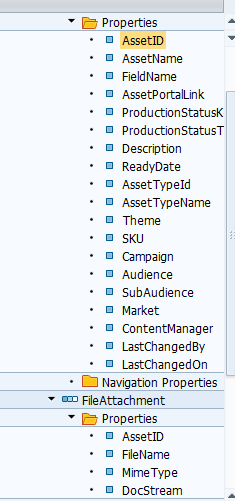
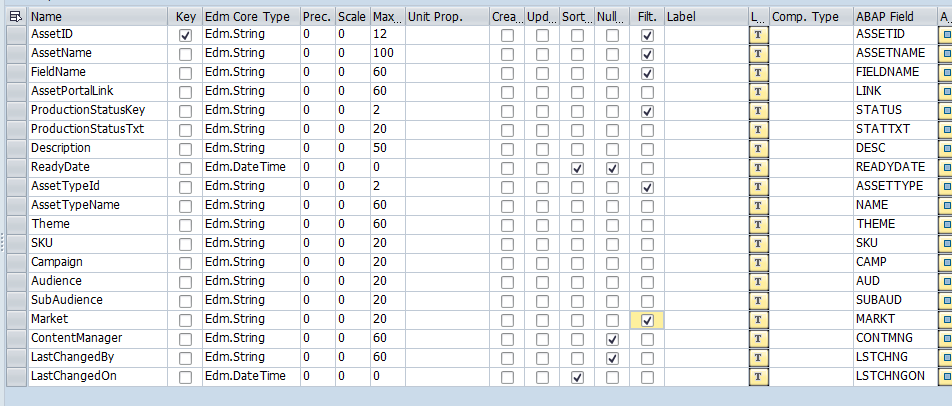
**Digital Asset Management – Data Model**

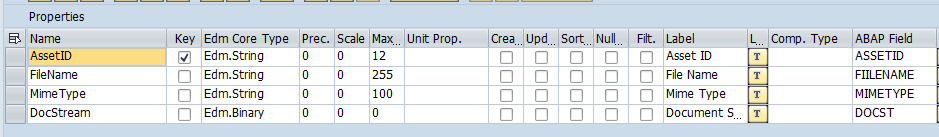
1. Entity – Asset ID and File Attachment



1. Entity Asset Properties



1. Entity FileAttachment Properties



**OData service method implementations:**

1. /IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME~CREATE\_STREAM

  METHOD /iwbep/if\_mgw\_appl\_srv\_runtime~create\_stream.  
  
    DATA: lv\_assetid(8),  
          lv\_filename    TYPE char255,  
          lv\_slug        LIKE iv\_slug,  
          lv\_docid       TYPE saeardoid,  
          ls\_asst\_entity TYPE zcl\_zasset\_mpc\_ext=>ts\_fileattachment.  
  
\*   Get file name and Asset ID details passed in SLUG  
    IF iv\_entity\_name EQ zcl\_zasset\_mpc\_ext=>gc\_fileattachment.  
      DATA(lv\_flen) = xstrlen( is\_media\_resource-value ).  
      SPLIT iv\_slug AT '@' INTO DATA(lv\_assid) lv\_filename.  
      lv\_assetid = |{ lv\_assid ALPHA = IN }|.  
      ls\_asst\_entity-assetid = lv\_assetid.  
      ls\_asst\_entity-fiilename = lv\_filename.  
      ls\_asst\_entity-docst = is\_media\_resource-value.  
      ls\_asst\_entity-mimetype = is\_media\_resource-mime\_type.  
  
\*     Make table entry for new file  
      INSERT INTO zasset\_file\_table VALUES ls\_asst\_entity.  
  
      IF sy-subrc <> 0.  
        DATA(lv\_msg) = 'Error in document upload'.  
        RAISE EXCEPTION TYPE /iwbep/cx\_mgw\_busi\_exception  
          EXPORTING  
            textid  = /iwbep/cx\_mgw\_busi\_exception=>business\_error  
            message = lv\_msg.  
      ELSE.  
        ls\_asst\_entity-assetid = lv\_assetid.  
        copy\_data\_to\_ref( EXPORTING is\_data = ls\_asst\_entity  
                          CHANGING  cr\_data = er\_entity ).  
      ENDIF.  
    ENDIF.  
  
  ENDMETHOD.

1. /IWBEP/IF\_MGW\_APPL\_SRV\_RUNTIME~GET\_STREAM

  METHOD /iwbep/if\_mgw\_appl\_srv\_runtime~get\_stream.  
  
    DATA : ls\_header   TYPE  ihttpnvp,  
           lv\_filename TYPE char255,  
           ls\_stream   TYPE /iwbep/if\_mgw\_appl\_types=>ty\_s\_media\_resource.  
    CONSTANTS : lc\_asseitid(40) VALUE 'ASSETID'.  
  
    LOOP AT it\_key\_tab ASSIGNING FIELD-SYMBOL(<lfs\_key\_tab>).  
      CASE <lfs\_key\_tab>-name.  
        WHEN lc\_asseitid.  
          DATA(lv\_assetid) = <lfs\_key\_tab>-value.   " get Asset ID  
        WHEN OTHERS.  
      ENDCASE.  
    ENDLOOP.  
  
\*   Read file details from custom table based of Asset id passed in get stream call  
    SELECT fiilename, mimetype, docst FROM zasset\_file\_table  
        INTO @DATA(ls\_file\_data)  
         WHERE assetid = @lv\_assetid.  
    ENDSELECT.  
  
    IF ls\_file\_data-fiilename IS NOT INITIAL.  
      ls\_stream-mime\_type =  ls\_file\_data-mimetype.  
      ls\_stream-value     = ls\_file\_data-docst.  
      lv\_filename = ls\_file\_data-fiilename.  
      lv\_filename = escape( val = lv\_filename  
                           format = cl\_abap\_format=>e\_url ).

\*    set header to http framework  
      ls\_header-name = 'Content-Disposition'.  
      ls\_header-value = |inline; filename="{ lv\_filename }"|.  
      set\_header( is\_header = ls\_header ).  
  
\*   Pass MIME type and stream value to result  
      copy\_data\_to\_ref( EXPORTING is\_data = ls\_stream  
                        CHANGING  cr\_data = er\_stream ).  
    ENDIF.  
  
  ENDMETHOD.

1. ASSETSET\_CREATE\_ENTITY

  METHOD assetset\_create\_entity.  
\*Work area ------------------------------------------------------------\*  
    DATA: ls\_entity    LIKE er\_entity,  
          ls\_asset     TYPE zasset\_details,  
          ls\_return    TYPE bapireturn1,  
          lv\_msg(100),  
          lv\_number(4) TYPE n.  
  
\*Object References ----------------------------------------------------\*  
    DATA lo\_message\_container TYPE REF TO /iwbep/if\_message\_container.  
  
    " Instantiate the Message Container  
    CALL METHOD me->/iwbep/if\_mgw\_conv\_srv\_runtime~get\_message\_container  
      RECEIVING  
        ro\_message\_container = lo\_message\_container.  
  
    io\_data\_provider->read\_entry\_data( IMPORTING es\_data = ls\_entity ).  
    ls\_asset = CORRESPONDING #( ls\_entity ).  
  
\*    generate number range for asset id  
    CALL FUNCTION 'NUMBER\_GET\_NEXT'  
      EXPORTING  
        nr\_range\_nr             = '01'  
        object                  = 'ZASST'  
      IMPORTING  
        number                  = lv\_number  
      EXCEPTIONS  
        interval\_not\_found      = 1  
        number\_range\_not\_intern = 2  
        object\_not\_found        = 3  
        quantity\_is\_0           = 4  
        quantity\_is\_not\_1       = 5  
        interval\_overflow       = 6  
        buffer\_overflow         = 7  
        OTHERS                  = 8.  
  
    IF sy-subrc EQ 0.  
      CONCATENATE 'AST' sy-datum+0(4) lv\_number INTO ls\_asset-assetid.  
    ENDIF.  
    ls-asset-contmng = sy-uname.  
    ls-asset-lstchng = sy-uname.  
    ls-asset-lstchngon = sy-datum.  
  
\*    Make entry in custom table  
    INSERT zasset\_details FROM ls\_asset.  
  
    IF sy-subrc <> 0.  
      lv\_msg =    'Error in creation'.  
      RAISE EXCEPTION TYPE /iwbep/cx\_mgw\_busi\_exception  
        EXPORTING  
          textid            = /iwbep/cx\_mgw\_busi\_exception=>business\_error  
          message\_container = lo\_message\_container  
          entity\_type       = iv\_entity\_name  
          message           = lv\_msg.  
    ELSE.  
      ls\_entity = CORRESPONDING #( ls-asset ).  
      er\_entity = ls\_entity.  
    ENDIF.  
  
  ENDMETHOD.

1. ASSETSET\_GET\_ENTITY

  METHOD assetset\_get\_entity.  
  
    DATA: lt\_keys              TYPE /iwbep/t\_mgw\_tech\_pairs,  
          ls\_key               TYPE /iwbep/s\_mgw\_tech\_pair,  
          ls\_entity            TYPE zcl\_zasset\_mpc=>ts\_asset,  
          ls\_return            TYPE bapiret2,  
          err\_msg              TYPE string,  
          lo\_message\_container TYPE REF TO /iwbep/if\_message\_container.  
  
    DATA: ls\_message TYPE scx\_t100key.  
  
\*   Cathch exception if key value is not provided  
    TRY.  
        DATA(lv\_assetid) = it\_key\_tab[ name = 'ASSET\_ID' ]-value.  
      CATCH cx\_sy\_itab\_line\_not\_found.  
    ENDTRY.  
  
\*   Read from table based on Asset ID.  
  SELECT SINGLE \* FROM zasset\_details INTO @DATA(ls\_item)  
      WHERE assetid = @lv\_assetid.  
  
    IF sy-subrc EQ 0.  
      ls\_entity = CORRESPONDING #( ls\_item ).  
    ENDIF.  
    er\_entity = ls\_entity.  
  
  ENDMETHOD.

1. ASSETSET\_GET\_ENTITYSET

  METHOD assetset\_get\_entityset.  
  
    DATA :ls\_entity    TYPE zcl\_zasset\_mpc=>ts\_asset,  
          lv\_assetid   LIKE ls\_entity-assetid,  
          lv\_assetname LIKE ls\_entity-assetname,  
          lv\_statuskey LIKE ls\_entity-status,  
          lv\_audience  LIKE ls\_entity-aud,  
          lv\_market    LIKE ls\_entity-markt,  
          lt\_entity    TYPE zcl\_zasset\_mpc=>tt\_asset.  
\*Object References ----------------------------------------------------\*  
  
\*   Get filter values  
    IF line\_exists( it\_filter\_select\_options[ property = 'AssetID' ]-select\_options[ 1 ] ).  
      lv\_assetid = it\_filter\_select\_options[ property = 'AssetID' ]-select\_options[ 1 ]-low.  
    ENDIF.  
  
    IF line\_exists( it\_filter\_select\_options[ property = 'AssetName' ]-select\_options[ 1 ] ).  
      lv\_assetname = it\_filter\_select\_options[ property = 'AssetName' ]-select\_options[ 1 ]-low.  
    ENDIF.  
  
    IF line\_exists( it\_filter\_select\_options[ property = 'ProductionStatusKey' ]-select\_options[ 1 ] ).  
      lv\_statuskey = it\_filter\_select\_options[ property = 'ProductionStatusKey' ]-select\_options[ 1 ]-low.  
    ENDIF.  
  
    IF line\_exists( it\_filter\_select\_options[ property = 'AssetTypeId' ]-select\_options[ 1 ] ).  
      lv\_audience = it\_filter\_select\_options[ property = 'AssetTypeId' ]-select\_options[ 1 ]-low.  
    ENDIF.  
  
    IF line\_exists( it\_filter\_select\_options[ property = 'Market' ]-select\_options[ 1 ] ).  
      lv\_market = it\_filter\_select\_options[ property = 'Market' ]-select\_options[ 1 ]-low.  
    ENDIF.

\*   Select all entries from table as per filter value given  
    SELECT \* FROM zaaset\_details INTO TABLE @DATA(lt\_asst\_data)  
           WHERE assetid EQ @lv\_assetid  
            AND assetname EQ @lv\_assetname  
            AND status EQ @lv\_statuskey  
            AND assettype EQ @lv\_audience  
            AND markt EQ @lv\_market.  
    IF sy-subrc = 0.  
      SORT lt\_asst\_data DESCENDING BY readydate.  
    ENDIF.  
  
\*   Pass selected data to result entity  
    et\_entityset[] = CORRESPONDING #( lt\_asst\_data ).  
  
  ENDMETHOD.

1. ASSETSET\_UPDATE\_ENTITY

  METHOD assetset\_update\_entity.  
  
    DATA: lt\_keys              TYPE /iwbep/t\_mgw\_tech\_pairs,  
          ls\_key               TYPE /iwbep/s\_mgw\_tech\_pair,  
          ls\_return            TYPE bapiret2,  
          err\_msg(100),  
          ls\_message           TYPE scx\_t100key,  
          lo\_message\_container TYPE REF TO /iwbep/if\_message\_container.  
  
\*   Read key from update call  
    TRY.  
        DATA(lv\_assetid) = it\_key\_tab[ name = 'ASSET\_ID' ]-value.  
      CATCH cx\_sy\_itab\_line\_not\_found.  
    ENDTRY.  
  
\*   Read asset details from custom table  
    SELECT SINGLE \* FROM zasset\_details INTO @DATA(ls\_item)  
    WHERE assetid EQ @lv\_assetid.  
  
    ls\_item-lastchangedby = sy-uname.  
    ls\_item-lastchangedon = sy-datum.  
  
\*   Update table entry  
    UPDATE zasset\_details FROM ls\_item.  
  
\*   If update is not successful handle exception  
    IF sy-subrc NE 0.  
      err\_msg = 'Error while updating'.  
      RAISE EXCEPTION TYPE /iwbep/cx\_mgw\_busi\_exception  
        EXPORTING  
          textid  = /iwbep/cx\_mgw\_busi\_exception=>business\_error  
          message = err\_msg.  
    ENDIF.  
  ENDMETHOD.