

]>

Although DB2 allows the definition of multi-level collections it is currently not possible to access elements of nested complex types **directly** if the nesting level is more than one.

Following is a sample for a multi-level collection in Oracle. The type and variable definitions are accepted by DB2. But not the statements marked in red.

```
SUBTYPE tObjName IS VARCHAR2(320);

TYPE tRecAttrValue IS RECORD (
  vDataType      VARCHAR2(320),
  vStr           VARCHAR2(320),
  vNum           integer,
  vDate          date
);

TYPE tTabAttrValue IS TABLE OF tRecAttrValue INDEX BY typAttrName;

TYPE tRecContext IS RECORD (
  vEnabled        BOOLEAN,
  vObjName        tObjName,
  tAttrValue      tTabAttrValue           -- this is a multi-level collection, a table of records within a record
);
vContext        tRecContext;

BEGIN
  ...
  vContext.tTabAttrValue(vAttrName).Pos := ...  

  if vContext.tTabAttrValue.COUNT > 0 ...
  ...
END;
```

Here you see the same code rewritten to run on DB2 too. The new code is marked in blue.

```
vTmpRecAttrValue  tRecAttrValue;

BEGIN
  ...
  -- vContext.tTabAttrValue(vAttrName).Pos := ...
  vTmpRecAttrValue := vContext.tTabAttrValue(vAttrName);
  vTmpRecAttrValue.Pos := ...

  -- if vContext.tTabAttrValue.COUNT > 0 ...
  if vTmpRecAttrValue.COUNT > 0 ...

  -- This last line is clearly necessary if any inner elements of the temporary variable have been changed
  vContext.tTabAttrValue(vAttrName) := vTmpRecAttrValue;
  ...
END;
```