncillaryInputPointer
      FROM   DsMdAncillaryInput
         WHERE  granuleId = @granuleId
   END
IF @@error!=0
  BEGIN
```

```
    RAISERROR 45028
    RETURN -999
  END
END
go
```

ProcGetAPAssocColl**Procedure ProcGetAPAssocColl**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAPAssocColl                */
/*  Release       :  B0                                */
/*  Module        :  Delivered Algorithm Package       */
/*  Created on    :  4/1/97                            */
/*  Purpose       :  Create stored procedure to get    */
/*                DsMdCollectionAPXref                 */
/*  Calls         :  None                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetAPAssocColl
(
   @apId        ID
)
AS
BEGIN
  SELECT a.ShortName,
         a.VersionID
  FROM   DsMdCollections a,
       DsMdCollectionAPXref b
  WHERE  a.dbID = b.collectionId
  AND    b.apId = @apId

   IF @@error!=0
   BEGIN
        RAISERROR 45001
        RETURN -999
   END
END
go
```

ProcGetAPContactXref**Procedure ProcGetAPContactXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAPContactXref              */
/*  Release       :  B0                                */
/*  Module        :  Delivered Algorithm Package       */
/*  Created on    :  4/1/97                            */
/*  Purpose       :  Create stored procedure to Delivered,   */
/*                Algorithm Package, and associate data   */
/*                tables.                              */
/*  Calls         :                                    */
/*                                                     */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcGetAPContactXref( @dapId  ID = 0)
AS
BEGIN
  IF (@dapId = 0)
    BEGIN
      RAISERROR 40020
      RETURN -999
    END
   ELSE
    BEGIN
     SELECT dapId, contactId
     FROM DsMdApContactXref
     WHERE dapId = @dapId
    END
IF @@error!=0
  BEGIN
    RAISERROR 45020
    RETURN -999
  END
END
go
```

ProcGetAPSAPComponentXref**Procedure ProcGetAPSAPComponentXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAPSAPComponentXref              */
/*  Release       :  B0                                     */
/*  Module        :  Delivered Algorithm Package            */
/*  Created on    :  4/1/97                                 */
/*  Purpose       :  Create stored procedure to Delivered,   */
/*                  Algorithm Package, and associate data   */
/*                  tables.                                 */
/*  Calls         :                                         */
/*                                                        */
/*==============================================================*/

CREATE PROCEDURE ProcGetAPSAPComponentXref(@apId ID = 0)
AS
BEGIN
 IF (@apId = 0)
    BEGIN
      RAISERROR 40021
      RETURN -999
    END
 ELSE
   BEGIN
    SELECT apId, ssapId
    FROM   DsMdAPSAPComponentXref
    WHERE  apId = @apId
   END
    IF @@error!=0
    BEGIN
```

```
        RAISERROR 45021
        RETURN -999
   END
END
go
```

ProcGetAssocGranules**Procedure ProcGetAssocGranules**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAssocGranules                       */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :                                             */
/* The purpose of this stored proc is to identify  submited   */
/* granule ( ASSUMPTION: granules are identified by BaseType, */
/* Subtype, ID) and return any granules associated by a       */
/* FOREIGN KEY reference.                                       */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROC ProcGetAssocGranules
(
        @baseType varchar(60) = NULL,
      @subType varchar(30) = NULL,
      @id ID = NULL
)
AS
BEGIN

DECLARE @retStat  int
DECLARE @lookupValue varchar(30)
DECLARE @type varchar(30)
DECLARE @myBaseType varchar(60)
DECLARE @myProcHistoryID ID

IF (@baseType IS NULL)
  BEGIN
    RAISERROR  40028
    RETURN -999

  END

/*
  Determine Base Table FROM BaseType
  NOTE: Since DsMdGranules(  SCIENCE GRANULES) is the only table
  with foreign key pointers, "Science" is the only type that needs to
  be tested for. This may change for future requirements.
*/

-- note that 2nd argument is only a lookup value
```

```
  EXECUTE @retStat = ProcGetSDSRVLookup "type", "SC", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not find parameter @type first
call."
      RETURN -999
    END

  SELECT @type = @lookupValue

-- its a science baseType
IF (@baseType=@type)
   BEGIN
   /*
      Test each pointer and return the corresponding handle(baseType,subType,ID)
   */

     CREATE TABLE #tempGeOIDResult
         (BaseType varchar(60) null,
          SubType  varchar(30) null,
          dbID     numeric null)

  IF @@error != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not create temp table."
      RETURN -999
    END

  EXECUTE @retStat = ProcGetSDSRVLookup "type", "BR", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not find parameter @type for
Browse."
      RETURN -999
    END

  SELECT @type = @lookupValue

  /* Browse data to Temp */
      INSERT INTO   #tempGeOIDResult
       SELECT "BaseType"=CONVERT(varchar, @type),
            subType,
            dbID
       FROM   DsMdBrowse b,
            DsMdBrowseGranuleXref x
       WHERE
                x.granuleId=@id
         AND  x.browseId = b.dbID

      IF @@error != 0
        BEGIN
          RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not insert DsMdBrowse into
```

```
#tempGeOIDResult temp table."
        RETURN -999
       END

  /* Get Processing History ID */
     SELECT
        @myProcHistoryID = processingHistoryId
     FROM DsMdGranules
     WHERE
          dbID = @id

  EXECUTE @retStat = ProcGetSDSRVLookup "type", "PH", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
     RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not find parameter @type for
Production History."
     RETURN -999
    END

  SELECT @type = @lookupValue

  /* Processinng History Data to Temp */
     IF (@myProcHistoryID is NOT NULL)
        INSERT INTO   #tempGeOIDResult
        SELECT "BaseType"=CONVERT(varchar, @type),
             subType,
             dbID
        FROM   DsMdProcessingHistory
        WHERE  dbID=@myProcHistoryID

     IF @@error != 0
       BEGIN
        RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not insert
DsMdProcessingHistory into #tempGeOIDResult temp table."
        RETURN -999
       END

  EXECUTE @retStat = ProcGetSDSRVLookup "type", "QA", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
     RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not find parameter @type for
QA."
     RETURN -999
    END

  SELECT @type = @lookupValue

   /* QA Data to Temp  */
     INSERT INTO   #tempGeOIDResult
       SELECT "BaseType"=CONVERT(varchar, @type),
            subType,
            dbID
```

```
      FROM   DsMdQaGranule
      WHERE  granuleId=@id

    IF @@error != 0
      BEGIN
        RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not insert DsMdQaGranule
into #tempGeOIDResult temp table."
        RETURN -999
      END

  EXECUTE @retStat = ProcGetSDSRVLookup "type", "AN", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not find parameter @type for
Ancillary Input."
      RETURN -999
    END

  SELECT @type = @lookupValue

   /* Ancillary Data to Temp  */
     INSERT INTO   #tempGeOIDResult
        SELECT "BaseType"=CONVERT(varchar, @type),
             subType,
             dbID
        FROM   DsMdAncillaryInputGranule a,
                    DsMdGranulesAncillaryInputXref x
        WHERE
                x.granuleId=@id
          AND  x.ancillaryInputGranuleId = a.dbID

     IF @@error != 0
       BEGIN
         RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not insert
DsMdAncillaryInputGranule into #tempGeOIDResult temp table."
         RETURN -999
       END

  EXECUTE @retStat = ProcGetSDSRVLookup "type", "OR", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not find parameter @type for
Orbit Parameters."
      RETURN -999
    END

  SELECT @type = @lookupValue

   /* OrbitParameter Data to Temp */
     INSERT INTO   #tempGeOIDResult
        SELECT "BaseType"=CONVERT(varchar, @type),
             subType,
```

```
            dbID
        FROM  DsMdOrbitParametersGranule o,
                DsMdGranulesOrbitParmXref x
        WHERE x.granuleId = @id
            AND x.orbitParamGranuleId = o.dbID

    IF @@error != 0
      BEGIN
        RAISERROR 20014 "Procedure ProcGetAssocGranules error - could not insert
DsMdOrbitParametersGranule into #tempGeOIDResult temp table."
        RETURN -999
      END

    SELECT BaseType, SubType, CONVERT(varchar(30),dbID)
    FROM #tempGeOIDResult
    ORDER by BaseType, SubType

  END
END
go
```

ProcGetAssocPlatInstrSensor**Procedure ProcGetAssocPlatInstrSensor**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAssocPlatInstrSensor            */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                   platform, instrument, sensor data      */
/*                   for granules.                          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcGetAssocPlatInstrSensor(@granuleId   ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40029
      RETURN -999
    END
  SELECT platformId   = CONVERT(varchar(40), t1.dbID),
        instrumentId = CONVERT(varchar(40), t2.dbID),
        sensorId     = CONVERT(varchar(40), t3.dbID),
        t1.PlatformShortName,
        t2.InstrumentShortName,
        t5.OperationMode,
        t3.SensorShortName
  FROM   DsMdPlatform t1,
            DsMdInstrument t2,
            DsMdSensor t3,
```

```
      DsMdGranuleSensorXref t4,
          DsMdGranuleInstrumentXref t5
WHERE
      t2.platformId   = t1.dbID          AND
      t3.instrumentId = t2.dbID          AND
      t4.sensorId     = t3.dbID          AND
      t5.instrumentId = t2.dbID          AND
      t3.instrumentId = t5.instrumentId  AND
      t5.granuleId    = t4.granuleId     AND
          t4.granuleId    = @granuleId       AND
      t5.granuleId    = @granuleId
GROUP BY t1.dbID,
             t2.dbID,
             t3.dbID,
        t1.PlatformShortName,
             t2.InstrumentShortName,
        t5.OperationMode,
             t3.SensorShortName

IF @@error != 0
BEGIN
  RAISERROR 45031
  RETURN -999
END

SELECT platformId   = CONVERT(varchar(40), t1.dbID),
      instrumentId = CONVERT(varchar(40), t2.dbID),
      sensorId     = NULL,
      t1.PlatformShortName,
      t2.InstrumentShortName,
      t3.OperationMode,
      SensorShortName = NULL
FROM   DsMdPlatform t1, DsMdInstrument t2, DsMdGranuleInstrumentXref t3
WHERE  t3.granuleId    = @granuleId    AND
      t3.instrumentId = t2.dbID       AND
      t2.platformId   = t1.dbID

IF @@error != 0
BEGIN
  RAISERROR 45032
  RETURN -999
END


SELECT platformId   = CONVERT(varchar(40), t1.dbID),
      instrumentId = NULL,
      sensorId     = NULL,
      t1.PlatformShortName,
      InstrumentShortName = NULL,
      OperationMode       = NULL,
      SensorShortName     = NULL
FROM   DsMdPlatform t1, DsMdGranulePlatformXref t2
WHERE  t2.granuleId = @granuleId    AND
      t1.dbID = t2.platformId
```

```
  IF @@error != 0
  BEGIN
    RAISERROR 45033
    RETURN -999
  END
END
go
```

ProcGetAttributeList**Procedure ProcGetAttributeList**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAttributeList                       */
/*  Release       :  B0                                         */
/*  Module        :  System Tables                              */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  select from DsMdAttributeList              */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROC ProcGetAttributeList
AS
SELECT          listName,
          glType,
          attributeName
FROM
          DsMdAttributeList
IF @@error!=0
  BEGIN
          RAISERROR 45036
          RETURN -999
  END
go
```

ProcGetAttrTable**Procedure ProcGetAttrTable**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetAttrTable                           */
/*  Release       :  B0                                         */
/*  Module        :  System Tables                              */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  selects from DsMdAttributeTableXref        */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetAttrTable
AS
SELECT
```

```
        attributeName,
        productName,
        tableName,
        columnName
FROM
        DsMdAttributeTableXref
IF @@error!=0
  BEGIN
        RAISERROR 45034
        RETURN -999
  END
go
```

ProcGetAttrTableCnt**Procedure ProcGetAttrTableCnt**

Code

```
/*================================================================*/
/*  Name          :  ProcGetAttrTableCnt                  */
/*  Release       :  B0                                */
/*  Module        :  System Tables                        */
/*  Created on    :  3/24/97                           */
/*  Purpose       :  selects from GetAttrTableCnt          */
/*                                                     */
/*  Calls         :  None                              */
/*================================================================*/

CREATE PROCEDURE ProcGetAttrTableCnt
AS
SELECT
        COUNT(*)
FROM
        DsMdAttributeTableXref
IF @@error!=0
  BEGIN
        RAISERROR 45035
        RETURN -999
  END
go
```

ProcGetBrowseGranuleXref**Procedure ProcGetBrowseGranuleXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetBrowseGranuleXref              */
/*  Release       :  B0                                */
/*  Module        :  Granules                             */
/*  Created on    :  04/17/97                             */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                the browseIds associated with a Granule.   */
/*  Calls         :  ProcGetSDSRVLookup                    */
/*================================================================*/
```

```
CREATE PROCEDURE ProcGetBrowseGranuleXref(@granuleId ID = 0)
AS
BEGIN

  DECLARE @retStat int
  DECLARE @lookupValue varchar(40)
  DECLARE @subType varchar(30)
  DECLARE @type varchar(30)

  IF (@granuleId = 0)
    BEGIN
       RAISERROR 40115
       RETURN -999
    END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "subType", "Browse", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetBrowseGranuleXref error - could not find parameter
@subType."
      RETURN -999
    END

  SELECT @subType = @lookupValue

  EXECUTE @retStat = ProcGetSDSRVLookup "type", "BR", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetBrowseGranuleXref error - could not find parameter @type."
      RETURN -999
    END

  SELECT @type = @lookupValue

  SELECT CONVERT(varchar, @type) +
        ":" +
        CONVERT(varchar, @subType) +
        ":" +
        CONVERT(varchar(50), browseId)
       FROM   DsMdBrowseGranuleXref
       WHERE  granuleId = @granuleId

  IF (@@error!=0)
   BEGIN
      RAISERROR 45141
      RETURN -999
   END
END
go
```

ProcGetBrowseMetadata**Procedure ProcGetBrowseMetadata**

Code

```
/*================================================================*/
/*  Name          :  ProcGetBrowseMetadata                        */
/*  Release       :  B0                                 */
/*  Module        :  Granules                             */
/*  Created on    :  04/17/97                              */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                 browse data for Granules.                */
/*  Calls         :  ProcGetFileStorage                     */
/*                 ProcGetSDSRVLookup                       */
/*================================================================*/

CREATE PROCEDURE ProcGetBrowseMetadata(@dbID ID = 0)
AS
BEGIN

  DECLARE @retStat int
  DECLARE @lookupValue varchar(40)
  DECLARE @subType varchar(30)
  DECLARE @type varchar(30)

  IF (@dbID = 0)
    BEGIN
      RAISERROR 40031
      RETURN -999
    END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "subType", "Browse", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetBrowseMetadata error - could not find parameter
@subType."
      RETURN -999
    END

  SELECT @subType = @lookupValue

  SELECT  subType = @subType,
          dbID = CONVERT(varchar(30), dbID),
          CONVERT(varchar(60), lastUpdate, 109),
          BrowsePointer,
          BrowseDescription,
          BrowseSize
   FROM   DsMdBrowse
   WHERE  dbID = @dbID


   IF @@error!=0 or @@rowcount != 1
```

```
  BEGIN
    RAISERROR 45038
    RETURN -999
  END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "type", "BR", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
   BEGIN
    RAISERROR 20014 "Procedure ProcGetBrowseMetadata error - could not find parameter @type ."
    RETURN -999
   END

  SELECT @type = @lookupValue

  SELECT       ShortName,
               VersionID = CONVERT(int, VersionID)
   FROM        DsMdCollections
   WHERE       type = @type

   IF @@error!=0 or @@rowcount != 1
   BEGIN
    RAISERROR 45039
    RETURN -999
   END

   EXECUTE @retStat = ProcGetFileStorage @dbID

   IF @retStat != 0
   BEGIN
    RAISERROR 32000 "ProcGetBrowseMetadata: Failed select on DsMdFileStorage"
    RETURN -999
   END

END
go
```

ProcGetCampaign**Procedure ProcGetCampaign**

Code

```
/*================================================================*/
/*  Name            :  ProcGetCampaign                            */
/*  Release         :  B0                                         */
/*  Module          :  DataOriginator                             */
/*  Created on      :  3/30/97                                    */
/*  Purpose         :  Create Stored Procedure for selecting      */
/*                     data from DsMdCampaign                     */
/*                     table                                      */
/*                                                                */
/*  Calls           :  None                                       */
/*                                                                */
/*================================================================*/
```

```
CREATE PROC ProcGetCampaign(@dbID ID = 0)
AS
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40033
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT CampaignShortName,
         CampaignLongName,
         CampaignStartDate,
         CampaignEndDate
   FROM   DsMdCampaign
   WHERE  dbID = @dbID
 END
    IF @@error!=0
    BEGIN
           RAISERROR 45042
           RETURN -999
    END
END
go
```

ProcGetCollAddnlAttribsXref**Procedure ProcGetCollAddnlAttribsXref**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetCollAddnlAttribsXref                  */
/*  Release       :  B0                                        */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                 the additional attribute cross reference data */
/*                 associated with Collections.                 */
/*  Calls         :                                          */
/*=================================================================*/

CREATE PROCEDURE ProcGetCollAddnlAttribsXref(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40035
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT collectionId,
            attributeId
   FROM   DsMdCollectionAddnlAttribsXref
```

```
  WHERE  collectionId = @collectionId
 END

  IF @@error!=0
  BEGIN
    RAISERROR 45044
    RETURN -999
  END
END
go
```

ProcGetCollAnalysisXref**Procedure ProcGetCollAnalysisXref**

Code

```
/*=====================================================================*/
/*  Name          :  ProcGetCollAnalysisXref                           */
/*  Release       :  B0                                                */
/*  Module        :  Collections                                       */
/*  Created on    :  3/25/97                                           */
/*  Purpose       :  Create stored procedure to retrieve               */
/*                   the analysisSourceId associated with              */
/*                   Collections.                                      */
/*  Calls         :                                                    */
/*=====================================================================*/

CREATE PROCEDURE ProcGetCollAnalysisXref(@collectionId ID = 0)
AS
BEGIN
 IF (@collectionId = 0)
   BEGIN
    RAISERROR 40036
    RETURN -999
   END
 ELSE
  BEGIN
  SELECT collectionId,
         analysisSourceId
  FROM   DsMdCollectionAnalysisXref
  WHERE  collectionId = @collectionId
 END

  IF @@error!=0
  BEGIN
    RAISERROR 45045
    RETURN -999
  END
END
go
```

ProcGetCollAPXref**Procedure ProcGetCollAPXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetCollAPXref                            */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the apId associated with Collections.        */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetCollAPXref(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40034
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT
            CONVERT(varchar(30), apId)
   FROM
            DsMdCollectionAPXref
   WHERE
            collectionId = @collectionId
  END

   IF @@error!=0
   BEGIN
     RAISERROR 45043
   RETURN -999
   END
END
go
```

ProcGetCollAssociation**Procedure ProcGetCollAssociation**

Code

```
/*================================================================*/
/*  Name          :  ProcGetCollAssociation                       */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   association data for Collections.            */
/*  Calls         :                                               */
/*================================================================*/
```

```
CREATE PROCEDURE ProcGetCollAssociation(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40037
      RETURN -999
    END
  ELSE
    BEGIN

    SELECT AssociatedShortName,
              AssociatedVersionId,
          CollectionType,
          CollectionUse1,
          CollectionUse2
    FROM   DsMdCollectionAssociation
    WHERE  collectionId = @collectionId
  END

    IF @@error!=0
    BEGIN
      RAISERROR 45046
      RETURN -999
    END
END
go
```

ProcGetCollBrowseXref**Procedure ProcGetCollBrowseXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetCollBrowseXref                        */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                the browseId associated with Collections.  */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetCollBrowseXref(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40038
      RETURN -999
    END
  ELSE
    BEGIN
```

```
  SELECT
          CONVERT(varchar(30), browseId)
  FROM
          DsMdCollBrowseXref
  WHERE
          collectionId = @collectionId
 END
  IF @@error!=0
  BEGIN
    RAISERROR 45047
    RETURN -999
  END
END
go
```

ProcGetCollCampaignXref**Procedure ProcGetCollCampaignXref**

Code

```
/*=====================================================================*/
/*  Name          :  ProcGetCollCampaignXref                  */
/*  Release       :  B0                                    */
/*  Module        :  Collections                             */
/*  Created on    :  3/25/97                                 */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                 the campaignId associated with Collections.   */
/*  Calls         :                                        */
/*=====================================================================*/

CREATE PROCEDURE ProcGetCollCampaignXref(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40039
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT collectionId,
          campaignId
   FROM   DsMdCollectionCampaignXref
   WHERE  collectionId = @collectionId
 END
  IF @@error!=0
  BEGIN
    RAISERROR 45048
    RETURN -999
  END
END
go
```

ProcGetCollContactXref**Procedure ProcGetCollContactXref**

Code

```
/*=================================================================*/
/*  Name           :  ProcGetCollContactXref                       */
/*  Release        :  B0                                  */
/*  Module         :  Collections                            */
/*  Created on     :  3/25/97                                */
/*  Purpose        :  Create stored procedure to retrieve       */
/*                 the contactId associated with Collections. */
/*  Calls          :                                      */
/*=================================================================*/

CREATE PROCEDURE ProcGetCollContactXref(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40040
      RETURN -999
    END
  ELSE
    BEGIN
    SELECT collectionId,
              contactId
    FROM   DsMdCollectionContactXref
    WHERE  collectionId = @collectionId
  END

    IF @@error!=0
    BEGIN
      RAISERROR 45049
      RETURN -999
    END
END
go
```

ProcGetCollDiscXref**Procedure ProcGetCollDiscXref**

Code

```
/*=================================================================*/
/*  Name           :  ProcGetCollDiscXref                       */
/*  Release        :  B0                                  */
/*  Module         :  Collections                            */
/*  Created on     :  5/15/97                                */
/*  Purpose        :  Create stored procedure to retrieve       */
/*                 the sensorId associated with Collections.  */
/*  Calls          :                                      */
/*=================================================================*/

CREATE PROCEDURE ProcGetCollDiscXref
```

```
(
 @collectionId ID = 0
)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40041
      RETURN -999
    END

  SELECT
            CONVERT(varchar(30), dbID)
  FROM   DsMdCollDisciplineKeywordXref x,
         DsMdDisciplineKeywords s
  WHERE  x.collectionId = @collectionId
    AND  x.disciplineKeywordId = s.dbID

  IF @@error!=0
  BEGIN
    RAISERROR 45050
    RETURN -999
  END
END
go
```

ProcGetCollectionContact**Procedure ProcGetCollectionContact**

Code

```
/*================================================================*/
/*  Name           :  ProcGetCollectionContact              */
/*  Release        :  B0                                 */
/*  Module         :  Collections                          */
/*  Created on     :  3/25/97                              */
/*  Purpose        :  Create stored procedure to retrieve     */
/*                 the contactIds for a Collections.       */
/*  Calls          :                                     */
/*================================================================*/

CREATE PROCEDURE ProcGetCollectionContact(@collectionIdID = 0)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    /* Mimic Sybase error. */
   RAISERROR 40053
   RETURN -999
  END
  ELSE
  BEGIN
    SELECT     b.contactId,
               type = "P"
    FROM       DsMdCollectionContactXref a,  DsMdContactPersons b
```

```
    WHERE       a.collectionId = @collectionId AND
                a.contactId = b.contactId
    IF @@error!=0
    BEGIN
      RAISERROR 45062
          RETURN -999
    END

    SELECT      b.contactId,
                type = "O"
    FROM        DsMdCollectionContactXref a,  DsMdContactOrganizations b
    WHERE       a.collectionId = @collectionId AND
                a.contactId = b.contactId
    IF @@error!=0
    BEGIN
      RAISERROR 45063
          RETURN -999
    END
  END
END
go
```

ProcGetCollectionID**Procedure ProcGetCollectionID**

Code

```
/*================================================================*/
/*  Name          :  ProcGetCollectionID                          */
/*  Release       :  B0                                 */
/*  Module        :  Collections                           */
/*  Created on    :  3/25/97                               */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                the id for a Collection based on subType. */
/*  Calls         :                                     */
/*================================================================*/

CREATE PROCEDURE ProcGetCollectionID(@subType   varchar(30))
AS
BEGIN
  IF (@subType = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40054
    RETURN -999
  END
  ELSE
  BEGIN
    SELECT dbID = CONVERT(varchar(30), dbID)
    FROM   DsMdCollections
    WHERE  subType = @subType

    IF (@@error != 0)
    BEGIN
      RAISERROR 45064
```

```
      RETURN -999
    END
  END
END
go
```

ProcGetCollectionMetadata**Procedure ProcGetCollectionMetadata**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetCollectionMetadata              */
/*  Release       :  B0                                  */
/*  Module        :  Collections                           */
/*  Created on    :  04/25/97                              */
/*  Purpose       :  Create stored procedure to retrieve     */
/*                collection metadata.                     */
/*  Calls         :  ProcGetCollAssociation                  */
/*                ProcGetAdditionalAttributes              */
/*                ProcGetCollIntInfoContent                */
/*                ProcGetCollStrngInfoContent              */
/*                ProcGetCollFloatInfoContent              */
/*                ProcGetDisciplineKeywords                */
/*                ProcGetCollReview                     */
/*                ProcGetSpatialKeyword                    */
/*                ProcGetTemporalKeyword                   */
/*                ProcGetLocality                       */
/*                ProcGetCSDTDescription                   */
/*                ProcGetCollStorageMedium                 */
/*                ProcGetCollBrowseXref                    */
/*                ProcGetCollAPXref                     */
/*==============================================================*/

CREATE PROCEDURE ProcGetCollectionMetadata(
       @dbID  ID = 0)
AS
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40055
      RETURN -999
    END
   /*********************************************************************/
   /*   Select data for following GlParameterLists:                     */
   /*   CollectionMetadata, CollectionDescriptionClass, ECSCollection,  */
   /*   SingleTypeCollection                                            */
   /*********************************************************************/
   DECLARE @colldbID              varchar(30)
   DECLARE @colltype              varchar(30)
   DECLARE @collsubType           varchar(30)
   DECLARE @colllastUpdate        varchar(60)
   DECLARE @ShortName             char(8)
   DECLARE @LongName              varchar(80)
   DECLARE @CollectionDesc        varchar(255)
```

```
DECLARE @VersionID              tinyint
DECLARE @VersionDescription     varchar(255)
DECLARE @RevisionDate           datetime
DECLARE @SuggestedUsage1        varchar(255)
DECLARE @SuggestedUsage2        varchar(245)
DECLARE @ProcessingCenter       varchar(20)
DECLARE @ArchiveCenter          varchar(20)
DECLARE @CitationForExternalPub  varchar(255)
DECLARE @CollectionState        char(10)
DECLARE @MaintenanceUpdFreq     varchar(80)
DECLARE @AccessConstraints      varchar(255)
DECLARE @procLevelId            char(6)
DECLARE @procLevelDesc          varchar(80)


SELECT @colldbID = CONVERT(varchar(30), dbID),
      @colltype = type,
      @collsubType = subType,
      @colllastUpdate = CONVERT(varchar(60), lastUpdate, 109),
          @ShortName = ShortName,
          @LongName = LongName,
      @CollectionDesc = CollectionDescription,
      @VersionID = VersionID,
          @VersionDescription = VersionDescription,
      @RevisionDate = RevisionDate,
      @SuggestedUsage1 = SuggestedUsage1,
      @SuggestedUsage2 = SuggestedUsage2,
      @ProcessingCenter = ProcessingCenter,
      @ArchiveCenter = ArchiveCenter,
      @CitationForExternalPub = CitationForExternalPublication,
      @CollectionState = CollectionState,
      @MaintenanceUpdFreq = MaintenanceUpdateFrequency,
      @AccessConstraints = AccessConstraints,
      @procLevelId = ProcessingLevelID,
      @procLevelDesc = ProcessingLevelDescription
FROM   DsMdCollections
WHERE  dbID = @dbID
IF @@error!=0
BEGIN
  RAISERROR 45065
  RETURN -999
END


/* Reselect to match the GI view */

SELECT dbID = @colldbID,
      type = @colltype,
      subType = @collsubType,
      lastUpdate = @colllastUpdate

SELECT ShortName = @ShortName,
      VersionID = convert(int, @VersionID),
      LongName = @LongName,
```

```
        CollectionDescription = @CollectionDesc

    SELECT RevisionDate = @RevisionDate,
        SuggestedUsage1 = @SuggestedUsage1,
        SuggestedUsage2 = @SuggestedUsage2,
        ProcessingCenter = @ProcessingCenter,
        ArchiveCenter = @ArchiveCenter,
        VersionDescription = @VersionDescription

    SELECT CitationForExternalPublication = @CitationForExternalPub,
        CollectionState = @CollectionState,
        MaintenanceUpdateFrequency = @MaintenanceUpdFreq,
        AccessConstraints = @AccessConstraints

    SELECT ProcessingLevelID = @procLevelId,
        ProcessingLevelDescription = @procLevelDesc


    DECLARE @retStat  int

    /*************************************************/
    /* Select data for following GlParameterList:    */
    /*   CollectionAssociation                       */
    /*************************************************/
    EXECUTE @retStat = ProcGetCollAssociation @dbID
    IF (@retStat != 0)
    BEGIN
      RAISERROR 56003
      RETURN -999
    END


    /*************************************************/
    /* Select data for following GlParameterList:    */
    /*   AdditionalAttributes                        */
    /*************************************************/


    EXECUTE @retStat = ProcGetAdditionalAttributes @dbID
    IF (@retStat != 0)
    BEGIN
      RAISERROR 56004
      RETURN -999
    END
   SELECT CONVERT(varchar(30), attributeId),
        clStringValue
   FROM   DsMdCollStringInfoContent
   WHERE collectionId = @dbID
 UNION
   SELECT CONVERT(varchar(30), attributeId),
        CONVERT(VARCHAR(30),clFloatValue)
   FROM   DsMdCollFloatInfoContent
   WHERE collectionId = @dbID
 UNION
```

```
SELECT CONVERT(varchar(30), attributeId),
      CONVERT(varchar(30), clIntValue)
FROM   DsMdCollIntegerInfoContent
WHERE collectionId = @dbID

IF @@error!=0
 BEGIN
   RAISERROR 45066
   RETURN -999
 END


 /*****************************************************/
 /* Select data for following GlParameterList:     */
 /*   Review                                      */
 /*****************************************************/
 EXECUTE @retStat = ProcGetCollReview @dbID
 IF (@retStat != 0)
 BEGIN
   RAISERROR 56006
   RETURN -999
 END

 /*****************************************************/
 /* Select data for following GlParameterList:     */
 /*   SpatialKeywordClass                         */
 /*****************************************************/
 EXECUTE @retStat = ProcGetSpatialKeyword @dbID
 IF (@retStat != 0)
 BEGIN
   RAISERROR 56007
   RETURN -999
 END

 /*****************************************************/
 /* Select data for following GlParameterList:     */
 /*   TemporalKeywordClass                        */
 /*****************************************************/
 EXECUTE @retStat = ProcGetTemporalKeyword @dbID
 IF (@retStat != 0)
 BEGIN
   RAISERROR 56008
   RETURN -999
 END

 /*****************************************************/
 /* Select data for following GlParameterList:     */
 /*   Locality                                    */
 /*****************************************************/
 EXECUTE @retStat = ProcGetLocality @dbID
 IF (@retStat != 0)
 BEGIN
   RAISERROR 56009
   RETURN -999
```

```
END

/*****************************************************/
/* Select data for following GlParameterList:    */
/*   AnalysisSource                                  */
/*****************************************************/
EXECUTE @retStat = ProcGetAnalysisSource @dbID
IF (@retStat != 0)
BEGIN
  RAISERROR 32000 "ProcGetCollectionMetadata: Failed select on DsMdLocality"
  RETURN -999
END

/*****************************************************/
/* Select data for following GlParameterList:    */
/*   Campaign                                        */
/*****************************************************/
EXECUTE @retStat = ProcGetCampaign @dbID
IF (@retStat != 0)
BEGIN
  RAISERROR 32000 "ProcGetCollectionMetadata: Failed select on DsMdLocality"
  RETURN -999
END

/*****************************************************/
/* Select data for following GlParameterList:    */
/*   CSDTDescription                                 */
/*****************************************************/
EXECUTE @retStat = ProcGetCSDTDescription @dbID
IF (@retStat != 0)
BEGIN
  RAISERROR 56010
  RETURN -999
END

/*****************************************************/
/* Select data for following GlParameterList:    */
/*   StorageMedium                                   */
/*****************************************************/
EXECUTE @retStat = ProcGetCollStorageMedium @dbID
IF (@retStat != 0)
BEGIN
  RAISERROR 56011
  RETURN -999
END

/*****************************************************/
/* Select data for following GlParameterList:    */
/*   browseIds                                       */
/*****************************************************/
EXECUTE @retStat = ProcGetCollBrowseXref @dbID
IF (@retStat != 0)
BEGIN
  RAISERROR 56012
```

```
    RETURN -999
  END

  /*****************************************************/
  /* Select data for following GlParameterList:     */
  /*   apIds                                        */
  /*****************************************************/
  EXECUTE @retStat = ProcGetCollAPXref @dbID
  IF (@retStat != 0)
  BEGIN
    RAISERROR 56013
    RETURN -999
  END
 END
go
```

ProcGetCollFloatInfoContent**Procedure ProcGetCollFloatInfoContent**

Code

```
/*================================================================*/
/*  Name          :  ProcGetCollFloatInfoContent            */
/*  Release       :  B0                                     */
/*  Module        :  Collections                              */
/*  Created on    :  3/25/97                                  */
/*  Purpose       :  Create stored procedure to retrieve     */
/*                   float name/value pairs for Collections. */
/*  Calls         :                                         */
/*================================================================*/

CREATE PROCEDURE ProcGetCollFloatInfoContent(@collectionId ID = 0)
AS
BEGIN
 IF (@collectionId = 0)
   BEGIN
    RAISERROR 40042
    RETURN -999
   END
 ELSE
  BEGIN
  SELECT
        x.attributeId,
            clFloatValue=CONVERT(varchar(60), c.clFloatValue)
  FROM
            DsMdCollectionAddnlAttribsXref  x,
            DsMdCollFloatInfoContent c
  WHERE
            x.collectionId = @collectionId
    AND  x.attributeId = c.attributeId
  ORDER BY
        x.attributeId
 END
  IF @@error!=0
  BEGIN
```

```
    RAISERROR 45051
    RETURN -999
  END
END
go
```

ProcGetCollGranuleXref**Procedure ProcGetCollGranuleXref**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetCollGranuleXref                        */
/*  Release       :  B0                                            */
/*  Module        :  Collections                                   */
/*  Created on    :  3/25/97                                       */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                   the granuleId associated with Collections. */
/*  Calls         :                                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetCollGranuleXref(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40043
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT collectionId,
            granuleId
   FROM   DsMdCollectionGranuleXref
   WHERE  collectionId = @collectionId
  END

   IF @@error!=0
   BEGIN
     RAISERROR 45052
     RETURN -999
   END
END
go
```

ProcGetCollInstrXref**Procedure ProcGetCollInstrXref**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetCollInstrXref                          */
/*  Release       :  B0                                            */
/*  Module        :  Collections                                   */
/*  Created on    :  3/25/97                                       */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                   the instrumentId associated with Collections. */
/*  Calls         :                                                */
/*=================================================================*/
```

```
CREATE PROCEDURE ProcGetCollInstrXref
(
 @collectionId ID = 0,
 @platformId ID = 0
)
AS
BEGIN
  IF (@collectionId = 0 OR @platformId = 0)
    BEGIN
      RAISERROR 40044
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT
            CONVERT(varchar(30), instrumentId)
   FROM   DsMdCollectionInstrumentXref x,
            DsMdInstrument i
   WHERE  x.collectionId = @collectionId
     AND  x.instrumentId = i.dbID
     AND  i.platformId = @platformId
  END
   IF @@error!=0
   BEGIN
      RAISERROR 45053
      RETURN -999
   END
END
go
```

ProcGetCollIntInfoContent**Procedure ProcGetCollIntInfoContent**

Code

```
/*================================================================*/
/*  Name          :  ProcGetCollIntInfoContent                    */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                integer name/value pairs for Collections. */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetCollIntInfoContent(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40045
      RETURN -999
    END
  ELSE
```

```
  BEGIN
  SELECT
          x.attributeId,
          clIntValue= CONVERT(varchar(60), c.clIntValue)
  FROM
          DsMdCollectionAddnlAttribsXref  x,
          DsMdCollIntegerInfoContent c
  WHERE
          x.collectionId = @collectionId
    AND  x.attributeId = c.attributeId
  ORDER BY
          x.attributeId
 END
  IF @@error!=0
  BEGIN
    RAISERROR 45054
    RETURN -999
  END
END
go
```

ProcGetCollPlatformXref**Procedure ProcGetCollPlatformXref**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetCollPlatformXref                    */
/*  Release       :  B0                                    */
/*  Module        :  Collections                              */
/*  Created on    :  3/25/97                                  */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                 the platformId associated with Collections.   */
/*  Calls         :                                        */
/*=================================================================*/

CREATE PROCEDURE ProcGetCollPlatformXref(@collectionId ID = 0)
AS
BEGIN
 IF (@collectionId = 0)
  BEGIN
    RAISERROR 40046
    RETURN -999
  END

  SELECT
          CONVERT(varchar(30), platformId)
  FROM   DsMdCollectionPlatformXref
  WHERE  collectionId = @collectionId

  IF @@error!=0
  BEGIN
    RAISERROR 45055
    RETURN -999
  END
```

```
END
go
```

ProcGetCollReview**Procedure ProcGetCollReview**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetCollReview                          */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   review data for Collections.               */
/*  Calls         :                                             */
/*==============================================================*/

CREATE PROCEDURE ProcGetCollReview(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40047
      RETURN -999
    END
  ELSE
    BEGIN
     SELECT ScienceReviewStatus,
            ScienceReviewDate,
         FutureReviewDate
   FROM   DsMdCollReview
   WHERE collectionId = @collectionId
  END
   IF @@error!=0
   BEGIN
      RAISERROR 45056
      RETURN -999
   END
END
go
```

ProcGetCollSensorXref**Procedure ProcGetCollSensorXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetCollSensorXref                      */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   the sensorId associated with Collections.  */
/*  Calls         :                                             */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcGetCollSensorXref
(
 @collectionId ID = 0,
 @instrumentId ID = 0
)
AS
BEGIN
  IF (@collectionId = 0 OR @instrumentId = 0)
    BEGIN
      RAISERROR 40048
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT
            CONVERT(varchar(30), sensorId)
   FROM   DsMdCollectionSensorXref x,
          DsMdSensor s
   WHERE  x.collectionId = @collectionId
     AND  x.sensorId = s.dbID
     AND  s.instrumentId = @instrumentId
  END
   IF @@error!=0
   BEGIN
      RAISERROR 45057
      RETURN -999
   END
END
go
```

ProcGetCollStorageMedium**Procedure ProcGetCollStorageMedium**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetCollStorageMedium                   */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   storage medium data for Collections.      */
/*  Calls         :                                             */
/*==============================================================*/

CREATE PROCEDURE ProcGetCollStorageMedium(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40049
      RETURN -999
    END
  ELSE
```

```
  BEGIN

  SELECT
          StorageMedium
  FROM
          DsMdCollStorageMedium
  WHERE
          collectionId = @collectionId
 END
  IF @@error!=0
  BEGIN
    RAISERROR 45058
    RETURN -999
  END
END
go
```

ProcGetCollStrngInfoContent**Procedure ProcGetCollStrngInfoContent**

Code

```
/*==============================================================*/
/*  Name           :  ProcGetCollStrngInfoContent             */
/*  Release        :  B0                                      */
/*  Module         :  Collections                             */
/*  Created on     :  3/25/97                                 */
/*  Purpose        :  Create stored procedure to retrieve     */
/*                 string name/value pairs for Collections.*/
/*  Calls          :                                          */
/*==============================================================*/

CREATE PROCEDURE ProcGetCollStrngInfoContent(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40050
      RETURN -999
    END
  ELSE
   BEGIN
   SELECT
        x.attributeId,
          clStringValue
   FROM
          DsMdCollStringInfoContent c,
        DsMdCollectionAddnlAttribsXref  x
   WHERE
          x.collectionId = @collectionId
    AND  x.attributeId = c.attributeId
   ORDER BY
        x.attributeId
  END
   IF @@error!=0
```

```
  BEGIN
    RAISERROR 45059
    RETURN -999
  END
END
go
```

ProcGetCollVertSpatialDomn**Procedure ProcGetCollVertSpatialDomn**

Code

```
/*====================================================================*/
/*  Name            :  ProcGetCollVertSpatialDomn             */
/*  Release         :  B0                                     */
/*  Module          :  Spatial                                */
/*  Created on      :  3/30/97                                */
/*  Purpose         :  Create Stored Procedure for retrieving  */
/*                  data from DsMdCollVerticalSpatialDomain */
/*                                                            */
/*                                                            */
/*  Calls           :  None                                   */
/*                                                            */
/*====================================================================*/

CREATE PROCEDURE ProcGetCollVertSpatialDomn(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40051
      RETURN -999
    END
  ELSE
   BEGIN
    SELECT VerticalSpatialDomainValue, VerticalSpatialDomainType
    FROM DsMdCollVerticalSpatialDomain
    WHERE  collectionId = @collectionId
  END
    IF (@@error!= 0)
    BEGIN
       RAISERROR 45060
       RETURN -999
    END
END
go
```

ProcGetCollZoneIdentifier**Procedure ProcGetCollZoneIdentifier**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetCollZoneIdentifier            */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                               */
/*  Created on      :  3/30/97                                */
/*  Purpose         :  Create Stored Procedure to retrieve   */
/*                  :  data from DsMdCollZoneIdentifier      */
/*                  :  table l                               */
/*                                                           */
/*  Calls           :  None                                  */
/*                                                           */
/*==============================================================*/

CREATE PROCEDURE ProcGetCollZoneIdentifier(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40052
      RETURN -999
    END
  ELSE
   BEGIN
    SELECT ZoneIdentifier
    FROM   DsMdCollZoneIdentifier
    WHERE  collectionId = @collectionId
  END
    IF (@@error!= 0)
    BEGIN
          RAISERROR 45061
          RETURN -999
    END
END
go
```

ProcGetContact**Procedure ProcGetContact**

Code

```
/*==============================================================*/
/*  Name           :  ProcGetContactPerson                  */
/*  Release        :  B0                                    */
/*  Module         :  Contact                                */
/*  Created on     :  3/24/97                                */
/*  Purpose        :  Create stored procedure to Contact,     */
/*                  and associate data tables.               */
/*  Calls          :                                         */
/*                                                           */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcGetContact(@dbID ID = 0)
AS
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40056
      RETURN -999
    END
    SELECT dbID = CONVERT(varchar(30), dbID),
          ContactRole,
          HoursOfService,
          ContactInstructions,
          CONVERT(varchar(60), lastUpdate, 109)
    FROM   DsMdContact
    WHERE  dbID = @dbID

    IF (@@error!= 0)
    BEGIN
           RAISERROR 45067
           RETURN -999
    END
END
go
```

ProcGetContactAddress**Procedure ProcGetContactAddress**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactPerson                       */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  Create stored procedure to Contact,        */
/*                   and associate data tables.                 */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetContactAddress(@contactId ID = 0)
AS
BEGIN
  IF (@contactId = 0)
    BEGIN
      RAISERROR 40057
      RETURN -999
    END
  ELSE
  BEGIN
    SELECT
           StreetAddress,
          City,
          StateProvince,
```

```
        PostalCode,
        Country
   FROM   DsMdContactAddress
   WHERE  contactId = @contactId
  END

   IF (@@error!= 0)
   BEGIN
         RAISERROR 45068
         RETURN -999
   END
END
go
```

ProcGetContactIDs**Procedure ProcGetContactIDs**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactIDs                          */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  Create stored procedure to get Contact  */
/*                   and associate data tables.                 */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/

CREATE PROC ProcGetContactIDs(
     @collectionId          ID = 0)
AS
BEGIN

  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40058
      RETURN -999
    END
  ELSE
  BEGIN
    SELECT convert(varchar(30), c.contactId)
    FROM   DsMdCollectionContactXref a,
           DsMdContact b,
           DsMdContactPersons c
    WHERE  a.collectionId = @collectionId     AND
           a.contactId    = b.dbID           AND
           b.dbID         = c.contactId
  END
    IF (@@error!= 0)
    BEGIN
          RAISERROR 45069
          RETURN -999
    END
```

```
   SELECT convert(varchar(30), c.contactId)
   FROM   DsMdCollectionContactXref a,
          DsMdContact b,
          DsMdContactOrganizations c
   WHERE  a.collectionId = @collectionId     AND
          a.contactId    = b.dbID            AND
          b.dbID         = c.contactId

   IF (@@error!= 0)
   BEGIN
          RAISERROR 45070
          RETURN -999
   END
END
go
```

ProcGetContactOrganizationID**Procedure ProcGetContactOrganizationID**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactOrganizationID          */
/*  Release       :  B0                                    */
/*  Module        :  Collections                           */
/*  Created on    :  3/24/97                               */
/*  Purpose       :  Create stored procedure to Contact,     */
/*                   and associate data tables.             */
/*  Calls         :                                        */
/*                                                         */
/*==============================================================*/
CREATE PROC ProcGetContactOrganizationID(
   @ContactRole                varchar(20),
   @ContactOrganizationName    varchar(200) )
AS
BEGIN
   SELECT convert(varchar(30), a.dbID)
   FROM   DsMdContact a,
          DsMdContactOrganizations b
   WHERE  a.ContactRole           = @ContactRole           AND
          b.ContactOrganizationName = @ContactOrganizationName   AND
          a.dbID                  = b.contactId

   IF @@error!=0
   BEGIN
      RAISERROR 45072
      RETURN -999
   END
END
go
```

ProcGetContactOrgInfo**Procedure ProcGetContactOrgInfo**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactOrgInfo                      */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  Create stored procedure to Contact,        */
/*                   and associate data tables.                 */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetContactOrgInfo
        (@dbID ID = 0)
AS
BEGIN

   IF  (@dbID = 0)
   BEGIN
      RAISERROR 40059
      RETURN -999
   END

        SELECT
          a.ContactRole,
                a.HoursOfService,
                a.ContactInstructions,
                b.ContactOrganizationName
        FROM   DsMdContact a,
                DsMdContactOrganizations b
        WHERE           a.dbID = @dbID
         AND   a.dbID = b.contactId

        IF @@error!=0
          BEGIN
        RAISERROR 45071
                RETURN -999
          END



      DECLARE @retStat  int

      EXECUTE @retStat = ProcGetContactAddress @dbID
      IF @retStat != 0
      BEGIN
        RAISERROR 56014
        RETURN -999
      END
```

```
      EXECUTE @retStat = ProcGetTelephoneNumbers @dbID
      IF @retStat != 0
      BEGIN
        RAISERROR 56015
        RETURN -999
      END

      EXECUTE @retStat = ProcGetEmailAddress @dbID
      IF @retStat != 0
      BEGIN
        RAISERROR 56016
        RETURN -999
      END
END
go
```

ProcGetContactPersonID**Procedure ProcGetContactPersonID**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactPersonID                     */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  4/3/97                                     */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                the id for a contact person.                  */
/*  Calls         :  None                                       */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetContactPersonID(
  @ContactRole        varchar(20),
  @ContactFirstName   varchar(200),
  @ContactLastName    varchar(200),
  @ContactMiddleName  varchar(200),
  @ContactJobPosition varchar(200))
AS
BEGIN
  SELECT convert(varchar(30), a.dbID)
  FROM   DsMdContact a,
         DsMdContactPersons b
  WHERE  a.ContactRole           = @ContactRole       AND
         b.ContactFirstName      = @ContactFirstName  AND
         b.ContactLastName       = @ContactLastName   AND
         b.ContactMiddleName     = @ContactMiddleName AND
         b.ContactJobPosition    = @ContactJobPosition AND
         a.dbID                  = b.contactId

  IF @@error!=0
  BEGIN
    RAISERROR 45073
    RETURN -999
  END
```

```
END
go
```

ProcGetContactPersonInfo**Procedure ProcGetContactPersonInfo**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactPersonInfo              */
/*  Release       :  B0                                 */
/*  Module        :  Contact                             */
/*  Created on    :  3/24/97                             */
/*  Purpose       :  Create stored procedure to Contact,     */
/*                   and associate data tables.             */
/*  Calls         :  ProcGetContactAddress                 */
/*                   ProcGetTelephoneNumbers                */
/*                   ProcGetEmailAddress                  */
/*                                          */
/*==============================================================*/

CREATE PROCEDURE ProcGetContactPersonInfo
        (@contactId      ID = 0)
AS
BEGIN

  IF (@contactId = 0)
    BEGIN
      RAISERROR 40062
      RETURN -999
    END

  SELECT
        a.ContactRole,
        a.HoursOfService,
        a.ContactInstructions,
        b.ContactJobPosition,
        b.ContactFirstName,
        b.ContactMiddleName,
        b.ContactLastName
  FROM DsMdContact a,
        DsMdContactPersons b
  WHERE         a.dbID = @contactId AND
        a.dbID = b.contactId

  IF @@error!=0
     BEGIN
       RAISERROR 45074
       RETURN -999
      END

  DECLARE @RetStat Int
  EXECUTE @RetStat = ProcGetContactAddress  @contactId
  IF @RetStat!=0
     BEGIN
```

```
    RAISERROR 56017
    RETURN -999
   END


 EXECUTE @RetStat = ProcGetTelephoneNumbers  @contactId
 IF @RetStat!=0
   BEGIN
    RAISERROR 56018
    RETURN -999
   END

 EXECUTE @RetStat = ProcGetEmailAddress  @contactId
 IF @RetStat!=0
   BEGIN
    RAISERROR 56019
    RETURN -999
   END

END
go
```

ProcGetCSDTDescription**Procedure ProcGetCSDTDescription**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetCSDTDescription                        */
/*  Release       :  B0                                            */
/*  Module        :  Collections                                   */
/*  Created on    :  3/25/97                                       */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                CSDT description data for Collections.           */
/*  Calls         :                                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetCSDTDescription(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
     RAISERROR 40032
     RETURN -999
    END
  ELSE
   BEGIN
   SELECT
         PrimaryCSDT,
         IndirectReference,
         Implementation,
         CSDTComments
   FROM
            DsMdCSDTDescription
   WHERE
```

```
            collectionId = @collectionId
  END
  IF @@error!=0
  BEGIN
    RAISERROR 45041
    RETURN -999
  END
END
go
```

ProcGetDAP**Procedure ProcGetDAP**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetDAP                                 */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to Delivered       */
/*                   Algorithm Package and associate data       */
/*                   tables.                                    */
/*  Calls         :                                             */
/*                   ProcGetSDSRVLookup                         */
/*                   ProcGetFileStorage                         */
/*                   ProcGetPGEGroup                            */
/*==============================================================*/

CREATE PROCEDURE ProcGetDAP(@dbID    ID = 0)
AS
BEGIN

  DECLARE @retStat  int
  DECLARE @lookupValue varchar(40)
  DECLARE @subType varchar(30)
  DECLARE @type varchar(30)

  IF (@dbID = 0)
    BEGIN
      RAISERROR 40063
      RETURN -999
    END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "type", "DP", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetDAP error - could not find parameter @type ."
      RETURN -999
    END

  SELECT @type = @lookupValue
```

```
 EXECUTE @retStat = ProcGetSDSRVLookup "subType", "DAP", "varchar", @lookupValue OUTPUT

 IF @retStat != 0
  BEGIN
    RAISERROR 20014 "Procedure ProcGetDAP error - could not find parameter @subType."
    RETURN -999
  END

 SELECT @subType = @lookupValue

 SELECT  type = CONVERT(varchar, @type),
        subType = CONVERT(varchar, @subType),
        dbID = CONVERT(varchar(30), dbID),
        lastUpdate = CONVERT(varchar(60), lastUpdate, 109),
        DAPID,
        DAPInsertDate
  FROM   DsMdDAP
  WHERE  dbID = @dbID

  IF @@error!=0 OR @@rowcount != 1
  BEGIN
    RAISERROR 45075
    RETURN -999
  END

  SELECT ShortName,
        VersionID = CONVERT(int, VersionID)
  FROM   DsMdCollections
  WHERE  type = @type

  IF @@error!=0 OR @@rowcount != 1
  BEGIN
    RAISERROR 45076
    RETURN -999
  END
  EXECUTE @retStat = ProcGetPGEGroup @dbID
  IF @retStat != 0
  BEGIN
    RAISERROR 56020
    RETURN -999
  END

  EXECUTE @retStat = ProcGetFileStorage @dbID
  IF @retStat != 0
  BEGIN
    RAISERROR 56021
    RETURN -999
  END
END
go
```

ProcGetDepthResolution**Procedure ProcGetDepthResolution**

Code

```
/*=================================================================*/
/*  Name            :  ProcGetDepthResolution              */
/*  Release         :  B0                                  */
/*  Module          :  Spatial                             */
/*  Created on      :  3/25/97                             */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                  data into DsMdDepthResolution          */
/*  Calls           :  None                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetDepthResolution(@collectionId  ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40064
      RETURN -999
    END
  ELSE
  BEGIN
   SELECT collectionId,
         DepthResolution
   FROM   DsMdDepthResolution
   WHERE  collectionId = @collectionId
  END
   IF @@error!=0
   BEGIN
      RAISERROR 45077
      RETURN -999
   END
END
go
```

ProcGetDepthSysDef**Procedure ProcGetDepthSysDef**

Code

```
/*=================================================================*/
/*  Name            :  ProcGetDepthSysDef                  */
/*  Release         :  B0                                  */
/*  Module          :  Spatial                             */
/*  Created on      :  3/25/97                             */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                  data from DsMdSpatial table            */
/*                                                         */
/*                                                         */
/*  Calls           :  None                                */
/*                                                         */
/*=================================================================*/
```

```
CREATE PROCEDURE ProcGetDepthSysDef(@collectionId ID = 0)
AS
BEGIN

   IF  (@collectionId = 0)
   BEGIN
      RAISERROR 40065
      RETURN -999
   END

   SELECT DepthDatumName,
         DepthDistanceUnits,
         DepthEncodingMethod
   FROM   DsMdSpatial
   WHERE  collectionId = @collectionId

   IF (@@error!= 0)
   BEGIN
          RAISERROR 45078
          RETURN -999
   END

   SELECT DepthResolution
   FROM   DsMdDepthResolution
   WHERE  collectionId = @collectionId

   IF (@@error!= 0)
   BEGIN
          RAISERROR 45079
          RETURN -999
   END
END
go
```

ProcGetDictionaryAttribute**Procedure ProcGetDictionaryAttribute**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetDictionaryAttribute            */
/*  Release       :  B0                                    */
/*  Module        :  Dictionary                            */
/*  Created on    :  11/4/97                               */
/*  Purpose       :  Create stored procedure to Dictionary,  */
/*                 and associate data tables.              */
/*  Calls         :  None                                  */
/*                                                         */
/*==============================================================*/

CREATE PROCEDURE ProcGetDictionaryAttribute
AS
BEGIN
          SELECT          CONVERT(varchar(30), attributeID),
                    qualifiedAttrName,
```

```
                type,
                length,
                seqNum,
                seqType,
                CONVERT(int, optionalIndicator),
                CONVERT(int, maxOccurances),
                glType,
                CONVERT(int, psaIndicator),
                operator
        FROM  DsDeDictionaryAttribute
    ORDER BY attributeID

        IF @@error!=0
          BEGIN
        RAISERROR 45080 "ProcGetDictionaryAttribute: Failed select on DsDeDictionaryAttribute"
                RETURN -999
          END
END
go
```

ProcGetDictionaryContent**Procedure ProcGetDictionaryContent**

Code

```
/*================================================================*/
/*  Name          :  ProcGetDictionaryContent              */
/*  Release       :  B0                                 */
/*  Module        :  Dictionary                           */
/*  Created on    :  4/17/97                              */
/*  Purpose       :  Create stored procedure to Dictionary,  */
/*                 and associate data tables.             */
/*  Calls         :                                     */
/*                                                      */
/*================================================================*/

CREATE PROCEDURE ProcGetDictionaryContent
AS
BEGIN
        SELECT          qualifiedGroupName,
                CONVERT(int, numLevels),
                CONVERT(int, numAttribs),
                CONVERT(int, skips),
                CONVERT(int, numMultiples),
                CONVERT(int, containerIndicator),
                CONVERT(int, optionalIndicator)
        FROM
                DsDeDictionaryContent
        IF @@error!=0
          BEGIN
        RAISERROR 45081
                RETURN -999
          END
END
go
```

ProcGetDictionaryRule**Procedure ProcGetDictionaryRule**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetDictionaryRule                      */
/*  Release       :  B0                                         */
/*  Module        :  Dictionary                                 */
/*  Created on    :  11/4/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                DsDeDictionaryAttribute and associated        */
/*                data tables.                                  */
/*  Calls         :  None                                       */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetDictionaryRule
AS
BEGIN

      DECLARE @ruleText varchar(255)
      DECLARE @buffer varchar(255)
      DECLARE @qualifiedAttrName varchar(255)
      DECLARE @attributeID ID
      DECLARE @tempid ID
      DECLARE @secondaryID int

      CREATE TABLE #temp(qualifiedAttrName varchar(255),
                  secondaryID int,
                  ruleText varchar(255))

      DECLARE rule_cur cursor for
      SELECT attributeID, ruleText
      FROM DsDeDictionaryRule
      FOR READ ONLY


      OPEN rule_cur
      FETCH rule_cur into @attributeID, @ruleText

      SELECT @tempid = @attributeID
      SELECT @secondaryID = 1
      SELECT @qualifiedAttrName = qualifiedAttrName FROM DsDeDictionaryAttribute
            WHERE attributeID = @attributeID

      SELECT @buffer = "("

      WHILE (@@sqlstatus = 0)
            BEGIN
                  IF ((char_length(@buffer) + char_length(@ruleText) + 2) > 255)
                  BEGIN
                        INSERT #temp values(@qualifiedAttrName, @secondaryID, @buffer)
                        SELECT @buffer = ""
                        SELECT @secondaryID = @secondaryID + 1
```

```
                END

                SELECT @buffer = @buffer + @ruleText + ", "
                FETCH rule_cur into @attributeID, @ruleText


                -- when the last rule for a given attributeID was read,
                -- close with ")" and start a new line.
                IF (@tempid != @attributeID)
                BEGIN
                        SELECT @buffer = substring(@buffer,1,
                                char_length(@buffer)-2) + ")"

                        INSERT #temp values(@qualifiedAttrName, @secondaryID, @buffer)

                        SELECT @tempid = @attributeID

                        SELECT @qualifiedAttrName = qualifiedAttrName FROM DsDeDictionaryAttribute
                                WHERE attributeID = @attributeID

                        SELECT @secondaryID = 1
                        SELECT @buffer = "("
                END
        END

        CLOSE rule_cur
        DEALLOCATE cursor rule_cur


        SELECT qualifiedAttrName, ruleText
        FROM #temp
        ORDER BY qualifiedAttrName, secondaryID
END
go
```

ProcGetDisciplineKeywords**Procedure ProcGetDisciplineKeywords**

Code

```
/*=============================================================*/
/*  Name          :  ProcGetDisciplineKeywords              */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  4/28/97                                */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                discipline keyword data for Collections.*/
/*  Calls         :                                         */
/*=============================================================*/

CREATE PROCEDURE ProcGetDisciplineKeywords(@dbID  ID = 0)
AS
BEGIN
   IF  (@dbID = 0)
   BEGIN
```

```
      RAISERROR 40067
      RETURN -999
   END

  SELECT ECSDisciplineKeyword,
         ECSTopicKeyword,
       ECSTermKeyword,
       ECSVariableKeyword
  FROM   DsMdDisciplineKeywords
  WHERE  dbID = @dbID

  IF @@error!=0
  BEGIN
    RAISERROR 45083
    RETURN -999
  END

  SELECT ECSParameterKeyword
  FROM   DsMdECSParameterKeywords
  WHERE  parameterKeywordId = @dbID

  IF @@error!=0
  BEGIN
    RAISERROR 45084
    RETURN -999
  END

END
go
```

ProcGetDisciTopParaID**Procedure ProcGetDisciTopParaID**

Code

```
/*================================================================*/
/*  Name          :  ProcGetDisciTopParaID                        */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  05/05/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the id for a Discipline Keyword.             */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcGetDisciTopParaID(@ECSDiscKeyword   varchar(25),
                           @ECSTopicKeyword  varchar(32),
                           @ECSTermKeyword   varchar(50),
                           @ECSVarKeyword    varchar(80))
AS
BEGIN
  SELECT dbID = CONVERT(varchar(30), dbID)
  FROM   DsMdDisciplineKeywords
  WHERE  ECSDisciplineKeyword = @ECSDiscKeyword
  AND    ECSTopicKeyword = @ECSTopicKeyword
```

```
  AND    ECSTermKeyword = @ECSTermKeyword
  AND    ECSVariableKeyword = @ECSVarKeyword

  IF (@@error != 0)
  BEGIN
    RAISERROR 45082
    RETURN -999
  END
END
go
```

ProcGetDisconMulti**Procedure ProcGetDisconMulti**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetDisconMulti                       */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for selecting */
/*                  data from DsMdTemporal                      */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetDisconMulti(@collectionId ID = 0)
AS
BEGIN
   IF  (@collectionId = 0)
   BEGIN
      RAISERROR 40068
      RETURN -999
   END
   SELECT EndsatPresentFlag
   FROM   DsMdTemporal
   WHERE  collectionId = @collectionId AND EndsatPresentFlag != NULL

   IF (@@error!= 0)
   BEGIN
         RAISERROR 45085
         RETURN -999
   END
END
go
```

ProcGetECSParameterKeywords**Procedure ProcGetECSParameterKeywords**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetECSParameterKeywords            */
/*  Release       :  B0                                  */
/*  Module        :  Collections                           */
/*  Created on    :  4/28/97                               */
/*  Purpose       :  Stored procedure to retrieve          */
/*                ECS Parameter Keyword data.             */
/*  Calls         :                                      */
/*==============================================================*/

CREATE PROCEDURE ProcGetECSParameterKeywords(@parameterKeywordId  ID = 0)
AS
BEGIN
  IF (@parameterKeywordId = 0)
    BEGIN
      RAISERROR 40069
      RETURN -999
    END

    SELECT ECSParameterKeyword
    FROM   DsMdECSParameterKeywords
    WHERE  parameterKeywordId  = @parameterKeywordId

   IF @@error!=0
   BEGIN
     RAISERROR 45086
     RETURN -999
   END

END
go
```

ProcGetEmailAddress**Procedure ProcGetEmailAddress**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactPerson                  */
/*  Release       :  B0                                  */
/*  Module        :  Contact                               */
/*  Created on    :  3/24/97                               */
/*  Purpose       :  Create stored procedure to Contact,     */
/*                and associate data tables.              */
/*  Calls         :                                      */
/*                                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetEmailAddress(@contactId ID = 0)
AS
```

```
BEGIN
  IF (@contactId = 0)
    BEGIN
      RAISERROR 40007
      RETURN -999
    END
    SELECT ElectronicMailAddress
    FROM   DsMdEmailAddress
    WHERE  contactId = @contactId

    IF (@@error!= 0)
    BEGIN
          RAISERROR 45007
          RETURN -999
    END
END
go
```

ProcGetESDTConfiguredType**Procedure ProcGetESDTConfiguredType**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetOneESDTConfiguredType          */
/*  Release       :  B0                                    */
/*  Module        :                                        */
/*  Created on    :  8/20/97                               */
/*  Purpose       :                                        */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcGetESDTConfiguredType
AS
BEGIN
    SELECT configuredName,
          versionID = CONVERT(VARCHAR(20), versionID),
          archiveIdentifier,
          backupIdentifier,
          offsiteIdentifier,
          description,
          descriptorFileName,
          dllFileName,
          eventFileName
    FROM DsGeESDTConfiguredType
    ORDER BY configuredName, versionID

    IF @@error!=0
    BEGIN
       RAISERROR 45145
       RETURN -999
    END
END
go
```

ProcGetFileStorage**Procedure ProcGetFileStorage**

Code

```
/*================================================================*/
/*  Name          :  ProcGetFileStorage                           */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   file storage data for Granules.              */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetFileStorage(@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40002
      RETURN -999
    END

   DECLARE @internalFileName             varchar(80)
   DECLARE @filePath               varchar(80)
   DECLARE @userDataFile       varchar(80)
   DECLARE @checkSum                     int
   DECLARE @fileSize           int
   DECLARE @creationDate       DsTMdModDate
   DECLARE @archiveIdentifier   varchar(20)
   DECLARE @backupIdentifier    varchar(20)
   DECLARE @offsiteIdentifier   varchar(20)
   DECLARE @ShortName           char(8)
   DECLARE @VersionID          tinyint
   DECLARE @primaryCollectionId ID

   SELECT @internalFileName = internalFileName,
         @filePath = filePath,
         @userDataFile = userDataFile,
         @checkSum = checkSum,
         @fileSize = fileSize,
         @creationDate =  creationDate
--         @ArchiveID
   FROM   DsMdFileStorage
   WHERE  granuleId = @granuleId

   IF @@error!=0
   BEGIN
     RAISERROR 45002
     RETURN -999
   END

   SET ROWCOUNT 1
```

```
  SELECT @ShortName =  ShortName,
        @VersionID = VersionID
  FROM DsMdGranules
  WHERE dbID = @granuleId

  IF @@error!=0
  BEGIN
--      RAISERROR 20014 "Error fromm ProcGetFileStorage getting DsMdGranules"
     RAISERROR 45283
     RETURN -999
  END

  SET ROWCOUNT 0

  SELECT @archiveIdentifier = archiveIdentifier,
        @backupIdentifier = backupIdentifier,
        @offsiteIdentifier = offsiteIdentifier
  FROM
          DsGeESDTConfiguredType g
  WHERE g.configuredName = @ShortName
    AND g.versionID = @VersionID

  IF @@error!=0
  BEGIN
--      RAISERROR 20014 "Error fromm ProcGetFileStorage getting DsGeESDTConfiguredType"
     RAISERROR 45284
     RETURN -999
  END

  SELECT @internalFileName,
        @filePath,
        @userDataFile,
        @checkSum,
        @fileSize,
        @creationDate,
            @archiveIdentifier,
            @backupIdentifier,
        @offsiteIdentifier

  IF @@error!=0
  BEGIN
--      RAISERROR 20014 "Error from ProcGetFileStorage selecting File Storage data"
     RAISERROR 45285
     RETURN -999
  END


END
go
```

ProcGetGrAnalysisShortName**Procedure ProcGetGrAnalysisShortName**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrAnalysisShortName                   */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the AnalysisSourceShortName associated       */
/*                   with Granules.                               */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGrAnalysisShortName(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40070
      RETURN -999
    END
   SELECT t1.AnalysisShortName
   FROM   DsMdAnalysisSource t1,
          DsMdGranuleAnalysisXref t2
   WHERE  t2.granuleId = @granuleId AND
          t1.dbID = t2.analysisSourceId
   IF @@error!=0
   BEGIN
     RAISERROR 45087
     RETURN -999
   END
END
go
```

ProcGetGranReview**Procedure ProcGetGranReview**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGranReview                            */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   granule review data associated with Granules. */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGranReview(@granuleId ID = 0)
AS
BEGIN
```

```
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40082
      RETURN -999
    END
   SELECT ScienceReviewStatus,
         ScienceReviewDate,
         FutureReviewDate
   FROM   DsMdGranuleReview
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
     RAISERROR 45099
     RETURN -999
   END
END
go
```

ProcGetGranuleAnalysisXref**Procedure ProcGetGranuleAnalysisXref**

Code

```
/*=================================================================*/
/*  Name           :  ProcGetGranuleAnalysisXref                   */
/*  Release        :  B0                                           */
/*  Module         :  Granules                                     */
/*  Created on     :  04/17/97                                     */
/*  Purpose        :  Create stored procedure to retrieve          */
/*                    the analysisSourceId associated with         */
/*                    Granules.                                    */
/*  Calls          :                                               */
/*=================================================================*/

CREATE PROCEDURE ProcGetGranuleAnalysisXref(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40083
      RETURN -999
    END
   SELECT CONVERT(varchar(30), granuleId),
            CONVERT(varchar(30), analysisSourceId)
   FROM   DsMdGranuleAnalysisXref
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
     RAISERROR 45100
     RETURN -999
   END
END
go
```

ProcGetGranuleCampaignXref**Procedure ProcGetGranuleCampaignXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGranuleCampaignXref                   */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the campaignId associated with Granules.     */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGranuleCampaignXref(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40084
      RETURN -999
    END
   SELECT CONVERT(varchar(30), granuleId),
            CONVERT(varchar(30), campaignId)
   FROM   DsMdGranuleCampaignXref
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
     RAISERROR 45101
     RETURN -999
   END
END
go
```

ProcGetGranuleLocality**Procedure ProcGetGranuleLocality**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGranuleLocality                       */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   granule locality data.                       */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGranuleLocality(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
```

```
    RAISERROR 40085
    RETURN -999
   END
  SELECT LocalityValue
  FROM   DsMdGranuleLocality
  WHERE  granuleId = @granuleId
  IF @@error!=0
  BEGIN
    RAISERROR 45102
    RETURN -999
  END
END
go
```

ProcGetGranuleMetadata**Procedure ProcGetGranuleMetadata**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetGranuleMetadata                     */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  04/18/97                                   */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   ALL metadata associated with a Science     */
/*                   Granule.                                   */
/*  Calls         :  ProcGetGranuleLocality                     */
/*                   ProcGetGrVerticalSpatialDom                */
/*                   ProcGetOrbitCalcSpatialDom                 */
/*                   ProcGetMeasuredParameter                   */
/*                   ProcGetProcessingQA                        */
/*                   ProcGetGrStorageMedium                     */
/*                   ProcGetGranReview                          */
/*                   ProcGetGrSensorCharacteristics             */
/*                   ProcGetAssocPlatInstrSensor                */
/*                   ProcGetGrAnalysisShortName                 */
/*                   ProcGetGrCampaignShortName                 */
/*                   ProcGetGrIntInfoContent                    */
/*                   ProcGetGrStringInfoContent                 */
/*                   ProcGetGrFloatInfoContent                  */
/*                   ProcGetInputGranule                        */
/*                   ProcGetAncillaryInput                      */
/*                   ProcGetOrbitParamGranule                   */
/*                   ProcGetBrowseGranuleXref                   */
/*                   ProcGetQaRef                               */
/*                   ProcGetFileStorage                         */
/*                   ProcGetSDSRVLookup                         */
/*==============================================================*/


CREATE PROCEDURE ProcGetGranuleMetadata(@dbID  ID = 0)
AS
BEGIN
  IF (@dbID = 0)
```

```
 BEGIN
   RAISERROR 40117
   RETURN -999
 END
/*********************************************************/
/* Select data for following GlParameterLists:           */
/*    GranuleMetadata, CollectionDescriptionClass,       */
/*    ECSDataGranule, PGEVersionClass, RangeDateTime,    */
/*    SingleDateTime, and HorizontalSpatialDomainContainer */
/*********************************************************/

DECLARE @grdbId          varchar(40)
DECLARE @grtype          varchar(30)
DECLARE @grsubtype       varchar(30)
DECLARE @primCollId      varchar(40)
DECLARE @procHistId      varchar(40)
DECLARE @procHistTypeCode int
DECLARE @grlastUpd       varchar(60)
DECLARE @delEffectDate   datetime

DECLARE @ShortName       char(8)
DECLARE @VersionID       tinyint

DECLARE @SizeMBECS       float
DECLARE @ReprocPlanned   varchar(45)
DECLARE @ReprocActual    varchar(20)
DECLARE @LocalGrId       varchar(80)
DECLARE @DayNightFlag    char(5)
DECLARE @ProdDateTime    datetime
DECLARE @LocalVersId     varchar(60)

DECLARE @PGEVersion      char(10)

DECLARE @RangeEndTime    DsTMdTime
DECLARE @RangeEndDate    datetime
DECLARE @RangeBegTime    DsTMdTime
DECLARE @RangeBegDate    datetime

DECLARE @TimeOfDay       DsTMdTime
DECLARE @CalendarDate    datetime

DECLARE @BeginningDateTime datetime
DECLARE @EndingDateTime   datetime

DECLARE @ZoneIdent       varchar(64)

DECLARE @retStat int
DECLARE @type varchar(40)
DECLARE @subType varchar(40)
DECLARE @lookupValue varchar(255)

SELECT  @grdbId          = CONVERT(varchar(40), a.dbID),
       @grtype          = b.type,
       @grsubtype       = b.subType,
```

```
        @primCollId       = CONVERT(varchar(40), a.primaryCollectionId),
        @procHistId       = CONVERT(varchar(40), processingHistoryId),
        @procHistTypeCode = processingHistoryTypeCode,
        @grlastUpd        = CONVERT(varchar(60), a.lastUpdate, 109),
        @delEffectDate    = a.deleteEffectiveDate,

        @ShortName = a.ShortName,
        @VersionID = b.VersionID,

        @SizeMBECS     = SizeMBECSDataGranule,
        @ReprocPlanned = ReprocessingPlanned,
        @ReprocActual  = ReprocessingActual,
        @LocalGrId     = LocalGranuleID,
        @DayNightFlag  = DayNightFlag,
        @ProdDateTime  = ProductionDateTime,
        @LocalVersId   = LocalVersionID,

        @PGEVersion = PGEVersion,

        @RangeEndTime = RangeEndingTime,
        @RangeEndDate = RangeEndingDate,
        @RangeBegTime = RangeBeginningTime,
        @RangeBegDate = RangeBeginningDate,

        @TimeOfDay    = TimeOfDay,
        @CalendarDate = CalendarDate,

        @BeginningDateTime = BeginningDateTime,
        @EndingDateTime    = EndingDateTime,

        @ZoneIdent = ZoneIdentifier

  FROM    DsMdGranules a, DsMdCollections b
  WHERE   a.dbID = @dbID and a.primaryCollectionId = b.dbID

  IF (@@error != 0 OR @@rowcount != 1)
  BEGIN
    RAISERROR 45143
    RETURN -999
  END


  IF (@procHistId != NULL)
  BEGIN
      EXECUTE @retStat = ProcGetSDSRVLookup "type", "PH", "varchar", @lookupValue OUTPUT

      IF @retStat != 0
        BEGIN
          RAISERROR 20014 "Procedure ProcGetGranuleMetadata error - could not find parameter @type
for Production History."
          RETURN -999
        END

      SELECT @type =  @lookupValue
```

```
    EXECUTE @retStat = ProcGetSDSRVLookup "subType", "PH", "varchar", @lookupValue OUTPUT

    IF @retStat != 0
      BEGIN
        RAISERROR 20014 "Procedure ProcGetGranuleMetadata error - could not find parameter
@subType."
        RETURN -999
      END

    SELECT @subType = @lookupValue

    SELECT @procHistId = @type + ":" + @subType + ":" + @procHistId
  END

  /* Reorganize the Granule table columns to match the data model */
  /* first process the system attributes */
  SELECT dbID                  = @grdbId,
       type                  = @grtype,
       subType               = @grsubtype,
       primaryCollectionId   = @primCollId,
       productionHistoryId   = @procHistId,
       lastUpdate            = @grlastUpd,
       deleteEffectiveDate   = @delEffectDate,
       BeginningDateTime     = @BeginningDateTime,
       EndingDateTime        = @EndingDateTime

  /* next process the CollectionDescription */
  SELECT ShortName = @ShortName,
       VersionID = convert(int, @VersionID)

  /* next process ECSDataGranule */
  SELECT SizeMBECSDataGranule = @SizeMBECS,
       ReprocessingPlanned  = @ReprocPlanned,
       ReprocessingActual   = @ReprocActual,
       LocalGranuleID       = @LocalGrId,
       DayNightFlag         = @DayNightFlag,
       ProductionDateTime   = @ProdDateTime,
       LocalVersionID       = @LocalVersId

  /* next process PGEVersionClass */
  SELECT PGEVersion = @PGEVersion

  /* next process the RangeDateTime */
  IF (@RangeEndTime != NULL  AND  @RangeEndDate != NULL  AND
     @RangeBegTime != NULL  AND  @RangeBegDate != NULL)
    BEGIN
      SELECT RangeEndingTime    = @RangeEndTime,
           RangeEndingDate    = @RangeEndDate,
           RangeBeginningTime = @RangeBegTime,
           RangeBeginningDate = @RangeBegDate
    END
  ELSE  /* generate an empty row set for the wrapper to process */
    BEGIN
```

```
    SELECT RangeEndingTime    = @RangeEndTime,
           RangeEndingDate    = @RangeEndDate,
           RangeBeginningTime = @RangeBegTime,
           RangeBeginningDate = @RangeBegDate
    WHERE  1 = 2
  END

/* next process the SingleDateTime */
IF (@TimeOfDay != NULL  AND  @CalendarDate != NULL)
  BEGIN
    SELECT TimeOfDay    = @TimeOfDay,
           CalendarDate = @CalendarDate
  END
ELSE  /* generate an empty row set for the wrapper to process */
  BEGIN
    SELECT TimeOfDay    = @TimeOfDay,
           CalendarDate = @CalendarDate
    WHERE  1 = 2
  END

/* next process the ZoneIdentifier */
SELECT @ZoneIdent

/*******************************************************/
/* Select data for following GlParameterList:       */
/*    GranuleLocality                                  */
/*******************************************************/
EXECUTE @retStat = ProcGetGranuleLocality  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56039
  RETURN -999
END

/*******************************************************/
/* Select data for following GlParameterList:       */
/*    VerticalSpatialDomainContainer                */
/*******************************************************/
EXECUTE @retStat = ProcGetGrVerticalSpatialDom  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56040
  RETURN -999
END


/*************************************************************/
/* Select data for following GlParameterList:                */
/*    OrbitCalculatedSpatialDomainContainer                    */
/*************************************************************/
EXECUTE @retStat = ProcGetOrbitCalcSpatialDom  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56041
```

```
  RETURN -999
END

/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    MeasuredParameterContainer                                 */
/*****************************************************************/
EXECUTE @retStat = ProcGetMeasuredParameter  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56042
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    ProcessingQAContainer                                      */
/*****************************************************************/
EXECUTE @retStat = ProcGetProcessingQA  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56043
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    StorageMedium                                              */
/*****************************************************************/
EXECUTE @retStat = ProcGetGrStorageMedium  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56044
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    Review                                                     */
/*****************************************************************/
EXECUTE @retStat = ProcGetGranReview  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56045
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    SensorCharacteristic                                       */
```

```
/*****************************************************************/
EXECUTE @retStat = ProcGetGrSensorCharacteristic  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56046
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    AssociatedPlatformInstrumentSensorContainer                */
/*****************************************************************/
EXECUTE @retStat = ProcGetAssocPlatInstrSensor  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56047
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    AnalysisShortName                                          */
/*****************************************************************/
EXECUTE @retStat = ProcGetGrAnalysisShortName  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56048
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    CampaignShortName                                          */
/*****************************************************************/
EXECUTE @retStat = ProcGetGrCampaignShortName  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56049
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    AdditionalAttributes                                       */
/*****************************************************************/
EXECUTE @retStat = ProcGetGrIntInfoContent  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56050 "ProcGetGranuleMetadata: Failed select on DsMdGrIntegerInfoContent"
  RETURN -999
```

```
END


EXECUTE @retStat = ProcGetGrStringInfoContent  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56051
  RETURN -999
END


EXECUTE @retStat = ProcGetGrFloatInfoContent  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56052
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    InputGranule                                               */
/*****************************************************************/
EXECUTE @retStat = ProcGetInputGranule  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56053
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    AncillaryInputGranule                                      */
/*****************************************************************/
EXECUTE @retStat = ProcGetAncillaryInput  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56054
  RETURN -999
END


/*****************************************************************/
/* Select data for following GlParameterList:                    */
/*    OrbitParametersGranule                                     */
/*****************************************************************/
EXECUTE @retStat = ProcGetOrbitParamGranule  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56055
  RETURN -999
END
```

```
  /*************************************************************/
  /* Select data for following GlParameterList:                */
  /*    browseGranules                                         */
  /*************************************************************/
  EXECUTE @retStat = ProcGetBrowseGranuleXref  @dbID
  IF @retStat != 0
  BEGIN
    RAISERROR 56056
    RETURN -999
  END


  /*************************************************************/
  /* Select data for following GlParameterList:                */
  /*    qaGranules                                             */
  /*************************************************************/
  EXECUTE @retStat = ProcGetGrQaRef  @dbID
  IF @retStat != 0
  BEGIN
    RAISERROR 56057
    RETURN -999
  END


  /*************************************************************/
  /* Select data for following GlParameterList:                */
  /*    nonScienceData                                         */
  /*************************************************************/
  EXECUTE @retStat = ProcGetFileStorage  @dbID
  IF @retStat != 0
  BEGIN
    RAISERROR 56058
    RETURN -999
  END
END
go
```

ProcGetGranuleOrbitParmXref**Procedure ProcGetGranuleOrbitParmXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGranuleOrbitParmXref                  */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the orbitParmGranuleId associated with       */
/*                   Granules.                                    */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGranuleOrbitParmXref(@granuleId ID = 0)
```

```
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40086
      RETURN -999
    END
   SELECT CONVERT(varchar(30), granuleId),
            CONVERT(varchar(30), orbitParamGranuleId)
   FROM   DsMdGranulesOrbitParmXref
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
      RAISERROR 45103
      RETURN -999
   END
END
go
```

ProcGetGranuleVersions**Procedure ProcGetGranuleVersions**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetGranuleVersions                        */
/*  Release       :  B0                                            */
/*  Module        :  Granules                                      */
/*  Created on    :  04/22/97                                      */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                   granule version data associated with          */
/*                   Granules.                                     */
/*  Calls         :                                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetGranuleVersions(@currentGranuleId ID = 0)
AS
BEGIN
  IF (@currentGranuleId = 0)
    BEGIN
      RAISERROR 40087
      RETURN -999
    END
   SELECT currentVersion,
         previousGranuleId = convert(varchar(30), previousGranuleId),
         previousVersion
   FROM   DsMdGranuleVersions
   WHERE  currentGranuleId = @currentGranuleId
   IF @@error!=0
   BEGIN
      RAISERROR 45104
      RETURN -999
   END
END
go
```

ProcGetGrCampaignShortName**Procedure ProcGetGrCampaignShortName**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrCampaignShortName                   */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the CampaignShortName associated with        */
/*                   Granules.                                    */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGrCampaignShortName(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40071
      RETURN -999
    END
   SELECT t1.CampaignShortName
   FROM   DsMdCampaign t1,
          DsMdGranuleCampaignXref t2
   WHERE  t2.granuleId = @granuleId AND
          t1.dbID = t2.campaignId
   IF @@error!=0
   BEGIN
     RAISERROR 45088
     RETURN -999
   END
END
go
```

ProcGetGrFloatInfoContent**Procedure ProcGetGrFloatInfoContent**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrFloatInfoContent                    */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   float name/value pairs for Granules.        */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGrFloatInfoContent(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
```

```
   BEGIN
     RAISERROR 40072
     RETURN -999
    END
   SELECT
            a.AdditionalAttributeName,
            CONVERT(varchar(60), grFloatValue)
   FROM
            DsMdGrFloatInfoContent c,
         DsMdAdditionalAttributes a
   WHERE
            c.granuleId = @granuleId
     AND  c.attributeId = a.attributeId
   ORDER BY
         a.AdditionalAttributeName

   IF @@error!=0
   BEGIN
     RAISERROR 45089
     RETURN -999
   END
END
go
```

ProcGetGrInstrumentXref**Procedure ProcGetGrInstrumentXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrInstrumentXref                      */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the instrumentId and OperationMode           */
/*                   associated with Granules.                    */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGrInstrumentXref(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40073
      RETURN -999
    END
   SELECT granuleId    = CONVERT(varchar(30), granuleId),
            instrumentId = CONVERT(varchar(30), instrumentId),
         OperationMode
   FROM   DsMdGranuleInstrumentXref
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
```

```
    RAISERROR 45090
    RETURN -999
  END
END
go
```

ProcGetGrIntInfoContent**Procedure ProcGetGrIntInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetGrIntInfoContent                    */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  04/17/97                                   */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   integer name/value pairs for Granules.     */
/*  Calls         :                                             */
/*==============================================================*/

CREATE PROCEDURE ProcGetGrIntInfoContent(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
   BEGIN
     RAISERROR 40074
     RETURN -999
   END

  SELECT
        a.AdditionalAttributeName,
            CONVERT(varchar(60), grIntValue)
  FROM
            DsMdGrIntegerInfoContent c,
        DsMdAdditionalAttributes a
  WHERE
            c.granuleId = @granuleId
    AND  c.attributeId = a.attributeId
  ORDER BY
        a.AdditionalAttributeName

  IF @@error!=0
  BEGIN
    RAISERROR 45091
    RETURN -999
  END
END
go
```

ProcGetGrLastUpdate**Procedure ProcGetGrLastUpdate**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrLastUpdate                  */
/*  Release       :  B0                                 */
/*  Module        :  Granules                             */
/*  Created on    :  4/3/97                               */
/*  Purpose       :  Create stored procedure to retrieve     */
/*                   the last update for a Granule.          */
/*  Calls         :  None                                 */
/*                                                        */
/*================================================================*/

CREATE PROCEDURE ProcGetGrLastUpdate(@dbId  ID = 0)
AS
BEGIN
  IF (@dbId = 0)
    BEGIN
      RAISERROR 40075
      RETURN -999
    END
   SELECT convert(varchar(60), lastUpdate, 109)
   FROM             DsMdGranules
   WHERE  dbID = @dbId
   IF @@error!=0
   BEGIN
     RAISERROR 45092
     RETURN -999
   END
END
go
```

ProcGetGrPlatformXref**Procedure ProcGetGrPlatformXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrPlatformXref                  */
/*  Release       :  B0                                 */
/*  Module        :  Granules                             */
/*  Created on    :  04/17/97                               */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                   the platformId associated with Granules.   */
/*  Calls         :                                       */
/*================================================================*/

CREATE PROCEDURE ProcGetGrPlatformXref(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
```

```
    RAISERROR 40076
    RETURN -999
  END
  SELECT granuleId  = CONVERT(varchar(30), granuleId),
         platformId = CONVERT(varchar(30), platformId)
  FROM   DsMdGranulePlatformXref
  WHERE  granuleId = @granuleId
  IF @@error!=0
  BEGIN
    RAISERROR 45093
    RETURN -999
  END
END
go
```

ProcGetGrQaRef**Procedure ProcGetGrQaRef**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrQaRef                               */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  07/29/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the qaGranuleIds associated with the         */
/*                   Granules specified by @granuleId             */
/*  Calls         :                                               */
/*================================================================*/


CREATE PROCEDURE ProcGetGrQaRef(@granuleId ID = 0)
AS
BEGIN

DECLARE @retStat  int
DECLARE @lookupValue varchar(40)
DECLARE @subType varchar(30)
DECLARE @type varchar(30)

  IF (@granuleId = 0)
    BEGIN
       RAISERROR 40116
       RETURN -999
    END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "type", "QA", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetGrQaRef error - could not find parameter @type ."
      RETURN -999
    END
```

```
  SELECT @type = @lookupValue

  EXECUTE @retStat = ProcGetSDSRVLookup "subType", "QA", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetGrQaRef error - could not find parameter @subType."
      RETURN -999
    END

  SELECT @subType = @lookupValue

  SELECT qaGranuleId = @type + ":" + @subType + ":" + CONVERT(varchar(50), dbID)
       FROM   DsMdQaGranule
       WHERE  granuleId = @granuleId

  IF (@@error != 0)
   BEGIN
     RAISERROR 45142
     RETURN -999
   END

END
go
```

ProcGetGrSensorCharacteristic**Procedure ProcGetGrSensorCharacteristic**

Code

```
/*==============================================================*/
/*  Name           :  ProcGetGrSensorCharacteristic           */
/*  Release        :  B0                                      */
/*  Module         :  Data Originator                         */
/*  Created on     :  4/03/97                                 */
/*  Purpose        :  Create stored procedure to retrieve     */
/*                    sensor characteristic data.             */
/*  Calls          :  None                                    */
/*==============================================================*/

CREATE PROCEDURE ProcGetGrSensorCharacteristic(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40077
      RETURN -999
    END
    SELECT SensorCharacteristicName,
          SensorCharacteristicValue
    FROM   DsMdGrSensorCharacteristics
    WHERE  granuleId = @granuleId

    IF @@error!=0
```

```
   BEGIN
     RAISERROR 45094
     RETURN -999
   END
END
go
```

ProcGetGrSensorXref**Procedure ProcGetGrSensorXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrSensorXref                          */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   the sensorId associated with Granules.       */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetGrSensorXref(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40078
      RETURN -999
    END
   SELECT granuleId = CONVERT(varchar(30), granuleId),
            sensorId = CONVERT(varchar(30), sensorId)
   FROM   DsMdGranuleSensorXref
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
     RAISERROR 45095
     RETURN -999
   END
END
go
```

ProcGetGrStorageMedium**Procedure ProcGetGrStorageMedium**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrStorageMedium                       */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   storage medium data for Granules.            */
/*  Calls         :                                               */
/*================================================================*/
```

```
CREATE PROCEDURE ProcGetGrStorageMedium(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40079
      RETURN -999
    END
   SELECT StorageMedium
   FROM   DsMdGrStorageMedium
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
      RAISERROR 45096
      RETURN -999
   END
END
go
```

ProcGetGrStringInfoContent**Procedure ProcGetGrStringInfoContent**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrStringInfoContent              */
/*  Release       :  B0                                  */
/*  Module        :  Granules                              */
/*  Created on    :  04/17/97                               */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                   string name/value pairs for Granules.     */
/*  Calls         :                                      */
/*================================================================*/

CREATE PROCEDURE ProcGetGrStringInfoContent(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40080
      RETURN -999
    END

   SELECT
            a.AdditionalAttributeName,
            c.grStringValue
   FROM
            DsMdGrStringInfoContent c,
            DsMdAdditionalAttributes a
   WHERE
            c.granuleId = @granuleId
     AND  c.attributeId = a.attributeId
   ORDER BY
         a.AdditionalAttributeName
```

```
  IF @@error!=0
  BEGIN
    RAISERROR 45097
    RETURN -999
  END
END
go
```

ProcGetGrVerticalSpatialDom**Procedure ProcGetGrVerticalSpatialDom**

Code

```
/*================================================================*/
/*  Name          :  ProcGetGrVerticalSpatialDom              */
/*  Release       :  B0                                       */
/*  Module        :  Granules                                 */
/*  Created on    :  04/17/97                                 */
/*  Purpose       :  Create stored procedure to retrieve      */
/*                   Granule Vertical Spatial Domain data.    */
/*  Calls         :                                           */
/*================================================================*/

CREATE PROCEDURE ProcGetGrVerticalSpatialDom(@granuleId ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40081
      RETURN -999
    END
  SELECT VerticalSpatialDomainType,
         VerticalSpatialDomainValue
  FROM   DsMdGrVerticalSpatialDomain
  WHERE  granuleId = @granuleId

  IF @@error!=0
  BEGIN
    RAISERROR 45098
    RETURN -999
  END
END
go
```

ProcGetInputGranule**Procedure ProcGetInputGranule**

Code

```
/*================================================================*/
/*  Name          :  ProcGetInputGranule                      */
/*  Release       :  B0                                       */
/*  Module        :  Granules                                 */
/*  Created on    :  04/17/97                                 */
/*  Purpose       :  Create stored procedure to retrieve      */
/*                   input granule data.                      */
/*  Calls         :                                           */
/*================================================================*/
```

```
CREATE PROCEDURE ProcGetInputGranule(@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
     RAISERROR 40016
     RETURN -999
  END

-- put rows in order that they were inserted
-- to match how they were input (MCF)
   SELECT inputGranule,
          InputPointer,
          inputGranuleFileCount,
          insertDatetime
   INTO   #tempInputGranule
   FROM
          DsMdInputGranule
   WHERE  granuleId = @granuleId
   ORDER BY inputGranule,
           insertDatetime

-- ignore the inputGranuleFileCount for now
-- can/will be used to return multiple "logical" rows
-- for multi-file granules as needed.
   SELECT InputPointer
   FROM #tempInputGranule

   IF @@error!=0
   BEGIN
     RAISERROR 45016
     RETURN -999
   END

END
go
```

ProcGetInsertSequence**Procedure ProcGetInsertSequence**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetInsertSequence                      */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  selects from DsMdInsertSequence            */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetInsertSequence
```

```
AS

SELECT
        BaseType,
        StoredProcName,
        ListName,
        CommandType,
        DataType,
        SequenceNo
FROM
        DsMdInsertSequence
ORDER BY BaseType,
        SequenceNo

IF @@error!=0
  BEGIN
        RAISERROR 45105
        RETURN -999
  END
go
```

ProcGetInstrument**Procedure ProcGetInstrument**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetInstrument                        */
/*  Release         :  B0                                   */
/*  Module          :  DataOriginator                           */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for selecting */
/*                     data from DsMdInstrument table        */
/*  Calls           :  ProcGetInstrumentChar                    */
/*                     ProcGetOperationMode                    */
/*==============================================================*/

CREATE PROCEDURE ProcGetInstrument(@dbID  ID = 0)
AS
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40088
      RETURN -999
    END
    SELECT
          InstrumentShortName,
          InstrumentLongName,
          InstrumentTechnique,
          NumberOfSensors
    FROM    DsMdInstrument
    WHERE   dbID = @dbID
    IF @@error!=0
    BEGIN
          RAISERROR 45106
```

```
        RETURN -999
    END

    DECLARE @retStat  int
    EXECUTE @retStat = ProcGetOperationMode @dbID
    IF @retStat!=0
    BEGIN
      RAISERROR 56022
      RETURN -999
    END

    EXECUTE @retStat = ProcGetInstrumentChar @dbID
    IF @retStat!=0
    BEGIN
      RAISERROR 56023
      RETURN -999
    END
END
go
```

ProcGetInstrumentChar**Procedure ProcGetInstrumentChar**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetInstrumentChar                    */
/*  Release         :  B0                                       */
/*  Module          :  DataOriginator                           */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for selecting    */
/*                  data from DsMdInstrumentCharacteristic */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/

CREATE PROC  ProcGetInstrumentChar(@instrumentId  ID = 0)
AS
BEGIN
  IF (@instrumentId = 0)
    BEGIN
      RAISERROR 40018
      RETURN -999
    END
   SELECT
         InstrumentCharacteristicName,
         InstrumentCharacteristicDesc,
         InstrumentCharacteristicUnit,
         InstrumentCharDataType,
         InstrumentCharacteristicValue

   FROM  DsMdInstrumentCharacteristic
   WHERE instrumentId = @instrumentId
```

```
   IF @@error!=0
   BEGIN
         RAISERROR 45018
         RETURN -999
   END
END
go
```

ProcGetInstrumentID**Procedure ProcGetInstrumentID**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetInstrumentID                           */
/*  Release       :  B0                                            */
/*  Module        :  Data Originator                               */
/*  Created on    :  04/22/97                                      */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                   the id for an instrument.                     */
/*  Calls         :                                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetInstrumentID(@platformId          ID = 0,
                              @InstrumentShortName  varchar(20))
AS
BEGIN
  IF (@platformId = 0)
    BEGIN
      RAISERROR 40089
      RETURN -999
    END
   SELECT dbID = CONVERT(varchar(30), dbID)
   FROM   DsMdInstrument
   WHERE  platformId = @platformId AND
          InstrumentShortName = @InstrumentShortName

   IF (@@error != 0)
   BEGIN
     RAISERROR 45107
     RETURN -999
   END

END
go
```

ProcGetJoins**Procedure ProcGetJoins**

Code

```
/*================================================================*/
/*  Name          :  ProcGetJoins                                 */
/*  Release       :  B0                                           */
/*  Module        :  System                                       */
/*  Created on    :  11/14/97                                     */
/*  Purpose       :  Create stored procedure to get all the       */
/*                   entries from the DsMdJoins table             */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcGetJoins
AS
BEGIN
   SELECT table1,
          column1,
          table2,
          column2
   FROM
          DsMdJoins

   IF @@error!=0
   BEGIN
      RAISERROR 45282
          /*  "ProcGetJoins: Failed select on DsMdJoins"  */
      RETURN -999
   END
END
go
```

ProcGetLimitedMetadata**Procedure ProcGetLimitedMetadata**

Code

```
/*================================================================*/
/*  Name          :  ProcGetLimitedMetadata                       */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  05/06/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   limited Granule metadata.                    */
/*  Calls         :  ProcGetVerticalSpatialDom                    */
/*                   ProcGetGrIntInfoContent                      */
/*                   ProcGetGrStringInfoContent                   */
/*                   ProcGetGrFloatInfoContent                    */
/*                   ProcGetFileStorage                           */
/*================================================================*/

CREATE PROCEDURE ProcGetLimitedMetadata(@dbID  ID = 0)
AS
```

```
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40090
      RETURN -999
    END
   DECLARE @grtype          varchar(30)
   DECLARE @grsubtype       varchar(30)
   DECLARE @grdbId          varchar(40)
   DECLARE @grlastUpd       varchar(60)
   DECLARE @primCollId      varchar(40)
   DECLARE @delEffectDate   datetime

   DECLARE @ShortName       char(8)
   DECLARE @VersionID       tinyint

   DECLARE @SizeMBECS       float
   DECLARE @ReprocPlanned   varchar(45)
   DECLARE @ReprocActual    varchar(20)
   DECLARE @LocalGrId       varchar(80)
   DECLARE @DayNightFlag    char(5)
   DECLARE @ProdDateTime    datetime
   DECLARE @LocalVersId     varchar(60)

   DECLARE @PGEVersion      char(10)

   DECLARE @RangeEndTime    DsTMdTime
   DECLARE @RangeEndDate    datetime
   DECLARE @RangeBegTime    DsTMdTime
   DECLARE @RangeBegDate    datetime

   DECLARE @TimeOfDay       DsTMdTime
   DECLARE @CalendarDate    datetime

   DECLARE @BeginningDateTime datetime
   DECLARE @EndingDateTime   datetime

   DECLARE @ZoneIdent       varchar(64)



   SELECT  @grdbId       = CONVERT(varchar(40), a.dbID),
         @grtype       = b.type,
         @grsubtype    = b.subType,
         @primCollId   = CONVERT(varchar(40), a.primaryCollectionId),
         @grlastUpd    = a.lastUpdate,
         @delEffectDate = a.deleteEffectiveDate,

         @ShortName    = a.ShortName,
         @VersionID    = b.VersionID,

         @SizeMBECS    = a.SizeMBECSDataGranule,
         @ReprocPlanned = ReprocessingPlanned,
         @ReprocActual  = ReprocessingActual,
```

```
        @LocalGrId     = LocalGranuleID,
        @DayNightFlag  = DayNightFlag,
        @ProdDateTime  = ProductionDateTime,
        @LocalVersId   = LocalVersionID,

        @PGEVersion    = PGEVersion,

        @RangeEndTime  = RangeEndingTime,
        @RangeEndDate  = RangeEndingDate,
        @RangeBegTime  = RangeBeginningTime,
        @RangeBegDate  = RangeBeginningDate,

        @TimeOfDay     = TimeOfDay,
        @CalendarDate  = CalendarDate,

        @BeginningDateTime = BeginningDateTime,
        @EndingDateTime    = EndingDateTime,

        @ZoneIdent     = a.ZoneIdentifier

FROM    DsMdGranules a, DsMdCollections b
WHERE   a.dbID = @dbID and a.primaryCollectionId = b.dbID

IF (@@error != 0 OR @@rowcount != 1)
BEGIN
  RAISERROR 45108
  RETURN -999
END

/* Reorganize the Granule table columns to match the data model */
/* first process the system attributes */
SELECT dbID                     = @grdbId,
       type                     = @grtype,
       subType                  = @grsubtype,
       primaryCollectionId      = @primCollId,
       lastUpdate               = @grlastUpd,
       deleteEffectiveDate      = @delEffectDate,
       BeginningDateTime        = @BeginningDateTime,
       EndingDateTime           = @EndingDateTime


/* next process the CollectionDescription */
SELECT ShortName = @ShortName,
       VersionID = convert(int, @VersionID)

/* next process ECSDataGranule */
SELECT SizeMBECSDataGranule = @SizeMBECS,
       ReprocessingPlanned  = @ReprocPlanned,
       ReprocessingActual   = @ReprocActual,
       LocalGranuleID       = @LocalGrId,
       DayNightFlag         = @DayNightFlag,
       ProductionDateTime   = @ProdDateTime,
       LocalVersionID       = @LocalVersId
```

```
/* next process PGEVersionClass */
SELECT PGEVersion = @PGEVersion

/* next process the RangeDateTime */
IF (@RangeEndTime != NULL  AND  @RangeEndDate != NULL  AND
   @RangeBegTime != NULL  AND  @RangeBegDate != NULL)
 BEGIN
    SELECT RangeEndingTime    = @RangeEndTime,
           RangeEndingDate    = @RangeEndDate,
           RangeBeginningTime = @RangeBegTime,
           RangeBeginningDate = @RangeBegDate
 END
ELSE  /* generate an empty row set for the wrapper to process */
 BEGIN
    SELECT RangeEndingTime    = @RangeEndTime,
           RangeEndingDate    = @RangeEndDate,
           RangeBeginningTime = @RangeBegTime,
           RangeBeginningDate = @RangeBegDate
    WHERE  1 = 2
 END

/* next process the SingleDateTime */
IF (@TimeOfDay != NULL  AND  @CalendarDate != NULL)
 BEGIN
    SELECT TimeOfDay    = @TimeOfDay,
           CalendarDate = @CalendarDate
 END
ELSE  /* generate an empty row set for the wrapper to process */
 BEGIN
    SELECT TimeOfDay    = @TimeOfDay,
           CalendarDate = @CalendarDate
    WHERE  1 = 2
 END

DECLARE @retStat int

EXECUTE @retStat = ProcGetGrIntInfoContent  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56024
  RETURN -999
END


EXECUTE @retStat = ProcGetGrStringInfoContent  @dbID
IF @retStat != 0
BEGIN
  RAISERROR 56025
  RETURN -999
END


EXECUTE @retStat = ProcGetGrFloatInfoContent  @dbID
IF @retStat != 0
```

```
  BEGIN
    RAISERROR 56026
    RETURN -999
  END


  EXECUTE @retStat = ProcGetFileStorage  @dbID
  IF @retStat != 0
  BEGIN
    RAISERROR 56027
    RETURN -999
  END
END
go
```

ProcGetLocality**Procedure ProcGetLocality**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetLocality                               */
/*  Release       :  B0                                            */
/*  Module        :  Collections                                   */
/*  Created on    :  3/25/97                                       */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                locality data for Collections.                   */
/*  Calls         :                                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetLocality(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
   BEGIN
    RAISERROR 40005
    RETURN -999
   END
  SELECT LocalityType,
       LocalityDescription
  FROM   DsMdLocality
  WHERE  collectionId = @collectionId
  IF @@error!=0
  BEGIN
    RAISERROR 45005
    RETURN -999
  END
END
go
```

ProcGetMeasuredParameter**Procedure ProcGetMeasuredParameter**

Code

```
/*================================================================*/
/*  Name          :  ProcGetMeasuredParameter                     */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   measured parameter data for Granules.        */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetMeasuredParameter(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40014
      RETURN -999
    END
   SELECT ParameterName,
         QAPercentMissingData,
         QAPercentOutOfBoundsData,
         QAPercentInterpolatedData,
         QAPercentCloudCover,
         AutomaticQualityFlag,
         AutomaticQualityFlagExplan,
         OperationalQualityFlag,
         OperationalQualityFlagExplan,
             ScienceQualityFlag,
             ScienceQualityFlagExplan
   FROM   DsMdMeasuredParameter
   WHERE  granuleId = @granuleId
   IF @@error!=0
   BEGIN
     RAISERROR 45014
     RETURN -999
   END
END
go
```

ProcGetMultiDateTime**Procedure ProcGetMultiDateTime**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetMultiDateTime                     */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                  data from DsMdMultipleDateTimePeriod   */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetMultiDateTime(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40092
      RETURN -999
    END

    SELECT MultipleDateName
    FROM   DsMdMultipleDateTimePeriod
    WHERE  collectionId = @collectionId

    IF (@@error!= 0)
    BEGIN
          RAISERROR 45110
          RETURN -999
    END

   DECLARE @retStat  int

   EXECUTE @retStat = ProcGetSingleDateTimeInMultPd @collectionId
   IF (@retStat != 0)
   BEGIN
     RAISERROR 56028
     RETURN -999
   END



END
go
```

ProcGetMultTypeCollection**Procedure ProcGetMultTypeCollection**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetMultTypeCollection                  */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   multiple type collection data for          */
/*                   Collections.                               */
/*  Calls         :                                             */
/*==============================================================*/

CREATE PROCEDURE ProcGetMultTypeCollection(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40091
      RETURN -999
    END
   SELECT collectionId,
         AggregationRelationship,
         AggregationType,
         AggregationValue
   FROM   DsMdMultipleTypeCollection
   WHERE  collectionId = @collectionId
   IF @@error!=0
   BEGIN
     RAISERROR 45109
     RETURN -999
   END
END
go
```

ProcGetOneESDTConfiguredType**Procedure ProcGetOneESDTConfiguredType**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetOneESDTConfiguredType               */
/*  Release       :  B0                                         */
/*  Module        :                                             */
/*  Created on    :  2/08/98                                    */
/*  Purpose       :                                             */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetOneESDTConfiguredType
(
   @configuredName     varchar(20) = NULL,
```

```
   @versionID          tinyint = NULL
)
AS
BEGIN
   IF (@configuredName = NULL) OR (@configuredName = "")
      BEGIN
        RAISERROR 40118
        RETURN -999
      END

   IF @versionID = NULL
      BEGIN
        RAISERROR 30000 "ProcGetOneESDTConfiguredType: Missing parameter @versionID"
        RETURN -999
      END

   SELECT configuredName,
        versionID=CONVERT(VARCHAR(20), versionID),
        archiveIdentifier,
        backupIdentifier,
        offsiteIdentifier,
        description,
        descriptorFileName,
        dllFileName,
        eventFileName
   FROM DsGeESDTConfiguredType
   WHERE configuredName = @configuredName
   AND versionID = @versionID
   IF @@error!=0
   BEGIN
      RAISERROR 45144
      RETURN -999
   END
END
go
```

ProcGetOperationMode**Procedure ProcGetOperationMode**

Code

```
/*==============================================================*/
/*  Name           :  ProcGetOperationMode                      */
/*  Release        :  B0                                        */
/*  Module         :  Data Originator                           */
/*  Created on     :  4/03/97                                   */
/*  Purpose        :  Create stored procedure to retrieve       */
/*                 operation mode data.                         */
/*  Calls          :  None                                      */
/*==============================================================*/

CREATE PROCEDURE ProcGetOperationMode(@instrumentId  ID = 0)
AS
BEGIN
  IF (@instrumentId = 0)
```

```
   BEGIN
     RAISERROR 40009
     RETURN -999
   END
   SELECT OperationMode
   FROM   DsMdOperationMode
   WHERE instrumentId = @instrumentId

   IF @@error!=0
      BEGIN
           RAISERROR 45009
                    RETURN -999
      END
END
go
```

ProcGetOrbitCalcSpatialDom**Procedure ProcGetOrbitCalcSpatialDom**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetOrbitCalcSpatialDom             */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                 */
/*  Created on    :  04/17/97                                 */
/*  Purpose       :  Create stored procedure to retrieve     */
/*                Orbit Calculated Spatial Domain data.   */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcGetOrbitCalcSpatialDom(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40013
      RETURN -999
    END

  SELECT OrbitModelName,
        OrbitNumber,
        StartOrbitNumber,
        StopOrbitNumber,
        EquatorCrossingLongitude,
        EquatorCrossingDate,
        EquatorCrossingTime
  FROM   DsMdOrbitCalcSpatialDomain
  WHERE  granuleId = @granuleId

  IF @@error!=0
  BEGIN
    RAISERROR 45013
    RETURN -999
  END
```

```
END
go
```

ProcGetOrbitCalculatedSpatial**Procedure ProcGetOrbitCalculatedSpatial**

Code

```
/*================================================================*/
/*  Name          :  ProcGetOrbitCalculatedSpatial          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  04/17/97                               */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                   Orbit Calculated Spatial data.         */
/*  Calls         :  None                                   */
/*================================================================*/

CREATE PROC ProcGetOrbitCalculatedSpatial(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40093
      RETURN -999
    END

   SELECT roundedEqCrossingLongitude,
        platInstCode
   FROM   DsMdOrbitCalculatedSpatial
   WHERE  granuleId = @granuleId

   IF @@error!=0
   BEGIN
      RAISERROR 45111
      RETURN -999
   END
END
go
```

ProcGetOrbitParamGranule**Procedure ProcGetOrbitParamGranule**

Code

```
/*================================================================*/
/*  Name          :  ProcGetOrbitParamGranule               */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  04/17/97                               */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                   the GeOID of the OrbitParamtersGranule */
/*                   associated with a Granule.             */
/*  Calls         :  ProcGetSDSRVLookup                     */
/*================================================================*/

CREATE PROCEDURE ProcGetOrbitParamGranule(@granuleId ID = 0)
```

```
AS
BEGIN

  DECLARE @retStat int
  DECLARE @lookupValue varchar(40)
  DECLARE @subType varchar(30)
  DECLARE @type varchar(30)

  IF (@granuleId = 0)
    BEGIN
       RAISERROR 40114
       RETURN -999
    END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "type", "OR", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetOrbitParamGranule error - could not find parameter @type."
      RETURN -999
    END

  SELECT @type = @lookupValue

  EXECUTE @retStat = ProcGetSDSRVLookup "subType", "ORBPARM", "varchar", @lookupValue
OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetOrbitParamGranule error - could not find parameter
@subType."
      RETURN -999
    END

  SELECT @subType = @lookupValue

   SELECT CONVERT(varchar, @type) +
         ":" +
         CONVERT(varchar, @subType) +
         ":" +
         CONVERT(varchar(50), t1.dbID)
   FROM   DsMdOrbitParametersGranule t1,
         DsMdGranulesOrbitParmXref t2
   WHERE  t2.granuleId = @granuleId AND
         t1.dbID = t2.orbitParamGranuleId

   IF (@@error != 0)
     BEGIN
       RAISERROR 45140
       RETURN -999
     END
END
go
```

ProcGetParmToTableVector**Procedure ProcGetParmToTableVector**

Code

```
/*================================================================*/
/*  Name          :  ProcGetParmToTableVector                     */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   parameter (datatype) to table data for       */
/*                   Collections.                                 */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetParmToTableVector(
                              @DataType char(10))
AS
BEGIN
  SELECT DataType,
       grTableName,
       clTableName
  FROM   DsMdParmToTableVector
  WHERE  DataType = @DataType
  IF @@error!=0
  BEGIN
    RAISERROR 45112
    RETURN -999
  END
END
go
```

ProcGetPersonOrgXref**Procedure ProcGetPersonOrgXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetContactPerson                         */
/*  Release       :  B0                                           */
/*  Module        :  Contact                                      */
/*  Created on    :  3/24/97                                      */
/*  Purpose       :  Create stored procedure to Contact,          */
/*                   and associate data tables.                   */
/*  Calls         :                                               */
/*                                                                */
/*================================================================*/

CREATE PROCEDURE ProcGetPersonOrgXref(@personId ID = 0)
AS
BEGIN
 IF (@personId = 0)
   BEGIN
     RAISERROR 40095
     RETURN -999
```

```
   END
   SELECT personId = CONVERT(varchar(30), personId),
        organizationId = CONVERT(varchar(30), personId)
   FROM   DsMdPersonOrganizationXref
   WHERE  personId = @personId

   IF (@@error!= 0)
   BEGIN
          RAISERROR 45113
          RETURN -999
   END
END
go
```

ProcGetPGEGroup**Procedure ProcGetPGEGroup**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetPGEGroup                            */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to Delivered,      */
/*                   Algorithm Package, and associate data      */
/*                   tables.                                    */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetPGEGroup(@dapDBId   ID = 0)
AS
BEGIN
  IF (@dapDBId = 0)
    BEGIN
      RAISERROR 40008
      RETURN -999
    END
    SELECT DAPPGEName,
         DAPPGEVersion,
         DAPSWVersion
    FROM   DsMdPGEGroup
    WHERE  dapDBId = @dapDBId

    IF @@error!=0
    BEGIN
          RAISERROR 45008 "ProcGetPGEGroup: Failed select on DsMdPGEGroup"
          RETURN -999
    END
END
go
```

ProcGetPlanCoor**Procedure ProcGetPlanCoor**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetPlanCoor                          */
/*  Release         :  B0                                       */
/*  Module          :  Spatial                                  */
/*  Created on      :  3/25/97                                  */
/*  Purpose         :  Create Stored Procedure for retrieving*/
/*                  data from DsMdPlanarCoordinateSystems */
/*                  table                                       */
/*  Calls           :  None                                     */
/*==============================================================*/

CREATE PROCEDURE ProcGetPlanCoor(@collectionId ID = 0)
AS
BEGIN

   IF  (@collectionId = 0)
   BEGIN
      RAISERROR 40096
      RETURN -999
   END

   SELECT PlanarDistanceUnits,
         PlanarCoordinateEncodingMeth,
         DistanceResolution,
         BearingResolution,
         BearingUnits,
         BearingReferenceDirection,
         BearingReferenceMeridian,
            AbscissaResolution,
         OrdinateResolution,
         MapProjectionName,
         MapProjectionPointer,
         LocalPlanarCoordinateSystemDes,
         LocalPlanarGeoreferenceInforma,
         GridCoordinateSystemName
   FROM   DsMdPlanarCoordinateSystems
   WHERE  collectionId = @collectionId
   ORDER BY planarCoordinateSystemsID

   IF (@@error!= 0)
   BEGIN
         RAISERROR 45114
         RETURN -999
   END
END
go
```

ProcGetPlatform**Procedure ProcGetPlatform**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetPlatform                          */
/*  Release         :  B0                                       */
/*  Module          :  DataOriginator                           */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for selecting */
/*                  data from DsMdPlatform table                */
/*  Calls           :  ProcGetPlatformChar                      */
/*==============================================================*/

CREATE PROC  ProcGetPlatform(@dbID ID = 0)
AS
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40098
      RETURN -999
    END
   SELECT PlatformShortName, PlatformLongName, PlatformType
   FROM   DsMdPlatform
   WHERE  dbID = @dbID

    IF @@error!=0
    BEGIN
          RAISERROR 45116
          RETURN -999
    END

    DECLARE @retStat  int
    EXECUTE @retStat = ProcGetPlatformChar @dbID
    IF @retStat!=0
    BEGIN
      RAISERROR 56029
      RETURN -999
    END
END
go
```

ProcGetPlatformChar**Procedure ProcGetPlatformChar**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetPlatformChar                      */
/*  Release         :  B0                                       */
/*  Module          :  DataOriginator                           */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for selecting */
/*                  data into DsMdPlatformCharacteristic  */
/*                  table                                   */
/*                                                          */
/*  Calls           :  None                                     */
/*                                                          */
/*==============================================================*/

CREATE PROC  ProcGetPlatformChar(@platformId ID = 0)
AS
BEGIN
  IF (@platformId = 0)
    BEGIN
      RAISERROR 40011
      RETURN -999
    END

  SELECT
            PlatformCharacteristicName,
        PlatformCharacteristicDesc,
        PlatformCharacteristicUnit,
        PlatformCharacteristicDataType,
        PlatformCharacteristicValue
  FROM   DsMdPlatformCharacteristic
  WHERE  platformId = @platformId

    IF @@error!=0
    BEGIN
          RAISERROR 45011
          RETURN -999
    END
END
go
```

ProcGetPlatformID**Procedure ProcGetPlatformID**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetPlatformID                          */
/*  Release       :  B0                                         */
/*  Module        :  Data Originator                            */
/*  Created on    :  04/22/97                                   */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                the id for a platform.                        */
/*  Calls         :                                             */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcGetPlatformID(@PlatformShortName   varchar(20) = NULL)
AS
BEGIN
  IF (@PlatformShortName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40099
    RETURN -999
  END
  ELSE
  BEGIN
    SELECT dbID = CONVERT(varchar(30), dbID)
    FROM   DsMdPlatform
    WHERE  PlatformShortName = @PlatformShortName

    IF (@@error != 0)
    BEGIN
      RAISERROR 45117
      RETURN -999
    END

  END
END
go
```

ProcGetPlatInstCode**Procedure ProcGetPlatInstCode**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetPlatInstCode                      */
/*  Release         :  B0                                       */
/*  Module          :  Spatial                                  */
/*  Created on      :  3/26/97                                  */
/*  Purpose         :  Create Stored Procedure for selecting */
/*                  data from DsMdPlatInstrCode          */
/*                  table                                */
/*                                                       */
/*  Calls           :  None                                     */
/*                                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetPlatInstCode(@platformInstrumentCode int = 0)
AS
BEGIN
  IF (@platformInstrumentCode = 0)
    BEGIN
      RAISERROR 40097
      RETURN -999
    END
    SELECT  platformShortName,
          instrumentName,
          resolution
```

```
   FROM    DsMdPlatInstrCode
   WHERE   platformInstrumentCode = @platformInstrumentCode

   IF @@error!=0
   BEGIN
         RAISERROR 45115
         RETURN -999
   END
END
go
```

ProcGetProcessingHistory**Procedure ProcGetProcessingHistory**

Code

```
/*================================================================*/
/*  Name          :  ProcGetProcessingHistory                     */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  05/08/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                   processing history data for Granules.        */
/*  Calls         :  ProcGetSDSRVLookup                           */
/*                   ProcGetFileStorage                           */
/*================================================================*/

CREATE PROCEDURE ProcGetProcessingHistory(@dbID ID = 0)
AS
BEGIN

  DECLARE @retStat  int
  DECLARE @lookupValue varchar(40)
  DECLARE @subType varchar(30)

  IF (@dbID = 0)
    BEGIN
      RAISERROR 40100
      RETURN -999
    END

  EXECUTE @retStat = ProcGetSDSRVLookup "subType", "PH", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetProcessingHistory error - could not find parameter
@subType."
      RETURN -999
    END

  SELECT @subType = @lookupValue

   SELECT subType = CONVERT(varchar, @subType),
         dbID = CONVERT(varchar(30), dbID),
            CONVERT(varchar(60), lastUpdate, 109),
```

```
            ProductionHistoryPointer
  FROM   DsMdProcessingHistory
  WHERE  dbID = @dbID

  IF @@error!=0 or @@rowcount != 1
  BEGIN
    RAISERROR 45118
    RETURN -999
  END

  SELECT        ShortName,
                VersionID = CONVERT(int, VersionID)
  FROM                  DsMdCollections
  WHERE         type = @subType

  IF @@error!=0 or @@rowcount != 1
  BEGIN
    RAISERROR 45119
    RETURN -999
  END

  EXECUTE @retStat = ProcGetFileStorage @dbID

  IF @retStat != 0
  BEGIN
    RAISERROR 56030
    RETURN -999
  END

END
go
```

ProcGetProcessingQA**Procedure ProcGetProcessingQA**

Code

```
/*================================================================*/
/*  Name          :  ProcGetProcessingQA                          */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                processing QA data for Granules.                */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetProcessingQA(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40015
      RETURN -999
    END
```

```
  SELECT ProcessingQADescription,
         ProcessingQAAttribute
  FROM   DsMdProcessingQA
  WHERE  granuleId = @granuleId
  IF @@error!=0
  BEGIN
    RAISERROR 45015
    RETURN -999
  END
END
go
```

ProcGetProdSpecType**Procedure ProcGetProdSpecType**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetProdSpecType                        */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  7/21/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   the datatypes for an attribute name.       */
/*  Calls         :  ProcGetSDSRVLookup                         */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetProdSpecType
         (@AdditionalAttributeName  varchar(30))
AS
BEGIN

   DECLARE @attributeId                varchar(20)
   DECLARE @AdditionalAttributeDataType  varchar(10)
   DECLARE @grTableName                varchar(40)
   DECLARE @clTableName                varchar(40)
   DECLARE @grColumnName               varchar(40)
   DECLARE @clColumnName               varchar(40)
   DECLARE @retStat                    int
   DECLARE @lookupValue                varchar(255)


  SELECT @attributeId               = convert(varchar(20),attributeId),
       @AdditionalAttributeDataType = AdditionalAttributeDataType
  FROM   DsMdAdditionalAttributes
  WHERE  AdditionalAttributeName = @AdditionalAttributeName
  IF (@@error != 0 OR @@rowcount != 1)
  BEGIN
    RAISERROR 45120
    RETURN -999
  END

/* find the tables used to store this datatype */
  SELECT @grTableName = grTableName,
```

```
        @clTableName = clTableName
   FROM   DsMdParmToTableVector
   WHERE  DataType = @AdditionalAttributeDataType
   IF (@@error != 0 OR @@rowcount != 1)
   BEGIN
     RAISERROR 45121
     RETURN -999
   END

/* set the column name used to store values of this datatype;     */
/* for now just set column names directly, need to add the column */
/* names to the DsMdParmToTableVector in the future               */

-- get the grColumnName
    EXECUTE @retStat = ProcGetSDSRVLookup "grColumnName", @grTableName, "varchar",
@lookupValue OUTPUT

    IF @retStat != 0
       BEGIN
         RAISERROR 20014 "Procedure ProcGetProdSpecType error - could not find parameter
@grColumnName."
         RETURN -999
       END

    SELECT @grColumnName =  @lookupValue

-- get the clColumnName
    EXECUTE @retStat = ProcGetSDSRVLookup "clColumnName", @clTableName, "varchar",
@lookupValue OUTPUT

    IF @retStat != 0
       BEGIN
         RAISERROR 20014 "Procedure ProcGetProdSpecType error - could not find parameter
@clColumnName."
         RETURN -999
       END

   SELECT @clColumnName =  @lookupValue

/* return the following to the wrapper layer code */

   SELECT @attributeId, @AdditionalAttributeDataType, @grColumnName, @clColumnName

END
go
```

ProcGetProductDbXref**Procedure ProcGetProductDbXref**

Code

```
/*================================================================*/
/*  Name          :  ProcGetProductDbXref                         */
/*  Release       :  B0                                           */
/*  Module        :  System Tables                                */
/*  Created on    :  3/24/97                                      */
/*  Purpose       :  selects from DsMdProductDbXref               */
/*                                                                */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcGetProductDbXref
AS
SELECT
        productName,
        dbName,
        startDate
        endDate
FROM
        DsMdProductDbXref

IF @@error!=0
  BEGIN
        RAISERROR 45122
        RETURN -999
  END
go
```

ProcGetQaGranule**Procedure ProcGetQaGranule**

Code

```
/*================================================================*/
/*  Name          :  ProcGetQaGranule                             */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  04/17/97                                     */
/*  Purpose       :  Create stored procedure to retrieve          */
/*                QA granule data associated with Granules.       */
/*  Calls         :                                               */
/*================================================================*/

CREATE PROCEDURE ProcGetQaGranule(@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40017
      RETURN -999
    END
```

```
  SELECT qaGranuleId = CONVERT(varchar(30), dbID)
  FROM   DsMdQaGranule
  WHERE  granuleId = @granuleId

  IF @@error!=0
  BEGIN
    RAISERROR 45017
    RETURN -999
  END

END
go
```

ProcGetQualityTextComment**Procedure ProcGetQualityTextComment**

Code

```
/*=================================================================*/
/*  Name          :  ProcGetQualityTextComment                     */
/*  Release       :  B0                                            */
/*  Module        :  Collections                                   */
/*  Created on    :  3/25/97                                       */
/*  Purpose       :  Create stored procedure to retrieve           */
/*                   quality text comment data for Collections.    */
/*  Calls         :                                                */
/*=================================================================*/

CREATE PROCEDURE ProcGetQualityTextComment(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40102
      RETURN -999
    END
  SELECT collectionId,
         QualityTextCommentPointer
  FROM   DsMdQualityTextComment
  WHERE  collectionId = @collectionId
  IF @@error!=0
  BEGIN
    RAISERROR 45123
    RETURN -999
  END
END
go
```

ProcGetRangeDateTime**Procedure ProcGetRangeDateTime**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetRangeDateTime                     */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                     data from DsMdRangeDateTime              */
/*                     table                                    */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetRangeDateTime(@collectionId ID = 0)
AS
BEGIN

   IF  (@collectionId = 0)
   BEGIN
      RAISERROR 40103
      RETURN -999
   END

   SELECT RangeEndingTime,
         RangeEndingDate,
         RangeBeginningTime,
         RangeBeginningDate
   FROM   DsMdRangeDateTime
   WHERE  collectionId = @collectionId

   IF (@@error!= 0)
   BEGIN
          RAISERROR 45124
          RETURN -999
   END
END
go
```

ProcGetRegPeriodic**Procedure ProcGetRegPeriodic**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetRegPeriodic                       */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                  data from DsMdRegularPeriodic               */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetRegPeriodic(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40104
      RETURN -999
    END
    SELECT PeriodName,
          Period1stDate,
          Period1stTime,
          PeriodDurationUnit,
          PeriodDurationValue,
          PeriodCycleDurationUnit,
          PeriodCycleDurationValue
    FROM   DsMdRegularPeriodic
    WHERE  collectionId = @collectionId

    IF (@@error!= 0)
    BEGIN
          RAISERROR 45125
          RETURN -999
    END
END
go
```

ProcGetReqDomain**Procedure ProcGetReqDomain**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetReqDomain                           */
/*  Release       :  B0                                         */
/*  Module        :  System Information                         */
/*  Created on    :  11/04/1997                                 */
/*  Purpose       :  Create stored procedure to get ReqDomain */
/*                information                                   */
/*  Calls         :                                             */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcGetReqDomain (
   @requestID    char(100) = null)
AS
BEGIN

   IF (@requestID = null)
     BEGIN
       RAISERROR 40399
       /*  "Procedure ProcGetReqDomain expects parameter @requestID, which was not supplied."  */
       RETURN -999
     END

   SELECT domainEntry
   FROM   DsSiReqDomain
   WHERE  requestID = @requestID

   IF @@error!=0
     BEGIN
       RAISERROR 45280
           /*  "ProcGetReqDomain: Failed select on DsSiReqDomain"  */
       RETURN -999
     END

END
go
```

ProcGetRequest**Procedure ProcGetRequest**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetRequest                             */
/*  Release       :  B0                                 */
/*  Module        :  System Information                         */
/*  Created on    :  11/04/1997                                 */
/*  Purpose       :  Create stored procedure to get Request  */
/*                   and Domain information                     */
/*  Calls         :  ProcGetReqDomain                           */
/*==============================================================*/


CREATE PROCEDURE ProcGetRequest (
   @requestID    char(100) = null)
AS
BEGIN
    DECLARE @retStat  int

   IF (@requestID = null)
     BEGIN
       RAISERROR 40340
           /*  "Procedure ProcGetRequest expects parameter @requestID, which was not supplied."  */
       RETURN -999
     END
```

```
    SELECT requestID,
           userID,
           convert(int, priority),
           createTime,
           clientCDSEntry,
           convert(int, status),
           sessionID,
           convert(int, asyncFlag)
    FROM   DsSiRequest
    WHERE  requestID = @requestID

    IF @@error!=0 or @@rowcount != 1
     BEGIN
       RAISERROR 45281
            /*  "ProcGetRequest: Failed select on DsSiRequest"  */
       RETURN -999
     END

    EXECUTE @retStat = ProcGetReqDomain @requestID

    IF @retStat != 0
      BEGIN
        RAISERROR 56099
            /*  "ProcGetRequest: Failed execution of ProcGetReqDomain"  */
        RETURN -999
      END

END
go
```

ProcGetSDSRVLookup**Procedure ProcGetSDSRVLookup**

Code

```
/*==============================================================*/
/*  Name           :  ProcGetSDSRVLookup                       */
/*  Release        :  B0                                       */
/*  Module         :  DB Wrappers System Tables                */
/*  Created on     :  12/21/97                                 */
/*  Purpose        :                                           */
/* selects:  1  if row logically deleted already               */
/*                                                             */
/*  Calls          :  ProcGetSDSRVLookup                       */
/*==============================================================*/

CREATE PROC ProcGetSDSRVLookup
(
   @lookupType          varchar(40) = "",
   @lookupValue         varchar(40) = "",
   @lookupValueType     char(2) = "",
   @lookupString        varchar(255)   OUTPUT
)
```

```
AS
BEGIN

DECLARE @errorMsg varchar(255)

IF (@lookupType = "")
  BEGIN
    RAISERROR 20014 "Procedure ProcGetSDSRVLookup expects parameter @lookupType, which was
not supplied."
    RETURN -999
  END

IF (@lookupValue = "")
  BEGIN
    RAISERROR 20014 "Procedure ProcGetSDSRVLookup expects parameter @lookupValue, which was
not supplied."
    RETURN -999
  END

SELECT @lookupString = NULL

-- Strings only supported now
-- update later to allow more data types
-- IF (@lookupValueType = "")
--    BEGIN
--      RAISERROR 20014 "Procedure ProcGetSDSRVLookup expects parameter @lookupValue, which
was not supplied."
--      RETURN -999
--    END

-- IF @lookupValueType = "varchar"
  BEGIN
    SELECT @lookupString = lookupString
      FROM
           DsMdSDSRVLookup
      WHERE lookupType = @lookupType
        AND lookupValue = @lookupValue
  END

IF ( @@rowcount != 1 or @lookupString IS NULL )
  BEGIN
    SELECT @errorMsg =  "Procedure ProcGetSDSRVLookup error - could not find row for lookupType =
" + CONVERT (varchar, @lookupType) +  " lookupValue = " + CONVERT (varchar, @lookupValue)
    RAISERROR 20014 @errorMsg
    RETURN -999
  END



RETURN

END
go
```

ProcGetSensor**Procedure ProcGetSensor**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetSensor                              */
/*  Release       :  B0                                         */
/*  Module        :  Data Originator                            */
/*  Created on    :  4/03/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                sensor data.                                  */
/*  Calls         :  ProcGetSensorCharacteristic                */
/*==============================================================*/

CREATE PROCEDURE ProcGetSensor(@dbID  ID = 0)
AS
BEGIN
  IF (@dbID = 0)
    BEGIN
      RAISERROR 40106
      RETURN -999
    END

    SELECT
          SensorTechnique,
          SensorShortName,
          SensorLongName
    FROM   DsMdSensor
    WHERE  dbID = @dbID
    IF @@error!=0
    BEGIN
      RAISERROR 45127
      RETURN -999
    END

    DECLARE @retStat  int
    EXECUTE @retStat = ProcGetSensorCharacteristic @dbID
    IF @retStat!=0
    BEGIN
      RAISERROR 56033
      RETURN -999
    END
END
go
```

ProcGetSensorCharacteristic**Procedure ProcGetSensorCharacteristic**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetSensorCharacteristic            */
/*  Release       :  B0                                     */
/*  Module        :  Data Originator                        */
/*  Created on    :  4/03/97                                */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                   sensor characteristic data.            */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcGetSensorCharacteristic(@sensorId ID = 0)
AS
BEGIN
  IF (@sensorId = 0)
    BEGIN
      RAISERROR 40012
      RETURN -999
    END
    SELECT
          SensorCharacteristicName,
          SensorCharacteristicDesc,
          SensorCharacteristicUnit,
          SensorCharacteristicDataType,
          SensorCharacteristicValue
    FROM   DsMdSensorCharacteristic
    WHERE  sensorId = @sensorId
    IF @@error!=0
      BEGIN
           RAISERROR 45012
                    RETURN -999
      END
END
go
```

ProcGetSensorID**Procedure ProcGetSensorID**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetSensorID                        */
/*  Release       :  B0                                     */
/*  Module        :  Data Originator                        */
/*  Created on    :  04/22/97                               */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                   the id for a sensor.                   */
/*  Calls         :                                         */
/*==============================================================*/

CREATE PROCEDURE ProcGetSensorID(@instrumentId     ID = 0,
                         @SensorShortName  varchar(20))
AS
```

```
BEGIN
  IF (@instrumentId = 0)
    BEGIN
      RAISERROR 40107
      RETURN -999
    END
   SELECT dbID = CONVERT(varchar(30), dbID)
   FROM   DsMdSensor
   WHERE  instrumentId = @instrumentId AND
          SensorShortName = @SensorShortName

   IF (@@error != 0)
   BEGIN
     RAISERROR 45128
     RETURN -999
   END

END
go
```

ProcGetSingleDateTime**Procedure ProcGetSingleDateTime**

Code

```
/*================================================================*/
/*  Name            :  ProcGetSingleDateTime                      */
/*  Release         :  B0                                         */
/*  Module          :  Temporal                                   */
/*  Created on      :  3/30/97                                    */
/*  Purpose         :  Create Stored Procedure for retrieving     */
/*                     data from DsMdSingleDateTime               */
/*                     table                                      */
/*                                                                */
/*  Calls           :  None                                       */
/*                                                                */
/*================================================================*/

CREATE PROCEDURE ProcGetSingleDateTime(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40108
      RETURN -999
    END
    SELECT TimeOfDay,
           CalendarDate
    FROM   DsMdSingleDateTime
    WHERE  collectionId = @collectionId
    AND MultipleDateName IS null

    IF (@@error!= 0)
    BEGIN
           RAISERROR 45129
```

```
        RETURN -999
    END
END
go
```

ProcGetSingleDateTimeInMultPd**Procedure ProcGetSingleDateTimeInMultPd**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetSingleDateTimeInMultPd        */
/*  Release         :  B0                                   */
/*  Module          :  Temporal                             */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for retrieving*/
/*                  data from DsMdSingleDateTime            */
/*                  andMdMultipleDateTimePeriod             */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/

CREATE PROCEDURE ProcGetSingleDateTimeInMultPd(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40010
      RETURN -999
    END

    SELECT s.TimeOfDay,
          s.CalendarDate
    FROM   DsMdSingleDateTime s
    WHERE s.collectionId = @collectionId
      AND s.MultipleDateName IS NOT null

    IF (@@error!= 0)
    BEGIN
          RAISERROR 45010
          RETURN -999
    END
END
go
```

ProcGetSpatial**Procedure ProcGetSpatial**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetSpatial                           */
/*  Release         :  B0                                       */
/*  Module          :  Spatial                                  */
/*  Created on      :  3/25/97                                  */
/*  Purpose         :  Create Stored Procedure to retrieve      */
/*                  :  data from DsMdSpatial                    */
/*                  :  table                                    */
/*  Calls           :  ProcGetPlanCoor                          */
/*                     ProcGetCollVertSpatialDomn               */
/*                     ProcGetCollZoneIdentifier                */
/*==============================================================*/

CREATE PROCEDURE ProcGetSpatial(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40109
      RETURN -999
    END
   DECLARE @SpatialCoverageType          char(10)
   DECLARE @AltitudeDatumName            varchar(40)
   DECLARE @AltitudeDistanceUnits        varchar(20)
   DECLARE @AltitudeEncodingMethod       varchar(200)
   DECLARE @AltitudeResolution           float
   DECLARE @DepthDatumName               varchar(80)
   DECLARE @DepthDistanceUnits           varchar(20)
   DECLARE @DepthEncodingMethod          varchar(200)
   DECLARE @DepthResolution              float
   DECLARE @HorizontalDatumName          varchar(30)
   DECLARE @EllipsoidName                varchar(200)
   DECLARE @SemiMajorAxis                float
   DECLARE @DenominatorOfFlatteningRatio  float
   DECLARE @LatitudeResolution           float
   DECLARE @LongitudeResolution          float
   DECLARE @GeographicCoordinateUnits    varchar(80)
   DECLARE @LocalCoordSysDesc            varchar(200)
   DECLARE @LocalGeorefInfo              varchar(200)

   SELECT @SpatialCoverageType = t1.SpatialCoverageType,
        @AltitudeDatumName = t1.AltitudeDatumName,
        @AltitudeDistanceUnits = t1.AltitudeDistanceUnits,
        @AltitudeEncodingMethod = t1.AltitudeEncodingMethod,
        @DepthDatumName = t1.DepthDatumName,
        @DepthDistanceUnits = t1.DepthDistanceUnits,
        @DepthEncodingMethod = t1.DepthEncodingMethod,
        @HorizontalDatumName = t1.HorizontalDatumName,
        @EllipsoidName = t1.EllipsoidName,
```

```
      @SemiMajorAxis = t1.SemiMajorAxis,
      @DenominatorOfFlatteningRatio = t1.DenominatorOfFlatteningRatio,
      @LatitudeResolution = t1.LatitudeResolution,
      @LongitudeResolution = t1.LongitudeResolution,
      @GeographicCoordinateUnits = t1.GeographicCoordinateUnits,
      @LocalCoordSysDesc = t1.LocalCoordinateSystemDesc,
      @LocalGeorefInfo = t1.LocalGeoreferenceInformation
 FROM   DsMdSpatial t1
 WHERE  t1.collectionId = @collectionId

 IF (@@error!= 0)
 BEGIN
   RAISERROR 45130
   RETURN -999
 END

 SELECT @DepthResolution = DepthResolution
 FROM   DsMdDepthResolution
 WHERE  collectionId = @collectionId

 IF (@@error!= 0)
 BEGIN
   RAISERROR 45131
   RETURN -999
 END


 SELECT @AltitudeResolution = AltitudeResolution
 FROM   DsMdAltitudeResolution
 WHERE  collectionId = @collectionId

 IF (@@error!= 0)
 BEGIN
   RAISERROR 45132
   RETURN -999
 END


 SELECT SpatialCoverageType = @SpatialCoverageType

IF (@AltitudeDatumName != NULL AND  @AltitudeDistanceUnits != NULL AND
    @AltitudeEncodingMethod != NULL AND @AltitudeResolution != NULL  )
BEGIN
 SELECT AltitudeDatumName      = @AltitudeDatumName,
       AltitudeDistanceUnits  = @AltitudeDistanceUnits,
       AltitudeEncodingMethod = @AltitudeEncodingMethod,
       AltitudeResolution     = @AltitudeResolution
END
ELSE
BEGIN
 SELECT AltitudeDatumName      = @AltitudeDatumName,
       AltitudeDistanceUnits  = @AltitudeDistanceUnits,
       AltitudeEncodingMethod = @AltitudeEncodingMethod,
       AltitudeResolution     = @AltitudeResolution
```

```
 WHERE 1 = 2
END

IF (@DepthDatumName != NULL AND @DepthDistanceUnits != NULL AND
   @DepthEncodingMethod != NULL AND @DepthResolution!= NULL )
BEGIN
 SELECT DepthDatumName      = @DepthDatumName,
        DepthDistanceUnits  = @DepthDistanceUnits,
        DepthEncodingMethod = @DepthEncodingMethod,
        DepthResolution     = @DepthResolution
END
ELSE
BEGIN
 SELECT DepthDatumName      = @DepthDatumName,
        DepthDistanceUnits  = @DepthDistanceUnits,
        DepthEncodingMethod = @DepthEncodingMethod,
        DepthResolution     = @DepthResolution
 WHERE 1 = 2
END

IF (@HorizontalDatumName != NULL AND @EllipsoidName != NULL AND
   @SemiMajorAxis != NULL AND @DenominatorOfFlatteningRatio != NULL)
BEGIN
 SELECT HorizontalDatumName = @HorizontalDatumName,
        EllipsoidName = @EllipsoidName,
        SemiMajorAxis = @SemiMajorAxis,
        DenominatorOfFlatteningRatio = @DenominatorOfFlatteningRatio
END
ELSE
BEGIN
 SELECT HorizontalDatumName = @HorizontalDatumName,
        EllipsoidName = @EllipsoidName,
        SemiMajorAxis = @SemiMajorAxis,
        DenominatorOfFlatteningRatio = @DenominatorOfFlatteningRatio
 WHERE 1 = 2
END

IF (@LatitudeResolution != NULL AND @LongitudeResolution != NULL AND
   @GeographicCoordinateUnits != NULL)
BEGIN
 SELECT LatitudeResolution = @LatitudeResolution,
        LongitudeResolution = @LongitudeResolution,
        GeographicCoordinateUnits = @GeographicCoordinateUnits
END
ELSE
BEGIN
 SELECT LatitudeResolution = @LatitudeResolution,
        LongitudeResolution = @LongitudeResolution,
        GeographicCoordinateUnits = @GeographicCoordinateUnits
 WHERE 1 = 2
END

 DECLARE @retStat  int
 EXECUTE @retStat = ProcGetPlanCoor @collectionId
```

```
  IF @retStat != 0
  BEGIN
    RAISERROR 56034
    RETURN -999
  END

  SELECT LocalCoordinateSystemDesc = @LocalCoordSysDesc,
       LocalGeoreferenceInformation = @LocalGeorefInfo

  EXECUTE @retStat = ProcGetCollVertSpatialDomn @collectionId
  IF @retStat != 0
  BEGIN
    RAISERROR 56035
    RETURN -999
  END

 EXECUTE @retStat = ProcGetCollZoneIdentifier @collectionId
  IF @retStat != 0
  BEGIN
    RAISERROR 45133
    RETURN -999
  END
END
go
```

ProcGetSpatialKeyword**Procedure ProcGetSpatialKeyword**

Code

```
/*====================================================================*/
/*  Name          :  ProcGetSpatialKeyword                    */
/*  Release       :  B0                                  */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                 */
/*  Purpose       :  Create stored procedure to retrieve       */
/*                spatial keyword data for Collections.     */
/*  Calls         :                                      */
/*====================================================================*/

CREATE PROCEDURE ProcGetSpatialKeyword(@collectionId ID = 0)
AS
BEGIN
 IF (@collectionId = 0)
   BEGIN
    RAISERROR 40003
    RETURN -999
   END
  SELECT SpatialKeyword
  FROM            DsMdSpatialKeyword
  WHERE  collectionId = @collectionId
  IF @@error!=0
  BEGIN
    RAISERROR 45003
    RETURN -999
```

```
  END
END
go
```

ProcGetSSAPComponent**Procedure ProcGetSSAPComponent**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetSSAPComponent                       */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to Delivered,      */
/*                   Algorithm Package, and associate data      */
/*                   tables.                                    */
/*  Calls         :  ProcGetSDSRVLookup                         */
/*                   ProcGetSSAPComponentAPVersion              */
/*                   ProcGetFileStorage                         */
/*==============================================================*/

CREATE PROCEDURE ProcGetSSAPComponent(@dbID  ID = 0)
AS
BEGIN

  DECLARE @retStat int
  DECLARE @lookupValue varchar(40)
  DECLARE @subType varchar(30)
  DECLARE @type varchar(30)

  IF (@dbID = 0)
    BEGIN
      RAISERROR 40105
      RETURN -999
    END

-- note that 2nd argument is only a lookup value
  EXECUTE @retStat = ProcGetSDSRVLookup "type", "AC", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetSSAPComponent error - could not find parameter @type."
      RETURN -999
    END

  SELECT @type = @lookupValue

  EXECUTE @retStat = ProcGetSDSRVLookup "subType", "SSAPC", "varchar", @lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcGetSSAPComponent error - could not find parameter
@subType."
      RETURN -999
```

```
   END

  SELECT @subType = @lookupValue

  SELECT   type = CONVERT(varchar, @type),
           subType = CONVERT(varchar, @subType),
           dbID = CONVERT(varchar(30), dbID),
           lastUpdate = CONVERT(varchar(60), lastUpdate, 109),
           ComponentType,
           ComponentName,
           SSAPAlgorithmPackageName,
           SSAPInsertDate
    FROM   DsMdSSAPComponent
    WHERE  dbID = @dbID

   IF @@error!=0 or @@rowcount != 1
   BEGIN
     RAISERROR 45126
     RETURN -999
   END

   EXECUTE @retStat = ProcGetSSAPComponentAPVersion @dbID

   IF @retStat != 0
   BEGIN
      RAISERROR 32000 "ProcGetSSAPComponent: Failed select on DsMdSSAPComponentAPVersion"
      RETURN -999
   END

   EXECUTE @retStat = ProcGetFileStorage @dbID

   IF @retStat != 0
   BEGIN
      RAISERROR 32000 "ProcGetSSAPComponent: Failed select on DsMdFileStorage"
      RETURN -999
   END

END
go
```

Procedure ProcGetSSAPComponentAPVersion

Code

```
/*================================================================*/
/*  Name          :  ProcGetSSAPComponentAPVersion          */
/*  Release       :  B0                                     */
/*  Module        :  Delivered Algorithm Package            */
/*  Created on    :  4/1/97                                 */
/*  Purpose       :  Create stored procedure to Delivered,  */
/*                   Algorithm Package, and associate data  */
/*                   tables.                                */
/*  Calls         :  None                                   */
/*================================================================*/
```

```
CREATE PROCEDURE ProcGetSSAPComponentAPVersion(@ssapId   ID = 0)
AS
BEGIN
  IF (@ssapId = 0)
    BEGIN
      RAISERROR 40019
      RETURN -999
    END
    SELECT SSAPAlgPackageVersion
    FROM DsMdSSAPComponentAPVersion
    WHERE ssapId = @ssapId

    IF @@error!=0
    BEGIN
          RAISERROR 45019
          RETURN -999
    END
END
go
```

ProcGetTelephoneNumbers**Procedure ProcGetTelephoneNumbers**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetContactPerson                       */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  Create stored procedure to Contact,        */
/*                   and associate data tables.                 */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetTelephoneNumbers(@contactId ID = 0)
AS
BEGIN
  IF (@contactId = 0)
    BEGIN
      RAISERROR 40006
      RETURN -999
    END
    SELECT TelephoneNumber,
          TelephoneNumberType
    FROM   DsMdTelephoneNumbers
    WHERE  contactId = @contactId

    IF (@@error!= 0)
    BEGIN
          RAISERROR 45006
          RETURN -999
    END
```

```
END
go
```

ProcGetTemporal**Procedure ProcGetTemporal**

Code

```
/*==============================================================*/
/*  Name            :  ProcGetTemporal                          */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for retrieving */
/*                  data from DsMdTemporal                      */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/

CREATE PROCEDURE ProcGetTemporal(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40110
      RETURN -999
    END
    SELECT TimeType,
          DateType,
          TemporalRangeType,
          PrecisionOfSeconds,
          EndsatPresentFlag
    FROM   DsMdTemporal
    WHERE  collectionId = @collectionId

    IF (@@error!= 0)
    BEGIN
           RAISERROR 45134
           RETURN -999
    END

   DECLARE @retStat  int

   EXECUTE @retStat = ProcGetRegPeriodic @collectionId
   IF (@retStat != 0)
   BEGIN
     RAISERROR 56036
     RETURN -999
   END

   EXECUTE @retStat = ProcGetSingleDateTime @collectionId
   IF (@retStat != 0)
   BEGIN
```

```
    RAISERROR 56037
    RETURN -999
  END

  EXECUTE @retStat =ProcGetRangeDateTime  @collectionId
  IF (@retStat != 0)
  BEGIN
    RAISERROR 56038
    RETURN -999
  END

END
go
```

ProcGetTemporalKeyword**Procedure ProcGetTemporalKeyword**

Code

```
/*==================================================================*/
/*  Name          :  ProcGetTemporalKeyword                         */
/*  Release       :  B0                                             */
/*  Module        :  Collections                                    */
/*  Created on    :  3/25/97                                        */
/*  Purpose       :  Create stored procedure to retrieve            */
/*                   temporal keyword data for Collections.         */
/*  Calls         :                                                 */
/*==================================================================*/

CREATE PROCEDURE ProcGetTemporalKeyword(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40004
      RETURN -999
    END
  SELECT TemporalKeyword
  FROM   DsMdTemporalKeyword
  WHERE  collectionId = @collectionId
  IF @@error!=0
  BEGIN
    RAISERROR 45004
    RETURN -999
  END
END
go
```

ProcGetUniqueID**Procedure ProcGetUniqueID**

Code

```
/*================================================================*/
/*  Name          :  ProcGetUniqueID                              */
/*  Release       :  B0                                           */
/*  Module        :  System Tables                                */
/*  Created on    :  3/24/97                                      */
/*  Purpose       :  update from DsMdIdentifier, add 1            */
/*                   selects from DsMdIdentifier                  */
/*                                                                */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcGetUniqueID
(
 @identifierType           char(30)="ID",
 @identifierObjectType     char(30)="MDWrappers"
)
AS
BEGIN TRANSACTION ID_UPDATE

-- hardcoded "if" to run fast
          IF @identifierType = "ID"
      BEGIN
            UPDATE       DsMdIdentifier
            SET   lastIdentifier = lastIdentifier + 1
            WHERE identifierType = @identifierType
          AND identifierObjectType = @identifierObjectType
          END
      ELSE
        IF @identifierType = "int"
            BEGIN
              UPDATE     DsMdIdentifier
              SET        lastIntIdentifier = lastIntIdentifier + 1
              WHERE identifierType = @identifierType
            AND identifierObjectType = @identifierObjectType
            END
      ELSE
        IF @identifierType = "smallint"
            BEGIN
              UPDATE     DsMdIdentifier
              SET        lastSmallIntIdentifier = lastSmallIntIdentifier + 1
              WHERE identifierType = @identifierType
            AND identifierObjectType = @identifierObjectType
            END
          ELSE
          BEGIN
                RAISERROR 32500 "ProcGetUniqueID: invalid identifierType value"
                ROLLBACK
                RETURN -999
          END
```

```
        IF @identifierType = "ID"
    BEGIN
        SELECT      lastIdentifier
        FROM        DsMdIdentifier
        WHERE identifierType = @identifierType
      AND identifierObjectType = @identifierObjectType
    END
    ELSE
      IF @identifierType = "int"
        BEGIN
          SELECT    lastIntIdentifier
          FROM      DsMdIdentifier
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
        END
    ELSE
      IF @identifierType = "smallint"
        BEGIN
          SELECT    lastSmallIntIdentifier
          FROM      DsMdIdentifier
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
        END


COMMIT ID_UPDATE
go
```

ProcGetUniqueIDOutput**Procedure ProcGetUniqueIDOutput**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetUniqueIDOutput                      */
/*  Release       :  B0                                         */
/*  Module        :  System Tables                              */
/*  Created on    :  1/16/98                                    */
/*  Purpose       :  update from DsMdIdentifier, add 1          */
/*                   selects from DsMdIdentifier into           */
/*                   OUTPUT variable                            */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetUniqueIDOutput
(
 @identifierType          char(30)="ID",
 @identifierObjectType    char(30)="MDWrappers",
 @outputID                ID  OUTPUT
)
AS
BEGIN TRANSACTION ID_UPDATE_OUTPUT
```

```
-- hardcoded "if" to run fast
        IF @identifierType = "ID"
    BEGIN
        UPDATE      DsMdIdentifier
        SET   lastIdentifier = lastIdentifier + 1
        WHERE identifierType = @identifierType
      AND identifierObjectType = @identifierObjectType
      END
    ELSE
      IF @identifierType = "int"
        BEGIN
          UPDATE    DsMdIdentifier
          SET       lastIntIdentifier = lastIntIdentifier + 1
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
        END
    ELSE
      IF @identifierType = "smallint"
        BEGIN
          UPDATE    DsMdIdentifier
          SET       lastSmallIntIdentifier = lastSmallIntIdentifier + 1
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
        END
      ELSE
      BEGIN
            RAISERROR 32500 "ProcGetUniqueIDOutput: invalid identifierType value"
            ROLLBACK
            RETURN -999
      END


      IF @@error!=0
            BEGIN
                  RAISERROR 32500 "ProcGetUniqueIDOutput: Failed update on DsMdIdentifier"
                  ROLLBACK
                  RETURN -999
            END

      IF @identifierType = "ID"
    BEGIN
        SELECT      @outputID = lastIdentifier
        FROM        DsMdIdentifier
        WHERE identifierType = @identifierType
      AND identifierObjectType = @identifierObjectType
    END
    ELSE
      IF @identifierType = "int"
        BEGIN
          SELECT    @outputID = lastIntIdentifier
          FROM      DsMdIdentifier
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
        END
```

```
    ELSE
      IF @identifierType = "smallint"
          BEGIN
            SELECT      @outputID = lastSmallIntIdentifier
            FROM        DsMdIdentifier
            WHERE identifierType = @identifierType
          AND identifierObjectType = @identifierObjectType
          END

        IF @@error!=0
        BEGIN
          RAISERROR 33000 "ProcGetUniqueIDOutput: Failed select on DsMdIdentifier"
          ROLLBACK
          RETURN -999
        END

COMMIT ID_UPDATE_OUTPUT
go
```

ProcGetUniqueIDString**Procedure ProcGetUniqueIDString**

Code

```
/*================================================================*/
/*  Name          :  ProcGetUniqueIDString                */
/*  Release       :  B0                                   */
/*  Module        :  System Tables                        */
/*  Created on    :  3/24/97                              */
/*  Purpose       :  update from DsMdIdentifier, add 1    */
/*                   selects from DsMdIdentifier          */
/*                                                        */
/*  Calls         :  None                                 */
/*================================================================*/

CREATE PROCEDURE ProcGetUniqueIDString
(
 @identifierType          char(30)="ID",
 @identifierObjectType    char(30)="MDWrappers"
)
AS
BEGIN TRANSACTION ID_UPDATE

        IF @identifierType = "ID"
    BEGIN
          UPDATE        DsMdIdentifier
          SET   lastIdentifier = lastIdentifier + 1
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
        END
    ELSE
      IF @identifierType = "int"
          BEGIN
            UPDATE      DsMdIdentifier
            SET         lastIntIdentifier = lastIntIdentifier + 1
```

```
            WHERE identifierType = @identifierType
          AND identifierObjectType = @identifierObjectType
          END
      ELSE
        IF @identifierType = "smallint"
          BEGIN
            UPDATE     DsMdIdentifier
            SET        lastSmallIntIdentifier = lastSmallIntIdentifier + 1
            WHERE identifierType = @identifierType
          AND identifierObjectType = @identifierObjectType
          END

        ELSE
        BEGIN
                RAISERROR 32500 "ProcGetUniqueIDString: invalid identifierType value"
                ROLLBACK
                RETURN -999
        END


        IF @identifierType = "ID"
      BEGIN
          SELECT       CONVERT(varchar(30), lastIdentifier)
          FROM         DsMdIdentifier
          WHERE identifierType = @identifierType
        AND identifierObjectType = @identifierObjectType
      END
      ELSE
        IF @identifierType = "int"
          BEGIN
            SELECT     CONVERT(varchar(30), lastIntIdentifier)
            FROM       DsMdIdentifier
            WHERE identifierType = @identifierType
          AND identifierObjectType = @identifierObjectType
          END
      ELSE
        IF @identifierType = "smallint"
          BEGIN
            SELECT     CONVERT(varchar(30), lastSmallIntIdentifier)
            FROM       DsMdIdentifier
            WHERE identifierType = @identifierType
          AND identifierObjectType = @identifierObjectType
          END

        IF @@error!=0
      BEGIN
                RAISERROR 32500 "ProcGetUniqueIDString: failed select on DsMdIdentifier"
                ROLLBACK
                RETURN -999
      END

COMMIT ID_UPDATE
go
```

ProcGetUniqueIDStringDDM**Procedure ProcGetUniqueIDStringDDM**

Code

```
CREATE PROCEDURE ProcGetUniqueIDStringDDM (@idstring ID output)
AS
BEGIN TRANSACTION ID_UPDATE

      UPDATE  DsMdIdentifier
      SET     lastIdentifier = lastIdentifier + 1
      IF @@error!=0
            BEGIN
                  RAISERROR 32500 "ProcGetUniqueIDString: Failed update on DsMdIdentifier"
                  ROLLBACK
                  RETURN -999
            END

      SELECT  @idstring=lastIdentifier
      FROM    DsMdIdentifier

      SELECT  convert(varchar(30), lastIdentifier)
      FROM    DsMdIdentifier
      IF @@error!=0
            BEGIN
                  RAISERROR 33000 "ProcGetUniqueIDString: Failed select on DsMdIdentifier"
                  ROLLBACK
                  RETURN -999
            END
COMMIT ID_UPDATE
go
```

ProcGetUserCommentDocument**Procedure ProcGetUserCommentDocument**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetUserCommentDocument             */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to retrieve    */
/*                   user comment document data for         */
/*                   Collections.                           */
/*  Calls         :                                         */
/*==============================================================*/

CREATE PROCEDURE ProcGetUserCommentDocument(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40111
      RETURN -999
    END
```

```
  SELECT collectionId,
       UserCommentDocumentPointer
  FROM   DsMdUserCommentDocument
  WHERE  collectionId = @collectionId
  IF @@error!=0
  BEGIN
    RAISERROR 45137
    RETURN -999
  END
END
go
```

ProcGetValidationDocument**Procedure ProcGetValidationDocument**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetValidationDocument                  */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to retrieve        */
/*                   validation document data for Collections. */
/*  Calls         :                                             */
/*==============================================================*/

CREATE PROCEDURE ProcGetValidationDocument(@collectionId ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
    BEGIN
      RAISERROR 40112
      RETURN -999
    END
  SELECT collectionId,
       ValidationDocumentPointer
  FROM   DsMdValidationDocument
  WHERE  collectionId = @collectionId
  IF @@error!=0
  BEGIN
    RAISERROR 45138
    RETURN -999
  END
END
go
```

ProcGetXAR**Procedure ProcGetXAR**

Code

```
/*==============================================================*/
/*  Name          :  ProcGetXAR                                 */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  5/07/97                                    */
/*  Purpose       :  Create stored procedure to get             */
/*                XAR granule order data.                       */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcGetXAR
            (@granuleId  ID = 0)
AS
BEGIN
  IF (@granuleId = 0)
    BEGIN
      RAISERROR 40113
      RETURN -999
    END

    SELECT XARId = convert(varchar(30), XARId),
         XARType
      FROM DsMdXAR
     WHERE granuleId = @granuleId

   IF @@error!=0
   BEGIN
     RAISERROR 45139
     RETURN -999
   END


END
go
```

ProcHdDelSSAPCompAPVersion**Procedure ProcHdDelSSAPCompAPVersion**

Code

```
/*==============================================================*/
/*  Name          :  ProcHdDelSSAPCompAPVersion                 */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  hard delete DsMdSSAPCompAPVersion          */
/*                        DsMdAPSAPComponentXref                */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHdDelSSAPCompAPVersion (@ssapId int = 0)
AS
BEGIN
	IF  (@ssapId = 0)
	    BEGIN
		RAISERROR 40266
		RETURN -999
	    END
	    BEGIN
		DELETE
			DsMdAPSAPComponentXref
		WHERE ssapId = @ssapId

		IF @@error != 0
		BEGIN
			RAISERROR 52044
			RETURN -999
		END

		DELETE
			DsMdSSAPComponentAPVersion
		WHERE ssapId = @ssapId

		IF @@error != 0
		BEGIN
			RAISERROR 52045
			RETURN -999
		END

	    END

	    RETURN
END
go
```

ProcHDeleteAlgorithmPackage**Procedure ProcHDeleteAlgorithmPackage**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteAlgorithmPackage          */
/*  Release       :  B0                                */
/*  Module        :  Algorithm Package                    */
/*  Created on    :  6/13/97                           */
/*  Purpose       :  unconditional delete                 */
/*                DsMdAlgorithmPackage                 */
/*                                                     */
/*                                                     */
/*  Calls         :  None                              */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteAlgorithmPackage (@dbID int = 0)
AS
```

```
BEGIN
      IF  (@dbID = 0)
          BEGIN
                    RAISERROR 40267
                    RETURN -999
          END

          BEGIN
                    DELETE
                              DsMdAlgorithmPackage
                    WHERE dbID = @dbID

                    IF @@error != 0
                    BEGIN
                              RAISERROR 52046
                              RETURN -999
                    END

          END

          RETURN
END
go
```

ProcHDeleteAltitudeResolution**Procedure ProcHDeleteAltitudeResolution**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteAltitudeResolution          */
/*  Release       :  B0                                     */
/*  Module        :  Spatial                                */
/*  Created on    :  6/26/97                                */
/*  Purpose       :  Delete contents of                     */
/*                DsMdAltitudeResolution                    */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteAltitudeResolution
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
          BEGIN
        /* Mimic Sybase error. */
            RAISERROR 40294
            RETURN -999
          END
    ELSE
          BEGIN
            DELETE DsMdAltitudeResolution
            WHERE collectionId = @collectionId
```

```
            IF (@@error != 0)
              BEGIN
                    RAISERROR 52073
                    RETURN -999
              END
           END
    RETURN
END
go
```

ProcHDeleteAnalysisSource**Procedure ProcHDeleteAnalysisSource**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteAnalysisSource              */
/*  Release       :  B0                                  */
/*  Module        :  Data Originator                        */
/*  Created on    :  6/13/97                             */
/*  Purpose       :  unconditional delete                   */
/*                DsMdAnalysisSource                      */
/*                                                  */
/*                                                  */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteAnalysisSource (@dbID int = 0)
AS
BEGIN
      IF  (@dbID = 0)
          BEGIN
                    RAISERROR 40287
                    RETURN -999
          END

          BEGIN
                    DELETE
                              DsMdAnalysisSource
                    WHERE dbID = @dbID

                    IF @@error != 0
                    BEGIN
                              RAISERROR 52066
                              RETURN -999
                    END

          END

          RETURN
END
go
```

ProcHDeleteAncillaryInpGranule**Procedure ProcHDeleteAncillaryInpGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteAncillaryInpGranule         */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/22/97                                */
/*  Purpose       :  Delete contents of                     */
/*                DsMdAncillaryInputGranule                 */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteAncillaryInpGranule
                    (@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
    BEGIN
      /* Mimic Sybase error. */
        RAISERROR 40252
        RETURN -999
    END
  ELSE
    BEGIN
        DELETE DsMdAncillaryInputGranule
        WHERE dbID = @granuleId

        IF @@error != 0
          BEGIN
                RAISERROR 52030
                RETURN -999
          END
        END
END
go
```

ProcHDeleteAncillaryInput**Procedure ProcHDeleteAncillaryInput**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteAncillaryInput              */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdAncillaryInput                        */
/*  Calls         :  None                                */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteAncillaryInput
            (@granuleId  ID = 0)
AS
BEGIN


  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40224
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdAncillaryInput
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52002
       RETURN -999
    END

END
go
```

ProcHDeleteBrBrowseGranuleXref**Procedure ProcHDeleteBrBrowseGranuleXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteBrBrowseGranuleXref          */
/*  Release       :  B0                                      */
/*  Module        :  Granules                                */
/*  Created on    :  8/01/97                                 */
/*  Purpose       :  Delete contents of table                */
/*                   DsMdBrowseGranuleXref using browseId    */
/*  Calls         :  None                                    */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteBrBrowseGranuleXref
            (@browseId  ID = 0)
AS
BEGIN

  IF (@browseId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40223
     RETURN -999
   END

    DECLARE @Message varchar(100)
```

```
   DELETE DsMdBrowseGranuleXref
     WHERE browseId = @browseId

   IF @@error!=0
   BEGIN
      RAISERROR 52001
      RETURN -999
   END

END
go
```

ProcHDeleteBrowse**Procedure ProcHDeleteBrowse**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteBrowse                          */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  Delete contents of table                   */
/*                   DsMdBrowse                                 */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteBrowse
            (@dbID  ID = 0)
AS
BEGIN

  IF (@dbID = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40225
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdBrowse
      WHERE dbID = @dbID

    IF @@error!=0
    BEGIN
       RAISERROR 52003
       RETURN -999
    END

END
go
```

ProcHDeleteBrowseGranuleXref**Procedure ProcHDeleteBrowseGranuleXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteBrowseGranuleXref          */
/*  Release       :  B0                                    */
/*  Module        :  Granules                                 */
/*  Created on    :  6/13/97                                  */
/*  Purpose       :  Delete contents of table              */
/*                   DsMdBrowseGranuleXref                   */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteBrowseGranuleXref
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40226
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdBrowseGranuleXref
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52004
       RETURN -999
    END

END
go
```

ProcHDeleteCampaign**Procedure ProcHDeleteCampaign**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteCampaign                   */
/*  Release       :  B0                                    */
/*  Module        :  Data Originator                          */
/*  Created on    :  6/13/97                                  */
/*  Purpose       :  unconditional delete                     */
/*                   DsMdCampaign                            */
/*                                                  */
/*                                                  */
/*  Calls         :  None                                  */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteCampaign (@dbID int = 0)
AS
BEGIN
      IF  (@dbID = 0)
          BEGIN
                    RAISERROR 40268
                    RETURN -999
          END

          BEGIN
                    DELETE
                              DsMdCampaign
                    WHERE dbID = @dbID

                    IF @@error != 0
                    BEGIN
                              RAISERROR 52047
                              RETURN -999
                    END

          END

          RETURN
END
go
```

ProcHDeleteCollectionAPXref**Procedure ProcHDeleteCollectionAPXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteCollectionAPXref            */
/*  Release       :  B0                                  */
/*  Module        :  Collection                          */
/*  Created on    :  6/13/97                             */
/*  Purpose       :  unconditional delete                    */
/*                DsMdCollectionAPXref                  */
/*                                                     */
/*                                                     */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteCollectionAPXref (@apId int = 0)
AS
BEGIN
      IF  (@apId = 0)
          BEGIN
                    RAISERROR 40269
                    RETURN -999
          END

          BEGIN
                    DELETE
```

```
                DsMdCollectionAPXref
            WHERE apId = @apId

            IF @@error != 0
            BEGIN
                RAISERROR 52048
                RETURN -999
            END

        END

        RETURN
END
go
```

ProcHDeleteCollGranuleXref**Procedure ProcHDeleteCollGranuleXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteCollGranuleXref             */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/22/97                                */
/*  Purpose       :  Delete contents of                     */
/*                DsMdCollectionGranuleXref                 */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteCollGranuleXref
            (@granuleId int = 0)
AS
BEGIN

   IF (@granuleId = 0)
        BEGIN
       /* Mimic Sybase error. */
         RAISERROR 40253
         RETURN -999
        END
    ELSE
        BEGIN
          DELETE DsMdCollectionGranuleXref
          WHERE granuleId = @granuleId

          IF (@@error != 0)
            BEGIN
                RAISERROR 50231
                RETURN -999
            END
        END
   RETURN
END
go
```

ProcHDeleteCollVertSpatlDomn**Procedure ProcHDeleteCollVertSpatlDomn**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteCollVertSpatlDomn          */
/*  Release       :  B0                                */
/*  Module        :  Collection                          */
/*  Created on    :  6/26/97                             */
/*  Purpose       :  Delete contents of                  */
/*                DsMdCollVerticalSpatialDomain          */
/*  Calls         :  None                              */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteCollVertSpatlDomn
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
       /* Mimic Sybase error. */
          RAISERROR 40290
          RETURN -999
         END
    ELSE
         BEGIN
          DELETE DsMdCollVerticalSpatialDomain
          WHERE collectionId = @collectionId

          IF (@@error != 0)
            BEGIN
                RAISERROR 52069
                RETURN -999
            END
         END
   RETURN
END
go
```

ProcHDeleteCollZoneIdentifier**Procedure ProcHDeleteCollZoneIdentifier**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteCollZoneIdentifier          */
/*  Release       :  B0                                */
/*  Module        :  Collection                          */
/*  Created on    :  6/26/97                             */
/*  Purpose       :  Delete contents of                  */
/*                DsMdCollZoneIdentifier                 */
/*  Calls         :  None                              */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteCollZoneIdentifier
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
        /* Mimic Sybase error. */
           RAISERROR 40291
           RETURN -999
         END
    ELSE
         BEGIN
          DELETE DsMdCollZoneIdentifier
          WHERE collectionId = @collectionId

          IF (@@error != 0)
            BEGIN
                  RAISERROR 52070
                  RETURN -999
            END
         END
   RETURN
END
go
```

ProcHDeleteContact**Procedure ProcHDeleteContact**

Code

```
/*=============================================================*/
/*  Name          :  ProcHDeleteContact                        */
/*  Release       :  B0                                        */
/*  Module        :  Contact                                   */
/*  Created on    :  6/13/97                                   */
/*  Purpose       :  unconditional delete                      */
/*                DsMdContact                                  */
/*                                                             */
/*                                                             */
/*  Calls         :  None                                      */
/*=============================================================*/

CREATE PROCEDURE ProcHDeleteContact (@dbID int = 0)
AS
BEGIN
      IF  (@dbID = 0)
         BEGIN
                  RAISERROR 40270
                  RETURN -999
         END

         BEGIN
                  DELETE
                           DsMdContact
```

```
                WHERE dbID = @dbID

                IF @@error != 0
                BEGIN
                        RAISERROR 52049
                        RETURN -999
                END

        END

        RETURN
END
go
```

ProcHDeleteContactAddress**Procedure ProcHDeleteContactAddress**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteContactAddress              */
/*  Release       :  B0                                  */
/*  Module        :  Contact                              */
/*  Created on    :  6/13/97                              */
/*  Purpose       :  unconditional delete                    */
/*                DsMdContactAddress                      */
/*                                                      */
/*                                                      */
/*  Calls         :  ProcHDeleteContact                   */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteContactAddress (@contactId int = 0)
AS
BEGIN
      IF  (@contactId = 0)
        BEGIN
                RAISERROR 40271
                RETURN -999
        END

        BEGIN
                DELETE
                        DsMdContactAddress
                WHERE contactId = @contactId

                IF @@error != 0
                BEGIN
                        RAISERROR 52050
                        RETURN -999
                END

                DECLARE @retStat  int
                EXECUTE @retStat = ProcHDeleteContact @contactId
                IF @retStat != 0
                BEGIN
```

```
                RAISERROR 56079
                RETURN -999
            END

        END

        RETURN
END
go
```

ProcHDeleteContactOrg**Procedure ProcHDeleteContactOrg**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteContactOrg                      */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  unconditional delete                       */
/*                DsMdContactOrganizations                      */
/*                                                              */
/*                                                              */
/*  Calls         :  ProcHDeleteContact                         */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteContactOrg (@contactId int = 0)
AS
BEGIN
    IF  (@contactId = 0)
        BEGIN
            RAISERROR 40288
            RETURN -999
        END

        BEGIN
            DELETE
                DsMdContactOrganizations
            WHERE contactId = @contactId

            IF @@error != 0
            BEGIN
                RAISERROR 52067
                RETURN -999
            END

          DECLARE @retStat  int
          EXECUTE @retStat = ProcHDeleteContact @contactId
          IF @retStat != 0
          BEGIN
              RAISERROR 56082
              RETURN -999
          END
```

```
        END

        RETURN
END
go
```

ProcHDeleteContactPersons**Procedure ProcHDeleteContactPersons**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteContactPersons              */
/*  Release       :  B0                                  */
/*  Module        :  Contact                              */
/*  Created on    :  6/13/97                              */
/*  Purpose       :  unconditional delete                    */
/*                DsMdContactPersons                    */
/*                                                    */
/*                                                    */
/*  Calls         :  ProcHDeleteContact                    */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteContactPersons (@contactId int = 0)
AS
BEGIN
     IF  (@contactId = 0)
        BEGIN
                RAISERROR 40272
                RETURN -999
        END

        BEGIN
                DELETE
                        DsMdContactPersons
                WHERE contactId = @contactId

                IF @@error != 0
                BEGIN
                        RAISERROR 52051
                        RETURN -999
                END

          DECLARE @retStat  int
          EXECUTE @retStat = ProcHDeleteContact @contactId
          IF @retStat != 0
          BEGIN
               RAISERROR 56080
               RETURN -999
          END

        END

        RETURN
END
go
```

ProcHDeleteDAP**Procedure ProcHDeleteDAP**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteDAP                             */
/*  Release       :  B0                                         */
/*  Module        :  Algorithm Package                          */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  unconditional delete                       */
/*                DsMdDAP                                       */
/*                                                              */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteDAP (@dbID int = 0)
AS
BEGIN
      IF  (@dbID = 0)
          BEGIN
                RAISERROR 40273
                RETURN -999
          END

          BEGIN
                DELETE
                      DsMdDAP
                WHERE dbID = @dbID

                IF @@error != 0
                BEGIN
                      RAISERROR 52052
                      RETURN -999
                END

          END

          RETURN
END
go
```

ProcHDeleteDepthResolution**Procedure ProcHDeleteDepthResolution**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteDepthResolution                 */
/*  Release       :  B0                                         */
/*  Module        :  Spatial                                    */
/*  Created on    :  6/26/97                                    */
/*  Purpose       :  Delete contents of                         */
/*                DsMdDepthResolution                           */
/*  Calls         :  None                                       */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteDepthResolution
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
        /* Mimic Sybase error. */
           RAISERROR 40295
           RETURN -999
         END
    ELSE
         BEGIN
           DELETE DsMdDepthResolution
           WHERE collectionId = @collectionId

           IF (@@error != 0)
             BEGIN
                  RAISERROR 52074
                  RETURN -999
             END
         END
   RETURN
END
go
```

ProcHDeleteDisciplineKeywords**Procedure ProcHDeleteDisciplineKeywords**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteDisciplineKeywords          */
/*  Release       :  B0                                  */
/*  Module        :  Collections                            */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  unconditional delete                       */
/*                DsMdDisciplineKeywords                */
/*                                                      */
/*                                                      */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteDisciplineKeywords (@dbID int = 0)
AS
BEGIN
      IF  (@dbID = 0)
         BEGIN
                  RAISERROR 40274
                  RETURN -999
         END

         BEGIN
                  DELETE
```

```
                DsMdDisciplineKeywords
            WHERE dbID = @dbID

            IF @@error != 0
            BEGIN
                RAISERROR 52053
                RETURN -999
            END

        END

        RETURN
END
go
```

ProcHDeleteEmailAddress**Procedure ProcHDeleteEmailAddress**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteEmailAddress                    */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  unconditional delete                       */
/*                DsMdEmailAddress                              */
/*                                                              */
/*                                                              */
/*  Calls         :  ProcHDeleteContact                         */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteEmailAddress (@contactId int = 0)
AS
BEGIN
    IF  (@contactId = 0)
        BEGIN
            RAISERROR 40275
            RETURN -999
        END

        BEGIN
            DELETE
                DsMdEmailAddress
            WHERE contactId = @contactId

            IF @@error != 0
            BEGIN
                RAISERROR 52054
                RETURN -999
            END

          DECLARE @retStat  int
          EXECUTE @retStat = ProcHDeleteContact @contactId
          IF @retStat != 0
```

```
            BEGIN
                RAISERROR 56081
                RETURN -999
            END

        END

        RETURN
END
go
```

ProcHDeleteESDTConfiguredType**Procedure ProcHDeleteESDTConfiguredType**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteESDTConfiguredType          */
/*  Release       :  B0                                     */
/*  Module        :                                         */
/*  Created on    :  8/21/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsGeESDTConfiguredType using esdtID     */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteESDTConfiguredType
(
   @configuredName     varchar(20) = NULL,
   @versionID          varchar(20) = NULL
)
AS
BEGIN

   DECLARE @tempID tinyint

   IF (@configuredName = NULL)
      BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40302
         RETURN -999
      END

   IF (@versionID is NULL)
      BEGIN
         /* Mimic Sybase error. */
         RAISERROR 20023 "Procedure ProcHDeleteESDTConfiguredType expects parameter
@versionID, which was not supplied."
         RETURN -999
      END

   SELECT @tempID = convert(tinyint, @versionID)

   DELETE DsGeESDTConfiguredType
   WHERE configuredName = @configuredName
```

```
   AND versionID = @tempID

   IF @@error!=0
   BEGIN
      RAISERROR 52080
      RETURN -999
   END
END
go
```

ProcHDeleteFileStorage**Procedure ProcHDeleteFileStorage**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteFileStorage                       */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  6/13/97                                      */
/*  Purpose       :  Delete contents of table                     */
/*                DsMdFileStorage                                 */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteFileStorage
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40227
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdFileStorage
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52005
       RETURN -999
    END

END
go
```

ProcHDeleteGrAncillaryInptXref**Procedure ProcHDeleteGrAncillaryInptXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGrAncillaryInptXref         */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                 */
/*  Created on    :  6/22/97                                  */
/*  Purpose       :  Delete contents of                     */
/*                DsMdGranulesAncillaryInputXref          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGrAncillaryInptXref
            (@granuleId  ID = 0)
AS
BEGIN
   IF (@granuleId = 0)
      BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40257
         RETURN -999
      END
   ELSE
      BEGIN
            DECLARE @Message varchar(100)

            DELETE DsMdGranulesAncillaryInputXref
            WHERE granuleId = @granuleId

            IF @@error!=0
            BEGIN
               RAISERROR 52035
               RETURN -999
            END
         END
END
go
```

ProcHDeleteGranOrbitParmXref**Procedure ProcHDeleteGranOrbitParmXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGranOrbitParmXref           */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                 */
/*  Created on    :  6/13/97                                  */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGranulesOrbitParmXref               */
/*  Calls         :  None                                   */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteGranOrbitParmXref
              (@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40237
     RETURN -999
   END

   DECLARE @Message varchar(100)

   DELETE DsMdGranulesOrbitParmXref
     WHERE granuleId = @granuleId

   IF @@error!=0
    BEGIN
       RAISERROR 52015
       RETURN -999
    END

END
go
```

ProcHDeleteGranuleAnalsisXref**Procedure ProcHDeleteGranuleAnalsisXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGranuleAnalsisXref          */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGranuleAnalysisXref                 */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGranuleAnalsisXref
              (@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40228
     RETURN -999
   END
```

```
   DECLARE @Message varchar(100)

   DELETE DsMdGranuleAnalysisXref
     WHERE granuleId = @granuleId

   IF @@error!=0
   BEGIN
      RAISERROR 52006
      RETURN -999
   END

END
go
```

ProcHDeleteGranuleCampgnXref**Procedure ProcHDeleteGranuleCampgnXref**

Code

```
/*================================================================*/
/*  Name           :  ProcHDeleteGranuleCampgnXref            */
/*  Release        :  B0                                  */
/*  Module         :  Granules                              */
/*  Created on     :  6/13/97                               */
/*  Purpose        :  Delete contents of table              */
/*                 DsMdGranuleCampaignXref                  */
/*  Calls          :  None                                */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteGranuleCampgnXref
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40229
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGranuleCampaignXref
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52007
       RETURN -999
    END

END
go
```

ProcHDeleteGranuleInstrXref**Procedure ProcHDeleteGranuleInstrXref**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteGranuleInstrXref            */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGranuleInstrumentXref                 */
/*  Calls         :  None                                */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteGranuleInstrXref
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40230
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGranuleInstrumentXref
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52008
       RETURN -999
    END

END
go
```

ProcHDeleteGranuleLocality**Procedure ProcHDeleteGranuleLocality**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteGranuleLocality             */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGranuleLocality                     */
/*  Calls         :  None                                */
/*================================================================*/
```

```
CREATE PROCEDURE ProcHDeleteGranuleLocality
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40231
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGranuleLocality
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52009
       RETURN -999
    END

END
go
```

ProcHDeleteGranulePlatfrmXref**Procedure ProcHDeleteGranulePlatfrmXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGranulePlatfrmXref          */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGranulePlatformXref                   */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGranulePlatfrmXref
            (@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40232
     RETURN -999
   END

    DECLARE @Message varchar(100)
```

```
   DELETE DsMdGranulePlatformXref
     WHERE granuleId = @granuleId

   IF @@error!=0
   BEGIN
      RAISERROR 52010
      RETURN -999
   END

END
go
```

ProcHDeleteGranuleReview**Procedure ProcHDeleteGranuleReview**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGranuleReview              */
/*  Release       :  B0                                    */
/*  Module        :  Granules                              */
/*  Created on    :  6/13/97                               */
/*  Purpose       :  Delete contents of table              */
/*                   DsMdGranuleReview                     */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGranuleReview
              (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40233
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGranuleReview
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52011
       RETURN -999
    END

END
go
```

ProcHDeleteGranules**Procedure ProcHDeleteGranules**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteGranules                          */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  6/13/97                                      */
/*  Purpose       :  Delete contents of table                     */
/*                DsMdGranules                                    */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteGranules
            (@dbID  ID = 0)
AS
BEGIN

  IF (@dbID = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40236
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGranules
      WHERE dbID = @dbID

    IF @@error!=0
    BEGIN
       RAISERROR 52014
       RETURN -999
    END

END
go
```

ProcHDeleteGranuleSensorXref**Procedure ProcHDeleteGranuleSensorXref**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteGranuleSensorXref                 */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  6/13/97                                      */
/*  Purpose       :  Delete contents of table                     */
/*                DsMdGranuleSensorXref                           */
/*  Calls         :  None                                         */
/*================================================================*/
```

```
CREATE PROCEDURE ProcHDeleteGranuleSensorXref
            (@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40234
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGranuleSensorXref
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52012
       RETURN -999
    END

END
go
```

ProcHDeleteGranuleVersions**Procedure ProcHDeleteGranuleVersions**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGranuleVersions             */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGranuleVersions                     */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGranuleVersions
            (@currentGranuleId  ID = 0)
AS
BEGIN

  IF (@currentGranuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40235
     RETURN -999
   END

    DECLARE @Message varchar(100)
```

```
   DELETE DsMdGranuleVersions
     WHERE currentGranuleId = @currentGranuleId

   IF @@error!=0
   BEGIN
      RAISERROR 52013
      RETURN -999
   END

END
go
```

ProcHDeleteGrFloatInfoContent**Procedure ProcHDeleteGrFloatInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGrFloatInfoContent          */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/22/97                                */
/*  Purpose       :  Delete Contents of                     */
/*                DsMdGrFloatInfoContent                    */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGrFloatInfoContent
            (@granuleId ID = 0)
AS
BEGIN
    IF (@granuleId = 0)
          BEGIN
        /* Mimic Sybase error. */
            RAISERROR 40254
            RETURN -999
          END
    ELSE
          BEGIN
            DELETE DsMdGrFloatInfoContent
            WHERE granuleId = @granuleId

            IF (@@error != 0)
              BEGIN
                   RAISERROR 52032
                   RETURN -999
              END
          END

    RETURN 0

END
go
```

ProcHDeleteGrIntInfoContent**Procedure ProcHDeleteGrIntInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGrIntInfoContent            */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/22/97                                */
/*  Purpose       :  Delete Contents of                     */
/*                DsMdGrIntegerInfoContent                  */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGrIntInfoContent
            (@granuleId ID = 0)
AS
BEGIN

   IF (@granuleId = 0)
         BEGIN
       /* Mimic Sybase error. */
           RAISERROR 40255
           RETURN -999
         END
   ELSE
         BEGIN
           DELETE DsMdGrIntegerInfoContent
           WHERE granuleId = @granuleId

           IF (@@error != 0)
             BEGIN
                 RAISERROR 52033
                 RETURN -999
           END
         END

    RETURN 0
END
go
```

ProcHDeleteGrSensorCharistics**Procedure ProcHDeleteGrSensorCharistics**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGrSensorCharistics          */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGrSensorCharacteristics               */
/*  Calls         :  None                                */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteGrSensorCharistics
        (@granuleId   ID = 0)

AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40238
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGrSensorCharacteristics
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52016

       RETURN -999
    END

END
go
```

ProcHDeleteGrStorageMedium**Procedure ProcHDeleteGrStorageMedium**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGrStorageMedium            */
/*  Release       :  B0                                */
/*  Module        :  Granules                             */
/*  Created on    :  6/13/97                              */
/*  Purpose       :  Delete contents of table             */
/*                DsMdGrStorageMedium                     */
/*  Calls         :  None                              */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGrStorageMedium
        (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40239
     RETURN -999
   END
```

```
   DECLARE @Message varchar(100)

   DELETE DsMdGrStorageMedium
     WHERE granuleId = @granuleId

   IF @@error!=0
   BEGIN
      RAISERROR 52017
      RETURN -999
   END

END
go
```

ProcHDeleteGrStringInfoContent**Procedure ProcHDeleteGrStringInfoContent**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteGrStringInfoContent          */
/*  Release       :  B0                                      */
/*  Module        :  Granules                                */
/*  Created on    :  6/22/97                                 */
/*  Purpose       :  Delete Contents of                      */
/*                DsMdGrStringInfoContent                    */
/*  Calls         :  None                                    */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteGrStringInfoContent
             (@granuleId ID = 0)
AS
BEGIN
   IF (@granuleId = 0)
          BEGIN
        /* Mimic Sybase error. */
           RAISERROR 40256
           RETURN -999
          END
   ELSE
          BEGIN
            DELETE DsMdGrStringInfoContent
            WHERE granuleId = @granuleId

            IF (@@error != 0)
              BEGIN
                    RAISERROR 52034
                    RETURN -999
              END
          END

    RETURN 0

END
go
```

ProcHDeleteGrVertSpatlDomn**Procedure ProcHDeleteGrVertSpatlDomn**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteGrVertSpatlDomn             */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdGrVerticalSpatialDomain             */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteGrVertSpatlDomn
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40240
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdGrVerticalSpatialDomain
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52018
       RETURN -999
    END

END
go
```

ProcHDeleteInputGranule**Procedure ProcHDeleteInputGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteInputGranule                */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdInputGranule                        */
/*  Calls         :  None                                */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteInputGranule
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40241
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdInputGranule
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52019
       RETURN -999
    END

END
go
```

ProcHDeleteInstrument**Procedure ProcHDeleteInstrument**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteInstrument                      */
/*  Release       :  B0                                         */
/*  Module        :  Data Originator                            */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  unconditional delete                       */
/*                DsMdInstrument                                */
/*                                                              */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteInstrument (@dbID int = 0)
AS
BEGIN
      IF  (@dbID = 0)
          BEGIN
                RAISERROR 40276
                RETURN -999
          END

          BEGIN
                DELETE
                      DsMdInstrument
```

```
                WHERE dbID = @dbID

                IF @@error != 0
                BEGIN
                        RAISERROR 52055
                        RETURN -999
                END

        END

        RETURN
END
go
```

ProcHDeleteInstrumentChar**Procedure ProcHDeleteInstrumentChar**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteInstrumentChar              */
/*  Release       :  B0                                  */
/*  Module        :  Data Originator                        */
/*  Created on    :  6/13/97                             */
/*  Purpose       :  unconditional delete                   */
/*                DsMdInstrumentCharacteristic           */
/*                                                       */
/*                                                       */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteInstrumentChar (@instrumentId int = 0)
AS
BEGIN
      IF  (@instrumentId = 0)
        BEGIN
                RAISERROR 40277
                RETURN -999
        END

        BEGIN
                DELETE
                        DsMdInstrumentCharacteristic
                WHERE instrumentId = @instrumentId

                IF @@error != 0
                BEGIN
                        RAISERROR 52056
                        RETURN -999
                END

        END

        RETURN
END
go
```

ProcHDeleteMeasuredParameter**Procedure ProcHDeleteMeasuredParameter**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteMeasuredParameter          */
/*  Release       :  B0                                    */
/*  Module        :  Granules                              */
/*  Created on    :  6/13/97                               */
/*  Purpose       :  Delete contents of table              */
/*                DsMdMeasuredParameter                    */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteMeasuredParameter
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40242
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdMeasuredParameter
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52020
       RETURN -999
    END

END
go
```

ProcHDeleteMultiDateTimePd**Procedure ProcHDeleteMultiDateTimePd**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteMultiDateTimePd            */
/*  Release       :  B0                                    */
/*  Module        :  Temporal                              */
/*  Created on    :  6/26/97                               */
/*  Purpose       :  Delete contents of                    */
/*                DsMdMultipleDateTimePeriod               */
/*  Calls         :  None                                  */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteMultiDateTimePd
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
        /* Mimic Sybase error. */
           RAISERROR 40298
           RETURN -999
         END
    ELSE
         BEGIN
           DELETE DsMdMultipleDateTimePeriod
           WHERE collectionId = @collectionId

           IF (@@error != 0)
             BEGIN
                  RAISERROR 52077
                  RETURN -999
             END
         END
   RETURN
END
go
```

ProcHDeleteOperationMode**Procedure ProcHDeleteOperationMode**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteOperationMode              */
/*  Release       :  B0                                 */
/*  Module        :  Data Originator                       */
/*  Created on    :  6/13/97                            */
/*  Purpose       :  unconditional delete                     */
/*                DsMdOperationMode                     */
/*                                                 */
/*                                                 */
/*  Calls         :  None                               */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteOperationMode (@instrumentId int = 0)
AS
BEGIN
      IF  (@instrumentId = 0)
         BEGIN
                  RAISERROR 40278
                  RETURN -999
         END

         BEGIN
                  DELETE
                        DsMdOperationMode
```

```
                WHERE instrumentId = @instrumentId

                IF @@error != 0
                BEGIN
                        RAISERROR 52057
                        RETURN -999
                END

        END

        RETURN
END
go
```

ProcHDeleteOrbitCalcSpatial**Procedure ProcHDeleteOrbitCalcSpatial**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteOrbitCalcSpatial            */
/*  Release       :  B0                                  */
/*  Module        :  Granules                              */
/*  Created on    :  6/13/97                               */
/*  Purpose       :  Delete contents of table              */
/*                DsMdOrbitCalculatedSpatial             */
/*  Calls         :  None                                */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteOrbitCalcSpatial
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40244
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdOrbitCalculatedSpatial
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52022
       RETURN -999
    END

END
go
```

ProcHDeleteOrbitCalcSpatlDomn**Procedure ProcHDeleteOrbitCalcSpatlDomn**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteOrbitCalcSpatlDomn          */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdOrbitCalcSpatialDomain              */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteOrbitCalcSpatlDomn
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40243
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdOrbitCalcSpatialDomain
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52021
       RETURN -999
    END

END
go
```

ProcHDeleteOrbitParamGranule**Procedure ProcHDeleteOrbitParamGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteOrbitParamGranule           */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  Delete contents of table               */
/*                DsMdOrbitParametersGranule              */
/*  Calls         :  None                                */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteOrbitParamGranule
              (@dbID  ID = 0)
AS
BEGIN

  IF (@dbID = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40245
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdOrbitParametersGranule
      WHERE dbID = @dbID

    IF @@error!=0
    BEGIN
       RAISERROR 52023
       RETURN -999
    END

END
go
```

ProcHDeleteOrbitPolygons**Procedure ProcHDeleteOrbitPolygons**

Code

```
/*==============================================================*/
/*  Name           :  ProcHDeleteOrbitPolygons                  */
/*  Release        :  B0                                        */
/*  Module         :  Granules                                  */
/*  Created on     :  6/13/97                                   */
/*  Purpose        :  Delete contents of table                  */
/*                 DsMdOrbitPolygons                           */
/*  Calls          :  None                                      */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteOrbitPolygons
              (@platInstCode  int = 0)
AS
BEGIN

  IF (@platInstCode = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40246
     RETURN -999
   END

    DECLARE @Message varchar(100)
```

```
   DELETE DsMdOrbitPolygons
     WHERE platInstCode = @platInstCode

   IF @@error!=0
   BEGIN
      RAISERROR 52024
      RETURN -999
   END

END
go
```

ProcHDeletePersonOrgXref**Procedure ProcHDeletePersonOrgXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeletePersonOrgXref              */
/*  Release       :  B0                                    */
/*  Module        :  Contact                               */
/*  Created on    :  6/13/97                               */
/*  Purpose       :  unconditional delete                  */
/*                DsMdPersonOrgXref                        */
/*                                                         */
/*                                                         */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcHDeletePersonOrgXref
          (@personId int = 0,
           @organizationId int = 0)
AS
BEGIN
      IF  (@personId = 0)
          BEGIN
                    RAISERROR 40280
                    RETURN -999
          END
      IF  (@organizationId = 0)
          BEGIN
                    RAISERROR 40280
                    RETURN -999
          END

          BEGIN
                    DELETE
                              DsMdPersonOrganizationXref
                    WHERE personId = @personId
                      AND organizationId = @organizationId

                    IF @@error != 0
                    BEGIN
                              RAISERROR 52059
                              RETURN -999
```

```
                END

        END

        RETURN
END
go
```

ProcHDeletePGEGroup**Procedure ProcHDeletePGEGroup**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeletePGEGroup                        */
/*  Release       :  B0                                         */
/*  Module        :  Algorithm Package                          */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  unconditional delete                       */
/*                DsMdPGEGroup                                  */
/*                                                              */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeletePGEGroup
                (@dapDBId int = 0)
AS
BEGIN
      IF  (@dapDBId = 0)
        BEGIN
                RAISERROR 40279
                RETURN -999
        END

        BEGIN
                DELETE          DsMdPGEGroup
                WHERE           dapDBId = @dapDBId

                IF @@error != 0
                BEGIN
                        RAISERROR 52058
                        RETURN -999
                END

        END

        RETURN
END
go
```

ProcHDeletePlanarCoordSys**Procedure ProcHDeletePlanarCoordSys**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeletePlanarCoordSys              */
/*  Release       :  B0                                     */
/*  Module        :  Spatial                                */
/*  Created on    :  6/26/97                                */
/*  Purpose       :  Delete contents of                     */
/*                DsMdPlanarCoordinateSystems               */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcHDeletePlanarCoordSys
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
        /* Mimic Sybase error. */
           RAISERROR 40292
           RETURN -999
         END
    ELSE
         BEGIN
           DELETE DsMdPlanarCoordinateSystems
           WHERE collectionId = @collectionId

           IF (@@error != 0)
             BEGIN
                  RAISERROR 52071
                  RETURN -999
             END
         END
   RETURN
END
go
```

ProcHDeletePlatform**Procedure ProcHDeletePlatform**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeletePlatform                    */
/*  Release       :  B0                                     */
/*  Module        :  Data Originator                        */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  unconditional delete                   */
/*                DsMdPlatform                              */
/*                                                          */
/*                                                          */
/*  Calls         :  None                                   */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeletePlatform (@dbID int = 0)
AS
BEGIN
	IF  (@dbID = 0)
	    BEGIN
		RAISERROR 40281
		RETURN -999
	    END

	    BEGIN
		DELETE
			DsMdPlatform
		WHERE dbID = @dbID

		IF @@error != 0
		BEGIN
			RAISERROR 52060
			RETURN -999
		END

	    END

	    RETURN
END
go
```

ProcHDeletePlatformChar**Procedure ProcHDeletePlatformChar**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeletePlatformChar              */
/*  Release       :  B0                                 */
/*  Module        :  Data Originator                      */
/*  Created on    :  6/13/97                            */
/*  Purpose       :  unconditional delete                 */
/*                DsMdPlatformChar                      */
/*                                                      */
/*                                                      */
/*  Calls         :  None                               */
/*==============================================================*/

CREATE PROCEDURE ProcHDeletePlatformChar (@platformId int = 0)
AS
BEGIN
	IF  (@platformId = 0)
	    BEGIN
		RAISERROR 40282
		RETURN -999
	    END

	    BEGIN
		DELETE
```

```
                DsMdPlatformCharacteristic
            WHERE platformId = @platformId

            IF @@error != 0
            BEGIN
                RAISERROR 52061
                RETURN -999
            END

        END

        RETURN
END
go
```

ProcHDeletePlatInstrCode**Procedure ProcHDeletePlatInstrCode**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeletePlatInstrCode              */
/*  Release       :  B0                                 */
/*  Module        :  Granules                             */
/*  Created on    :  6/13/97                              */
/*  Purpose       :  Delete contents of table             */
/*                DsMdPlatInstrCode                       */
/*  Calls         :  None                               */
/*================================================================*/

CREATE PROCEDURE ProcHDeletePlatInstrCode
            (@platformInstrumentCode  int = 0)
AS
BEGIN

  IF (@platformInstrumentCode = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40247
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdPlatInstrCode
      WHERE platformInstrumentCode = @platformInstrumentCode

    IF @@error!=0
    BEGIN
       RAISERROR 52025
       RETURN -999
    END

END
go
```

ProcHDeleteProcessingHistory**Procedure ProcHDeleteProcessingHistory**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteProcessingHistory          */
/*  Release       :  B0                                */
/*  Module        :  Granules                             */
/*  Created on    :  6/13/97                              */
/*  Purpose       :  Delete contents of table             */
/*                DsMdProcessingHistory                 */
/*  Calls         :  None                              */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteProcessingHistory
            (@dbID  ID = 0)
AS
BEGIN

  IF (@dbID = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40248
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdProcessingHistory
      WHERE dbID = @dbID

    IF @@error!=0
    BEGIN
       RAISERROR 52026
       RETURN -999
    END

END
go
```

ProcHDeleteProcessingQA**Procedure ProcHDeleteProcessingQA**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteProcessingQA               */
/*  Release       :  B0                                */
/*  Module        :  Granules                             */
/*  Created on    :  6/13/97                              */
/*  Purpose       :  Delete contents of table             */
/*                DsMdProcessingQA                      */
/*  Calls         :  None                              */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcHDeleteProcessingQA
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40249
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdProcessingQA
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52027
       RETURN -999
    END

END
go
```

ProcHDeleteProdSpecific**Procedure ProcHDeleteProdSpecific**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteProdSpecific                    */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  7/7/97                                     */
/*  Purpose       :  Create stored procedure to delete          */
/*                   product specific attributes and their      */
/*                   values for a given granuleId.              */
/*  Calls         :  ProcHDeleteGrFloatInfoContent              */
/*                   ProcHDeleteGrIntInfoContent                */
/*                   ProcHDeleteGrStringInfoContent             */
/*==============================================================*/
CREATE PROCEDURE ProcHDeleteProdSpecific
           (@granuleId ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40301
     RETURN -999
   END
```

```
  DECLARE @retStat  int

  EXECUTE @retStat = ProcHDeleteGrFloatInfoContent @granuleId
  IF (@retStat != 0)
    BEGIN
         RAISERROR 56084
         RETURN -999
    END

  EXECUTE @retStat = ProcHDeleteGrIntInfoContent @granuleId
  IF (@retStat != 0)
    BEGIN
      RAISERROR 56085
          RETURN -999
    END

  EXECUTE @retStat = ProcHDeleteGrStringInfoContent @granuleId
  IF (@retStat != 0)
    BEGIN
      RAISERROR 56086
          RETURN -999
    END

END
go
```

ProcHDeleteQaGranule**Procedure ProcHDeleteQaGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteQaGranule                       */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  Delete contents of table                   */
/*                DsMdQaGranule                                 */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteQaGranule
            (@granuleId  ID = 0)
AS
BEGIN


 IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40250
    RETURN -999
  END

   DECLARE @Message varchar(100)
```

```
   DELETE DsMdQaGranule
     WHERE granuleId = @granuleId

   IF @@error!=0
   BEGIN
      RAISERROR 52028
      RETURN -999
   END

END
go
```

ProcHDeleteRangeDateTime**Procedure ProcHDeleteRangeDateTime**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteRangeDateTime              */
/*  Release       :  B0                                 */
/*  Module        :  Temporal                              */
/*  Created on    :  6/26/97                               */
/*  Purpose       :  Delete contents of                    */
/*                DsMdRangeDateTime                        */
/*  Calls         :  None                               */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteRangeDateTime
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
       /* Mimic Sybase error. */
          RAISERROR 40299
          RETURN -999
         END
    ELSE
         BEGIN
           DELETE DsMdRangeDateTime
           WHERE collectionId = @collectionId

           IF (@@error != 0)
             BEGIN
                  RAISERROR 52078
                  RETURN -999
             END
         END
    RETURN
END
go
```

ProcHDeleteRegularPeriodic**Procedure ProcHDeleteRegularPeriodic**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteRegularPeriodic             */
/*  Release       :  B0                                  */
/*  Module        :  Temporal                               */
/*  Created on    :  6/26/97                                */
/*  Purpose       :  Delete contents of                     */
/*                DsMdRegularPeriodic                       */
/*  Calls         :  None                                */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteRegularPeriodic
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
        /* Mimic Sybase error. */
           RAISERROR 40297
           RETURN -999
         END
    ELSE
         BEGIN
           DELETE DsMdRegularPeriodic
           WHERE collectionId = @collectionId

           IF (@@error != 0)
             BEGIN
                  RAISERROR 52076
                  RETURN -999
             END
         END
   RETURN
END
go
```

ProcHDeleteSensor**Procedure ProcHDeleteSensor**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteSensor                      */
/*  Release       :  B0                                  */
/*  Module        :  Data Originator                        */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  unconditional delete                   */
/*                DsMdSensor                             */
/*                                                    */
/*                                                    */
/*  Calls         :  None                                */
/*================================================================*/
```

```
CREATE PROCEDURE ProcHDeleteSensor (@dbID int = 0)
AS
BEGIN
	IF  (@dbID = 0)
		BEGIN
			RAISERROR 40285
			RETURN -999
		END

		BEGIN
			DELETE
				DsMdSensor
			WHERE dbID = @dbID

			IF @@error != 0
			BEGIN
				RAISERROR 52064
				RETURN -999
			END

		END

		RETURN
END
go
```

ProcHDeleteSensorChar**Procedure ProcHDeleteSensorChar**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteSensorChar                      */
/*  Release       :  B0                                         */
/*  Module        :  Data Originator                            */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  unconditional delete                       */
/*                DsMdSensorCharacteristic                      */
/*                                                              */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteSensorChar (@sensorId int = 0)
AS
BEGIN
	IF  (@sensorId = 0)
		BEGIN
			RAISERROR 40286
			RETURN -999
		END

		BEGIN
			DELETE
```

```
                DsMdSensorCharacteristic
            WHERE sensorId = @sensorId

            IF @@error != 0
            BEGIN
                RAISERROR 52065
                RETURN -999
            END

        END

        RETURN
END
go
```

ProcHDeleteSingleDateTime**Procedure ProcHDeleteSingleDateTime**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteSingleDateTime              */
/*  Release       :  B0                                  */
/*  Module        :  Temporal                               */
/*  Created on    :  6/26/97                                */
/*  Purpose       :  Delete contents of                     */
/*                DsMdSingleDateTime                        */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteSingleDateTime
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
        BEGIN
       /* Mimic Sybase error. */
          RAISERROR 40300
          RETURN -999
        END
    ELSE
        BEGIN
          DELETE DsMdSingleDateTime
          WHERE collectionId = @collectionId

          IF (@@error != 0)
            BEGIN
                RAISERROR 52079
                RETURN -999
            END
        END
   RETURN
END
go
```

ProcHDeleteSpatial**Procedure ProcHDeleteSpatial**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteSpatial                           */
/*  Release       :  B0                                           */
/*  Module        :  Spatial                                      */
/*  Created on    :  6/26/97                                      */
/*  Purpose       :  Delete contents of                           */
/*                DsMdSpatial                                     */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROCEDURE ProcHDeleteSpatial
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
         BEGIN
       /* Mimic Sybase error. */
          RAISERROR 40293
          RETURN -999
         END
    ELSE
         BEGIN
          DELETE DsMdSpatial
          WHERE collectionId = @collectionId

          IF (@@error != 0)
            BEGIN
                RAISERROR 52072
                RETURN -999
            END
         END
    RETURN
END
go
```

ProcHDeleteSSAPComponent**Procedure ProcHDeleteSSAPComponent**

Code

```
/*================================================================*/
/*  Name          :  ProcHDeleteSSAPComponent                     */
/*  Release       :  B0                                           */
/*  Module        :  Algorithm Package                            */
/*  Created on    :  6/13/97                                      */
/*  Purpose       :  unconditional delete                         */
/*                DsMdSSAPComponent                               */
/*                                                                */
/*                                                                */
/*  Calls         :  None                                         */
/*================================================================*/
```

```
CREATE PROCEDURE ProcHDeleteSSAPComponent (@dbID int = 0)
AS
BEGIN
	IF  (@dbID = 0)
	    BEGIN
		RAISERROR 40283
		RETURN -999
	    END

	    BEGIN
		DELETE
			DsMdSSAPComponent
		WHERE dbID = @dbID

		IF @@error != 0
		BEGIN
			RAISERROR 52062
			RETURN -999
		END

	    END

	    RETURN
END
go
```

Procedure ProcHDeleteSSAPComponentAPVer

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteSSAPComponentAPVer          */
/*  Release       :  B0                                  */
/*  Module        :  Algorithm Package                   */
/*  Created on    :  6/13/97                             */
/*  Purpose       :  unconditional delete                */
/*                DsMdSSAPComponentAPVersion             */
/*                                                       */
/*                                                       */
/*  Calls         :  None                                */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteSSAPComponentAPVer (@ssapId int = 0)
AS
BEGIN
	IF  (@ssapId = 0)
	    BEGIN
		RAISERROR 40284
		RETURN -999
	    END

	    BEGIN
		DELETE
```

```
                    DsMdSSAPComponentAPVersion
                WHERE ssapId = @ssapId

                IF @@error != 0
                BEGIN
                        RAISERROR 52063
                        RETURN -999
                END

        END

        RETURN
END
go
```

ProcHDeleteTelephoneNumber**Procedure ProcHDeleteTelephoneNumber**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteTelephoneNumber              */
/*  Release       :  B0                                    */
/*  Module        :  Contact                                */
/*  Created on    :  6/13/97                                */
/*  Purpose       :  unconditional delete                    */
/*                DsMdTelephoneNumber                        */
/*                                                        */
/*                                                        */
/*  Calls         :  ProcHDeleteContact                      */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteTelephoneNumber (@contactId int = 0)
AS
BEGIN
      IF  (@contactId = 0)
        BEGIN
                RAISERROR 40289
                RETURN -999
        END

        BEGIN
                DELETE
                        DsMdTelephoneNumbers
                WHERE contactId = @contactId

                IF @@error != 0
                BEGIN
                        RAISERROR 52068
                        RETURN -999
                END

                DECLARE @retStat  int
          EXECUTE @retStat = ProcHDeleteContact @contactId
          IF @retStat != 0
```

```
            BEGIN
                  RAISERROR 56083
                  RETURN -999
            END

          END

          RETURN
END
go
```

ProcHDeleteTemporal**Procedure ProcHDeleteTemporal**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteTemporal                        */
/*  Release       :  B0                                         */
/*  Module        :  Temporal                                   */
/*  Created on    :  6/26/97                                    */
/*  Purpose       :  Delete contents of                         */
/*                DsMdTemporal                                  */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteTemporal
            (@collectionId int = 0)
AS
BEGIN

   IF (@collectionId = 0)
          BEGIN
        /* Mimic Sybase error. */
            RAISERROR 40296
            RETURN -999
          END
    ELSE
          BEGIN
            DELETE DsMdTemporal
            WHERE collectionId = @collectionId

            IF (@@error != 0)
              BEGIN
                    RAISERROR 52075
                    RETURN -999
              END
          END
   RETURN
END
go
```

ProcHDeleteXAR**Procedure ProcHDeleteXAR**

Code

```
/*==============================================================*/
/*  Name          :  ProcHDeleteXAR                             */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  6/13/97                                    */
/*  Purpose       :  Delete contents of table                   */
/*                DsMdXAR                                       */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcHDeleteXAR
            (@granuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40251
     RETURN -999
   END

    DECLARE @Message varchar(100)

    DELETE DsMdXAR
      WHERE granuleId = @granuleId

    IF @@error!=0
    BEGIN
       RAISERROR 52029
       RETURN -999
    END

END
go
```

ProcInsertAdditionalAttributes**Procedure ProcInsertAdditionalAttributes**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAdditionalAttributes             */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                additional attribute data for                 */
/*                Collections.                                  */
/*  Calls         :                                             */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertAdditionalAttributes
(
 @AdditionalAttributeName        varchar(40),
 @AdditionalAttributeDesc        varchar(255),
 @AdditionalAttributeDataType    char(10),
 @ParameterUnitsOfMeasure        varchar(20),
 @ParameterRangeBegin            varchar(40),
 @ParameterRangeEnd              varchar(40),
 @ParameterValueAccuracy         varchar(30),
 @ValueAccuracyExplanation       varchar(255),
 @MeasurementResolution          varchar(30),
 @attributeId                    int      OUTPUT
)
AS
BEGIN

IF NOT EXISTS (
                SELECT 1 FROM DsMdAdditionalAttributes
                WHERE AdditionalAttributeName = @AdditionalAttributeName
               )
BEGIN
 BEGIN TRANSACTION ATTR_INSERT

   DECLARE @newAttrId int
   DECLARE @retStat   int

   EXECUTE @retStat = ProcGetUniqueIDOutput "int", "DsMdAdditionalAttributes",  @newAttrId OUTPUT

   IF ( @retStat != 0  )
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertAdditionalAttributes error  on  call to
ProcGetUniqueIDOutput."
      RETURN -999
   END

   INSERT DsMdAdditionalAttributes
               (
                    attributeId,
                    AdditionalAttributeName,
            AdditionalAttributeDescription,
            AdditionalAttributeDataType,
            ParameterUnitsOfMeasure,
            ParameterRangeBegin,
                    ParameterRangeEnd,
                    ParameterValueAccuracy,
            ValueAccuracyExplanation,
            MeasurementResolution
                    )
   VALUES     (
                    @newAttrId,
                    @AdditionalAttributeName,
         @AdditionalAttributeDesc,
```

```
                @AdditionalAttributeDataType,
        @ParameterUnitsOfMeasure,
              @ParameterRangeBegin,
              @ParameterRangeEnd,
        @ParameterValueAccuracy,
        @ValueAccuracyExplanation,
        @MeasurementResolution
                )

 IF @@error!=0
  BEGIN
        RAISERROR 50032
        ROLLBACK
        RETURN -999
  END

SELECT @attributeId = @newAttrId

 COMMIT ATTR_INSERT
 END
END
go
```

ProcInsertAlgorithmPackage**Procedure ProcInsertAlgorithmPackage**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAlgorithmPackage          */
/*  Release       :  B0                                  */
/*  Module        :  Delivered Algorithm Package         */
/*  Created on    :  5/5/97                              */
/*  Purpose       :  Create stored procedure to insert data  */
/*                 into Algorithm Package                */
/*  Calls         :  none                                */
/*==============================================================*/


CREATE PROCEDURE ProcInsertAlgorithmPackage(
            @dbID                       ID = 0,
            @AlgorithmPackageName       varchar(80),
            @AlgorithmPackageVersion    varchar(20),
            @AlgorithmPackageMaturityCode   char(10),
            @AlgorithmPackageAcceptDate     datetime,
            @DeliveryPurpose            varchar(20),
            @PGEName                    varchar(20),
            @PGEVersion                 char(10),
            @PGEIdentifier              char(10),
            @PGEFunction                varchar(80),
            @PGEDateLastModified        datetime,
            @SWVersion                  varchar(12),
            @SWDateLastModified         datetime,
                  @ShortName                    varchar(8),
                  @VersionID                    tinyint = NULL)
AS
```

```
BEGIN
  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40142
    RETURN -999
  END

  IF (@VersionID = NULL)
  BEGIN
    RAISERROR 20023 "Procedure ProcInsertAPAssocColl expects parameter @VersionID of range 1 to
255."
    RETURN -999
  END

  INSERT  DsMdAlgorithmPackage (
              dbID,
              AlgorithmPackageName,
              AlgorithmPackageVersion,
              AlgorithmPackageMaturityCode,
              AlgorithmPackageAcceptDate,
              DeliveryPurpose,
              PGEName,
              PGEVersion,
              PGEIdentifier,
              PGEFunction,
              PGEDateLastModified,
              SWVersion,
              SWDateLastModified,
              lastUpdate,
              insertTime)
  VALUES( @dbID,
        @AlgorithmPackageName,
        @AlgorithmPackageVersion,
        @AlgorithmPackageMaturityCode,
        @AlgorithmPackageAcceptDate,
        @DeliveryPurpose,
        @PGEName,
        @PGEVersion,
        @PGEIdentifier,
        @PGEFunction,
        @PGEDateLastModified,
        @SWVersion,
        @SWDateLastModified,
        getdate(),
        getdate())

  IF @@error!=0
  BEGIN
    RAISERROR 50035
    RETURN -999
  END
END
go
```

ProcInsertAltitudeResolution**Procedure ProcInsertAltitudeResolution**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertAltitudeResolution         */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                              */
/*  Created on      :  3/25/97                              */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdAltitudeResolution        */
/*                  table                                   */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROCEDURE ProcInsertAltitudeResolution(
            @collectionId                   ID      = 0,
            @AltitudeResolution    float    = 0)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      RAISERROR 40136
      RETURN -999
   END

   IF (@AltitudeResolution = 0)
   BEGIN
      RAISERROR 40136
      RETURN -999
   END


   INSERT  DsMdAltitudeResolution (collectionId, AltitudeResolution)
   VALUES( @collectionId,
            @AltitudeResolution)

   IF @@error!=0
   BEGIN
         RAISERROR 50027
         RETURN -999
   END
END
go
```

ProcInsertAnalysisSource**Procedure ProcInsertAnalysisSource**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertAnalysisSource             */
/*  Release         :  B0                                */
/*  Module          :  DataOriginator                       */
/*  Created on      :  3/30/97                           */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdAnalysisSource         */
/*                  table                                */
/*  Calls           :  ProcInsertCollAnalysisXref           */
/*                                                       */
/*                                                       */
/*==============================================================*/


CREATE PROC ProcInsertAnalysisSource(
  @collectionId       ID              = 0,
  @AnalysisShortName  varchar(20)     = NULL,
  @AnalysisLongName   varchar(80)     = NULL,
  @AnalysisTechnique  varchar(80)     = NULL,
  @AnalysisType       varchar(20)     = NULL)
AS
BEGIN

   DECLARE @analysisSourceId ID
   DECLARE @retStat int

   IF (@collectionId = 0)
   BEGIN
      RAISERROR 40143
      RETURN -999
   END

   IF (@AnalysisShortName = NULL or @AnalysisShortName = "")
   BEGIN
      RAISERROR 40143
      RETURN -999
   END

   IF (@AnalysisType = NULL or @AnalysisType = "")
   BEGIN
      RAISERROR 40143
      RETURN -999
   END

   SELECT @analysisSourceId =
      dbID FROM DsMdAnalysisSource
          WHERE AnalysisShortName = @AnalysisShortName

 IF (@analysisSourceId IS NULL)
   BEGIN
```

```
   BEGIN TRANSACTION ID_RETURN_ANALYSISSOURCE

   EXECUTE @retStat = ProcGetUniqueIDOutput "ID", "DsMdAnalysisSource",  @analysisSourceId
OUTPUT

   IF ( @retStat != 0  )
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertAnalysisSource error  on  call to
ProcGetUniqueIDOutput."
      RETURN -999
   END
   COMMIT ID_RETURN_ANALYSISSOURCE

   INSERT  DsMdAnalysisSource(
         dbID,
         AnalysisShortName,
         AnalysisLongName,
         AnalysisTechnique,
         AnalysisType)
   VALUES(@analysisSourceId,
        @AnalysisShortName,
        @AnalysisLongName,
        @AnalysisTechnique,
        @AnalysisType)

   IF @@error!=0
   BEGIN
      RAISERROR 50036
      RETURN -999
   END


   EXECUTE @retStat = ProcInsertCollAnalysisXref @collectionId,  @analysisSourceId
   IF @retStat != 0
      BEGIN
        RAISERROR 56062
        RETURN -999
      END

  END
 ELSE
  BEGIN
   EXECUTE @retStat =  ProcInsertCollAnalysisXref @collectionId, @analysisSourceId
   IF @retStat != 0
      BEGIN
        RAISERROR 32001 'ProcInsertAnalysisSource:Failed execution of ProcInsertCollAnalysisXref'
        RETURN -999
      END
   END

END
go
```

ProcInsertAncillaryInput**Procedure ProcInsertAncillaryInput**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAncillaryInput              */
/*  Release       :  B0                                    */
/*  Module        :  Granules                              */
/*  Created on    :  3/31/97                               */
/*  Purpose       :  Create stored procedure to insert     */
/*                   Ancillary Input data for a Granule.   */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcInsertAncillaryInput
(@granuleId                 ID                  = 0,
 @AncillaryInputType        varchar(20)= NULL,
 @AncillaryInputPointer     varchar(255)    = NULL)
AS
BEGIN
  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40144
    RETURN -999
  END

  IF (@AncillaryInputType = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40144
    RETURN -999
  END

  IF (@AncillaryInputPointer = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40144
    RETURN -999
  END

  INSERT DsMdAncillaryInput(granuleId,
                    AncillaryInputType,
                    AncillaryInputPointer,
                    lastUpdate)
                VALUES(@granuleId,
                    @AncillaryInputType,
                    @AncillaryInputPointer,
                    getdate())

IF @@error!=0
  BEGIN
        RAISERROR 50037
        RETURN -999
```

```
  END

END
go
```

ProcInsertAncillaryInputGran**Procedure ProcInsertAncillaryInputGran**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertAncillaryInputGran          */
/*  Release       :  B0                                    */
/*  Module        :  Granules                              */
/*  Created on    :  3/31/97                               */
/*  Purpose       :  Create stored procedure to insert     */
/*                Ancillary Input data for a Granule.      */
/*  Calls         :  None                                  */
/*================================================================*/

CREATE PROCEDURE ProcInsertAncillaryInputGran
(@dbID                          ID = 0,
 @AncillaryInputType          varchar(20) = NULL,
 @AncillaryInputPointer       varchar(255) = NULL,
 @subType                     varchar(30),
 @insertTime                  datetime = NULL)
AS
BEGIN
  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40145
    RETURN -999
  END

IF (@AncillaryInputType = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40145
    RETURN -999
  END

  IF (@AncillaryInputPointer = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40145
    RETURN -999
  END

  IF (@insertTime = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40145
    RETURN -999
  END
```

```
  INSERT DsMdAncillaryInputGranule(dbID,
                        AncillaryInputType,
                        AncillaryInputPointer,
                        lastUpdate,
                                subType,
                                insertTime,
                                deleteEffectiveDate)
                    VALUES(@dbID,
                        @AncillaryInputType,
                        @AncillaryInputPointer,
                        getdate(),
                                @subType,
                                @insertTime,
                                NULL)

IF @@error!=0
  BEGIN
        RAISERROR 50038
        RETURN -999
  END

END
go
```

ProcInsertAPAssocColl**Procedure ProcInsertAPAssocColl**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAPAssocColl                      */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to insert          */
/*                DsMdCollectionAPXref                          */
/*  Calls         :  None                                       */
/*==============================================================*/


CREATE PROCEDURE ProcInsertAPAssocColl(
   @apId        ID,
   @ShortName    varchar(20),
   @VersionID    tinyint = NULL
)
AS
BEGIN
   DECLARE @collectionId ID

IF (@VersionID = NULL)
BEGIN
          RAISERROR 40120
          RETURN -999
END
```

```
--    IF (@VersionID < 1)
--    BEGIN
       /* Mimic Sybase error. */
--        RAISERROR 20023 "Procedure ProcInsertAPAssocColl expects parameter @VersionID of range 1
to 255."
--        RETURN -999
--    END

--    IF (@VersionID > 255)
--    BEGIN
       /* Mimic Sybase error. */
--        RAISERROR 20023 "Procedure ProcInsertAPAssocColl expects parameter @VersionID of range 1
to 255."
--        RETURN -999
--    END

    SELECT  @collectionId = dbID
    FROM    DsMdCollections
    WHERE   ShortName = @ShortName
    AND     VersionID = @VersionID

    IF @@error!=0
    BEGIN
            RAISERROR 45146
            RETURN -999
    END

    INSERT  DsMdCollectionAPXref
    VALUES( @collectionId,
           @apId)

    IF (@@error!= 0)
    BEGIN
            RAISERROR 50003
            RETURN -999
    END
END
go
```

ProcInsertAPContactXref**Procedure ProcInsertAPContactXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAPContactXref              */
/*  Release       :  B0                                   */
/*  Module        :  Delivered Algorithm Package          */
/*  Created on    :  4/1/97                               */
/*  Purpose       :  Create stored procedure to Delivered,   */
/*                Algorithm Package, and associate data   */
/*                tables.                                 */
/*  Calls         :                                       */
/*                                                        */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertAPContactXref(
            @dapId                          ID = 0,
            @contactId                  ID = 0)
AS
BEGIN
   IF (@dapId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40140
      RETURN -999
   END

   IF (@contactId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40140
      RETURN -999
   END


   INSERT  DsMdApContactXref
   VALUES( @dapId,
         @contactId )

   IF @@error!=0
   BEGIN
         RAISERROR 50033
         RETURN -999
   END
END
go
```

ProcInsertAPSAPComponentXref**Procedure ProcInsertAPSAPComponentXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAPSAPComponentXref          */
/*  Release       :  B0                                    */
/*  Module        :  Delivered Algorithm Package           */
/*  Created on    :  4/1/97                                */
/*  Purpose       :  Create stored procedure to Delivered,   */
/*                Algorithm Package, and associate data   */
/*                tables.                                  */
/*  Calls         :                                        */
/*                                                        */
/*==============================================================*/

CREATE PROCEDURE ProcInsertAPSAPComponentXref(
            @apId                           ID = 0,
            @ssapId                     ID = 0)
AS
BEGIN
```

```
   IF (@apId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40141
      RETURN -999
   END

   IF (@ssapId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40141
      RETURN -999
   END


   INSERT  DsMdAPSAPComponentXref
   VALUES( @apId,
         @ssapId )


   IF @@error!=0
   BEGIN
         RAISERROR 50034
         RETURN -999
   END
END
go
```

ProcInsertAssocPlatInstrSensor**Procedure ProcInsertAssocPlatInstrSensor**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertAssocPlatInstrSensor         */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   platform, instrument, sensor cross     */
/*                   reference tables for granules.         */
/*  Calls         :  ProcInsertGrPlatformXref               */
/*                   ProcInsertGrInstrumentXref             */
/*                   ProcInsertGrSensorXref                 */
/*==============================================================*/

CREATE PROCEDURE ProcInsertAssocPlatInstrSensor
            (@granuleId                    ID = 0,
             @AssociatedPlatformShortName    varchar(20) = NULL,
             @AssociatedInstrumentShortName  varchar(20) = NULL,
             @OperationMode                  varchar(20) = NULL,
             @AssociatedSensorShortName      varchar(20) = NULL)
AS
BEGIN
  IF (@granuleId = 0)
```

```
BEGIN
  /* Mimic Sybase error. */
  RAISERROR 40146
  RETURN -999
END

IF (@AssociatedPlatformShortName = NULL)
BEGIN
  /* Mimic Sybase error. */
  RAISERROR 40146
  RETURN -999
END

IF (@AssociatedInstrumentShortName = NULL)
BEGIN
  /* Mimic Sybase error. */
  RAISERROR 40146
  RETURN -999
END

IF (@AssociatedSensorShortName = NULL)
BEGIN
  /* Mimic Sybase error. */
  RAISERROR 40146
  RETURN -999
END

DECLARE @platDbId  ID
DECLARE @instDbId  ID
DECLARE @sensDbId  ID
DECLARE @retStat   int

SELECT @platDbId = dbID
FROM   DsMdPlatform
WHERE  PlatformShortName = @AssociatedPlatformShortName

EXECUTE @retStat = ProcInsertGranulePlatformXref @granuleId, @platDbId
IF (@retStat != 0)
BEGIN
  RAISERROR 56063
  RETURN -999
END

SELECT @instDbId = dbID
FROM   DsMdInstrument
WHERE  platformId = @platDbId
AND    InstrumentShortName = @AssociatedInstrumentShortName

EXECUTE @retStat = ProcInsertGranuleInstrmntXref @granuleId, @instDbId, @OperationMode
IF (@retStat != 0)
BEGIN
  RAISERROR 50039
  RETURN -999
END
```

```
  SELECT @sensDbId = dbID
  FROM   DsMdSensor
  WHERE  instrumentId = @instDbId
  AND    SensorShortName = @AssociatedSensorShortName

  EXECUTE @retStat = ProcInsertGranuleSensorXref @granuleId, @sensDbId
  IF (@retStat != 0)
  BEGIN
    RAISERROR 50040
    RETURN -999
  END
END
go
```

ProcInsertBrBrowseGranuleXref**Procedure ProcInsertBrBrowseGranuleXref**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertBrBrowseGranuleXref          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   Browse and Granule Xref data.          */
/*  Calls         :  None                                   */
/*================================================================*/

CREATE PROCEDURE ProcInsertBrBrowseGranuleXref
            (@browseId  ID = 0,
             @scienceGranule varchar(100) = NULL)
AS
BEGIN

  IF (@browseId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40149
    RETURN -999
  END


  IF (@scienceGranule = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40149
    RETURN -999
  END



   /* scienceGranule contains a GeOID, it must be converted to an    */
   /* ID for inserting into the DsMdBrowseGranuleXref table          */
```

```
  DECLARE @granuleId      ID
  DECLARE @backwards      varchar(100)

  SELECT  @backwards = reverse(@scienceGranule)
  SELECT  @granuleId = convert(numeric(16,0), reverse(substring(@backwards, 1, charindex(":",
@backwards) - 1 )))

 IF EXISTS (SELECT 1 FROM DsMdBrowseGranuleXref
                  WHERE granuleId =  @granuleId
              AND browseId  = @browseId
         )
   BEGIN
      RETURN
   END
 ELSE
   BEGIN

    INSERT DsMdBrowseGranuleXref(granuleId,
                        browseId)
                    VALUES(@granuleId,
                        @browseId)

    IF @@error!=0
      BEGIN
        RAISERROR 50043
        RETURN -999
      END
   END

END
go
```

ProcInsertBrowse**Procedure ProcInsertBrowse**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertBrowse                           */
/*  Release       :  B0                                 */
/*  Module        :  Granules                              */
/*  Created on    :  3/31/97                               */
/*  Purpose       :  Create stored procedure to insert      */
/*                Browse data for a Granule.             */
/*  Calls         :  ProcInsertCollBrowseXref              */
/*==============================================================*/

CREATE PROCEDURE ProcInsertBrowse
      (@dbID                        ID = 0,
       @subType                     varchar(30) = NULL,
          @BrowsePointer                  varchar(255),
          @BrowseDescription    varchar(255),
          @BrowseSize           float)
AS
```

```
BEGIN
  DECLARE @retStat int
  DECLARE @cid ID

  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40147
    RETURN -999
  END

  IF (@subType = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40147
    RETURN -999
  END

--  SELECT @cid = dbID FROM DsMdCollections WHERE subType = @subType

--  IF (@cid = NULL)
--  BEGIN
--    RAISERROR 32000 "Procedure ProcInsertBrowse: subType does not exist in DsMdCollections"
--    RETURN -999
--  END

-- Changed 10/24/97 - BrowseSize no longer requires conversion
--   INSERT DsMdBrowse(dbID,
--                     BrowsePointer,
--                     BrowseDescription,
--                     BrowseSize,
--                     lastUpdate,
--                     subType,
--                     insertTime,
--                     deleteEffectiveDate)
--              VALUES(@dbID,
--                     @BrowsePointer,
--                     @BrowseDescription,
--                     convert(integer, @BrowseSize),
--                     getdate(),
--                     @subType,
--                     getdate(),
--                     NULL)



  INSERT DsMdBrowse(dbID,
                    BrowsePointer,
                    BrowseDescription,
                    BrowseSize,
                    lastUpdate,
                    subType,
                    insertTime,
```

```
            deleteEffectiveDate)
        VALUES(@dbID,
            @BrowsePointer,
            @BrowseDescription,
            @BrowseSize,
            getdate(),
            @subType,
            getdate(),
            NULL)

  IF @@error!=0
  BEGIN
     RAISERROR 50041
     RETURN -999
  END

--  IF NOT EXISTS(SELECT 1 FROM DsMdCollBrowseXref
--                      WHERE collectionId = @cid
--                      AND browseId = @dbID)
--  BEGIN
--     EXECUTE @retStat = ProcInsertCollBrowseXref @cid, @dbID
--     IF @@error!=0
--     BEGIN
--                RAISERROR 32000 "ProcInsertCollBrowseXref: Failed Insert on DsMdCollBrowseXref"
--          RETURN -999
--     END
--  END
END
go
```

ProcInsertBrowseGranuleXref**Procedure ProcInsertBrowseGranuleXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertBrowseGranuleXref          */
/*  Release       :  B0                                   */
/*  Module        :  Granules                             */
/*  Created on    :  3/31/97                              */
/*  Purpose       :  Create stored procedure to insert    */
/*                Browse and Granule Xref data.           */
/*  Calls         :  None                                 */
/*==============================================================*/

CREATE PROCEDURE ProcInsertBrowseGranuleXref
            (@granuleId  ID = 0,
             @browseId   ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40148
```

```
    RETURN -999
  END

  IF (@browseId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40148
    RETURN -999
  END

 IF EXISTS (SELECT 1 FROM DsMdBrowseGranuleXref
                    WHERE granuleId =  @granuleId
               AND browseId  = @browseId
          )
    BEGIN
       RETURN
    END
 ELSE
    BEGIN
          INSERT DsMdBrowseGranuleXref(granuleId,
                       browseId)
                  VALUES(@granuleId,
                       @browseId)

      IF @@error!=0
      BEGIN
         RAISERROR 50042
         RETURN -999
      END
    END

END
go
```

ProcInsertCampaign**Procedure ProcInsertCampaign**

Code

```
/*================================================================*/
/*  Name            :  ProcInsertCampaign                  */
/*  Release         :  B0                              */
/*  Module          :  DataOriginator                      */
/*  Created on      :  3/30/97                         */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdCampaign              */
/*                  table                          */
/*                                                 */
/*  Calls           :                                 */
/*                  ProcInsertCollCampaignXref          */
/*                                                 */
/*================================================================*/


CREATE PROCEDURE ProcInsertCampaign
```

```
            (
        @collectionId           ID                  = 0,
        @CampaignShortName      varchar(20) = NULL,
        @CampaignLongName       varchar(80) = NULL,
        @CampaignStartDate      datetime    = NULL,
        @CampaignEndDate        datetime    = NULL
            )

AS
BEGIN

   DECLARE @campaignId ID
   DECLARE @retStat int

   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error */
      RAISERROR 20023 "Procedure ProcInsertCampaign expects parameter @dbID, which was not
supplied."
      RETURN -999
   END

   IF (@CampaignShortName = NULL)
   BEGIN
      /* Mimic Sybase error */
      RAISERROR 20023 "Procedure ProcInsertCampaign expects parameter @CampaignShortName,
which was not supplied."
      RETURN -999
   END

   SELECT @campaignId =
         dbID FROM DsMdCampaign
            WHERE CampaignShortName = @CampaignShortName

-- new message
   IF (@@error != 0)
   BEGIN
      /* Mimic Sybase error */
      RAISERROR 20023 "Procedure ProcInsertCampaign error getting  @campaignId with
CampaignShortName."
      RETURN -999
   END

 IF (@campaignId IS NULL)
   BEGIN

      BEGIN TRANSACTION ID_UPDATE_CAMPAIGN

      EXECUTE @retStat = ProcGetUniqueIDOutput "ID", "DsMdCampaign",  @campaignId OUTPUT

   IF ( @retStat != 0  )
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertCampaign error  on  call to ProcGetUniqueIDOutput."
```

```
    RETURN -999
  END

  COMMIT ID_UPDATE_CAMPAIGN

  INSERT DsMdCampaign(dbID,
                CampaignShortName,
                CampaignLongName,
                CampaignStartDate,
                CampaignEndDate)
  VALUES(@campaignId,
       @CampaignShortName,
       @CampaignLongName,
       @CampaignStartDate,
       @CampaignEndDate)

   IF @@error!=0
   BEGIN
     RAISERROR 32000 'ProcInsertCampaign:Failed Insert on DsMdCampaign'
     RETURN -999
   END

   EXECUTE @retStat = ProcInsertCollCampaignXref @collectionId,  @campaignId
   IF @retStat != 0
      BEGIN
        RAISERROR 32000 'ProcInsertCampaign:Failed execution of ProcInsertCollCampaignXref'
        RETURN -999
      END
  END
 ELSE
  BEGIN
  EXECUTE @retStat = ProcInsertCollCampaignXref @collectionId, @campaignId
  IF @retStat != 0
     BEGIN
       RAISERROR 32001 'ProcInsertCampaign:Failed execution of ProcInsertCollCampaignXref'
       RETURN -999
     END
  END

END
go
```

ProcInsertClDiscplnKeywrdXref**Procedure ProcInsertClDiscplnKeywrdXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertClDiscplnKeywrdXref          */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  4/28/97                                */
/*  Purpose       :  Procedure to insert Collection         */
/*                discipline keyword Xref data.             */
/*  Calls         :                                         */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertClDiscplnKeywrdXref
          (@collectionId         ID = 0,
       @disciplineKeywordId  ID = 0)
AS
BEGIN

  IF (@collectionId = 0)
    BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40154
         RETURN -999
    END

  IF (@disciplineKeywordId = 0)
    BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40154
         RETURN -999
    END


   INSERT DsMdCollDisciplineKeywordXref(collectionId,
                              disciplineKeywordId)
                         VALUES(@collectionId,
                              @disciplineKeywordId)
 IF @@error!=0
   BEGIN
     RAISERROR 50050
     RETURN -999
   END

END
go
```

ProcInsertClProdSpecific**Procedure ProcInsertClProdSpecific**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertClProdSpecific              */
/*  Release       :  B0                                 */
/*  Module        :  Collections                           */
/*  Created on    :  3/25/97                               */
/*  Purpose       :  Create stored procedure to insert      */
/*                   parameters into the appropriate       */
/*                   content tables based on the parameter's */
/*                   datatype.                             */
/*  Calls         :  ProcInsertCollFloatInfoContent         */
/*                   ProcInsertCollIntInfoContent          */
/*                   ProcInsertCollStrngInfoContent        */
/*==============================================================*/
```

```
CREATE PROC ProcInsertClProdSpecific
(@collectionId          ID              = 0,
 @clAttributeName       varchar(40)     = "",
 @clAttributeValue      varchar(255)    = "")
AS
BEGIN

   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40139
      RETURN -999
   END


   DECLARE @tname       varchar(32)
   DECLARE @attributeId  int
   DECLARE @retStat      int

   SELECT @tname = a.clTableName,
          @attributeId = b.attributeId
   FROM   DsMdParmToTableVector a,
             DsMdAdditionalAttributes b
   WHERE  a.DataType = b.AdditionalAttributeDataType AND
          b.AdditionalAttributeName = @clAttributeName

   IF @@error!=0
      BEGIN
         RAISERROR 45147
            RETURN -999
      END


-- PSA value is optional at the collection level,
-- so  if null and it took the default above (""), get out

   IF (@clAttributeValue IS NULL or @clAttributeValue = "")
      RETURN


  IF @tname = "DsMdCollFloatInfoContent"
      BEGIN
         EXECUTE @retStat = ProcInsertCollFloatInfoContent @collectionId, @attributeId,
@clAttributeValue
            IF @retStat != 0
               BEGIN
               RAISERROR 56059
                  RETURN -999
            END
      END
   ELSE IF @tname = "DsMdCollIntegerInfoContent"
      BEGIN
         EXECUTE @retStat = ProcInsertCollIntInfoContent @collectionId, @attributeId, @clAttributeValue
            IF @retStat != 0
```

```
            BEGIN
            RAISERROR 56060
               RETURN -999
         END
    END
  ELSE
    BEGIN
      EXECUTE @retStat = ProcInsertCollStrngInfoContent @collectionId, @attributeId,
@clAttributeValue
           IF @retStat != 0
             BEGIN
            RAISERROR 56061
               RETURN -999
         END
    END

END
go
```

ProcInsertCollAddnlAttribs**Procedure ProcInsertCollAddnlAttribs**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollAddnlAttribs              */
/*  Release       :  B0                                  */
/*  Module        :  Collections                           */
/*  Created on    :  05/09/97                              */
/*  Purpose       :  Create stored procedure to insert       */
/*                   additional attribute and information    */
/*                   content data for Collections.           */
/*  Calls         :  ProcInsertAdditionalAttributes          */
/*                   ProcInsertClProdSpecific                */
/*==============================================================*/

CREATE PROCEDURE ProcInsertCollAddnlAttribs
(@collectionId                 ID,
 @AdditionalAttributeName        varchar(40),
 @AdditionalAttributeDesc        varchar(255),
 @AdditionalAttributeDataType     char(10),
 @ParameterUnitsOfMeasure         varchar(20),
 @ParameterRangeBegin             varchar(40),
 @ParameterRangeEnd                                    varchar(40),
 @ParameterValueAccuracy          varchar(30),
 @ValueAccuracyExplanation        varchar(255),
 @MeasurementResolution           varchar(30),
 @ParameterValue                varchar(255))
AS
BEGIN

   DECLARE @retStat    int

-- not used here; tossed
-- but must have a variable to complete call
```

```
  DECLARE @attributeId int

  EXECUTE @retStat=ProcInsertAdditionalAttributes @AdditionalAttributeName,
                                    @AdditionalAttributeDesc,
                                    @AdditionalAttributeDataType,
                                    @ParameterUnitsOfMeasure,
                                    @ParameterRangeBegin,
                                              @ParameterRangeEnd,
                                    @ParameterValueAccuracy,
                                    @ValueAccuracyExplanation,
                                    @MeasurementResolution,
                                              @attributeId OUTPUT
  IF @retStat != 0
  BEGIN
    RAISERROR 56065
    RETURN -999
  END

  SELECT @attributeId = attributeId
  FROM   DsMdAdditionalAttributes
  WHERE  AdditionalAttributeName = @AdditionalAttributeName

  IF @@rowcount != 1
  BEGIN
    RAISERROR 45150
    RETURN -999
  END


  /*************************************************/
  /* insert the cross ref table                    */
  /*************************************************/
  EXECUTE @retStat=ProcInsertCollAddnlAttribsXref @collectionId,
                                    @attributeId

  IF @retStat != 0
  BEGIN
    RAISERROR 56066
    RETURN -999
  END


  EXECUTE @retStat = ProcInsertClProdSpecific @collectionId,
                                  @AdditionalAttributeName,
                                  @ParameterValue

  IF @retStat != 0
  BEGIN
    RAISERROR 56067
    RETURN -999
  END
END
go
```

ProcInsertCollAddnlAttribsXref**Procedure ProcInsertCollAddnlAttribsXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollAddnlAttribsXref         */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   cross reference data for Collections   */
/*                   and additional attributes.             */
/*  Calls         :                                         */
/*==============================================================*/

CREATE PROCEDURE ProcInsertCollAddnlAttribsXref
          (@collectionId  ID = 0,
           @attributeId     ID = 0)
AS
BEGIN

  IF (@collectionId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40126
    RETURN -999
  END

  IF (@attributeId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40126
    RETURN -999
  END


  INSERT  DsMdCollectionAddnlAttribsXref(collectionId,
                                attributeId)
  VALUES  (@collectionId,
         @attributeId)

  IF @@error!=0
  BEGIN
    RAISERROR 50011
    RETURN -999
  END

END
go
```

ProcInsertCollAnalysisXref**Procedure ProcInsertCollAnalysisXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollAnalysisXref             */
/*  Release       :  B0                                  */
/*  Module        :  Collections                           */
/*  Created on    :  3/25/97                               */
/*  Purpose       :  Create stored procedure to insert       */
/*                cross reference data for Collections    */
/*                and Analysis.                           */
/*  Calls         :                                      */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollAnalysisXref
(@collectionId      ID = 0,
 @analysisSourceId  ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    RAISERROR 40131
    RETURN -999
  END

  IF (@analysisSourceId = 0)
  BEGIN
    RAISERROR 40131
    RETURN -999
  END

  INSERT  DsMdCollectionAnalysisXref(collectionId,analysisSourceId)
  VALUES  (@collectionId,
        @analysisSourceId)
  IF @@error!=0
  BEGIN
    RAISERROR 50016
    RETURN -999
  END
END
go
```

ProcInsertCollAPXref**Procedure ProcInsertCollAPXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollAPXref                       */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                   cross reference data for Collections       */
/*                   and Algorithm Packages.                    */
/*  Calls         :                                             */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollAPXref
(@collectionId      ID = 0,
 @apId              ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    RAISERROR 40125
    RETURN -999
  END

  IF (@apId = 0)
  BEGIN
    RAISERROR 40125
    RETURN -999
  END

 IF EXISTS (SELECT 1 FROM DsMdCollectionAPXref
                  WHERE collectionId =  @collectionId
              AND apId  = @apId
         )
   BEGIN
      RETURN
   END
 ELSE
   BEGIN
         INSERT  DsMdCollectionAPXref
                  (
                  collectionId,
                  apId
                  )
         VALUES  (
                  @collectionId,
         @apId
                  )

      IF @@error!=0
```

```
      BEGIN
        RAISERROR 50010
        RETURN -999
      END
    END

END
go
```

ProcInsertCollAssociation**Procedure ProcInsertCollAssociation**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollAssociation              */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   association data for Collections.      */
/*  Calls         :                                         */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollAssociation
(
 @collectionId            ID                      = 0,
 @AssociatedShortName     char(8)      = NULL,
 @AssociatedVersionId     tinyint  = NULL,
 @CollectionType          varchar(20)      = NULL,
 @CollectionUse1          varchar(255),
 @CollectionUse2          varchar(255)
)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40152
      RETURN -999
  END

  IF (@AssociatedShortName = NULL)
  BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40152
      RETURN -999
  END

--    IF (@AssociatedVersionId < 1)
--    BEGIN
       /* Mimic Sybase error. */
--       RAISERROR 20023 "Procedure ProcInsertAPAssocColl expects parameter @VersionID of range 1
to 255."
```

```
--       RETURN -999
--    END

--    IF (@AssociatedVersionId > 255)
--    BEGIN
       /* Mimic Sybase error. */
--       RAISERROR 20023 "Procedure ProcInsertAPAssocColl expects parameter @VersionID of range 1
to 255."
--       RETURN -999
--    END

   IF (@CollectionType = NULL)
   BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40152
       RETURN -999
   END

   IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
       INSERT   DsMdCollectionAssociation(collectionId,
                                   AssociatedShortName,
                                   AssociatedVersionId,
                                   CollectionType,
                                   CollectionUse1,
                                   CollectionUse2)
       VALUES  (@collectionId,
              @AssociatedShortName,
              @AssociatedVersionId,
              @CollectionType,
              @CollectionUse1,
              @CollectionUse2)
       IF @@error!=0
       BEGIN
          RAISERROR 50047
              RETURN -999
       END
   END
   ELSE
   BEGIN
       /* Mimic Sybase error. */
       RAISERROR 50048
       RETURN -999
   END
END
go
```

ProcInsertCollBrowseXref**Procedure ProcInsertCollBrowseXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollBrowseXref              */
/*  Release       :  B0                                 */
/*  Module        :  Collections                          */
/*  Created on    :  3/25/97                              */
/*  Purpose       :  Create stored procedure to insert       */
/*                cross reference data for Collections    */
/*                and Browse.                             */
/*  Calls         :                                     */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollBrowseXref
(@collectionId      ID = 0,
 @browseId          ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    RAISERROR 40119
    RETURN -999
  END

  IF (@browseId = 0)
  BEGIN
    RAISERROR 40119
    RETURN -999
  END

  INSERT  DsMdCollBrowseXref(collectionId,browseId)
  VALUES  (@collectionId,
         @browseId)
  IF @@error!=0
  BEGIN
    RAISERROR 50002
    RETURN -999
  END
END
go
```

ProcInsertCollCampaignXref**Procedure ProcInsertCollCampaignXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollCampaignXref             */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   cross reference data for Collections   */
/*                   and Campaigns.                         */
/*  Calls         :                                         */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollCampaignXref
(@collectionId      ID = 0,
 @campaignId        ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    RAISERROR 40130
    RETURN -999
  END

  IF (@campaignId = 0)
  BEGIN
    RAISERROR 40130
    RETURN -999
  END

  INSERT  DsMdCollectionCampaignXref(collectionId,campaignId)
  VALUES  (@collectionId,
          @campaignId)
  IF @@error!=0
  BEGIN
    RAISERROR 50015
    RETURN -999
  END
END
go
```

ProcInsertCollContactXref**Procedure ProcInsertCollContactXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollContactXref              */
/*  Release       :  B0                                  */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   cross reference data for Collections   */
/*                   and Contacts.                          */
/*  Calls         :                                      */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollContactXref
(@collectionId  ID = 0,
 @contactId     ID = 0)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      RAISERROR 40153
      RETURN -999
   END

   IF (@contactId = 0)
   BEGIN
      RAISERROR 40153
      RETURN -999
   END

   INSERT DsMdCollectionContactXref(collectionId, contactId)
   VALUES (@collectionId,
         @contactId)

   IF @@error!=0
   BEGIN
      RAISERROR 50049
      RETURN -999
   END
END
go
```

ProcInsertCollections**Procedure ProcInsertCollections**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollections                      */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                   collection data.                           */
/*  Calls         :                                             */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollections
(
 @dbID                        ID                    = 0,
 @type                        varchar(30)   = NULL,
 @subType                     varchar(30),
 @ShortName                   char(8)       = NULL,
 @LongName                    varchar(80)= NULL,
 @CollectionDescription       varchar(255)  = NULL,
 @VersionID                   tinyint = NULL,
 @VersionDescription          varchar(255),
 @RevisionDate                datetime,
 @SuggestedUsage1             varchar(255),
 @SuggestedUsage2             varchar(245),
 @ProcessingCenter            varchar(20),
 @ArchiveCenter               varchar(20)   = NULL,
 @CitationForExtPublication   varchar(255),
 @CollectionState             char(10),
 @MaintenanceUpdateFrequency  varchar(80)   = NULL,
 @AccessConstraints           varchar(255),
 @ProcessingLevelID           char(6),
 @ProcessingLevelDescription  varchar(80)
)
--  @primaryCollectionFlag          char(1))
AS
BEGIN
--  IF (@primaryCollectionFlag = '')
--     select @primaryCollectionFlag = 'Y'

  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40163
    RETURN -999
  END

  IF (@type = NULL)
  BEGIN
    /* Mimic Sybase error. */
```

```
    RAISERROR 40163
    RETURN -999
  END

  IF (@ShortName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40163
    RETURN -999
  END

  IF (@LongName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    SELECT @LongName = "none"
/*
    RAISERROR 40163
    RETURN -999
*/
  END

  IF (@CollectionDescription = NULL)
  BEGIN

    SELECT @CollectionDescription = "none"

    /* Mimic Sybase error. */
/*
    RAISERROR 40163
    RETURN -999
*/
  END
  IF (@VersionID = NULL)
  BEGIN
    RAISERROR 40163
    RETURN -999
  END
--  IF (@VersionID = 0)
--  BEGIN
--    /* Mimic Sybase error. */
--    RAISERROR 40163
--    RETURN -999
--  END

--  IF (@VersionID < 1)
--  BEGIN
     /* Mimic Sybase error. */
--     RAISERROR 40163
--     RETURN -999
--  END

--  IF (@VersionID > 255)
--  BEGIN
     /* Mimic Sybase error. */
```

```
--      RAISERROR 40163
--      RETURN -999
--  END

  IF (@ArchiveCenter = NULL)
  BEGIN
     SELECT @ArchiveCenter = "none"

     /* Mimic Sybase error. */
/*
     RAISERROR 40163
     RETURN -999
*/
  END

  IF (@MaintenanceUpdateFrequency = NULL)
  BEGIN
     SELECT @MaintenanceUpdateFrequency = "none"

     /* Mimic Sybase error. */
/*
     RAISERROR 40163
     RETURN -999
*/
  END

     INSERT  DsMdCollections(dbID,
                        type,
                        subType,
                        lastUpdate,
                        ShortName,
                        LongName,
                        CollectionDescription,
                        VersionID,
                            VersionDescription,
                        RevisionDate,
                        SuggestedUsage1,
                        SuggestedUsage2,
                        ProcessingCenter,
                        ArchiveCenter,
                        CitationForExternalPublication,
                        CollectionState,
                        MaintenanceUpdateFrequency,
                        AccessConstraints,
                        ProcessingLevelID,
                        ProcessingLevelDescription,
--                        logicalDataServerName,
                        primaryCollectionFlag,
                        spatialType)
     VALUES  (@dbID,
           @type,
           @subType,
           getdate(),
           @ShortName,
```

```
            @LongName,
          @CollectionDescription,
          @VersionID,
          @VersionDescription,
          @RevisionDate,
          @SuggestedUsage1,
          @SuggestedUsage2,
          @ProcessingCenter,
          @ArchiveCenter,
          @CitationForExtPublication,
          @CollectionState,
          @MaintenanceUpdateFrequency,
          @AccessConstraints,
          @ProcessingLevelID,
          @ProcessingLevelDescription,
--          NULL,
--  @primaryCollectionFlag,
          "Y",
          NULL)
    IF @@error!=0
    BEGIN
      RAISERROR 50061
      RETURN -999
    END
END
go
```

ProcInsertCollFloatInfoContent**Procedure ProcInsertCollFloatInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollFloatInfoContent          */
/*  Release       :  B0                                      */
/*  Module        :  Collections                             */
/*  Created on    :  3/25/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                   float name/value pairs for Collections. */
/*  Calls         :                                          */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollFloatInfoContent
(@collectionId            ID  = 0,
 @attributeId                  int = 0,
 @clFloatValue           varchar(255))
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40133
    RETURN -999
  END
```

```
 IF (@attributeId = 0)
 BEGIN
   /* Mimic Sybase error. */
   RAISERROR 40133
   RETURN -999
 END

 IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
     IF exists(SELECT 1 FROM DsMdAdditionalAttributes WHERE attributeId = @attributeId)
     BEGIN
       INSERT  DsMdCollFloatInfoContent(collectionId,
                                 attributeId,
                                 clFloatValue)
       VALUES  (@collectionId,
              @attributeId,
              CONVERT(float, @clFloatValue))
       IF @@error!=0
       BEGIN
         RAISERROR 50020
         RETURN -999
       END
     END
     ELSE
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 50022
       RETURN -999
     END
   END
 ELSE
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 50021
     RETURN -999
   END

END
go
```

ProcInsertCollGranuleXref**Procedure ProcInsertCollGranuleXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollGranuleXref              */
/*  Release       :  B0                                  */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   cross reference data for Collections    */
/*                   and Granules.                           */
/*  Calls         :                                      */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertCollGranuleXref
(@collectionId      ID = 0,
 @granuleId         ID = 0)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      RAISERROR 40155
      RETURN -999
   END

   IF (@granuleId = 0)
   BEGIN
      RAISERROR 40155
      RETURN -999
   END


  INSERT  DsMdCollectionGranuleXref(collectionId,granuleId)
  VALUES  (@collectionId,
         @granuleId)
  IF @@error!=0
  BEGIN
    RAISERROR 50051
    RETURN -999
  END
END
go
```

ProcInsertCollInstrXref**Procedure ProcInsertCollInstrXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollInstrXref                    */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                   cross reference data for Collections       */
/*                   and Instruments.                           */
/*  Calls         :                                             */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollInstrXref
(@collectionId      ID = 0,
 @instrumentId      ID = 0)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
```

```
      RAISERROR 40156
      RETURN -999
   END

   IF (@instrumentId = 0)
   BEGIN
      RAISERROR 40156
      RETURN -999
   END


  INSERT  DsMdCollectionInstrumentXref(collectionId,instrumentId)
  VALUES  (@collectionId,
         @instrumentId)
  IF @@error!=0
  BEGIN
    RAISERROR 50052
    RETURN -999
  END
END
go
```

ProcInsertCollIntInfoContent**Procedure ProcInsertCollIntInfoContent**

Code

```
/*=================================================================*/
/*  Name          :  ProcInsertCollIntInfoContent            */
/*  Release       :  B0                                  */
/*  Module        :  Collections                           */
/*  Created on    :  3/25/97                               */
/*  Purpose       :  Create stored procedure to insert       */
/*                integer name/value pairs for Collections. */
/*  Calls         :                                      */
/*=================================================================*/


CREATE PROCEDURE ProcInsertCollIntInfoContent
(@collectionId         ID  = 0,
 @attributeId               int = 0,
 @clIntValue          varchar(255))
AS
BEGIN

  IF (@collectionId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40134
    RETURN -999
  END

  IF (@attributeId = 0)
  BEGIN
    /* Mimic Sybase error. */
```

```
   RAISERROR 40134
   RETURN -999
  END

  IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
    BEGIN
      IF exists(SELECT 1 FROM DsMdAdditionalAttributes WHERE attributeId = @attributeId)
      BEGIN
        INSERT  DsMdCollIntegerInfoContent(collectionId,
                                    attributeId,
                                    clIntValue)
        VALUES  (@collectionId,
                 @attributeId,
                 CONVERT(int, @clIntValue))
        IF @@error!=0
        BEGIN
          RAISERROR 50023
          RETURN -999
        END
      END
      ELSE
      BEGIN
        /* Mimic Sybase error. */
        RAISERROR 50025
        RETURN -999
      END
    END
  ELSE
    BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50024
      RETURN -999
    END

END
go
```

ProcInsertCollPlatformXref**Procedure ProcInsertCollPlatformXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollPlatformXref             */
/*  Release       :  B0                                  */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   cross reference data for Collections    */
/*                   and Platforms.                          */
/*  Calls         :                                      */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollPlatformXref
```

```
(@collectionId      ID = 0,
 @platformId        ID = 0)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
     RAISERROR 40157
     RETURN -999
  END

  IF (@platformId = 0)
  BEGIN
     RAISERROR 40157
     RETURN -999
  END

  INSERT  DsMdCollectionPlatformXref(collectionId,platformId)
  VALUES  (@collectionId,
         @platformId)
  IF @@error!=0
  BEGIN
    RAISERROR 50053
    RETURN -999
  END
END
go
```

ProcInsertCollReview**Procedure ProcInsertCollReview**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertCollReview                      */
/*  Release        :  B0                                 */
/*  Module         :  Collections                           */
/*  Created on     :  3/25/97                               */
/*  Purpose        :  Create stored procedure to insert       */
/*                 service data for Collections.          */
/*  Calls          :                                     */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollReview
(@collectionId         ID       = 0,
 @ScienceReviewStatus  varchar(20) = NULL,
 @ScienceReviewDate    datetime            = NULL,
 @FutureReviewDate     datetime)
AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40158
    RETURN -999
```

```
  END

  IF (@ScienceReviewStatus = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40158
    RETURN -999
  END

  IF (@ScienceReviewDate = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40158
    RETURN -999
  END

  IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
  BEGIN
    INSERT       DsMdCollReview(collectionId,
                        ScienceReviewStatus,
                        ScienceReviewDate,
                        FutureReviewDate)
    VALUES  (@collectionId,
          @ScienceReviewStatus,
          @ScienceReviewDate,
          @FutureReviewDate)
    IF @@error!=0
    BEGIN
      RAISERROR 50054
      RETURN -999
    END
  END
  ELSE
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 50055
    RETURN -999
  END
END
go
```

ProcInsertCollSensorXref**Procedure ProcInsertCollSensorXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollSensorXref              */
/*  Release       :  B0                                    */
/*  Module        :  Collections                           */
/*  Created on    :  3/25/97                               */
/*  Purpose       :  Create stored procedure to insert     */
/*                   cross reference data for Collections  */
/*                   and Sensors.                          */
/*  Calls         :                                        */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertCollSensorXref
(@collectionId      ID = 0,
 @sensorId          ID = 0)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40159
      RETURN -999
   END

   IF (@sensorId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40159
      RETURN -999
   END


   INSERT  DsMdCollectionSensorXref(collectionId,sensorId)
   VALUES  (@collectionId,
            @sensorId)
   IF @@error!=0
   BEGIN
      RAISERROR 50056
      RETURN -999
   END
END
go
```

ProcInsertCollStorageMedium**Procedure ProcInsertCollStorageMedium**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertCollStorageMedium            */
/*  Release        :  B0                                     */
/*  Module         :  Collections                            */
/*  Created on     :  3/25/97                                */
/*  Purpose        :  Create stored procedure to insert      */
/*                    storage medium data for Collections.   */
/*  Calls          :                                         */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollStorageMedium
( @collectionId      ID      = 0,
  @StorageMedium     varchar(30) = NULL)
AS
BEGIN
   IF (@collectionId = 0)
```

```
  BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40160
      RETURN -999
  END

  IF (@StorageMedium = NULL)
  BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40160
      RETURN -999
  END

  IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
  BEGIN
    INSERT  DsMdCollStorageMedium(StorageMedium,collectionId)
    VALUES  (@StorageMedium,
           @collectionId)
    IF @@error!=0
    BEGIN
          RAISERROR 50057
          RETURN -999
    END
  END
  ELSE
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 50058
    RETURN -99
  END
END
go
```

ProcInsertCollStrngInfoContent**Procedure ProcInsertCollStrngInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertCollStrngInfoContent         */
/*  Release       :  B0                                     */
/*  Module        :  Collections                              */
/*  Created on    :  3/25/97                                  */
/*  Purpose       :  Create stored procedure to insert       */
/*                   string name/value pairs for Collections.*/
/*  Calls         :                                         */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollStrngInfoContent
(@collectionId            ID  = 0,
 @attributeId                  int = 0,
 @clStringValue           varchar(255))
AS
BEGIN
```

```
 IF (@collectionId = 0)
 BEGIN
  /* Mimic Sybase error. */
  RAISERROR 40132
  RETURN -999
 END

 IF (@attributeId = 0)
 BEGIN
  /* Mimic Sybase error. */
  RAISERROR 40132
  RETURN -999
 END


 IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
  BEGIN
    IF exists(SELECT 1 FROM DsMdAdditionalAttributes WHERE attributeId = @attributeId)
    BEGIN
      INSERT  DsMdCollStringInfoContent(collectionId,
                                attributeId,
                                clStringValue)
      VALUES  (@collectionId,
            @attributeId,
            @clStringValue)

      IF @@error!=0
      BEGIN
        RAISERROR 50017
        RETURN -999
      END
    END
    ELSE
    BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50019
      RETURN -999
    END
  END
 ELSE
  BEGIN
   /* Mimic Sybase error. */
   RAISERROR 50018
   RETURN -999
  END

END
go
```

ProcInsertCollVertSpatialDomn**Procedure ProcInsertCollVertSpatialDomn**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertCollVertSpatialDomn          */
/*  Release         :  B0                                     */
/*  Module          :  Spatial                                */
/*  Created on      :  3/30/97                                */
/*  Purpose         :  Create Stored Procedure for inserting   */
/*                  data into DsMdCollVerticalSpatialDomain */
/*                  table                                     */
/*                                                            */
/*  Calls           :  None                                   */
/*                                                            */
/*==============================================================*/


CREATE PROCEDURE ProcInsertCollVertSpatialDomn(
            @collectionId               ID = 0,
            @VerticalSpatialDomainValue    varchar(20),
            @VerticalSpatialDomainType     varchar(20))

AS
BEGIN
  IF (@collectionId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40161
    RETURN -999
  END


    INSERT  DsMdCollVerticalSpatialDomain(collectionId, VerticalSpatialDomainValue,
VerticalSpatialDomainType)
    VALUES( @collectionId,
          @VerticalSpatialDomainValue,
          @VerticalSpatialDomainType)


    IF @@error!=0
    BEGIN
          RAISERROR 50059
          RETURN -999
    END
END
go
```

ProcInsertCollZoneIdentifier**Procedure ProcInsertCollZoneIdentifier**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertCollZoneIdentifier         */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                               */
/*  Created on      :  3/25/97                                */
/*  Purpose         :  Create Stored Procedure to insert     */
/*                  Collection Zone Identifier data       */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROC ProcInsertCollZoneIdentifier(
        @collectionId               ID                  = 0,
        @ZoneIdentifier             varchar(64)     = NULL)

AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40162
      RETURN -999
   END

   IF (@ZoneIdentifier = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40162
      RETURN -999
   END

   INSERT  DsMdCollZoneIdentifier(collectionId, ZoneIdentifier)
   VALUES( @collectionId,
        @ZoneIdentifier)

   IF @@error!=0
   BEGIN
      RAISERROR 50060
      RETURN -999
   END
END
go
```

ProcInsertContact**Procedure ProcInsertContact**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertContactPerson              */
/*  Release       :  B0                                 */
/*  Module        :  Contact                             */
/*  Created on    :  3/24/97                              */
/*  Purpose       :  Inserts data into DsDbContactPerson.    */
/*                                                      */
/*  Calls         :                                       */
/*                                                      */
/*==============================================================*/


CREATE PROCEDURE ProcInsertContact
      (@dbID                        ID            = 0,
       @ContactRole                 varchar(20) = NULL,
       @HoursOfService              varchar(255),
           @ContactInstructions     varchar(255))
AS
BEGIN
   IF (@dbID = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40164
      RETURN -999
   END

   IF (@ContactRole = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40164
      RETURN -999
   END

   INSERT INTO DsMdContact
         (dbID,
          ContactRole,
          HoursOfService,
          ContactInstructions,
          lastUpdate)
     VALUES
         (@dbID,
          @ContactRole,
          @HoursOfService,
          @ContactInstructions,
          getdate())

   IF (@@error!= 0)
   BEGIN
          RAISERROR 50062
```

```
        RETURN -999
   END
END
go
```

ProcInsertContactAddress**Procedure ProcInsertContactAddress**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertContactAddress              */
/*  Release       :  B0                                 */
/*  Module        :  Contact                             */
/*  Created on    :  3/24/97                             */
/*  Purpose       :  Create stored procedure to Contact,     */
/*                   and associate data tables.             */
/*  Calls         :                                     */
/*                                                     */
/*================================================================*/


CREATE PROCEDURE ProcInsertContactAddress
          ( @contactId            ID            = 0,
            @StreetAddress        varchar(80)       = NULL,
        @City                 varchar(30)       = NULL,
        @StateProvince        varchar(30) = NULL,
        @PostalCode           varchar(20)  = NULL,
        @Country              varchar(10)    = NULL)
AS
BEGIN
   IF (@contactId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40165
      RETURN -999
   END

   IF (@StreetAddress = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40165
      RETURN -999
   END

   IF (@City = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40165
      RETURN -999
   END

   IF (@StateProvince = NULL)
   BEGIN
      /* Mimic Sybase error. */
```

```
      RAISERROR 40165
      RETURN -999
   END

   IF (@PostalCode = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40165
      RETURN -999
   END

   IF (@Country = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40165
      RETURN -999
   END

   INSERT DsMdContactAddress ( contactId,
                        StreetAddress,
                            City,
                        StateProvince,
                        PostalCode,
                        Country )
   VALUES( @contactId,
         @StreetAddress,
            @City,
            @StateProvince,
            @PostalCode,
            @Country)

   IF @@error!=0
   BEGIN
         RAISERROR 50063
         RETURN -999
   END
END
go
```

ProcInsertContactOrganizations**Procedure ProcInsertContactOrganizations**

Code

```
/*=============================================================*/
/*  Name          :  ProcInsertContactOrganization          */
/*  Release       :  B0                                     */
/*  Module        :  Contact                                */
/*  Created on    :  3/24/97                                */
/*  Purpose       :  Create stored procedure to Contact,    */
/*                   and associate data tables.             */
/*  Calls         :                                         */
/*                                                          */
/*=============================================================*/
```

```
CREATE PROC ProcInsertContactOrganizations(
   @contactId               ID            = 0,
   @ContactRole             varchar(20),
   @HoursOfService          varchar(255),
   @ContactInstructions     varchar(255),
   @ContactOrganizationName   varchar(255) = NULL)
AS
BEGIN
   IF (@contactId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40166
      RETURN -999
   END

   IF (@ContactOrganizationName = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40166
      RETURN -999
   END


      DECLARE @RetStat int
      EXECUTE @RetStat = ProcInsertContact @dbID=@contactId,
                  @ContactRole=@ContactRole,
                  @HoursOfService=@HoursOfService,
                  @ContactInstructions=@ContactInstructions

      IF @RetStat!=0
            BEGIN
                  RAISERROR 56068
               RETURN -999
            END

         INSERT DsMdContactOrganizations( contactId,
                                    ContactOrganizationName )
         VALUES(          @contactId,
                  @ContactOrganizationName )

      IF @@error!=0
            BEGIN
                  RAISERROR 50064
               RETURN -999
            END
END
go
```

ProcInsertContactPersons**Procedure ProcInsertContactPersons**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertContactPersons              */
/*  Release       :  B0                                  */
/*  Module        :  Contact                               */
/*  Created on    :  3/24/97                               */
/*  Purpose       :  Create stored procedure to contact,     */
/*                   and associate data tables.             */
/*  Calls         :                                        */
/*                                                         */
/*==============================================================*/


CREATE PROCEDURE ProcInsertContactPersons
          (@contactId             ID              = 0,
       @ContactRole           varchar(20),
       @HoursOfService        varchar(255),
       @ContactInstructions   varchar(255),
           @ContactJobPosition    varchar(255),
           @ContactFirstName      varchar(255)    = null,
           @ContactMiddleName     varchar(255),
           @ContactLastName       varchar(255)   = null)
AS
BEGIN
   IF (@contactId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40167
      RETURN -999
   END

   IF (@ContactFirstName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40167
      RETURN -999
   END

   IF (@ContactLastName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40167
      RETURN -999
   END


  DECLARE @RetStat int
  EXECUTE @RetStat = ProcInsertContact @dbID=@contactId,
                     @ContactRole=@ContactRole,
                     @HoursOfService=@HoursOfService,
```

```
                @ContactInstructions=@ContactInstructions

  IF @RetStat!=0
  BEGIN
    RAISERROR 56069
    RETURN -999
  END

  INSERT DsMdContactPersons(contactId,
                            ContactFirstName,
                            ContactMiddleName,
                            ContactLastName,
                            ContactJobPosition)
  VALUES(@contactId,
         @ContactFirstName,
         @ContactMiddleName,
         @ContactLastName,
         @ContactJobPosition)

  IF @@error!=0
  BEGIN
    RAISERROR 50065
    RETURN -999
  END
END
go
```

ProcInsertCSDTDescription**Procedure ProcInsertCSDTDescription**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertCSDTDescription                    */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to insert            */
/*                CSDT description data for Collections.          */
/*  Calls         :                                               */
/*================================================================*/


CREATE PROCEDURE ProcInsertCSDTDescription
(@collectionId       ID = 0,
 @PrimaryCSDT              varchar(30) = NULL,
 @IndirectReference  varchar(100),
 @Implementation     varchar(100),
 @CSDTComments       varchar(255))
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40150
```

```
      RETURN -999
   END

   IF (@PrimaryCSDT = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40150
      RETURN -999
   END

   IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
      INSERT  DsMdCSDTDescription(collectionId,
                           PrimaryCSDT,
                           IndirectReference,
                           Implementation,
                           CSDTComments)
            VALUES (@collectionId,
                  @PrimaryCSDT,
              @IndirectReference,
              @Implementation,
              @CSDTComments)

            IF @@error!=0
      BEGIN
         RAISERROR 50044
            RETURN -999
      END
   END
   ELSE
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50045
      RETURN -999
   END
END
go
```

ProcInsertDAP**Procedure ProcInsertDAP**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertDAP                              */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to insert data     */
/*                   for Deliverd Algorithm Package             */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcInsertDAP(
            @dbID          ID = 0,
```

```
            @DAPID          varchar(12),
            @DAPInsertDate  datetime)
AS
BEGIN
  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40168
    RETURN -999
  END

  INSERT  DsMdDAP (dbID,
              DAPID,
              DAPInsertDate,
              lastUpdate,
              insertTime)
  VALUES(@dbID,
       @DAPID,
       @DAPInsertDate,
       getdate(),
       getdate())

  IF @@error!=0
  BEGIN
    RAISERROR 50066
    RETURN -999
  END
END
go
```

ProcInsertDepthResolution**Procedure ProcInsertDepthResolution**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertDepthResolution            */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                              */
/*  Created on      :  3/25/97                              */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdDepthResolution           */
/*                                                          */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROCEDURE ProcInsertDepthResolution(
            @collectionId                   ID    = 0,
            @DepthResolution       float = null)
AS
BEGIN
   IF (@collectionId = 0)
```

```
    BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40137
      RETURN -999
    END

  IF (@DepthResolution = null)
    BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40137
      RETURN -999
    END


  INSERT  DsMdDepthResolution (collectionId, DepthResolution)
  VALUES( @collectionId,
          @DepthResolution)

  IF @@error!=0
  BEGIN
        RAISERROR 50028
        RETURN -999
  END
END
go
```

ProcInsertDictionaryAttribute**Procedure ProcInsertDictionaryAttribute**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertDictionaryAttribute          */
/*  Release       :  B0                                  */
/*  Module        :  Dictionary                            */
/*  Created on    :  10/28/97                              */
/*  Purpose       :  Create stored procedure to Dictionary,  */
/*                   and associate data tables.             */
/*  Calls         :                                       */
/*                                                       */
/*==============================================================*/


CREATE PROCEDURE ProcInsertDictionaryAttribute
          (
          @attributeID          ID = 0,
          @contentID            ID = 0,
          @qualifiedAttrName    varchar(255),
          @type                 varchar(15),
          @length               int,
          @seqNum               int,
          @seqType              int,
          @Rule1                varchar(255),
          @optionalIndicator    tinyint,
          @maxOccurances        smallint,
```

```
            @glType              char(10),
            @psaIndicator          tinyint
           )
AS
BEGIN

   DECLARE @newAttrId          ID
   DECLARE @operator           varchar(255)
   DECLARE @pos       int
   DECLARE @end       int
   DECLARE @retStat    int

   IF (EXISTS (
             SELECT 1 FROM DsDeDictionaryAttribute
             WHERE qualifiedAttrName = @qualifiedAttrName
            ))
     BEGIN
        RETURN
     END

   IF (@contentID = 0)
     BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40127
        RETURN -999
     END

BEGIN TRAN  ATTR_INSERT_DICT

IF (@attributeID = 0) OR EXISTS (
             SELECT 1 FROM DsDeDictionaryAttribute
             WHERE attributeID = @attributeID
            )
  BEGIN
      EXECUTE @retStat = ProcGetUniqueIDOutput "int", "DsDeDictionaryAttribute",  @newAttrId
OUTPUT

    IF ( @retStat != 0  )
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 20029 "Procedure ProcInsertDictionaryAttribute error  on  call to
ProcGetUniqueIDOutput."
       RETURN -999
    END
  END
ELSE
  BEGIN
     SELECT @newAttrId = @attributeID
  END

IF (@Rule1 IS NULL) OR (@Rule1 = "NONE")
  BEGIN
     SELECT @operator = "NONE"
  END
```

```
ELSE
  BEGIN
     SELECT @end = char_length(@Rule1)
     SELECT @pos = charindex('(', substring(@Rule1, 1, @end))
     SELECT @operator = substring(@Rule1, 1, @pos - 1)
     SELECT @operator = ltrim(rtrim(@operator))
  END

INSERT DsDeDictionaryAttribute(attributeID,
                        contentID,
                        qualifiedAttrName,
                        type,
                        length,
                        seqNum,
                        seqType,
                        operator,
                        optionalIndicator,
                        maxOccurances,
                        glType,
                        psaIndicator )
          VALUES(       @newAttrId,
                  @contentID,
                  @qualifiedAttrName,
                  @type,
                  @length,
                  @seqNum,
                  @seqType,
                  @operator,
                  @optionalIndicator,
                  @maxOccurances,
                  @glType,
                  @psaIndicator )

          IF @@error!=0
            BEGIN
          RAISERROR 50012
                  ROLLBACK
                  RETURN -999
            END

  COMMIT ATTR_INSERT_DICT

END
go
```

ProcInsertDictionaryContent**Procedure ProcInsertDictionaryContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertDictionaryContent          */
/*  Release       :  B0                                   */
/*  Module        :  Dictionary                           */
/*  Created on    :  4/17/97                              */
/*  Purpose       :  Create stored procedure to Dictionary,  */
/*                   and associate data tables.           */
/*  Calls         :                                       */
/*                                                       */
/*==============================================================*/


CREATE PROCEDURE ProcInsertDictionaryContent
          (@contentID             ID = 0,
           @qualifiedGroupName     varchar(255),
           @numlevels                smallint,
           @numAttribs             smallint,
           @skips                  tinyint,
           @numMultiples           smallint,
           @containerIndicator     tinyint,
           @optionalIndicator      tinyint)
AS
BEGIN
   IF (@contentID = 0)
    BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40128
     RETURN -999
    END


          INSERT DsDeDictionaryContent
          VALUES(          @contentID,
                    @qualifiedGroupName,
                    @numlevels,
                    @numAttribs,
                    @skips,
                    @numMultiples,
                    @containerIndicator,
                    @optionalIndicator)

          IF @@error!=0
            BEGIN
          RAISERROR 50013
                    RETURN -999
            END
END
go
```

ProcInsertDictionaryRule**Procedure ProcInsertDictionaryRule**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertDictionaryRule              */
/*  Release       :  B0                                 */
/*  Module        :  Dictionary                           */
/*  Created on    :  10/28/97                              */
/*  Purpose       :  Create stored procedure to Dictionary,  */
/*                   and associate data tables.             */
/*  Calls         :                                     */
/*                                                     */
/*==============================================================*/


CREATE PROCEDURE ProcInsertDictionaryRule
       (
        @attributeID          ID = 0,
        @sequenceNo           int = 0,
        @ruleText             varchar(255) = NULL
       )
AS
BEGIN

   IF (@attributeID = 0)
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40341
       RETURN -999
     END

   IF (@sequenceNo = 0)
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40341
       RETURN -999
     END

   IF (@ruleText IS NULL or @ruleText = "")
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40341
       RETURN -999
     END

BEGIN TRANSACTION RULE_INSERT_DICT

INSERT DsDeDictionaryRule(attributeID,
                    sequenceNo,
                    ruleText)

               VALUES(@attributeID,
                    @sequenceNo,
```

```
                @ruleText)


IF @@error!=0
  BEGIN
     RAISERROR 32000 "ProcInsertDictionaryRule: Failed Insert on DsDeDictionaryRule"
     RETURN -999
  END

COMMIT RULE_INSERT_DICT

END
go
```

ProcInsertDisciplineKeywords**Procedure ProcInsertDisciplineKeywords**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertDisciplineKeywords           */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  4/28/97                                */
/*  Purpose       :  Stored procedure to insert discipline   */
/*                   keyword data for Collections.          */
/*  Calls         :                                         */
/*==============================================================*/

CREATE PROCEDURE ProcInsertDisciplineKeywords
          (@dbID                        ID  = 0 ,
           @ECSDisciplineKeyword  varchar(24) = null,
           @ECSTopicKeyword       varchar(32) = null,
           @ECSTermKeyword        varchar(50),
           @ECSVariableKeyword    varchar(80))

AS
BEGIN

  IF (@dbID = 0)
    BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40169
         RETURN -999
    END

  IF (@ECSDisciplineKeyword = null)
    BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40169
      RETURN -999
    END

  IF (@ECSTopicKeyword = null)
    BEGIN
```

```
      /* Mimic Sybase error. */
      RAISERROR 40169
      RETURN -999
    END


  INSERT  DsMdDisciplineKeywords(dbID,
                        ECSDisciplineKeyword,
                        ECSTopicKeyword,
                        ECSTermKeyword,
                        ECSVariableKeyword)
  VALUES  (@dbID,
         @ECSDisciplineKeyword,
            @ECSTopicKeyword,
         @ECSTermKeyword,
         @ECSVariableKeyword)

  IF @@error!=0
    BEGIN
      RAISERROR 50067
      RETURN -999
    END

END
go
```

ProcInsertDiscontMulti**Procedure ProcInsertDiscontMulti**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertDiscontMulti                   */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for updating  */
/*                  data into DsMdTemporal                      */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertDiscontMulti(
             @collectionId                    ID,
             @EndsatPresentFlag          char(8) )
AS
BEGIN
    UPDATE  DsMdTemporal
    SET     EndsatPresentFlag = @EndsatPresentFlag
    WHERE   collectionId = @collectionId

    IF @@error!=0
```

```
   BEGIN
         RAISERROR 54005
         RETURN -999
   END
END
go
```

Procedure ProcInsertECSParameterKeywords

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertECSParameterKeywords         */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  4/28/97                                */
/*  Purpose       :  Procedure to insert ECS Parameter      */
/*                   keyword data.                          */
/*  Calls         :                                         */
/*==============================================================*/

CREATE PROCEDURE ProcInsertECSParameterKeywords
            (@parameterKeywordId   ID          = 0,
             @ECSParameterKeyword  varchar(80) = null)
AS
BEGIN

  IF (@parameterKeywordId = 0)
    BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40171
         RETURN -999
    END

  IF (@ECSParameterKeyword = null)
    BEGIN
         /* Mimic Sybase error. */
         RAISERROR 40171
         RETURN -999
    END

 INSERT DsMdECSParameterKeywords(parameterKeywordId,
                          ECSParameterKeyword)
                   VALUES(@parameterKeywordId,
                          @ECSParameterKeyword)
 IF @@error!=0
    BEGIN
      RAISERROR 50068
      RETURN -999
    END

END
go
```

ProcInsertEmailAddress**Procedure ProcInsertEmailAddress**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertEmailAddress                     */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  Create stored procedure to Collection,  */
/*                   and associate data tables.                 */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertEmailAddress
            (@contactId             ID = 0,
        @ElectronicMailAddress  varchar(255) = null)
AS
BEGIN
-- @emailAddressId internal PK with contactId because
-- ElectronicMailAddress is too long with contactId

  DECLARE @emailAddressId  int

  IF (@contactId = 0)
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40172
       RETURN -999
     END

  IF (@ElectronicMailAddress = null)
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40172
       RETURN -999
     END

  IF NOT EXISTS
       ( SELECT 1
         FROM  DsMdEmailAddress
         WHERE contactId = @contactId
       )
    BEGIN
       SELECT @emailAddressId = 1
    END
    ELSE
    BEGIN
       SELECT @emailAddressId = MAX(emailAddressId) + 1
            FROM   DsMdEmailAddress
       WHERE contactId = @contactId
    END
```

```
 /* Check to see if primary key exists before inserting record. */
 IF exists(SELECT dbID FROM DsMdContact WHERE dbID = @contactId)
   BEGIN
        INSERT DsMdEmailAddress( contactId,
                                 emailAddressId,
                                 ElectronicMailAddress)
        VALUES(          @contactId,
                 @emailAddressId,
                 @ElectronicMailAddress)
        IF @@error!=0
          BEGIN
        RAISERROR 50069
                 RETURN -999
          END
   END
  ELSE
   BEGIN
        /* Mimic Sybase error. */
        RAISERROR 50070
        RETURN -999
   END
END
go
```

ProcInsertESDTConfiguredType**Procedure ProcInsertESDTConfiguredType**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertESDTConfiguredType          */
/*  Release       :  B0                                    */
/*  Module        :                                        */
/*  Created on    :  8/20/97                               */
/*  Purpose       :                                        */
/*  Calls         :                                        */
/*==============================================================*/


CREATE PROCEDURE ProcInsertESDTConfiguredType
(
   @configuredName           varchar(20)     = NULL,
   @versionID                varchar(20)     = NULL,
   @archiveIdentifier        varchar(20)     = NULL,
   @backupIdentifier         varchar(20)     = NULL,
   @offsiteIdentifier        varchar(20)     = NULL,
   @description              varchar(255)    = NULL,
   @descriptorFileName       varchar(80)     = NULL,
   @dllFileName              varchar(80)     = NULL,
   @eventFileName            varchar(80)     = NULL
)
AS
BEGIN
```

```
  IF (@configuredName IS NULL OR @configuredName = "")
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40353
      /*  "Procedure ProcInsertESDTConfiguredType expects parameter @configuredName, which was not
supplied."  */
      RETURN -999
   END

  IF (@versionID IS NULL OR @versionID = "")
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40354
      /*  "Procedure ProcInsertESDTConfiguredType expects parameter @versionID, which was not
supplied."  */
      RETURN -999
   END

    INSERT DsGeESDTConfiguredType
          (
          configuredName,
          versionID,
          archiveIdentifier,
          description,
          descriptorFileName,
          dllFileName,
          eventFileName,
          backupIdentifier,
          offsiteIdentifier
         )

    VALUES( @configuredName,
           CONVERT(tinyint, CONVERT(FLOAT, @versionID)),
           @archiveIdentifier,
           @description,
           @descriptorFileName,
           @dllFileName,
           @eventFileName,
           @backupIdentifier,
           @offsiteIdentifier)

    IF @@error!=0
       BEGIN
          RAISERROR 50134
          RETURN -999
       END
END
go
```

ProcInsertFileStorage**Procedure ProcInsertFileStorage**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertFileStorage                      */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  3/31/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                FileStorage data.                             */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE  ProcInsertFileStorage(@granuleId              ID = 0,
                                  @internalFileName     varchar(80) = NULL,
                                  @filePath         varchar(80),
                                    @userDataFile         varchar(40),
                                @checkSum           int,
                                @fileSize           int,
                                @creationDate         datetime,
                                @ArchiveID          varchar(255) = NULL)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40173
    RETURN -999
  END

  IF (@internalFileName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40173
    RETURN -999
  END

  IF (@ArchiveID = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40173
    RETURN -999
  END

   INSERT DsMdFileStorage(granuleId,
                      internalFileName,
                      filePath,
                      userDataFile,
                      checkSum,
                      fileSize,
                      creationDate,
```

```
                ArchiveID,
                deleteEffectiveDate)
            VALUES(@granuleId,
                @internalFileName,
                @filePath,
                @userDataFile,
                @checkSum,
                @fileSize,
                @creationDate,
                @ArchiveID,
                NULL)
  IF @@error!=0
    BEGIN
      RAISERROR 50071
          RETURN -999
    END
END
go
```

ProcInsertGranAncillaryInpXref**Procedure ProcInsertGranAncillaryInpXref**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertGranAncillaryInpXref          */
/*  Release       :  B0                                      */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                   Granule Ancillary Input Xref data.      */
/*  Calls         :  None                                    */
/*================================================================*/
CREATE PROCEDURE ProcInsertGranAncillaryInpXref
            (@granuleId              ID = 0,
             @ancillaryInputGranuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40182
    RETURN -999
  END

  IF (@ancillaryInputGranuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40182
    RETURN -999
  END

  INSERT DsMdGranulesAncillaryInputXref(granuleId,
                                ancillaryInputGranuleId)
                        VALUES(@granuleId,
```

```
                                @ancillaryInputGranuleId)
   IF @@error!=0
   BEGIN
      RAISERROR 50084
      RETURN -999
   END

END
go
```

ProcInsertGranOrbitParmXref**Procedure ProcInsertGranOrbitParmXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranOrbitParmXref            */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                   Granule Orbit Parameter Xref data.      */
/*  Calls         :  None                                   */
/*==============================================================*/
CREATE PROCEDURE ProcInsertGranOrbitParmXref
            (@granuleId           ID = 0,
             @orbitParamGranuleId  ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40185
    RETURN -999
  END

  IF (@orbitParamGranuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40185
    RETURN -999
  END

   INSERT DsMdGranulesOrbitParmXref(granuleId,
                           orbitParamGranuleId)
                       VALUES(@granuleId,
                           @orbitParamGranuleId)
   IF @@error!=0
   BEGIN
      RAISERROR 50087
      RETURN -999
   END

END
go
```

ProcInsertGranuleAnalysisXref**Procedure ProcInsertGranuleAnalysisXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranuleAnalysisXref          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                Granule and Analysis Source Xref data.  */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcInsertGranuleAnalysisXref
            (@granuleId        ID = 0,
             @AnalysisShortName varchar(20) = NULL)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40181
    RETURN -999
  END

  IF (@AnalysisShortName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40181
    RETURN -999
  END

  DECLARE @analysisId   ID

  SELECT @analysisId = dbID
  FROM   DsMdAnalysisSource
  WHERE  AnalysisShortName = @AnalysisShortName

  IF ( @@error!=0 OR @@rowcount != 1 )
  BEGIN
     RAISERROR 32000 "ProcInsertGranuleAnalysisXref: Failed select on DsMdAnalysisSource"
     RETURN -999
  END

  INSERT DsMdGranuleAnalysisXref(granuleId,
                          analysisSourceId)
                    VALUES(@granuleId,
                          @analysisId)
  IF @@error!=0
  BEGIN
     RAISERROR 50083
     RETURN -999
```

```
  END

END
go
```

ProcInsertGranuleCampaignXref**Procedure ProcInsertGranuleCampaignXref**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertGranuleCampaignXref          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   Granule and Campaign Xref data.        */
/*  Calls         :  None                                   */
/*================================================================*/

CREATE PROCEDURE ProcInsertGranuleCampaignXref
            (@granuleId         ID = 0,
             @CampaignShortName  varchar(20) = NULL)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40183
    RETURN -999
  END

  IF (@CampaignShortName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40183
    RETURN -999
  END

  DECLARE @campId  ID

  SELECT @campId = dbID
  FROM   DsMdCampaign
  WHERE  CampaignShortName = @CampaignShortName

  IF ( @@error!=0 OR @@rowcount != 1 )
  BEGIN
     RAISERROR 32000 "ProcInsertGranuleCampaignXref: Failed select on DsMdCampaign"
     RETURN -999
  END

  INSERT DsMdGranuleCampaignXref(granuleId,
                                 campaignId)
                         VALUES(@granuleId,
```

```
                        @campId)

   IF @@error!=0
   BEGIN
      RAISERROR 50085
      RETURN -999
   END

END
go
```

ProcInsertGranuleInstrmntXref**Procedure ProcInsertGranuleInstrmntXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranuleInstrmntXref          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                Granule and Instrument Xref data.       */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcInsertGranuleInstrmntXref
            (@granuleId        ID = 0,
             @instrumentId     ID = 0,
                    @OperationMode     varchar(20) )
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40121
    RETURN -999
  END

  IF (@instrumentId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40121
    RETURN -999
  END

-- added to handle multiple associated platform instrument sensor containers
-- per granule
-- NOTE: only one op mode per granule instrument combination allowed

 IF EXISTS (SELECT 1 FROM DsMdGranuleInstrumentXref
                    WHERE granuleId =  @granuleId
              AND instrumentId  = @instrumentId
         )
```

```
    BEGIN
       RETURN
    END
  ELSE
  BEGIN
    INSERT DsMdGranuleInstrumentXref(granuleId,
                                instrumentId,
                                        OperationMode)
                        VALUES(@granuleId,
                                @instrumentId,
                                        @OperationMode)
    IF @@error!=0
    BEGIN
       RAISERROR 50005
       RETURN -999
    END
  END

END
go
```

ProcInsertGranuleLocality**Procedure ProcInsertGranuleLocality**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranuleLocality              */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                Granule Locality data.                    */
/*  Calls         :  None                                */
/*==============================================================*/
CREATE PROCEDURE ProcInsertGranuleLocality(
            @referenceId          ID = 0,
            @LocalityValue        varchar(80) = NULL)
AS
BEGIN
  IF (@referenceId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40184
    RETURN -999
  END
  IF (@LocalityValue = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40184
    RETURN -999
  END


   INSERT  DsMdGranuleLocality(granuleId,
```

```
                    LocalityValue)
                VALUES(@referenceId,
                        @LocalityValue)

   IF @@error!=0
   BEGIN
        RAISERROR 50086
        RETURN -999
   END
END
go
```

ProcInsertGranulePlatformXref**Procedure ProcInsertGranulePlatformXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranulePlatformXref          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                Granule and Platform Xref data.         */
/*  Calls         :  None                                   */
/*==============================================================*/
CREATE PROCEDURE ProcInsertGranulePlatformXref
            (@granuleId        ID = 0,
             @platformId       ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40121
    RETURN -999
  END

  IF (@platformId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40121
    RETURN -999
  END

 IF EXISTS (SELECT 1 FROM DsMdGranulePlatformXref
                  WHERE granuleId =  @granuleId
              AND platformId  = @platformId
         )
   BEGIN
      RETURN
   END
 ELSE
   BEGIN
```

```
        INSERT DsMdGranulePlatformXref(granuleId,
                        platformId)
          VALUES(
                    @granuleId,
               @platformId
              )

        IF @@error!=0
        BEGIN
       RAISERROR 50004
       RETURN -999
        END
   END

END
go
```

ProcInsertGranuleReview**Procedure ProcInsertGranuleReview**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranuleReview              */
/*  Release       :  B0                                   */
/*  Module        :  Granules                             */
/*  Created on    :  3/31/97                              */
/*  Purpose       :  Create stored procedure to insert    */
/*                Granule level Review data.              */
/*  Calls         :  None                                 */
/*==============================================================*/

CREATE PROCEDURE ProcInsertGranuleReview
(@granuleId                     ID = 0,
 @ScienceReviewStatus                   varchar(20) = NULL,
 @ScienceReviewDate             datetime = NULL,
 @FutureReviewDate              datetime)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40180
    RETURN -999
  END

  IF (@ScienceReviewDate = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40180
    RETURN -999
  END

  IF (@ScienceReviewStatus = NULL)
```

```
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40180
    RETURN -999
  END


   INSERT DsMdGranuleReview(granuleId,
                        ScienceReviewStatus,
                        ScienceReviewDate,
                        FutureReviewDate)
                    VALUES(@granuleId,
                        @ScienceReviewStatus,
                        @ScienceReviewDate,
                        @FutureReviewDate)
IF @@error!=0
  BEGIN
          RAISERROR 50082
          RETURN -999
  END

END
go
```

ProcInsertGranules**Procedure ProcInsertGranules**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertGranules                   */
/*  Release       :  B0                                */
/*  Module        :  Granules                             */
/*  Created on    :  3/31/97                              */
/*  Purpose       :  Create stored procedure to insert       */
/*                a Granule.                           */
/*  Calls         :  ProcCheckDsTMdTime                   */
/*================================================================*/

CREATE PROCEDURE ProcInsertGranules (
              @dbID                    ID      = 0,
            @type                          varchar(60) = NULL,
                 @subType                        varchar(30) = NULL,
              @ShortName               char(8) = NULL,
              @SizeMBECSDataGranule      float,
              @ReprocessingPlanned       varchar(45),
              @ReprocessingActual        varchar(20),
              @LocalGranuleID            varchar(80),
              @DayNightFlag              char(5),
              @ProductionDateTime        datetime,
              @LocalVersionID            varchar(60),
              @PGEVersion                char(10),
              @RangeEndingTime           DsTMdTime,
              @RangeEndingDate           datetime,
              @RangeBeginningTime        DsTMdTime,
              @RangeBeginningDate        datetime,
```

```
            @TimeOfDay              DsTMdTime,
            @CalendarDate           datetime,
            @ZoneIdentifier         varchar(64),
            @processingHistoryId    ID,
            @processingHistoryTypeCode  int)
AS
BEGIN

  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40187
    RETURN -999
  END

  IF (@type = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40187
    RETURN -999
  END

  IF (@subType = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40187
    RETURN -999
  END

  IF (@ShortName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40187
    RETURN -999
  END

   DECLARE @retStat            int
   DECLARE @primaryCollectionId ID
   DECLARE @VersionID          tinyint
   DECLARE @OUTPUT_DsTMdTime    DsTMdTime

   SELECT  @primaryCollectionId = dbID,
         @VersionID          = VersionID
   FROM    DsMdCollections
   WHERE   type = @type AND subType = @subType

   IF (@@error!=0 OR @@rowcount!=1)
   BEGIN
          RAISERROR 50089
          RETURN -999
   END

   IF (@RangeBeginningTime IS NOT NULL)
   BEGIN
```

```
    EXECUTE @retStat = ProcCheckDsTMdTime @RangeBeginningTime, @OUTPUT_DsTMdTime
OUTPUT
     IF (@retStat != 0)
     BEGIN
       RAISERROR 40374
       RETURN -999
     END
   SELECT @RangeBeginningTime = @OUTPUT_DsTMdTime
   END

   IF (@RangeEndingTime IS NOT NULL)
   BEGIN
    EXECUTE @retStat = ProcCheckDsTMdTime @RangeEndingTime, @OUTPUT_DsTMdTime
OUTPUT
     IF (@retStat != 0)
     BEGIN
       RAISERROR 40375
       RETURN -999
     END
   SELECT @RangeEndingTime = @OUTPUT_DsTMdTime
   END

   IF (@TimeOfDay IS NOT NULL)
   BEGIN
    EXECUTE @retStat = ProcCheckDsTMdTime @TimeOfDay , @OUTPUT_DsTMdTime OUTPUT
     IF (@retStat != 0)
     BEGIN
       RAISERROR 20014 "Procedure ProcCheckDsTMdTime error - check @TimeOfDay . "
       RETURN -999
     END
   SELECT @TimeOfDay = @OUTPUT_DsTMdTime
   END

   INSERT DsMdGranules(dbID,
                       VersionID,
                primaryCollectionId,
                ShortName,
                processingHistoryId,
                processingHistoryTypeCode,
                SizeMBECSDataGranule,
                PGEVersion,
                ReprocessingActual,
                ReprocessingPlanned,
                RangeBeginningDate,
                RangeBeginningTime,
                RangeEndingDate,
                RangeEndingTime,
                CalendarDate,
                TimeOfDay,
                lastUpdate,
                BeginningDateTime,
                EndingDateTime,
                insertTime,
                ZoneIdentifier,
```

```
                LocalGranuleID,
                DayNightFlag,
                ProductionDateTime,
                LocalVersionID,
                deleteEffectiveDate)
            VALUES(@dbID,
                  @VersionID,
                @primaryCollectionId,
                @ShortName,
                @processingHistoryId,
                @processingHistoryTypeCode,
                @SizeMBECSDataGranule,
                @PGEVersion,
                @ReprocessingActual,
                @ReprocessingPlanned,
                @RangeBeginningDate,
                @RangeBeginningTime,
                @RangeEndingDate,
                @RangeEndingTime,
                @CalendarDate,
                @TimeOfDay,
                getdate(),
                NULL,
                NULL,
                getdate(),
                @ZoneIdentifier,
                @LocalGranuleID,
                @DayNightFlag,
                @ProductionDateTime,
                @LocalVersionID,
                NULL)


   IF @@error!=0
   BEGIN
         RAISERROR 50090
         RETURN -999
   END

END
go
```

ProcInsertGranuleSensorXref**Procedure ProcInsertGranuleSensorXref**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertGranuleSensorXref            */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                Granule and Sensor Xref data.             */
/*  Calls         :  None                                   */
/*================================================================*/
```

```
CREATE PROCEDURE ProcInsertGranuleSensorXref
            (@granuleId        ID = 0,
             @sensorId         ID = 0)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40123
    RETURN -999
  END

  IF (@sensorId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40123
    RETURN -999
  END

-- added to handle multiple associated platform instrument sensor containers
-- per granule

  IF EXISTS (
              SELECT 1 FROM DsMdGranuleSensorXref
                    WHERE granuleId = @granuleId
              AND sensorId = @sensorId
          )
    BEGIN
         RETURN
    END
  ELSE
  BEGIN
   INSERT DsMdGranuleSensorXref(granuleId,
                        sensorId)
                   VALUES(@granuleId,
                        @sensorId)

   IF @@error!=0
   BEGIN
      RAISERROR 50006 "ProcInsertGranuleSensorXref: Failed Insert on DsMdGranuleSensorXref"
      RETURN -999
   END

  END

END
go
```

ProcInsertGranuleVersions**Procedure ProcInsertGranuleVersions**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGranuleVersions              */
/*  Release       :  B0                                  */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   Granule Version data.                  */
/*  Calls         :  None                                */
/*==============================================================*/
CREATE PROCEDURE ProcInsertGranuleVersions
             (@currentGranuleId   ID = 0,
              @currentVersion     char(16),
              @previousGranuleId  ID,
              @previousVersion    char(16))
AS
BEGIN
  IF (@currentGranuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40186
    RETURN -999
  END

  INSERT DsMdGranuleVersions(currentGranuleId,
                        currentVersion,
                        previousGranuleId,
                        previousVersion)
                   VALUES(@currentGranuleId,
                        @currentVersion,
                        @previousGranuleId,
                        @previousVersion)

   IF @@error!=0
   BEGIN
      RAISERROR 50088
      RETURN -999
   END

END
go
```

ProcInsertGrFloatInfoContent**Procedure ProcInsertGrFloatInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGrFloatInfoContent          */
/*  Release       :  B0                                    */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                   Granule level Additional Attribute      */
/*                   float name/value pairs.                 */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcInsertGrFloatInfoContent
              (@granuleId              ID = 0,
               @attributeId              int = 0,
               @grFloatValue           varchar(255))
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40174
    RETURN -999
  END

  IF (@attributeId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40174
    RETURN -999
  END


  IF exists(SELECT dbID FROM DsMdGranules WHERE dbID = @granuleId)
   BEGIN
    IF exists(SELECT attributeId FROM DsMdAdditionalAttributes WHERE attributeId = @attributeId)
     BEGIN
          INSERT DsMdGrFloatInfoContent(granuleId,
                          attributeId,
                              grFloatValue)
                    VALUES(@granuleId,
                              @attributeId,
                              CONVERT(float, @grFloatValue))
       IF @@error!=0
       BEGIN
         RAISERROR 50072
         RETURN -999
       END
     END
```

```
      ELSE
       BEGIN
         /* Mimic Sybase error. */
         RAISERROR 50074
         RETURN -999
       END
   END
  ELSE
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 50073
     RETURN -999
   END
END
go
```

ProcInsertGrIntInfoContent**Procedure ProcInsertGrIntInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGrIntInfoContent            */
/*  Release       :  B0                                    */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                   Granule level Additional Attribute      */
/*                   integer name/value pairs.               */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcInsertGrIntInfoContent
              (@granuleId              ID = 0,
               @attributeId            int = 0,
               @grIntValue             varchar(255))
AS
BEGIN

   IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40175
     RETURN -999
   END
   IF (@attributeId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40175
     RETURN -999
   END


   IF exists(SELECT dbID FROM DsMdGranules WHERE dbID = @granuleId)
    BEGIN
```

```
    IF exists(SELECT attributeId FROM DsMdAdditionalAttributes WHERE attributeId = @attributeId)
     BEGIN
      INSERT DsMdGrIntegerInfoContent(granuleId,
                                      attributeId,
                                      grIntValue)
                              VALUES(@granuleId,
                                      @attributeId,
                                      CONVERT(int, @grIntValue))
       IF @@error!=0
       BEGIN
         RAISERROR 50075
         RETURN -999
       END
     END
    ELSE
     BEGIN
       /* Mimic Sybase error. */
       RAISERROR 50077
       RETURN -999
     END
   END
  ELSE
   BEGIN
    /* Mimic Sybase error. */
    RAISERROR 50076
    RETURN -999
   END
END
go
```

ProcInsertGrProdSpecific**Procedure ProcInsertGrProdSpecific**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGrProdSpecific                   */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  3/31/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                   parameters into the appropriate            */
/*                   content tables based on an Attribute's     */
/*                   datatype.                                  */
/*  Calls         :  ProcInsertGrFloatInfoContent               */
/*                   ProcInsertGrIntInfoContent                 */
/*                   ProcInsertGrStringInfoContent              */
/*==============================================================*/
CREATE PROC ProcInsertGrProdSpecific
          (@granuleId              ID                = 0,
           @grAttributeName        varchar(40)       = "",
           @grAttributeValue       varchar(255)      = "")
AS
BEGIN
```

```
  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40176
     RETURN -999
   END

   DECLARE @tname        varchar(32)
   DECLARE @error        varchar(255)
   DECLARE @lookupValue  varchar(255)
   DECLARE @ProcToExec   varchar(255)
   DECLARE @attributeId  int
   DECLARE @retStat      int

   SELECT        @tname = a.grTableName,
                 @attributeId = b.attributeId
   FROM          DsMdParmToTableVector a,
                 DsMdAdditionalAttributes b
   WHERE         a.DataType = b.AdditionalAttributeDataType AND
                 b.AdditionalAttributeName = @grAttributeName

   IF ( @@error!=0 OR @@rowcount != 1)
     BEGIN
       RAISERROR 50078
           RETURN -999
     END

   EXECUTE @retStat = ProcGetSDSRVLookup "ProcInsertGrProdSpecific", @tname, "varchar",
@lookupValue OUTPUT

  IF @retStat != 0
    BEGIN
      RAISERROR 20014 "Procedure ProcInsertGrProdSpecific error - could not find parameter
@ProcToExec ."
      RETURN -999
    END

  SELECT @ProcToExec = @lookupValue

-- This is an exec of the stored proc name found in the prior call
  EXECUTE @retStat = @ProcToExec @granuleId, @attributeId, @grAttributeValue

  IF @retStat != 0
    BEGIN
      SELECT @error =  "Procedure ProcInsertGrProdSpecific error - failure to execute proc = " +
CONVERT(varchar, @ProcToExec)
      RAISERROR 20014 @error
      RETURN -999
    END

END
go
```

ProcInsertGrSensCharactristic**Procedure ProcInsertGrSensCharactristic**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGrSensCharacteristic         */
/*  Release       :  B0                                     */
/*  Module        :  Data Originator                        */
/*  Created on    :  4/03/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   granule sensor characteristic data.    */
/*  Calls         :  None                                   */
/*==============================================================*/
CREATE PROCEDURE ProcInsertGrSensCharactristic
(@granuleId                         ID          = 0,
 @SensorCharacteristicName                      varchar(40) = NULL,
 @SensorCharacteristicValue             varchar(80))
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40177
    RETURN -999
  END

  IF (@SensorCharacteristicName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40177
    RETURN -999
  END


  INSERT  DsMdGrSensorCharacteristics
        (granuleId,
            SensorCharacteristicName,
            SensorCharacteristicValue)
  VALUES (@granuleId,
            @SensorCharacteristicName,
            @SensorCharacteristicValue)

  IF @@error!=0
    BEGIN
      RAISERROR 50079 "ProcInsertGrSensCharactristic: Failed Insert on
DsMdGrSensorCharacteristics"
          RETURN -999
    END
END
go
```

ProcInsertGrStorageMedium**Procedure ProcInsertGrStorageMedium**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGrStorageMedium              */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  4/03/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   granule storage medium data.           */
/*  Calls         :  None                                   */
/*==============================================================*/


CREATE PROCEDURE ProcInsertGrStorageMedium
    (@granuleId        ID      = 0,
     @StorageMedium    varchar(30) = NULL)
AS
BEGIN
  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40178
     RETURN -999
   END

  IF (@StorageMedium = NULL)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40178
     RETURN -999
   END


   INSERT  DsMdGrStorageMedium(granuleId,
                               StorageMedium)
   VALUES  (@granuleId,
            @StorageMedium)

   IF @@error!=0
   BEGIN
     RAISERROR 50080
     RETURN -999
   END
END
go
```

ProcInsertGrStringInfoContent**Procedure ProcInsertGrStringInfoContent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertGrStringInfoContent          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  4/23/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   Granule level Additional Attribute     */
/*                   string name/value pairs.               */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROCEDURE ProcInsertGrStringInfoContent
              (@granuleId              ID = 0,
               @attributeId           int = 0,
               @grStringValue          varchar(255))
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40124
    RETURN -999
  END

  IF (@attributeId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40124
    RETURN -999
  END

  IF exists(SELECT dbID FROM DsMdGranules WHERE dbID = @granuleId)
   BEGIN
    IF exists(SELECT attributeId FROM DsMdAdditionalAttributes WHERE attributeId = @attributeId)
     BEGIN
          INSERT DsMdGrStringInfoContent(granuleId,
                          attributeId,
                              grStringValue)
                    VALUES(@granuleId,
                              @attributeId,
                              @grStringValue)
       IF @@error!=0
       BEGIN
         RAISERROR 50007
         RETURN -999
       END
     END
    ELSE
```

```
      BEGIN
        /* Mimic Sybase error. */
        RAISERROR 50008
        RETURN -999
      END
    END
   ELSE
    BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50009
      RETURN -999
    END
END
go
```

ProcInsertInputGranule**Procedure ProcInsertInputGranule**

Code

```
/*================================================================*/
/*  Name           :  ProcInsertInputGranule                      */
/*  Release        :  B0                                          */
/*  Module         :  Granules                                    */
/*  Created on     :  3/31/97                                     */
/*  Purpose        :  Create stored procedure to insert           */
/*                    Input Granule data.                         */
/*  Calls          :  None                                        */
/*================================================================*/
CREATE PROCEDURE ProcInsertInputGranule
              (
                    @granuleId ID = 0,
               @InputPointer  varchar(255) = NULL
                 )
-- @inputGranule ID = 0,
AS
BEGIN

    DECLARE @inputGranule ID
    SELECT @inputGranule = -1
-- can't be NULL because its in the PK

    DECLARE @inputGranuleFileCount smallint
    SELECT @inputGranuleFileCount = 1

   IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40188
     RETURN -999
   END

   IF ( @InputPointer IS NULL OR @InputPointer = "" )
   BEGIN
     /* Mimic Sybase error. */
```

```
    RAISERROR 24188
    RETURN -999
  END

--  IF (@inputGranule = 0)
--  BEGIN
--    /* Mimic Sybase error. */
--    RAISERROR 20024 "Procedure ProcInsertInputGranule expects parameter @inputGranule, which
was not supplied."
--    RETURN -999
--  END

  IF (@InputPointer = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 20025 "Procedure ProcInsertInputGranule expects parameter @InputPointer, which was
not supplied."
    RETURN -999
  END

  IF NOT EXISTS (
            SELECT 1
            FROM DsMdInputGranule
            WHERE granuleId = @granuleId
            AND   InputPointer = @InputPointer
            )
  BEGIN
   INSERT DsMdInputGranule(
                    granuleId,
                    inputGranule,
                    InputPointer,
                    inputGranuleFileCount,
                    insertDatetime
                   )
               VALUES(
                    @granuleId,
                    @inputGranule,
                    @InputPointer,
                    @inputGranuleFileCount,
                    getdate()
                   )
  END
  ELSE
-- for multi-file input granules, bump up the count
  BEGIN
   UPDATE DsMdInputGranule
      SET inputGranuleFileCount = ISNULL(inputGranuleFileCount, 1) + 1
      WHERE granuleId = @granuleId
      AND    InputPointer = @InputPointer
  END

  IF @@error!=0
   BEGIN
      RAISERROR 50091
```

```
      RETURN -999
   END

END
go
```

ProcInsertInstrument**Procedure ProcInsertInstrument**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertInstrument                     */
/*  Release         :  B0                                       */
/*  Module          :  DataOriginator                           */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                     data into DsMdInstrument                 */
/*                     table                                    */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertInstrument
                  (
            @dbID                ID      = 0,
            @platformId          ID              = 0,
            @InstrumentShortName   varchar(20) = NULL,
            @InstrumentLongName    varchar(80) = NULL,
            @InstrumentTechnique   varchar(80) = NULL,
            @NumberOfSensors       int         = NULL
                  )
AS
BEGIN
   IF (@dbID = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40189
      RETURN -999
   END

   IF (@platformId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40189
      RETURN -999
   END

   IF (@InstrumentShortName = NULL OR @InstrumentShortName = "")
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40189
      RETURN -999
```

```
   END


  INSERT DsMdInstrument(dbID,
                platformId,
                InstrumentShortName,
                InstrumentLongName,
                InstrumentTechnique,
                NumberOfSensors)
  VALUES(@dbID,
       @platformId,
       @InstrumentShortName,
       @InstrumentLongName,
       @InstrumentTechnique,
       @NumberOfSensors)

  IF @@error!=0
  BEGIN
    RAISERROR 50092
    RETURN -999
  END
END
go
```

ProcInsertInstrumentChar**Procedure ProcInsertInstrumentChar**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertInstrumentChar             */
/*  Release         :  B0                                   */
/*  Module          :  DataOriginator                       */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for inserting  */
/*                  data into DsMdInstrumentCharacteristic */
/*                  table                                   */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROC  ProcInsertInstrumentChar(
   @instrumentId                    ID              = 0,
   @InstrumentCharacteristicName      varchar(40) = null,
   @InstrumentCharacteristicDesc      varchar(80),
   @InstrumentCharacteristicUnit      varchar(20),
   @InstrumentCharDataType            char(8),
   @InstrumentCharacteristicValue     varchar(15) = null)
AS
BEGIN
   IF (@instrumentId = 0)
   BEGIN
      /* Mimic Sybase error. */
```

```
    RAISERROR 40190
    RETURN -999
  END

  IF (@InstrumentCharacteristicName = null)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40190
    RETURN -999
  END

  IF (@InstrumentCharacteristicValue = null)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40190
    RETURN -999
  END


  INSERT  DsMdInstrumentCharacteristic(
          instrumentId,
          InstrumentCharacteristicName,
          InstrumentCharacteristicDesc,
          InstrumentCharacteristicUnit,
          InstrumentCharDataType,
          InstrumentCharacteristicValue)
  VALUES(@instrumentId,
       @InstrumentCharacteristicName,
       @InstrumentCharacteristicDesc,
       @InstrumentCharacteristicUnit,
       @InstrumentCharDataType,
       @InstrumentCharacteristicValue)

  IF @@error!=0
  BEGIN
       RAISERROR 50093
       RETURN -999
  END
END
go
```

ProcInsertLocality**Procedure ProcInsertLocality**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertLocality                         */
/*  Release       :  B0                                         */
/*  Module        :  Collections                                */
/*  Created on    :  3/25/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                locality data for Collections.                */
/*  Calls         :                                             */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertLocality
(@collectionId          ID = 0,
 @LocalityType        varchar(20) = null,
 @LocalityDescription  varchar(255))
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40191
      RETURN -999
   END

   IF (@LocalityType = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40191
      RETURN -999
   END


   IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
      INSERT  DsMdLocality(collectionId,
                        LocalityType,
                        LocalityDescription)
           VALUES (@collectionId,
                     @LocalityType,
             @LocalityDescription)

           IF @@error!=0
      BEGIN
         RAISERROR 50094
            RETURN -999
      END
   END
   ELSE
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50095
      RETURN -999
   END
END
go
```

ProcInsertMeasuredParameter**Procedure ProcInsertMeasuredParameter**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertMeasuredParameter           */
/*  Release       :  B0                                    */
/*  Module        :  Granules                              */
/*  Created on    :  3/31/97                               */
/*  Purpose       :  Create stored procedure to insert     */
/*                   Measured Parameter data.              */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROCEDURE ProcInsertMeasuredParameter
            (@granuleId                    ID = 0,
             @ParameterName                varchar(40) = NULL,
             @QAPercentMissingData         int,
             @QAPercentOutOfBoundsData     int,
             @QAPercentInterpolatedData    int,
             @QAPercentCloudCover          int,
             @AutomaticQualityFlag         varchar(64),
             @AutomaticQualityFlagExplan   varchar(255),
             @OperationalQualityFlag       varchar(20),
             @OperationalQualityFlagExplan varchar(255),
             @ScienceQualityFlag           varchar(25),
             @ScienceQualityFlagExplan     varchar(255))
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40192
    RETURN -999
  END

  IF (@ParameterName = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40192
    RETURN -999
  END



   INSERT DsMdMeasuredParameter(granuleId,
                        ParameterName,
                        QAPercentMissingData,
                        QAPercentOutOfBoundsData,
                        QAPercentInterpolatedData,
                        QAPercentCloudCover,
                        AutomaticQualityFlag,
```

```
                    AutomaticQualityFlagExplan,
                    OperationalQualityFlag,
                    OperationalQualityFlagExplan,
                    ScienceQualityFlag,
                    ScienceQualityFlagExplan)
                VALUES(@granuleId,
                    @ParameterName,
                    @QAPercentMissingData,
                    @QAPercentOutOfBoundsData,
                    @QAPercentInterpolatedData,
                    @QAPercentCloudCover,
                    @AutomaticQualityFlag,
                    @AutomaticQualityFlagExplan,
                    @OperationalQualityFlag,
                    @OperationalQualityFlagExplan,
                    @ScienceQualityFlag,
                    @ScienceQualityFlagExplan)

  IF @@error!=0
   BEGIN
      RAISERROR 50096
      RETURN -999
   END

END
go
```

ProcInsertMultiDateTime**Procedure ProcInsertMultiDateTime**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertMultiDateTime              */
/*  Release         :  B0                                   */
/*  Module          :  Temporal                             */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdMultipleDateTimePeriod  */
/*                  table                                   */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROCEDURE ProcInsertMultiDateTime(
            @collectionId                   ID = 0,
            @MultipleDateName      varchar(30) = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40194
```

```
      RETURN -999
    END

   IF (@MultipleDateName = null)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40194
     RETURN -999
   END


    INSERT  DsMdMultipleDateTimePeriod(collectionId, MultipleDateName)
    VALUES( @collectionId,
          @MultipleDateName )

    IF @@error!=0
    BEGIN
          RAISERROR 50099
          RETURN -999
    END
END
go
```

ProcInsertMultTypeCollection**Procedure ProcInsertMultTypeCollection**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertMultTypeCollection              */
/*  Release        :  B0                                        */
/*  Module         :  Collections                               */
/*  Created on     :  3/25/97                                   */
/*  Purpose        :  Create stored procedure to insert         */
/*                    multiple type collection data for         */
/*                    Collections.                              */
/*  Calls          :                                            */
/*==============================================================*/


CREATE PROCEDURE ProcInsertMultTypeCollection
(@collectionId              ID = 0,
 @AggregationRelationship  varchar(2),
 @AggregationType          varchar(20),
 @AggregationValue         varchar(80))
AS
BEGIN
    IF (@collectionId = 0)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40193
       RETURN -999
    END
    ELSE
    BEGIN
```

```
      IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
      BEGIN
            INSERT  DsMdMultipleTypeCollection(collectionId,
                                 AggregationRelationship,
                                 AggregationType,
                                 AggregationValue)
            VALUES (@collectionId,
                     @AggregationRelationship,
              @AggregationType,
              @AggregationValue)

            IF @@error!=0
        BEGIN
           RAISERROR 50097
              RETURN -999
        END
      END
      ELSE
      BEGIN
        /* Mimic Sybase error. */
        RAISERROR 50098
        RETURN -999
      END
   END
END
go
```

ProcInsertOperationMode**Procedure ProcInsertOperationMode**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertOperationMode                    */
/*  Release       :  B0                                         */
/*  Module        :  Data Originator                            */
/*  Created on    :  4/03/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                   operation mode data.                       */
/*  Calls         :  None                                       */
/*==============================================================*/


CREATE PROCEDURE ProcInsertOperationMode
            (@instrumentId          ID  = 0,
             @OperationMode  varchar(20) = null)
AS
BEGIN
   IF (@instrumentId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40195
     RETURN -999
   END
```

```
  IF (@OperationMode = null)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40195
    RETURN -999
  END


   INSERT DsMdOperationMode(instrumentId,
                    OperationMode)
   VALUES(@instrumentId,
         @OperationMode)

   IF @@error!=0
   BEGIN
      RAISERROR 50100
      RETURN -999
   END
END
go
```

ProcInsertOrbitCalcSpatial**Procedure ProcInsertOrbitCalcSpatial**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertOrbitCalcSpatial             */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                Orbit Calculated Spatial data.          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcInsertOrbitCalcSpatial
            (@granuleId                ID = 0,
             @roundedEqCrossingLongitude  float,
             @platInstCode              int)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40196
    RETURN -999
  END

  INSERT DsMdOrbitCalculatedSpatial(granuleId,
                        roundedEqCrossingLongitude,
                        platInstCode)
                    VALUES(@granuleId,
                        @roundedEqCrossingLongitude,
```

```
                        @platInstCode)
   IF @@error!=0
   BEGIN
         RAISERROR 50101
         RETURN -999
   END
END
go
```

ProcInsertOrbitCalcSpatialDom**Procedure ProcInsertOrbitCalcSpatialDom**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertOrbitCalcSpatialDom          */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  04/17/97                               */
/*  Purpose       :  Create stored procedure to insert      */
/*                Orbit Calculated Spatial Domain data.   */
/*  Calls         :  ProcCheckDsTMdTime                     */
/*==============================================================*/

CREATE PROC ProcInsertOrbitCalcSpatialDom
            (@granuleId                ID = 0,
             @OrbitModelName           varchar(80),
             @OrbitNumber              int,
             @StartOrbitNumber         int,
             @StopOrbitNumber          int,
             @EquatorCrossingLongitude    float = null,
             @EquatorCrossingDate      datetime,
             @EquatorCrossingTime      DsTMdTime)
AS
BEGIN
   DECLARE @OUTPUT_DsTMdTime    DsTMdTime

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40197
    RETURN -999
  END

  IF (@EquatorCrossingLongitude = null)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40197
    RETURN -999
  END

   DECLARE @retStat   int

   IF (@EquatorCrossingTime IS NOT NULL)
   BEGIN
```

```
    EXECUTE @retStat = ProcCheckDsTMdTime @EquatorCrossingTime, @OUTPUT_DsTMdTime
OUTPUT
     IF (@retStat != 0)
     BEGIN
       RAISERROR 40377
       RETURN -999
     END
   SELECT @EquatorCrossingTime = @OUTPUT_DsTMdTime
   END

  INSERT DsMdOrbitCalcSpatialDomain(granuleId,
                              EquatorCrossingLongitude,
                              OrbitModelName,
                              OrbitNumber,
                              StartOrbitNumber,
                              StopOrbitNumber,
                              EquatorCrossingDate,
                              EquatorCrossingTime)
  VALUES(@granuleId,
        @EquatorCrossingLongitude,
        @OrbitModelName,
        @OrbitNumber,
        @StartOrbitNumber,
        @StopOrbitNumber,
        @EquatorCrossingDate,
        @EquatorCrossingTime)
   IF @@error!=0
   BEGIN
          RAISERROR 50102
          RETURN -999
   END
END
go
```

ProcInsertOrbitParmGranule**Procedure ProcInsertOrbitParmGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertOrbitParmGranule             */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   Orbit Parameter data.                  */
/*  Calls         :  None                                   */
/*==============================================================*/
CREATE PROCEDURE ProcInsertOrbitParmGranule
                (@dbID                    ID = 0,
              @OrbitParametersPointer    varchar(255))
AS
BEGIN

  IF (@dbID = 0)
```

```
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40198
    RETURN -999
  END

   INSERT DsMdOrbitParametersGranule(dbID,
                                OrbitParametersPointer,
                                lastUpdate,
                                subType,
                                insertTime,
                                deleteEffectiveDate)
                           VALUES(@dbID,
                                @OrbitParametersPointer,
                                getdate(),
                                NULL,
                                getdate(),
                                NULL)
    IF @@error!=0
          BEGIN
             RAISERROR 50103
          RETURN -999
          END
END
go
```

ProcInsertParmToTableVector**Procedure ProcInsertParmToTableVector**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertParmToTableVector              */
/*  Release       :  B0                                       */
/*  Module        :  Collections                              */
/*  Created on    :  3/25/97                                  */
/*  Purpose       :  Create stored procedure to insert        */
/*                parameter (datatype) to table data for     */
/*                Collections.                                */
/*  Calls         :                                           */
/*================================================================*/


CREATE PROCEDURE ProcInsertParmToTableVector
(@DataType     char(10) = null,
 @grTableName  varchar(32),
 @clTableName  varchar(32))
AS
BEGIN
   IF (@DataType = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40200
      RETURN -999
   END
```

```
  INSERT  DsMdParmToTableVector(DataType,
                          grTableName,
                          clTableName)
  VALUES (@DataType,
            @grTableName,
          @clTableName)

  IF @@error!=0
    BEGIN
       RAISERROR 50105
       RETURN -999
    END
  END
go
```

ProcInsertPersonOrgXref**Procedure ProcInsertPersonOrgXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertPersonOrgXref                    */
/*  Release       :  B0                                         */
/*  Module        :  Contact                                    */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  Create stored procedure to Contact,        */
/*                   and associate data tables.                 */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertPersonOrgXref
          ( @personId             ID = 0,
            @organizationId       ID = 0)
AS
BEGIN
   IF (@personId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40201
      RETURN -999
   END

   IF (@organizationId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40201
      RETURN -999
   END


         INSERT DsMdPersonOrganizationXref(personId,
```

```
                                        organizationId)
        VALUES(@personId,
             @organizationId)
        IF @@error!=0
          BEGIN
        RAISERROR 50106
                RETURN -999
          END
END
go
```

ProcInsertPGEGroup**Procedure ProcInsertPGEGroup**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertPGEGroup                         */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to Delivered,      */
/*                Algorithm Package, and associate data         */
/*                tables.                                       */
/*  Calls         :                                             */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertPGEGroup(
            @dapDBId                    ID     = 0,
            @DAPPGEName                 varchar(20) = null,
            @DAPPGEVersion              char(10) = null,
                @DAPSWVersion               varchar(12) = null)
AS
BEGIN
   IF (@dapDBId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40199
      RETURN -999
   END

   IF (@DAPPGEName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40199
      RETURN -999
   END

   IF (@DAPPGEVersion = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40199
      RETURN -999
```

```
   END

   IF (@DAPSWVersion = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40199
      RETURN -999
   END

   INSERT  DsMdPGEGroup (dapDBId,
                  DAPPGEName,
                  DAPPGEVersion,
                  DAPSWVersion )
   VALUES( @dapDBId,
         @DAPPGEName,
         @DAPPGEVersion,
            @DAPSWVersion )

   IF @@error!=0
   BEGIN
         RAISERROR 50104
         RETURN -999
   END
END
go
```

ProcInsertPhysicalParameters**Procedure ProcInsertPhysicalParameters**

Code

```
/*============================================================*/
/*  Name          :  ProcInsertPhysicalParameters          */
/*  Release       :  B0                                    */
/*  Module        :  Collections                           */
/*  Created on    :  3/25/97                               */
/*  Purpose       :  Create stored procedure to insert     */
/*                   additional attribute data for         */
/*                   Collections.                          */
/*  Calls         :  ProcInsertAdditionalAttributes        */
/*                   ProcInsertClProdSpecific              */
/*============================================================*/

CREATE PROCEDURE ProcInsertPhysicalParameters
(@collectionId                  ID,
 @AdditionalAttributeName       varchar(30),
 @AdditionalAttributeDesc       varchar(255),
 @AdditionalAttributeDatatype   char(10),
 @ParameterUnitsOfMeasure       varchar(20),
 @ParameterRangeBegin           varchar(40),
 @ParameterRangeEnd                                  varchar(40),
 @ParameterValueAccuracy        varchar(30),
 @ValueAccuracyExplanation      varchar(255),
 @MeasurementResolution         varchar(30),
 @clParameterValue              varchar(200))
```

```
AS
  DECLARE @retStat  int
  DECLARE @attributeId  int

  EXECUTE @retStat = ProcInsertAdditionalAttributes @AdditionalAttributeName,
                     @AdditionalAttributeDesc,
                     @AdditionalAttributeDatatype,
                     @ParameterUnitsOfMeasure,
                     @ParameterRangeBegin,
                                                       @ParameterRangeEnd,
                     @ParameterValueAccuracy,
                     @ValueAccuracyExplanation,
                     @MeasurementResolution,
                           @attributeId

  IF @retStat != 0
  BEGIN
    RAISERROR 56073
    RETURN -999
  END

  EXECUTE @retStat = ProcInsertClProdSpecific @collectionId,
                                  @AdditionalAttributeName,
                                  @clParameterValue

  IF @retStat != 0
  BEGIN
    RAISERROR 56074
    RETURN -999
  END
go
```

ProcInsertPlanarCoordSys**Procedure ProcInsertPlanarCoordSys**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertPlanarCoordSys             */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                               */
/*  Created on      :  3/25/97                                 */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdPlanarCoordinateSystems */
/*                  table                                 */
/*  Calls           :                                       */
/*==============================================================*/


CREATE PROC ProcInsertPlanarCoordSys (
        @collectionId                ID = 0,
        @PlanarDistanceUnits          varchar(80),
        @PlanarCoordinateEncodingMeth    varchar(80),
        @DistanceResolution           float,
        @BearingResolution            float,
```

```
        @BearingUnits                varchar(255),
        @BearingReferenceDirection   varchar(20),
        @BearingReferenceMeridian    varchar(255),
        @AbscissaResolution          float,
        @OrdinateResolution          float,
        @MapProjectionName           varchar(80),
        @MapProjectionPointer        varchar(255),
        @LocalPlanarCoordinateSysDes varchar(255),
        @LocalPlanarGeoreferenceInfo varchar(255),
        @GridCoordinateSystemName    varchar(80))
AS
BEGIN

   DECLARE @retStat int
   DECLARE @planarCoordinateSystemsID int

   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40135
      RETURN -999
   END

   EXECUTE @retStat = ProcGetUniqueIDOutput "int", "DsMdPlanarCoordinateSystems",
@planarCoordinateSystemsID OUTPUT

   IF ( @retStat != 0  )
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertPlanarCoordSys error  on  call to ProcGetUniqueID."
      RETURN -999
   END

   INSERT  DsMdPlanarCoordinateSystems (collectionId,
                        planarCoordinateSystemsID,
                              PlanarDistanceUnits,
                        PlanarCoordinateEncodingMeth,
                        DistanceResolution,
                        BearingResolution,
                        BearingUnits,
                        BearingReferenceDirection,
                        BearingReferenceMeridian,
                        AbscissaResolution,
                        OrdinateResolution,
                        MapProjectionName,
                        MapProjectionPointer,
                        LocalPlanarCoordinateSystemDes,
                        LocalPlanarGeoreferenceInforma,
                        GridCoordinateSystemName)
   VALUES( @collectionId,
        @planarCoordinateSystemsID,
        @PlanarDistanceUnits,
        @PlanarCoordinateEncodingMeth,
        @DistanceResolution,
```

```
        @BearingResolution,
        @BearingUnits,
        @BearingReferenceDirection,
        @BearingReferenceMeridian,
        @AbscissaResolution,
        @OrdinateResolution,
        @MapProjectionName,
        @MapProjectionPointer,
        @LocalPlanarCoordinateSysDes,
        @LocalPlanarGeoreferenceInfo,
        @GridCoordinateSystemName )

   IF @@error!=0
   BEGIN
      RAISERROR 50026
      RETURN -999
   END
END
go
```

ProcInsertPlatform**Procedure ProcInsertPlatform**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertPlatform                       */
/*  Release         :  B0                                       */
/*  Module          :  DataOriginator                           */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                     data into DsMdPlatform                   */
/*                     table                                    */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/


CREATE PROC  ProcInsertPlatform(
   @dbID               ID        = 0,
   @PlatformShortName    varchar(20) = null,
   @PlatformLongName     varchar(80) = null,
   @PlatformType         varchar(20) = null)
AS
BEGIN
   IF (@dbID = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40202
      RETURN -999
   END

   IF (@PlatformShortName = null)
   BEGIN
```

```
      /* Mimic Sybase error. */
      RAISERROR 40202
      RETURN -999
   END

   IF (@PlatformLongName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40202
      RETURN -999
   END

   IF (@PlatformType = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40202
      RETURN -999
   END


   INSERT  DsMdPlatform(dbID, PlatformShortName, PlatformLongName, PlatformType)
   VALUES(@dbID,
         @PlatformShortName,
         @PlatformLongName,
         @PlatformType)

   IF @@error!=0
   BEGIN
          RAISERROR 50108
          RETURN -999
   END
END
go
```

ProcInsertPlatformChar**Procedure ProcInsertPlatformChar**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertPlatformChar               */
/*  Release         :  B0                                   */
/*  Module          :  DataOriginator                       */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdPlatformCharacteristic  */
/*                  table                                   */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROC  ProcInsertPlatformChar(
   @platformId                    ID              = 0,
```

```
    @PlatformCharacteristicName        varchar(40) = null,
    @PlatformCharacteristicDesc        varchar(80) = null,
    @PlatformCharacteristicUnit        varchar(20) = null,
    @PlatformCharDataType              char(8) = null,
    @PlatformCharacteristicValue       varchar(20) = null)
AS
BEGIN
    IF (@platformId = 0)
    BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40203
        RETURN -999
    END

    IF (@PlatformCharacteristicName = null)
    BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40203
        RETURN -999
    END

    IF (@PlatformCharacteristicDesc = null)
    BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40203
        RETURN -999
    END

    IF (@PlatformCharacteristicUnit = null)
    BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40203
        RETURN -999
    END

    IF (@PlatformCharDataType = null)
    BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40203
        RETURN -999
    END

    IF (@PlatformCharacteristicValue = null)
    BEGIN
        /* Mimic Sybase error. */
        RAISERROR 40203
        RETURN -999
    END


    INSERT  DsMdPlatformCharacteristic(
            platformId,
            PlatformCharacteristicName,
            PlatformCharacteristicDesc,
```

```
            PlatformCharacteristicUnit,
            PlatformCharacteristicDataType,
            PlatformCharacteristicValue)
    VALUES(@platformId,
          @PlatformCharacteristicName,
          @PlatformCharacteristicDesc,
          @PlatformCharacteristicUnit,
          @PlatformCharDataType,
          @PlatformCharacteristicValue)

    IF @@error!=0
    BEGIN
          RAISERROR 50109
          RETURN -999
    END
END
go
```

ProcInsertPlatInstCode**Procedure ProcInsertPlatInstCode**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertPlatInstCode              */
/*  Release         :  B0                                  */
/*  Module          :  Spatial                              */
/*  Created on      :  3/26/97                                */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                     data into DsMdPlatInstrCode          */
/*                     table                                  */
/*                                                          */
/*  Calls           :  None                                 */
/*                                                          */
/*==============================================================*/


CREATE PROCEDURE ProcInsertPlatInstCode(
            @platformInstrumentCode    int = NULL,
            @platformShortName         varchar(20) = NULL,
                @instrumentName           varchar(80) = NULL,
                @resolution               float = NULL )
AS
BEGIN


    IF (@platformInstrumentCode IS NULL )
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 20029 "Procedure ProcInsertPlatInstCode expects parameter
@platformInstrumentCode, which was not supplied."
       RETURN -999
    END

    IF ( @platformShortName IS NULL OR @platformShortName  = "" )
```

```
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertPlatInstCode expects parameter @platformShortName,
which was not supplied."
      RETURN -999
   END

   IF ( @instrumentName IS NULL OR @instrumentName  = "" )
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertPlatInstCode expects parameter @instrumentName,
which was not supplied."
      RETURN -999
   END

   IF ( @resolution IS NULL )
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 20029 "Procedure ProcInsertPlatInstCode expects parameter @resolution, which was
not supplied."
      RETURN -999
   END

   INSERT DsMdPlatInstrCode(platformInstrumentCode,
                            platformShortName,
                            instrumentName,
                  resolution)
   VALUES(@platformInstrumentCode,
          @platformShortName,
          @instrumentName,
          @resolution)

   IF @@error!=0
   BEGIN
         RAISERROR 50107
         RETURN -999
   END

END
go
```

ProcInsertProcessingHistory**Procedure ProcInsertProcessingHistory**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertProcessingHistory            */
/*  Release       :  B0                                     */
/*  Module        :  Granules                               */
/*  Created on    :  3/31/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                Processing History data.                  */
/*  Calls         :  None                                   */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertProcessingHistory
            (@dbID                    ID = 0,
             @subType                 varchar(30),
             @ProductionHistoryPointer  varchar(255) = NULL)
AS
BEGIN

  IF (@dbID = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40204
    RETURN -999
  END

   INSERT DsMdProcessingHistory(dbID,
                        ProductionHistoryPointer,
                        lastUpdate,
                        subType,
                        insertTime,
                        deleteEffectiveDate)
                    VALUES(@dbID,
                        @ProductionHistoryPointer,
                        getdate(),
                        @subType,
                        getdate(),
                        NULL)
   IF @@error!=0
   BEGIN
      RAISERROR 50110
      RETURN -999
   END

END
go
```

ProcInsertProcessingQA**Procedure ProcInsertProcessingQA**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertProcessingQA                     */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  3/31/97                                    */
/*  Purpose       :  Create stored procedure to insert          */
/*                Granule Processing Quality Assurance          */
/*                data.                                         */
/*  Calls         :  None                                       */
/*==============================================================*/
CREATE PROCEDURE ProcInsertProcessingQA
            (@granuleId              ID = 0,
             @ProcessingQADescription  varchar(255),
             @ProcessingQAAttribute    varchar(80) = NULL)
```

```
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40205
    RETURN -999
  END


  IF (@ProcessingQAAttribute = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40205
    RETURN -999
  END


   INSERT DsMdProcessingQA(granuleId,
                    ProcessingQADescription,
                    ProcessingQAAttribute)
                VALUES(@granuleId,
                    @ProcessingQADescription,
                    @ProcessingQAAttribute)
   IF @@error!=0
   BEGIN
      RAISERROR 50111
      RETURN -999
   END

END
go
```

ProcInsertQaGranule**Procedure ProcInsertQaGranule**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertQaGranule                  */
/*  Release       :  B0                                 */
/*  Module        :  Granules                             */
/*  Created on    :  3/31/97                              */
/*  Purpose       :  Create stored procedure to insert       */
/*                Quality Assurance Granule data.         */
/*  Calls         :  None                                 */
/*================================================================*/
CREATE PROCEDURE ProcInsertQaGranule
            (@dbID                  ID = 0,
             @granuleId    ID = 0)
AS
BEGIN

  IF (@dbID = 0)
```

```
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40206
    RETURN -999
  END

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40206
    RETURN -999
  END

   INSERT DsMdQaGranule(dbID,
                granuleId,
                QAGranulePointer,
                lastUpdate,
                subType,
                insertTime,
                deleteEffectiveDate)
            VALUES(@dbID,
                @granuleId,
                NULL,
                getdate(),
                NULL,
                getdate(),
                NULL)
  IF @@error!=0
    BEGIN
        RAISERROR 50112
        RETURN -999
    END

END
go
```

ProcInsertQualityTextComment**Procedure ProcInsertQualityTextComment**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertQualityTextComment              */
/*  Release       :  B0                                        */
/*  Module        :  Collections                               */
/*  Created on    :  3/25/97                                   */
/*  Purpose       :  Create stored procedure to insert         */
/*                quality text comment data for Collections. */
/*  Calls         :                                            */
/*================================================================*/


CREATE PROCEDURE ProcInsertQualityTextComment
(@collectionId                ID             = 0,
 @QualityTextCommentPointer   varchar(255) = null)
```

```
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40207
      RETURN -999
   END

   IF (@QualityTextCommentPointer = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40207
      RETURN -999
   END

   IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
      INSERT  DsMdQualityTextComment(collectionId,
                              QualityTextCommentPointer)
          VALUES (@collectionId,
                    @QualityTextCommentPointer)

          IF @@error!=0
      BEGIN
         RAISERROR 50113
            RETURN -999
      END
   END
   ELSE
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50114
      RETURN -999
   END
END
go
```

ProcInsertRangeDateTime**Procedure ProcInsertRangeDateTime**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertRangeDateTime              */
/*  Release         :  B0                                   */
/*  Module          :  Temporal                             */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdRangeDateTime             */
/*                  table                                   */
/*                                                          */
/*  Calls           :  ProcCheckDsTMdTime                   */
/*                                                          */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertRangeDateTime(
            @collectionId                    ID              = 0,
            @RangeEndingTime        DsTMdTime        = null,
            @RangeEndingDate        datetime   = null,
            @RangeBeginningTime      DsTMdTime       = null,
            @RangeBeginningDate      datetime  = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40208
      RETURN -999
   END

   IF (@RangeEndingTime = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40208
      RETURN -999
   END

   IF (@RangeEndingDate = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40208
      RETURN -999
   END

   IF (@RangeBeginningTime = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40208
      RETURN -999
   END

  IF (@RangeBeginningDate = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40208
      RETURN -999
   END

   DECLARE @retStat   int
   DECLARE @OUTPUT_DsTMdTime    DsTMdTime

   IF (@RangeBeginningTime IS NOT NULL)
   BEGIN
    EXECUTE @retStat = ProcCheckDsTMdTime @RangeBeginningTime, @OUTPUT_DsTMdTime
OUTPUT
     IF (@retStat != 0)
     BEGIN
```

```
      RAISERROR 40374
      RETURN -999
    END
  SELECT @RangeBeginningTime = @OUTPUT_DsTMdTime
  END

  IF (@RangeEndingTime IS NOT NULL)
  BEGIN
   EXECUTE @retStat = ProcCheckDsTMdTime @RangeEndingTime, @OUTPUT_DsTMdTime
OUTPUT
    IF (@retStat != 0)
    BEGIN
      RAISERROR 40375
      RETURN -999
    END
  SELECT @RangeEndingTime = @OUTPUT_DsTMdTime
  END

  INSERT  DsMdRangeDateTime(collectionId,
                            RangeEndingTime,
                            RangeEndingDate,
                            RangeBeginningTime,
                            RangeBeginningDate)
  VALUES( @collectionId,
        @RangeEndingTime,
        @RangeEndingDate,
        @RangeBeginningTime,
        @RangeBeginningDate )

  IF @@error!=0
  BEGIN
        RAISERROR 50115
        RETURN -999
  END
END
go
```

ProcInsertRegPeriodic**Procedure ProcInsertRegPeriodic**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertRegPeriodic              */
/*  Release        :  B0                              */
/*  Module         :  Temporal                           */
/*  Created on     :  3/30/97                            */
/*  Purpose        :  Create Stored Procedure for inserting */
/*                 data into DsMdRegularPeriodic         */
/*                 table                                 */
/*                                                       */
/*  Calls          :  ProcCheckDsTMdTime                              */
/*                                                       */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertRegPeriodic(
        @collectionId                ID          = 0,
        @PeriodName          varchar(30) = null,
        @Period1stDate       datetime    = null,
        @Period1stTime       DsTMdTime   = null,
        @PeriodDurationUnit  varchar(15) = null,
        @PeriodDurationValue float       = null,
        @PeriodCycleDurationUnit  varchar(15)  = null,
        @PeriodCycleDurationValue float  = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40209
      RETURN -999
   END

   IF (@PeriodName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40209
      RETURN -999
   END

   IF (@Period1stDate = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40209
      RETURN -999
   END

   IF (@Period1stTime = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40209
      RETURN -999
   END

   IF (@PeriodDurationUnit = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40209
      RETURN -999
   END

   IF (@PeriodDurationValue = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40209
      RETURN -999
   END
```

```
    IF (@PeriodCycleDurationUnit = null)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40209
       RETURN -999
    END

    IF (@PeriodCycleDurationValue = null)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40209
       RETURN -999
    END

    DECLARE @retStat   int
    DECLARE @OUTPUT_DsTMdTime    DsTMdTime

    IF (@Period1stTime IS NOT NULL)
    BEGIN
     EXECUTE @retStat = ProcCheckDsTMdTime @Period1stTime , @OUTPUT_DsTMdTime OUTPUT
      IF (@retStat != 0)
      BEGIN
        RAISERROR 40378
        RETURN -999
      END
    SELECT @Period1stTime = @OUTPUT_DsTMdTime
    END

    INSERT  DsMdRegularPeriodic(collectionId,
                                    PeriodName,
                                    Period1stDate,
                                        Period1stTime,
                                        PeriodDurationUnit,
                                        PeriodDurationValue,
                                        PeriodCycleDurationUnit,
                                        PeriodCycleDurationValue)
    VALUES( @collectionId,
           @PeriodName,
           @Period1stDate,
           @Period1stTime,
           @PeriodDurationUnit,
           @PeriodDurationValue,
           @PeriodCycleDurationUnit,
           @PeriodCycleDurationValue )

    IF @@error!=0
    BEGIN
           RAISERROR 50116
           RETURN -999
    END
END
go
```

ProcInsertReqDomain**Procedure ProcInsertReqDomain**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertReqDomain                       */
/*  Release        :  B0                                        */
/*  Module         :  System Information                        */
/*  Created on     :  11/04/1997                                */
/*  Purpose        :  Create stored procedure to insert data  */
/*                 into DsSiReqDomain                           */
/*  Calls          :  none                                      */
/*==============================================================*/

CREATE PROCEDURE ProcInsertReqDomain(
   @requestID           char(100)     = null,
   @domainEntry          int          = -1)
AS
BEGIN
  IF (@requestID = null)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40351
    /*  "Procedure ProcInsertReqDomain expects parameter @requestID, which was not supplied."  */
    RETURN -999
  END

  IF (@domainEntry = -1)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40352
    /*   "Procedure ProcInsertReqDomain expects parameter @domainEntry, which was not supplied."  */
    RETURN -999
  END

  INSERT  DsSiReqDomain (requestID,
                  domainEntry)
  VALUES(@requestID,
        @domainEntry)

  IF @@error!=0
  BEGIN
    RAISERROR 50139
    /*  "ProcInsertReqDomain: Failed Inserting into DsSiReqDomain"  */
    RETURN -999
  END

END
go
```

ProcInsertRequest**Procedure ProcInsertRequest**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertRequest                          */
/*  Release       :  B0                                         */
/*  Module        :  System Information                         */
/*  Created on    :  11/04/1997                                 */
/*  Purpose       :  Create stored procedure to insert data  */
/*                   into DsSiRequest                           */
/*  Calls         :  none                                       */
/*==============================================================*/


CREATE PROCEDURE ProcInsertRequest(
   @requestID           char(100)     = null,
   @userID              char(40),
   @priority            smallint,
   @createTime          datetime,
   @clientCDSEntry      char(255),
   @status              smallint,
   @sessionID           char(40),
   @asyncFlag           tinyint)
AS
BEGIN
  IF (@requestID = null)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40350
    /*  "Procedure ProcInsertRequest expects parameter @requestID, which was not supplied."  */
    RETURN -999
  END

  INSERT  DsSiRequest (requestID,
                  userID,
                  priority,
                  createTime,
                  clientCDSEntry,
                  status,
                  sessionID,
                  asyncFlag)
  VALUES(@requestID,
       @userID,
       @priority,
       @createTime,
       @clientCDSEntry,
       @status,
       @sessionID,
       @asyncFlag)

  IF @@error!=0
  BEGIN
```

```
    RAISERROR 50138
    /*  "ProcInsertRequest: Failed Inserting into DsMdRequest"  */
    RETURN -999
  END

END
go
```

ProcInsertSensor**Procedure ProcInsertSensor**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertSensor                          */
/*  Release        :  B0                                        */
/*  Module         :  Data Originator                           */
/*  Created on     :  4/03/97                                   */
/*  Purpose        :  Create stored procedure to insert         */
/*                    sensor data.                              */
/*  Calls          :  None                                      */
/*==============================================================*/


CREATE PROCEDURE ProcInsertSensor
               (@dbID            ID                 = 0,
                @instrumentId    ID                 = 0,
                @SensorTechnique  varchar(80) ,
                @SensorShortName  varchar(20) = null,
                @SensorLongName   varchar(80) )
AS
BEGIN
   IF (@dbID = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40212
      RETURN -999
   END

   IF (@instrumentId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40212
      RETURN -999
   END


   IF (@SensorShortName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40212
      RETURN -999
   END
```

```
        INSERT DsMdSensor(dbID,
                    instrumentId,
                    SensorTechnique,
                    SensorShortName,
                    SensorLongName)
                VALUES(@dbID,
                    @instrumentId,
                    @SensorTechnique,
                    @SensorShortName,
                    @SensorLongName)

        IF @@error!=0
        BEGIN
           RAISERROR 50119
           RETURN -999
        END
END
go
```

ProcInsertSensorCharacteristic**Procedure ProcInsertSensorCharacteristic**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertSensorCharacteristic          */
/*  Release       :  B0                                      */
/*  Module        :  Data Originator                         */
/*  Created on    :  4/03/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                   sensor characteristic data.             */
/*  Calls         :  None                                    */
/*================================================================*/


CREATE PROCEDURE ProcInsertSensorCharacteristic
(@sensorId                          ID = 0,
 @SensorCharacteristicName                  varchar(40) = null,
 @SensorCharacteristicDesc          varchar(80) = null,
 @SensorCharacteristicUnit          varchar(20),
 @SensorCharacteristicDataType              char(8) = null,
 @SensorCharacteristicValue         varchar(80) = null)
AS

BEGIN
   IF (@sensorId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40213
      RETURN -999
   END

   IF (@SensorCharacteristicName = null)
   BEGIN
      /* Mimic Sybase error. */
```

```
      RAISERROR 40213
      RETURN -999
   END

   IF (@SensorCharacteristicDesc = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40213
      RETURN -999
   END

   IF (@SensorCharacteristicDataType = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40213
      RETURN -999
   END

   IF (@SensorCharacteristicValue = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40213
      RETURN -999
   END


   INSERT  DsMdSensorCharacteristic( sensorId,
                                     SensorCharacteristicName,
                                     SensorCharacteristicDesc,
                                     SensorCharacteristicUnit,
                                     SensorCharacteristicDataType,
                                     SensorCharacteristicValue)

   VALUES (@sensorId,
              @SensorCharacteristicName,
           @SensorCharacteristicDesc,
              @SensorCharacteristicUnit,
              @SensorCharacteristicDataType,
              @SensorCharacteristicValue)

   IF @@error!=0
   BEGIN
          RAISERROR 50120
          RETURN -999
   END
END
go
```

ProcInsertSequenceCount**Procedure ProcInsertSequenceCount**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertSequenceCount                    */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  select count of rows from                  */
/*                DsMdInsertSequence                            */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/


CREATE PROCEDURE ProcInsertSequenceCount
AS

SELECT COUNT(*)
FROM
        DsMdInsertSequence

IF @@error != 0
  BEGIN
        RAISERROR 45151
        RETURN -999
  END
go
```

ProcInsertSingleDateTime**Procedure ProcInsertSingleDateTime**

Code

```
/*==============================================================*/
/*  Name           :  ProcInsertSingleDateTime                  */
/*  Release        :  B0                                        */
/*  Module         :  Temporal                                  */
/*  Created on     :  3/30/97                                   */
/*  Purpose        :  Create Stored Procedure for inserting */
/*                 data into DsMdSingleDateTime                 */
/*                 table                                        */
/*                                                              */
/*  Calls          :  ProcCheckDsTMdTime                        */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertSingleDateTime(
            @collectionId                ID = 0,
            @TimeOfDay              DsTMdTime = null,
            @CalendarDate            datetime  = null)
AS
```

```
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40214
      RETURN -999
   END

   IF (@TimeOfDay = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40214
      RETURN -999
   END

   IF (@CalendarDate = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40214
      RETURN -999
   END

   DECLARE @retStat   int
   DECLARE @OUTPUT_DsTMdTime    DsTMdTime

   IF (@TimeOfDay IS NOT NULL)
   BEGIN
    EXECUTE @retStat = ProcCheckDsTMdTime @TimeOfDay , @OUTPUT_DsTMdTime OUTPUT
     IF (@retStat != 0)
     BEGIN
       RAISERROR 40376
       RETURN -999
     END
   SELECT @TimeOfDay = @OUTPUT_DsTMdTime
   END

   INSERT  DsMdSingleDateTime(collectionId,
                        TimeOfDay,
                        CalendarDate)
   VALUES(@collectionId,
         @TimeOfDay,
         @CalendarDate)

   IF @@error!=0
   BEGIN
          RAISERROR 50121
          RETURN -999
   END
END
go
```

ProcInsertSingleDateTimeInMPd**Procedure ProcInsertSingleDateTimeInMPd**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertSingleDateTimeInMPd        */
/*  Release         :  B0                                   */
/*  Module          :  Temporal                             */
/*  Created on      :  3/30/97                              */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdSingleDateTimeInMPd     */
/*                  table                                   */
/*                                                          */
/*  Calls           :  ProcCheckDsTMdTime                   */
/*                                                          */
/*==============================================================*/


CREATE PROCEDURE ProcInsertSingleDateTimeInMPd(
            @collectionId                   ID = 0,
            @TimeOfDay              DsTMdTime = null,
            @CalendarDate           datetime  = null,
            @MultipleDateName         varchar(30) = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40215
      RETURN -999
   END

   IF (@TimeOfDay = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40215
      RETURN -999
   END

   IF (@CalendarDate = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40215
      RETURN -999
   END

   IF (@MultipleDateName = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40215
      RETURN -999
   END
```

```
   DECLARE @retStat   int
   DECLARE @OUTPUT_DsTMdTime  DsTMdTime

   IF (@TimeOfDay IS NOT NULL)
   BEGIN
    EXECUTE @retStat = ProcCheckDsTMdTime @TimeOfDay , @OUTPUT_DsTMdTime OUTPUT
     IF (@retStat != 0)
     BEGIN
       RAISERROR 40376
       RETURN -999
     END
   SELECT @TimeOfDay = @OUTPUT_DsTMdTime
   END

   INSERT  DsMdSingleDateTime(collectionId,
                        TimeOfDay,
                        CalendarDate,
                        MultipleDateName)
   VALUES(@collectionId,
         @TimeOfDay,
         @CalendarDate,
         @MultipleDateName)

   IF @@error!=0
   BEGIN
          RAISERROR 50122
          RETURN -999
   END
END
go
```

ProcInsertSpatial**Procedure ProcInsertSpatial**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertSpatial                    */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                               */
/*  Created on      :  3/25/97                                */
/*  Purpose         :  Create Stored Procedure to insert     */
/*                  :  data into DsMdSpatial                 */
/*                  :  table                                */
/*  Calls           :  None                                  */
/*==============================================================*/


CREATE PROCEDURE ProcInsertSpatial(@collectionId              ID = 0,
                           @SpatialCoverageType         char(10),
                           @AltitudeDatumName           varchar(40),
                           @AltitudeDistanceUnits       varchar(20),
                           @AltitudeEncodingMethod      varchar(255),
                           @DepthDatumName              varchar(80),
                           @DepthDistanceUnits          varchar(20),
```

```
                        @DepthEncodingMethod        varchar(255),
                        @HorizontalDatumName        varchar(30),
                        @EllipsoidName              varchar(255),
                        @SemiMajorAxis              float,
                        @DenominatorOfFlatteningRatio float,
                        @LatitudeResolution         float,
                        @LongitudeResolution        float,
                        @GeographicCoordinateUnits  varchar(80),
                        @LocalCoordinateSystemDesc  varchar(255),
                        @LocalGeoreferenceInformation varchar(255),
                                @AltitudeResolution         float,
                        @DepthResolution            float)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40138
      RETURN -999
   END

  INSERT DsMdSpatial(collectionId,
                SpatialCoverageType,
                AltitudeDatumName,
                AltitudeDistanceUnits,
                AltitudeEncodingMethod,
                DepthDatumName,
                DepthDistanceUnits,
                DepthEncodingMethod,
                HorizontalDatumName,
                EllipsoidName,
                SemiMajorAxis,
                DenominatorOfFlatteningRatio,
                LatitudeResolution,
                LongitudeResolution,
                GeographicCoordinateUnits,
                LocalCoordinateSystemDesc,
                LocalGeoreferenceInformation)
  VALUES (@collectionId,
        @SpatialCoverageType,
        @AltitudeDatumName,
        @AltitudeDistanceUnits,
        @AltitudeEncodingMethod,
        @DepthDatumName,
        @DepthDistanceUnits,
        @DepthEncodingMethod,
        @HorizontalDatumName,
        @EllipsoidName,
        @SemiMajorAxis,
        @DenominatorOfFlatteningRatio,
        @LatitudeResolution,
        @LongitudeResolution,
        @GeographicCoordinateUnits,
        @LocalCoordinateSystemDesc,
```

```
        @LocalGeoreferenceInformation)

  IF (@@error!= 0)
  BEGIN
    RAISERROR 50029
    RETURN -999
  END

  IF @AltitudeResolution IS NOT NULL
  BEGIN
   INSERT DsMdAltitudeResolution(collectionId,
                          AltitudeResolution)
   VALUES (@collectionId,
         @AltitudeResolution)
   IF (@@error!= 0)
   BEGIN
     RAISERROR 50030
     RETURN -999
   END
  END

  IF @DepthResolution IS NOT NULL
  BEGIN
   INSERT DsMdDepthResolution(collectionId,
                        DepthResolution)
   VALUES (@collectionId,
         @DepthResolution)
   IF (@@error!= 0)
   BEGIN
     RAISERROR 50031
     RETURN -999
   END
  END


END
go
```

ProcInsertSpatialCoord**Procedure ProcInsertSpatialCoord**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertSpatialCoord               */
/*  Release         :  B0                                   */
/*  Module          :  Spatial                              */
/*  Created on      :  3/25/97                              */
/*  Purpose         :  Create Stored Procedure to insert    */
/*                  :  spatial coord data                   */
/*  Calls           :  ProcInsertSpatial                    */
/*                     ProcInsertAltitudeResolution         */
/*                     ProcInsertDepthResolution            */
/*==============================================================*/
```

```
CREATE PROCEDURE ProcInsertSpatialCoord(
      @collectionId               ID,
      @SpatialCoverageType        char(10),
      @AltitudeDatumName          varchar(40),
      @AltitudeDistanceUnits      varchar(20),
      @AltitudeEncodingMethod     varchar(255),
      @AltitudeResolution         float,
      @DepthDatumName             varchar(80),
      @DepthDistanceUnits         varchar(20),
      @DepthEncodingMethod        varchar(255),
      @DepthResolution            float,
      @HorizontalDatumName        varchar(30),
      @EllipsoidName              varchar(255),
      @SemiMajorAxis              float,
      @DenominatorOfFlatteningRatio float,
      @LatitudeResolution         float,
      @LongitudeResolution        float,
      @GeographicCoordinateUnits  varchar(80),
      @LocalCoordinateSystemDesc  varchar(255),
      @LocalGeoreferenceInformation varchar(255))
AS
BEGIN
  DECLARE @retStat int

  EXECUTE @retStat = ProcInsertSpatial @collectionId,
                     @SpatialCoverageType,
                     @AltitudeDatumName,
                     @AltitudeDistanceUnits,
                     @AltitudeEncodingMethod,
                     @DepthDatumName,
                     @DepthDistanceUnits,
                     @DepthEncodingMethod,
                     @HorizontalDatumName,
                     @EllipsoidName,
                     @SemiMajorAxis,
                     @DenominatorOfFlatteningRatio,
                     @LatitudeResolution,
                     @LongitudeResolution,
                     @GeographicCoordinateUnits,
                     @LocalCoordinateSystemDesc,
                     @LocalGeoreferenceInformation
  IF (@retStat != 0)
  BEGIN
    RAISERROR 56075
    RETURN -999
  END

  EXECUTE @retStat = ProcInsertAltitudeResolution @collectionId,
                                                  @AltitudeResolution
  IF (@retStat != 0)
  BEGIN
    RAISERROR 56076
    RETURN -999
  END
```

```
  EXECUTE @retStat = ProcInsertDepthResolution @collectionId,
                                  @DepthResolution
  IF (@retStat != 0)
  BEGIN
    RAISERROR 56077
    RETURN -999
  END
END
go
```

ProcInsertSpatialKeyword**Procedure ProcInsertSpatialKeyword**

Code

```
/*======================================================================*/
/*  Name          :  ProcInsertSpatialKeyword                 */
/*  Release       :  B0                                    */
/*  Module        :  Collections                             */
/*  Created on    :  3/25/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                spatial keyword data for Collections.     */
/*  Calls         :                                        */
/*======================================================================*/


CREATE PROCEDURE ProcInsertSpatialKeyword
(@collectionId              ID = 0,
 @SpatialKeyword        varchar(40) = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40216
      RETURN -999
   END

   IF (@SpatialKeyword = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40216
      RETURN -999
   END

   IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
      INSERT  DsMdSpatialKeyword(collectionId,
                        SpatialKeyword)
          VALUES (@collectionId,
                    @SpatialKeyword)

          IF @@error!=0
      BEGIN
```

```
        RAISERROR 50123
            RETURN -999
      END
   END
   ELSE
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50124
      RETURN -999
   END
END
go
```

ProcInsertSSAPComponent**Procedure ProcInsertSSAPComponent**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertSSAPComponent                    */
/*  Release       :  B0                                         */
/*  Module        :  Delivered Algorithm Package                */
/*  Created on    :  4/1/97                                     */
/*  Purpose       :  Create stored procedure to insert into  */
/*                SSAP Component.                               */
/*  Calls         :  ProcInsertSSAPComponentAPVer               */
/*==============================================================*/

CREATE PROCEDURE ProcInsertSSAPComponent(
            @dbID                           ID     = 0,
            @ComponentType                  varchar(40),
            @ComponentName                  varchar(80),
            @SSAPAlgorithmPackageName       varchar(80),
            @SSAPInsertDate                 datetime)
AS
BEGIN
   IF (@dbID = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40210
      RETURN -999
   END


   INSERT  DsMdSSAPComponent (dbID,
                     ComponentType,
                     ComponentName,
                     SSAPAlgorithmPackageName,
                     SSAPInsertDate,
                     lastUpdate,
                     insertTime)
   VALUES(@dbID,
        @ComponentType,
        @ComponentName,
        @SSAPAlgorithmPackageName,
```

```
        @SSAPInsertDate,
            getdate(),
            getdate())

    IF @@error!=0
    BEGIN
            RAISERROR 50117
            RETURN -999
    END
END
go
```

ProcInsertSSAPComponentAPVer**Procedure ProcInsertSSAPComponentAPVer**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertSSAPComponentAPVersion       */
/*  Release       :  B0                                     */
/*  Module        :  Delivered Algorithm Package            */
/*  Created on    :  4/1/97                                 */
/*  Purpose       :  Create stored procedure to Delivered,   */
/*                   Algorithm Package, and associate data   */
/*                   tables.                                 */
/*  Calls         :                                         */
/*                                                        */
/*==============================================================*/


CREATE PROCEDURE ProcInsertSSAPComponentAPVer(
            @ssapId                         ID     = 0,
            @SSAPAlgPackageVersion          varchar(20) = null)
AS
BEGIN
    IF (@ssapId = 0)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40211
       RETURN -999
    END

    IF (@SSAPAlgPackageVersion = NULL)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40211
       RETURN -999
    END


    INSERT  DsMdSSAPComponentAPVersion ( ssapId,
                                SSAPAlgPackageVersion )
    VALUES( @ssapId,
          @SSAPAlgPackageVersion )
```

```
   IF @@error!=0
   BEGIN
         RAISERROR 50118
         RETURN -999
   END
END
go
```

ProcInsertTelephoneNumbers**Procedure ProcInsertTelephoneNumbers**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertTelephoneNumbers            */
/*  Release       :  B0                                    */
/*  Module        :  Contact                                 */
/*  Created on    :  3/24/97                                 */
/*  Purpose       :  Create stored procedure to Contact,     */
/*                   and associate data tables.              */
/*  Calls         :                                        */
/*                                                         */
/*==============================================================*/


CREATE PROCEDURE ProcInsertTelephoneNumbers
          (@contactId             ID = 0,
           @TelephoneNumber       varchar(23) = null,
           @TelephoneNumberType   char(10) = null )
AS
BEGIN
   IF (@contactId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40217
      RETURN -999
   END

   IF (@TelephoneNumber = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40217
      RETURN -999
   END

   IF (@TelephoneNumberType = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40217
      RETURN -999
   END


         INSERT DsMdTelephoneNumbers( contactId,
                                      TelephoneNumber,
```

```
                                TelephoneNumberType)
        VALUES(         @contactId,
                @TelephoneNumber,
                @TelephoneNumberType)

        IF @@error!=0
          BEGIN
        RAISERROR 50125
                RETURN -999
          END
END
go
```

ProcInsertTemporal**Procedure ProcInsertTemporal**

Code

```
/*==============================================================*/
/*  Name            :  ProcInsertTemporal                       */
/*  Release         :  B0                                       */
/*  Module          :  Temporal                                 */
/*  Created on      :  3/30/97                                  */
/*  Purpose         :  Create Stored Procedure for inserting */
/*                  data into DsMdTemporal                      */
/*                  table                                       */
/*                                                              */
/*  Calls           :  None                                     */
/*                                                              */
/*==============================================================*/


CREATE PROCEDURE ProcInsertTemporal(
            @collectionId                   ID = 0,
            @TimeType           char(10)    = NULL,
            @DateType           char(10)    = NULL,
            @TemporalRangeType    varchar(30) = NULL,
            @PrecisionOfSeconds   int       = NULL,
            @EndsatPresentFlag    char(1))
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40218
      RETURN -999
   END

   IF (@TimeType = NULL)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40218
      RETURN -999
   END

   IF (@DateType = NULL)
```

```
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40218
       RETURN -999
    END

    IF (@TemporalRangeType = NULL)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40218
       RETURN -999
    END

    IF (@PrecisionOfSeconds = NULL)
    BEGIN
       /* Mimic Sybase error. */
       RAISERROR 40218
       RETURN -999
    END


    INSERT  DsMdTemporal( collectionId,
                      TimeType,
                      DateType,
                      TemporalRangeType,
                      PrecisionOfSeconds,
                      EndsatPresentFlag )
    VALUES( @collectionId,
           @TimeType,
           @DateType,
           @TemporalRangeType,
           @PrecisionOfSeconds,
           @EndsatPresentFlag)

    IF @@error!=0
    BEGIN
           RAISERROR 50126
           RETURN -999
    END
END
go
```

ProcInsertTemporalKeyword**Procedure ProcInsertTemporalKeyword**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertTemporalKeyword                    */
/*  Release       :  B0                                           */
/*  Module        :  Collections                                  */
/*  Created on    :  3/25/97                                      */
/*  Purpose       :  Create stored procedure to insert            */
/*                   temporal keyword data for Collections.       */
/*  Calls         :                                               */
/*================================================================*/
```

```
CREATE PROCEDURE ProcInsertTemporalKeyword
(@collectionId              ID = 0,
 @TemporalKeyword     varchar(40) = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40219
      RETURN -999
   END

   IF (@TemporalKeyword = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40219
      RETURN -999
   END


   IF exists(SELECT dbID FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
      INSERT  DsMdTemporalKeyword(collectionId,TemporalKeyword)
          VALUES (@collectionId,
                  @TemporalKeyword)

          IF @@error!=0
      BEGIN
         RAISERROR 50127
             RETURN -999
      END
   END
   ELSE
   BEGIN
        /* Mimic Sybase error. */
        RAISERROR 50128
        RETURN -999
   END
END
go
```

ProcInsertUserCommentDocument**Procedure ProcInsertUserCommentDocument**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertUserCommentDocument          */
/*  Release       :  B0                                     */
/*  Module        :  Collections                            */
/*  Created on    :  3/25/97                                */
/*  Purpose       :  Create stored procedure to insert      */
/*                   user comment document data for         */
/*                   Collections.                           */
/*  Calls         :                                         */
/*==============================================================*/

CREATE PROCEDURE ProcInsertUserCommentDocument
(@collectionId                      ID = 0,
 @UserCommentDocumentPointer        varchar(255) = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40220
      RETURN -999
   END

   IF (@UserCommentDocumentPointer = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40220
      RETURN -999
   END

   IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
         INSERT  DsMdUserCommentDocument(collectionId,
                           UserCommentDocumentPointer)
         VALUES (@collectionId,
               @UserCommentDocumentPointer)

         IF @@error!=0
      BEGIN
         RAISERROR 50129
            RETURN -999
      END
   END
   ELSE
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50130
      RETURN -999
   END
END
go
```

ProcInsertValidationDocument**Procedure ProcInsertValidationDocument**

Code

```
/*================================================================*/
/*  Name          :  ProcInsertValidationDocument            */
/*  Release       :  B0                                      */
/*  Module        :  Collections                             */
/*  Created on    :  3/25/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                validation document data for Collections. */
/*  Calls         :                                          */
/*================================================================*/

CREATE PROCEDURE ProcInsertValidationDocument
(@collectionId                  ID = 0,
 @ValidationDocumentPointer    varchar(255) = null)
AS
BEGIN
   IF (@collectionId = 0)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40221
      RETURN -999
   END

   IF (@ValidationDocumentPointer = null)
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 40221
      RETURN -999
   END

   IF exists(SELECT 1 FROM DsMdCollections WHERE dbID = @collectionId)
   BEGIN
      INSERT  DsMdValidationDocument(collectionId,
                              ValidationDocumentPointer)
      VALUES (@collectionId,
               @ValidationDocumentPointer)

          IF @@error!=0
      BEGIN
         RAISERROR 50131
             RETURN -999
      END
   END
   ELSE
   BEGIN
      /* Mimic Sybase error. */
      RAISERROR 50132
      RETURN -999
   END
END
go
```

ProcInsertVertSpatialDomain**Procedure ProcInsertVertSpatialDomain**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertVertSpatialDomain            */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                */
/*  Created on    :  3/31/97                                 */
/*  Purpose       :  Create stored procedure to insert       */
/*                Granule Vertical Spatial Domain data.   */
/*  Calls         :  None                                   */
/*==============================================================*/
CREATE PROC ProcInsertVertSpatialDomain
        (@granuleId                    ID       = 0,
         @VerticalSpatialDomainType    varchar(20) = NULL,
         @VerticalSpatialDomainValue   varchar(20) = NULL)
AS
BEGIN

  IF (@granuleId = 0)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40179
    RETURN -999
  END

  IF (@VerticalSpatialDomainType = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40179
    RETURN -999
  END

  IF (@VerticalSpatialDomainValue = NULL)
  BEGIN
    /* Mimic Sybase error. */
    RAISERROR 40179
    RETURN -999
  END


   INSERT DsMdGrVerticalSpatialDomain(granuleId,
                              VerticalSpatialDomainType,
                              VerticalSpatialDomainValue)
                       VALUES(@granuleId,
                              @VerticalSpatialDomainType,
                              @VerticalSpatialDomainValue)
   IF @@error!=0
   BEGIN
      RAISERROR 50081
      RETURN -999
   END
END
go
```

ProcInsertXAR**Procedure ProcInsertXAR**

Code

```
/*==============================================================*/
/*  Name          :  ProcInsertXAR                              */
/*  Release       :  B0                                         */
/*  Module        :  Granules                                   */
/*  Created on    :  5/07/97                                    */
/*  Purpose       :  Create stored procedure to insert       */
/*                XAR granule order data.                    */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcInsertXAR
             (@granuleId  ID = 0,
              @XARId      int = 0,
              @XARType    varchar(8))

AS
BEGIN


  IF (@granuleId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40222
     RETURN -999
   END

  IF (@XARId = 0)
   BEGIN
     /* Mimic Sybase error. */
     RAISERROR 40222
     RETURN -999
   END


  INSERT DsMdXAR(granuleId,
                 XARId,
                 XARType)
         VALUES(@granuleId,
                 @XARId,
                 @XARType)


   IF @@error!=0
    BEGIN
          RAISERROR 50133
          RETURN -999
    END

END
go
```

ProcKeyDependency**Procedure ProcKeyDependency**

Code

```
CREATE PROCEDURE ProcKeyDependency
AS

DECLARE @error int

SELECT
        TableName,
        TableKeyName,
        DependentTableName,
        DependentTableKeyName,
        XrefFlag,
        SequenceNo
FROM
        DsMdKeyDependency
ORDER BY TableName,
        SequenceNo

SELECT @error = @@error

IF @error!=0
  BEGIN
        RAISERROR 45170
        RETURN -999
  END
go
```

ProcKeyDependencyCount**Procedure ProcKeyDependencyCount**

Code

```
CREATE PROCEDURE ProcKeyDependencyCount
AS

SELECT COUNT(*)
FROM
        DsMdKeyDependency
IF @@error != 0
  BEGIN
        RAISERROR 45171
        RETURN -999
  END
go
```

ProcOperatorConv**Procedure ProcOperatorConv**

Code

```
CREATE PROCEDURE ProcOperatorConv
AS
```

```
SELECT
        Operator,
        DBString
FROM
        DsMdOperatorConversion
ORDER BY
        Operator,
        DBString
IF @@error!=0
  BEGIN
        RAISERROR 45172
        RETURN -999
  END
go
```

ProcOperatorConvCount**Procedure ProcOperatorConvCount**

Code

```
CREATE PROCEDURE ProcOperatorConvCount
AS

SELECT COUNT(*)
FROM
        DsMdOperatorConversion
IF @@error != 0
  BEGIN
        RAISERROR 45173
        RETURN -999
  END
go
```

ProcUnloadList**Procedure ProcUnloadList**

Code

```
CREATE PROCEDURE ProcUnloadList
AS

SELECT
        BaseType,
        SubType,
        ListName,
        CommandType,
        DataType,
        GroupName,
        ContainerGroupName,
        SequenceNo
FROM  DsMdUnloadList
ORDER BY BaseType,
        SubType,
        SequenceNo

IF @@error!=0
```

```
  BEGIN
        RAISERROR 45174
        RETURN -999
  END
go
```

ProcUnloadListCount**Procedure ProcUnloadListCount**

Code

```
CREATE PROCEDURE ProcUnloadListCount
AS
SELECT COUNT(*)
FROM
        DsMdUnloadList
IF @@error != 0
  BEGIN
        RAISERROR 45175
        RETURN -999
  END
go
```

ProcUpdateESDTConfiguredType**Procedure ProcUpdateESDTConfiguredType**

Code

```
/*================================================================*/
/*  Name          :  ProcUpdateESDTConfiguredType          */
/*  Release       :  B0                                    */
/*  Module        :                                        */
/*  Created on    :  8/21/97                               */
/*  Purpose       :                                        */
/*  Calls         :  None                                  */
/*================================================================*/

CREATE PROC ProcUpdateESDTConfiguredType
(
   @configuredName          char(8)        = NULL,
   @versionID               tinyint        = NULL,
   @archiveIdentifier       varchar(20)    = NULL,
   @description             varchar(255)   = NULL,
   @descriptorFileName      varchar(80)    = NULL,
   @dllFileName             varchar(80)    = NULL,
   @eventFileName           varchar(80)    = NULL
)
AS
BEGIN
   IF (@configuredName = NULL OR @configuredName = "")
      BEGIN
         RAISERROR 40321
         RETURN -999
      END

    IF (@versionID = NULL)
```

```
        BEGIN
           RAISERROR 40321
           RETURN -999
        END

    UPDATE DsGeESDTConfiguredType
    SET archiveIdentifier = @archiveIdentifier,
       description = @description,
       descriptorFileName = @descriptorFileName,
       dllFileName = @dllFileName,
       eventFileName = @eventFileName
    WHERE configuredName = @configuredName
    AND versionID = @versionID

    IF (@@error != 0)
    BEGIN
       RAISERROR 54024
       RETURN -999
    END
END
go
```

ProcUpdateProcHistLink**Procedure ProcUpdateProcHistLink**

Code

```
/*================================================================*/
/*  Name          :  ProcUpdateProcHistLink                       */
/*  Release       :  B0                                           */
/*  Module        :  Granules                                     */
/*  Created on    :  5/06/97                                      */
/*  Purpose       :  Create stored procedure to add               */
/*                   processing History link                      */
/*  Calls         :  None                                         */
/*================================================================*/

CREATE PROC ProcUpdateProcHistLink (@procHistId  ID = 0,
                          @scienceId varchar(100))
AS
BEGIN
    IF  (@procHistId = 0)
    BEGIN
       RAISERROR 40320
       RETURN -999
    END
    DECLARE @idOnly        int
    DECLARE @backwards     varchar(100)

    SELECT  @backwards = reverse(@scienceId)
    SELECT  @idOnly    = convert(int, reverse(substring(@backwards, 1, charindex(":", @backwards) - 1
)))

    UPDATE DsMdGranules
    SET    processingHistoryId = @procHistId, processingHistoryTypeCode = 2
```

```
   WHERE  dbID = @idOnly

   IF (@@error != 0)
   BEGIN
      RAISERROR 54023
      RETURN -999
   END
END
go
```

ProcUpdateUnlinkProcHistLink**Procedure ProcUpdateUnlinkProcHistLink**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdateUnlinkProcHistLink           */
/*  Release       :  B0                                     */
/*  Module        :  Granules                                */
/*  Created on    :  8/01/97                                 */
/*  Purpose       :  Create stored procedure to remove       */
/*                   processing History link from a granule  */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcUpdateUnlinkProcHistLink (@dbID  ID = 0)
AS
BEGIN
   IF  (@dbID = 0)
   BEGIN
      RAISERROR 40303
      RETURN -999
   END

   UPDATE DsMdGranules
   SET    processingHistoryId = NULL, processingHistoryTypeCode =  NULL
   WHERE  dbID = @dbID

   IF (@@error != 0)
   BEGIN
      RAISERROR 54006
      RETURN -999
   END
END
go
```

ProcUpdateXref**Procedure ProcUpdateXref**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdateXref                             */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  select from SELECT * from                  */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcUpdateXref
AS

SELECT
        BaseType,
        GroupName,
        DBCommand,
        ListName,
        CommandType,
        SequenceNo
FROM
        DsMdUpdateXref

IF @@error!=0
  BEGIN
        RAISERROR 45273
        RETURN -999
  END
go
```

ProcUpdateXrefCount**Procedure ProcUpdateXrefCount**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdateXrefCount                        */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  select count of rows from                  */
/*                DsMdUpdateXref                                */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROCEDURE ProcUpdateXrefCount
AS

SELECT
```

```
        COUNT(*)
FROM
        DsMdUpdateXref

IF @@error!=0
  BEGIN
        RAISERROR 45274
        RETURN -999
  END
go
```

ProcUpdLglDelAlgorithmPackage**Procedure ProcUpdLglDelAlgorithmPackage**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdLglDelAlgorithmPackage          */
/*  Release       :  B0                                     */
/*  Module        :  DB Wrappers System Tables              */
/*  Created on    :  3/24/97                                */
/*  Purpose       :  logical delete of                      */
/*                DsMdDeliveredAlgorithmPackage           */
/*                                                          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcUpdLglDelAlgorithmPackage (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40304
      RETURN -999
   END

   UPDATE DsMdAlgorithmPackage
        SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54007
      RETURN -999
   END
END
go
```

ProcUpdLglDelAncillaryGranule**Procedure ProcUpdLglDelAncillaryGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdLglDelAncillaryGranule          */
/*  Release       :  B0                                     */
/*  Module        :  DB Wrappers System Tables              */
/*  Created on    :  3/24/97                                */
/*  Purpose       :  logical delete of                      */
/*                DsMdAncillaryInputGranule                 */
/*                                                          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcUpdLglDelAncillaryGranule (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40305
      RETURN -999
   END

   UPDATE DsMdAncillaryInputGranule
         SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54008
      RETURN -999
   END
END
go
```

ProcUpdLglDelBrowseMetadata**Procedure ProcUpdLglDelBrowseMetadata**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdLglDelBrowseMetadata            */
/*  Release       :  B0                                     */
/*  Module        :  DB Wrappers System Tables              */
/*  Created on    :  3/24/97                                */
/*  Purpose       :  logical delete of                      */
/*                DsMdBrowse                                */
/*                                                          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcUpdLglDelBrowseMetadata (@dbID  ID = 0)
```

```
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40306
      RETURN -999
   END

   UPDATE DsMdBrowse
         SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54009
      RETURN -999
   END
END
go
```

ProcUpdLglDelFileStorage**Procedure ProcUpdLglDelFileStorage**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdLglDelFileStorage              */
/*  Release       :  B0                                    */
/*  Module        :  DB Wrappers System Tables             */
/*  Created on    :  3/24/97                               */
/*  Purpose       :  logical delete of                     */
/*                DsMdFileStorageData                      */
/*                                                         */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROC ProcUpdLglDelFileStorage (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40307
      RETURN -999
   END

   UPDATE DsMdFileStorage
         SET    deleteEffectiveDate = getdate()
   WHERE  granuleId = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54010
      RETURN -999
```

```
   END
END
go
```

ProcUpdLglDelGranuleMetadata**Procedure ProcUpdLglDelGranuleMetadata**

Code

```
/*==============================================================*/
/*  Name          :  rocUpdLglDelGranuleMetadata          */
/*  Release       :  B0                                   */
/*  Module        :  DB Wrappers System Tables            */
/*  Created on    :  3/24/97                              */
/*  Purpose       :  logical  delete of                   */
/*                DsMdGranules                            */
/*                                                        */
/*  Calls         :  None                                 */
/*==============================================================*/

CREATE PROC ProcUpdLglDelGranuleMetadata (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40308
      RETURN -999
   END

   UPDATE DsMdGranules
         SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54011
      RETURN -999
   END
END
go
```

ProcUpdLglDelOrbitParmGranule**Procedure ProcUpdLglDelOrbitParmGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdLglDelOrbitParmGranule          */
/*  Release       :  B0                                     */
/*  Module        :  DB Wrappers System Tables              */
/*  Created on    :  3/24/97                                */
/*  Purpose       :  logical delete of                      */
/*                MdOrbitParametersGranule                  */
/*                                                          */
/*  Calls         :  None                                   */
/*==============================================================*/
```

```
CREATE PROC ProcUpdLglDelOrbitParmGranule (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40309
      RETURN -999
   END

   UPDATE DsMdOrbitParametersGranule
         SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54012
      RETURN -999
   END
END
go
```

ProcUpdLglDelProcHistory**Procedure ProcUpdLglDelProcHistory**

Code

```
/*==============================================================*/
/*  Name           :  ProcUpdLglDelProcHistory                  */
/*  Release        :  B0                                        */
/*  Module         :  DB Wrappers System Tables                 */
/*  Created on     :  3/24/97                                   */
/*  Purpose        :  logical delete of DsDbProcessingHistory */
/*                                                              */
/*  Calls          :  None                                      */
/*==============================================================*/

CREATE PROC ProcUpdLglDelProcHistory (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40310
      RETURN -999
   END

   UPDATE DsMdProcessingHistory
         SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54013
```

```
      RETURN -999
   END
END
go
```

ProcUpdLglDelQaGranule**Procedure ProcUpdLglDelQaGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdLglDelQaGranule                     */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  logical delete of                          */
/*                DsMdQaGranule                                 */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROC ProcUpdLglDelQaGranule (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40311
      RETURN -999
   END

   UPDATE DsMdQaGranule
         SET    deleteEffectiveDate = getdate()
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54014
      RETURN -999
   END
END
go
```

ProcUpdUndelAlgorithmPackage**Procedure ProcUpdUndelAlgorithmPackage**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelAlgorithmPackage               */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  logical undelete of                        */
/*                DsMdDeliveredAlgorithmPackage                 */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/
```

```
CREATE PROC ProcUpdUndelAlgorithmPackage (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40312
      RETURN -999
   END

   UPDATE DsMdAlgorithmPackage
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54015
      RETURN -999
   END
END
go
```

ProcUpdUndelAncillaryGranule**Procedure ProcUpdUndelAncillaryGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelAncillaryGranule          */
/*  Release       :  B0                                    */
/*  Module        :  DB Wrappers System Tables             */
/*  Created on    :  3/24/97                               */
/*  Purpose       :  Logical undelete of                   */
/*                DsMdAncillaryInputGranule                */
/*                                                         */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROC ProcUpdUndelAncillaryGranule (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40313
      RETURN -999
   END

   UPDATE DsMdAncillaryInputGranule
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54016
```

```
      RETURN -999
   END
END
go
```

ProcUpdUndelBrowseMetadata**Procedure ProcUpdUndelBrowseMetadata**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelBrowseMetadata              */
/*  Release       :  B0                                      */
/*  Module        :  DB Wrappers System Tables               */
/*  Created on    :  3/24/97                                 */
/*  Purpose       :  logical undelete of                     */
/*                of  DsMdBrowse                             */
/*                                                           */
/*  Calls         :  None                                    */
/*==============================================================*/

CREATE PROC ProcUpdUndelBrowseMetadata (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40314
      RETURN -999
   END

   UPDATE DsMdBrowse
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54017
      RETURN -999
   END
END
go
```

ProcUpdUndelFileStorage**Procedure ProcUpdUndelFileStorage**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelFileStorage                 */
/*  Release       :  B0                                      */
/*  Module        :  DB Wrappers System Tables               */
/*  Created on    :  3/24/97                                 */
/*  Purpose       :  logical undelete of                     */
/*                DsMdFileStorage                            */
/*                                                           */
/*  Calls         :  None                                    */
/*==============================================================*/
```

```
CREATE PROC ProcUpdUndelFileStorage (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40315
      RETURN -999
   END

   UPDATE DsMdFileStorage
         SET    deleteEffectiveDate = NULL
   WHERE  granuleId = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54018
      RETURN -999
   END
END
go
```

ProcUpdUndelGranuleMetadata**Procedure ProcUpdUndelGranuleMetadata**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelGranuleMetadata            */
/*  Release       :  B0                                     */
/*  Module        :  DB Wrappers System Tables              */
/*  Created on    :  3/24/97                                */
/*  Purpose       :  logical undelete of                    */
/*                DsMdGranules                              */
/*                                                          */
/*  Calls         :  None                                   */
/*==============================================================*/

CREATE PROC ProcUpdUndelGranuleMetadata (@dbID  ID = 0)
AS
BEGIN
    IF  (@dbID = 0)
   BEGIN
      RAISERROR 40316
      RETURN -999
   END

   UPDATE DsMdGranules
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
```

```
      RAISERROR 54019
      RETURN -999
   END
END
go
```

ProcUpdUndelOrbitGranule**Procedure ProcUpdUndelOrbitGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelOrbitGranule              */
/*  Release       :  B0                                    */
/*  Module        :  DB Wrappers System Tables             */
/*  Created on    :  3/24/97                               */
/*  Purpose       :  logical undelete of                   */
/*                iDsMdOrbitParametersGranule             */
/*                                                         */
/*  Calls         :  None                                  */
/*==============================================================*/

CREATE PROC ProcUpdUndelOrbitGranule (@dbID  ID = 0)
AS
BEGIN

   IF  (@dbID = 0)
   BEGIN
      RAISERROR 40317
      RETURN -999
   END

   UPDATE DsMdOrbitParametersGranule
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54020
      RETURN -999
   END
END
go
```

ProcUpdUndelProcHistory**Procedure ProcUpdUndelProcHistory**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelProcHistory                    */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  logical undelete of                        */
/*                DsDbProcessingHistory                         */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROC ProcUpdUndelProcHistory (@dbID  ID = 0)
AS
BEGIN
   IF  (@dbID = 0)
   BEGIN
      RAISERROR 40318
      RETURN -999
   END

   UPDATE DsMdProcessingHistory
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
     AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54021
      RETURN -999
   END
END
go
```

ProcUpdUndelQaGranule**Procedure ProcUpdUndelQaGranule**

Code

```
/*==============================================================*/
/*  Name          :  ProcUpdUndelQaGranule                      */
/*  Release       :  B0                                         */
/*  Module        :  DB Wrappers System Tables                  */
/*  Created on    :  3/24/97                                    */
/*  Purpose       :  logical undelete of                        */
/*                DsMdQaGranule                                 */
/*                                                              */
/*  Calls         :  None                                       */
/*==============================================================*/

CREATE PROC ProcUpdUndelQaGranule (@dbID  ID = 0)
```

```
AS
BEGIN
   IF  (@dbID = 0)
   BEGIN
      RAISERROR 40319
      RETURN -999
   END

   UPDATE DsMdQaGranule
         SET    deleteEffectiveDate = NULL
   WHERE  dbID = @dbID
    AND deleteEffectiveDate IS NOT NULL

   IF (@@error != 0)
   BEGIN
      RAISERROR 54022
      RETURN -999
   END
END
go
```

4.3 Flat File Usage

A flat file is an operating system file that is written and subsequently read, generally independent of other files that exist, and usually static in nature. There are cases when the implementation of persistent data is better suited to a flat file than to a database (e.g., system configuration data, external interface data). SDSRV Subsystem file usage is detailed in this section via file, block, field, and domain definitions.

4.3.1 Files Descriptions

A summary listing of the files in the SDSRV Subsystem is given in Table 4-6 together with a brief description of the file usage. Many different record formats are used in ECS including ODL., HDF, HDF EOS, block, fixed length, variable length, etc.

Table 4-6. Flat File Descriptions

File Name	File Type	Record Format	File Description
EcDsScienceDataServer.ALOG.CCYYMMDDHHMMSS (where CC=century, YY=year, MM=month, DD=day, HH=hour,MM=minutes, SS=seconds. All 2 digit integers).	Unix flat file	variable length	Main Science Data Server output log file. Contains informational, warning, and error messages. Each message has a date and time stamp. Always produced
EcDsScienceDataServerDebug.log.CCYYMMDDHHMMSS	Unix flat file	variable length	Optional debug log file. Setup in Science Data Server configuration file. Contains messages showing server activity at key points in processing. Includes executed SQL, DCE information , streamed metadata.
EcDsScienceDataServer,ACFG	Unix flat file	variable length	Science Data Server Application Configuration
EcDsScienceDataServer.CFG	Unix flat file	variable length	Metrics file. Describes installation configuration information.
EcDsScienceDataServer.PCFG	Unix flat file	variable length	Science Data Server runtime configuration file. Contains configurable information needed at system start-up for both COTS and custom software.
EcDsScienceDataServerClient.CFG	Unix flat file	variable length	Science Data Server Program Configuration file. Contains Program configuration metrics and process performance metrics configuration data. Science Data Server Client configuration file.
<ShortName.<VersionID>.evt	Unix Binary B-Tree	RogueWave Tree-on-Disk file.	Events and event IDs returned by SBSRV when SDSRV installs an ESDT. These events are contained within the Descriptor file.
DsESDTCollectionTemplate.desc	Unix flat file	variable length	Used to validate ESDTs upon installation. Descriptor Template.
DsESDT<sub-dir	Unix flat file	variable length	Uniquely named ESDT file descriptors.

File Name	File Type	Record Format	File Description
name><Shortname>.<VersionID>.desc (sub-dir is name of subdirectory used to store Descriptors)			Describe collection level metadata for SDSRV installation.
libDsESDT<sub-dir name><ShortName>.<VersionID>Sh.so	ELF 32-bit MSB dynamic lib SPARC Version 1, dynamically linked	Dynamic Link Library (DLL)	Provides generic to ESDT specific processing capability to SDSRV. These Dynamic Linked Libraries (DLLs) are dynamically linked into the Data Server on an as needed basis.

4.3.2 Block Specifications

Brief definitions of each of the flat file block descriptions are contained herein.

Table 4-7. Flat File Block Descriptions

File Name	Block Name	Block Description
EcDsScienceDataServer.ALOG.CCYYMMDDHHMMSS	N/A	N/A
EcDsScienceDataServerDebug.log.CCYYMMDDHHMMSS	N/A	N/A
EcDsScienceDataServer,ACFG	APPLICATION PROGRAM	Denotes Application level parameters. Denotes Program level parameters.
EcDsScienceDataServer.CFG	General Attributes	SDSRV general description data.
	DCE RELATED ATTTRIBUTES	Describes SDSRV to DCE.
	SECURITY ATTRIBUTES	Contains SDSRV ACL profile Contains ACL Database information.
	DATABASE ATTRIBUTES	Turns Server Request Framework (SRF) off or on.
	SERVER REQUEST FRAMEWORK AND MESSAGE PASSING FLAGS	Message passing attributes needed for SRF.

File Name	Block Name	Block Description
	MESSAGE PASSING ATTRIBUTES	Contains RecUuid - the UUID needed for SRF message passing.
	HDF server info	
EcDsScienceDataServer.PCFG	DSSSrEnv2	Environment variables used by the SDSRV. Sourced at start-up
	CSS_MP APPLICATION	Server specific parameters for the MSS Deputy Agent.
	PROGRAM	SNMP parameters.
	PROCESS	MSS Agent performance metrics configuration.
EcDsScienceDataServerClient.CFG	N/A	N/A
<ShortName.<VersionID>>.evt	N/A	N/A
DsESDTCollectionTemplate.desc		
DsESDT<sur-dir name><ShortName>.<VerionsID>.desc		
LibDsESDT<sub-dir name><ShortName>.<VerionsID>Sh.so	N/A	N/A

4.3.3 Field Specifications

Brief specifications of the fields present within the SDSRV Subsystem flat files are contained in Table 4-8. The fields are ordered alphabetically by File Name.

Table 4-8. Flat File Field Specifications

File Name/Block Name	Field Name	Data Type	Field Description
EcDsScienceDataServer.ALOG.CCYYMMDDHHMMSS/ N/A		integer string integer string string integer datetime string	Unix Process ID of SDSRV : Name of object sending message Message Text from SDSRV MM/DD/CCYY HH:MM:SS
EcDsScienceDataServerDebug.log.CCYYMMDDHHMMSS/ N/A			SDSRV debug message text .
EcDsScienceDataServer.ACFG APPLICATION.ConfigurationMetrics/		quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string quoted string	Application ID. Value - ì4000000î Application Name. Value - ìEcDsScienceDataServerî App. Major Version. Value - ì1î App. Minor Version.Value - ì0î App. Revision Number. Value - ì0î App. Maint Level. Value - ì0î App Type. Value - ìcustomî App primary contact. Value - ìDSS Leadî App msg language. Value - ìEnglishî Unix Dir. Path.. Value - ì/usr/ecs/<MODE>/CUSTOM/utilitiesî Command Line. Value - ìEcDsScienceDataServer <MODE>î Max log size in bytes. Value - ì1000000î Log Level. Value - ì0î Subsystem name. Value - ìDSSî Value - ì2î Mode install time. Value -

File Name/Block Name	Field Name	Data Type	Field Description
PROGRAM.ConfigurationMetrics			ì04/01/1998î
		quoted string	MSS log file path. Value - ì/usr/ecs//<MODE>/CUSTOM/logsî
		numeric	Log File name. Value - ìEcDsScienceDataServer.ALOGî
		numeric	Number of Programs. Value - ì2î
		string	Scope of App.. Value - public
		numeric	Peer Name. Value - ìxxxxxxx.hitc.comî
		numeric	
EcDsScienceDataServer.CFG/ General Attributes		Numeric	
		Numeric	Name of the SDSRV application - EcDsScienceDataServer
		numeric	
		string	ECS program identifier of the EcDsScienceDataServer. - 4000000
		string	ECS application Identifier for the SDSRV application - 4000000
		string	Name of the DAAC site - ex. GSFC.
		string	ECS subsystem name - DSS (Data Server Subsystem).
		string	Major version number of the software release.
		string	Minor version number of the software release.
		string	Application log size in number of bytes.
		string	Application logging level. Values are 0, 1.
		integer	Debug logging level. Values are 0,

File Name/Block Name	Field Name	Data Type	Field Description
		string	1, 2, 3..
		string	ECS release identifier - ìBî.
DCE RELATED ATTTRIBUTES		string	
		string	SDSRV DCE name. Value - EcDsScienceDataServer.
		string	DCE groupname 1. Value - EcDsScienceDataServerG1.
		string	DCE groupname 2. Value - EcDsScienceDataServerG2.
		integer	DCE groupname 3. Value - EcDsScienceDataServerG3.
		string	
		string	DCE groupname 4. Value - EcDsScienceDataServerG4.
			- EcDsScienceDataServer - Values - multiple.
		string	- This is the EcDsScienceDataServer (Mode: <MODE>).
			Value - 4.
		string	Value - 6d2aa05c-c9a1-11d1-82f4-9b9d3035aa77
		string	Value - 6d3aa05c-c9a1-11d1-82f4-9b9d3035aa77
		string	Value - 6d4aa05c-c9a1-11d1-82f4-9b9d3035aa77
			Value - 6d5aa05c-c9a1-11d1-82f4-9b9d3035aa77
		integer	
		integer	
		integer	Value - 6d5aa05c-c9a1-11d1-82f4-9b9d3035aa77
		integer	Number of listen threads. Value - 7.
		integer	Communications Protocol. Values - TCP/IP, UDP, TCP/IP UDP, Null.

File Name/Block Name	Field Name	Data Type	Field Description
SECURITY ATTRIBUTES		string	
		string	Unix path and filename for the EcDsScienceDataServer.Keyfile
		string	. Value - Custom/security/EcDsScienceDataServer.keyfile
		string	
		string	EcDsScienceDataServer.
		string	Sybase database name of the appropriate ACL database. Value - b_MsgPsngDB_acl.
		string	
		string	Unix path and filename of the random data file. Value - /usr/ecs//<mode>/CUSTOM/security/EcCoRandomDataFile.
		integer	Value - 12345.
DATABASE ATTRIBUTES		string	. Value - PfAclKey.
		string	Symbolic name of the database client library for
SERVER REQUEST FRAMEWORK AND MESSAGE PASSING FLAGS		string	RogueWave DBTools. Value - Sybase_CT.
MESSAGE PASSING ATTRIBUTES		string	Sybase SQL server name for the ACL database. Value - OTIS_SERVER.
HDF server info		string	
DSSSrEnv2		string	Sybase login name for ACL processing. Value - acluser.
		string	Sybase login password for ACL processing.
EcDsScienceDataServer.PCFG/		string	Sybase database name of the ACL database.
CSS_MP		string	Value - EcsAclStorage.
APPLICATION		integer	How often (in seconds) updates to the ACL information are processing . Value - 3600.
PROGRAM		string	Turns Server Request Framework(SRF) on or off. Value - no.
PROCESS		string	
EcDsScienceDataServerClient.CFG/			

File Name/Block Name	Field Name	Data Type	Field Description
N/A		string	UUID needed for SRF.
		string	Name of file used to persist received message passing data.. Value - b_Rec.
		string	Name of file used to persist sending message passing data.
		string	If not set there will be no persistence for message passing. Value - b_Send.
		integer	Number of FTP threads used by the application.
		string	Value - Between 1 and 30.
		string	Determines persistence with FTP. Value - ftpfilename.
		string	Lists those subsystems to which SDSRV acts as a client - Data Distribution Server info. Value - EcDsDistributionServer.
		string	This is a group environment variable containing
		integer	the following environment variables whose
		string	value is described below.
		string	DSSSDSRV DSSEVTDIR DSSATTRFILE DSSCONTFILE DSSSTMGSTAGEINGSERVERUR
		string	DSSSHOULDADVERTISE
		string	DSSNOSTAGINGDISK
		string	DSSNOARCHIVE
		string	URDDICT
		string	SDSRV_MODEDDICT
		string	
		string	
		string	
		string	
		string	This is a group environment

File Name/Block Name	Field Name	Data Type	Field Description
			variable containing the following environment variables whose value is described below. Values - DSSCONFIG DSSDLLDIR DSSSDSRV DSSSBSRV DSSEVSRV DSSDESCDIR $DSSSrEnv2 $DSSSrEnv_DB DSSDLLINPUTDIR DSSDESCINPUTDIR
		string string string string string string string string string	This is a group environment variable containing the following environment variables whose value is described below. Values - DBUSERNAME DBPASSWD DBNAME DBMAXRESULTS SYBINTERFACES
		integer	This is a group environment variable containing the following environment variables whose value is described below. Values - SYBASE DSQUERY
		string string string string	Maximum number of results returned to client from DsDbInterface class. Value - 1500
		string	Sybase database name of

File Name/Block Name	Field Name	Data Type	Field Description
		string	the SDSRV metadata database for this server. Value - relbdss_DEV04
		string	Sybase login of the SDSRV user. Value - EcDsScienceDataServer
		string	Sybase password of the SDSRV Sybase login. Value - xxxxxxxxxxxxxxxxxxx
		string	Name of the Sybase SQL Server instance used by the SDSRV instance. Value - comanche_sqs222_srvr
		string	Value - "not used"
		string	Value - "not used"
		string	Value - "not used"
		string	Unix file path containing ESDT Descriptors. Value - /usr/ecs/<MODE>/CUSTOM/cfg/DsESDTDesc
		integer	
		string	
		string	Unix file path containing input ESDT Descriptors Value - usr/ecs/<MODEL>CUSTOM/bin/DSS/
		string	Unix file path containing Dynamic Link Libraries Value - /usr/ecs/<MODE>/CUSTOM/lib/DSS
		array of :	Unix file path containing input DLLs. Value - /usr/ecs/<MODE>/CUSTOM/bin/DSS
		quoted string	Unix file path of the Event files. Value - /usr/ecs/<MODE>/CUSTOM/cfg
		quoted string	
		quoted string	Value - Not Applicable(NULL)
		quoted string	

File Name/Block Name	Field Name	Data Type	Field Description
		quoted string	Value - Not Applicable(NULL)
		quoted string	Universal Reference of the Subscription Server communicating to this SDSRV. Value - UR:12:EcUrServerUr[:CSSBSRV]
		quoted string	UR of the server communicating with this SDSRV. Value - UR:12:EcUrServerUr[:CSEVSRV]
		quoted string	. Value - [RBD:DSSDSRV]
		quoted string	Determines . Value - 1
		integer	UR of the Staging Server communicating with this SDSRV. Value - SGI_<MODE>
		integer	
		integer	Unix file path of the Home directory of the Sybase SQL Server software. Value - /tools/sybOCv11.1.0
		quoted string	Unix file path location of the Sybase INTERFACES file. Value - tools/sybOCv11.1.0/interfaces
		quoted string	
		Array of integer	UR of the Data Dictionary Server communicating with this SDSRV. Value - UR:12:EcUrServerUr[:DMDDICT]
		integer	
		integer	
		integer	
		integer	
		integer	
		string	Not Applicable(NULL)
		string	Not Applicable(NULL)
		string	

File Name/Block Name	Field Name	Data Type	Field Description
		string string string string integer integer integer integer integer integer integer integer N/A	Series of bracket delimited quoted strings. . Value - ì4000000î . Value - ì4000000î . Value - ìEcDsScienceDataServerî . Value - ìEcDsScienceDataServerî Value - ì1î . Value - ì1î . Value - ì0î . Value - ì0î . Value - ìcustomî . Value - ìDSS Task Leadî . Value - ìEnglishî . Value - ì/usr/ecs//<MODE>/CUSTOM/utilities . Value - EcDsScienceDataServer <MODE> . Value - ì99î . Value - 20 . Value - 15 . Value - 10 . Value - ì04/01/1998î . Value - ì0î Series of bracket delimited quoted strings. . Value - 500, 300, 100, 400, 200, 150 . Value - 500, 300, 100, 400, 200, 150 . Value - 500, 300, 100, 400, 200, 150 . Value - 500, 300, 100, 400, 200, 150 . Value - 500, 300, 100, 400, 200, 150 . Value - 500, 300, 100, 400,

File Name/Block Name	Field Name	Data Type	Field Description
<ShortName.<VersionID>.evt			200, 150 . Value - 500, 300, 100, 400, 200, 150 . Value - 500, 300, 100, 400, 200, 150 . Value - G1 . Value - G2 . Value - G3 . Value - G4 DAAC name. Value - GSFC ECS Sub System Name. Value - DSS Name of the SDSRV Client application - EcDsScienceDataServerClient ECS program identifier of the EcDsScienceDataServerClient. - 4000000 ECS application Identifier for the SDSRV Client application - 4000001 SDSRV Client log size. Value - 1000000 SDSRV Client logging level. Value - 0 Major version number of the software release. Value - 1 Minor version number of the software release.Value - 0 Application Start Number. Value - 12345.

File Name/Block Name	Field Name	Data Type	Field Description
DsESDTCollectionTemplate.desc	N/A		
DsESDT<sub-dir name><ShortName>.<VersionID>.desc			
LibDsESDT<sub-dir name><ShortName>.<VersionID>.Sh.so	N/A		

4.3.4 Domain Definitions

Domain definitions specify the data type and valid content of fields within a file (e.g., specific values for a limited set of data, ranges of numeric data, units of measure for applicable data). This information is generally used by software to edit incoming data for validity prior to writing or changing data within the file. Use of domain values in updating (adding and changing) records within files preserves the integrity of the data within the file. The domain definitions for the SDSRV Subsystem are presented in Table 4-9.

Table 4-9. Flat File Domain Definitions

File Name/Block Name	Field Name	Domain Description
EcDsScienceDataServer.ALOG.CCYYMMDDHHMMSS/ N/A		5 digit integer 1 digit integer string. Value - DsSrSdsrvmainStartSrSdsrv string. 2 digit integer datetime. Format - 05/15/98 10:40:42
EcDsScienceDataServerDebug.log.CCYYMMDDHHMMSS/ N/A		Variable Message Text fro SDSRV debug code. Describes state of processing at key points in code. Value - Dss EcDsScienceDataServer Server Debug log on texas starting on Mon Apr 13

File Name/Block Name	Field Name	Domain Description
		14:05:32 EDT 1998
EcDsScienceDataServer,ACFG/ APPLICATION PROGRAM		
EcDsScienceDataServer.CFG/ General Attributes		
DCE RELATED ATTRIBUTES		
SECURITY ATTRIBUTES		
DATABASE ATTRIBUTES		
SERVER REQUEST FRAMEWORK AND MESSAGE PASSING FLAGS		
MESSAGE PASSING ATTRIBUTES		
HDF server info		
DSSSrEnv2		
EcDsScienceDataServer.PCFG/ CSS_MP APPLICATION PROGRAM PROCESS		
EcDsScienceDataServerClient.CFG		
EcDsScienceDataServer.ACFG		
<ShortName>.<VerionID>.evt		
DsESDTCollectionTemplate.desc		

File Name/Block Name	Field Name	Domain Description
DsESDT<sub-dir name><ShortName>.<VersionID>.desc LibDsEsEsdt<sub-dir name><ShortName>.<VersionID>Sh.so		

This page intentionally left blank.