  METHOD /iwbep/if\_mgw\_appl\_srv\_runtime~create\_deep\_entity.  
  
\*&---------------------------------------------------------------------\*  
\*\*--------------------------Data Declaration--------------------------\*\*  
\*&---------------------------------------------------------------------\*  
    DATA: lr\_deep\_entity TYPE zcl\_z\_google\_processed\_mpc\_ext=>ts\_deep\_entity,  
          lo\_instance    TYPE REF TO zcl\_z\_update\_ocr\_extract,  
  
          lv\_success     TYPE boolean,  
          lv\_failed      TYPE boolean,  
          lv\_allowed     TYPE boolean,  
          lv\_invoice\_no  TYPE rbkp-belnr,  
          ex\_invoice\_no  TYPE bapi\_incinv\_fld-inv\_doc\_no,  
  
          it\_processed   TYPE STANDARD TABLE OF zcl\_z\_google\_processed\_mpc\_ext=>ts\_processed1,  
          et\_output      TYPE STANDARD TABLE OF zcl\_z\_google\_processed\_mpc\_ext=>ts\_processed1,  
          et\_exceptions  TYPE STANDARD TABLE OF zcl\_z\_google\_processed\_mpc\_ext=>ts\_exceptions1.  
  
    io\_data\_provider->read\_entry\_data( IMPORTING es\_data = lr\_deep\_entity ).  
  
    CREATE OBJECT lo\_instance.  
  
    it\_processed = lr\_deep\_entity-finger\_to\_processed.  
  
    CALL METHOD lo\_instance->determine\_to\_processed  
      EXPORTING  
        im\_payload   = lr\_deep\_entity  
        it\_processed = it\_processed  
      IMPORTING  
        et\_exception = et\_exceptions  
        et\_output    = et\_output.  
  
    CLEAR:lr\_deep\_entity-finger\_to\_processed.  
  
    lr\_deep\_entity-finger\_to\_processed  = et\_output.  
    lr\_deep\_entity-finger\_to\_exceptions = et\_exceptions.  
  
    copy\_data\_to\_ref( EXPORTING is\_data = lr\_deep\_entity  
                      CHANGING  cr\_data = er\_deep\_entity ).  
  ENDMETHOD.