

2.4.3 DEFINE OBJECT Command

[Return to DidoDLL](#)

Purpose

[Return to Top](#)

Syntax

[Return to Top](#)

```
DEFINE OBJECT <ObjectName> AS
  ( { <AttributeName> : <DataType> [ NULL | NOT NULL] [ TOKEN | "" ] [ :=
<InitialValue | "" ] [, | "" ] }
  );
```

Where

- **ObjectName** The ObjectName can be qualified with a namespace by using periods (i.e., '.') within the name. Each namespace and the ObjectName are limited to 30 characters in length.

ObjectName ::= textual name associated with the Object.

- **AttributeName** The AttributeName can not be associated with a namespace and is limited to 30 characters in length.

AttributeName ::= textual name associated with the Attribute.

- **DataType** The Data Type can be one of the <PrimitiveTypes> or a Type created within the DIDO.

DataType ::= textual name of the Data Type associated with the attribute.

- **InitialValue (Optional)** The initial values are those required by the specific Data Type. For example, the GeoPosition requires three parameters (Lat, Long and Elevation). When multiple values are required, the values need to be surrounded by parenthesis.

InitialValue ::= An expression that is used to initialize the attribute value.

Examples

| Examples of Defining Constants in DidoLL

```
-- Setup some domain specific types to use when creating DIDO objects

--- PersonNameType
DEFINE TYPE PersonNameType
( VALUE      : IN Text NOT NULL NOT EMPTY
) AS
TEXT
WITH CONSTRAINTS
( DEFAULT    = NULL,
  COMPRESION = ONE,
  CASE       = TITLE,
  MAX_LENGTH = 30,
  VALUE      = VALUE
);

--- IdType
DEFINE TYPE IdType
( algorithm : IN Text NOT NULL NOT EMPTY := DEFAULT_ALGORITHM
  message   : IN Text NOT NULL NOT EMPTY := CURRENT_DATE,
  options   : IN Text                    := "Hex"
) AS
HASH
WITH CONSTRAINTS
( DEFAULT = algorithm,
  MESSAGE = message,
  OPTIONS = options
);

--- Create an object that doesn't have any tokens in it
DEFINE OBJECT MyDido.Student
( firstName : IN PersonNameType NOT NULL NOT EMPTY,
  lastName  : IN PersonNameType NOT NULL NOT EMPTY
) RETURN IdType AS
( StudentId : IdType          RETURN := NEW IdType ( message =
FirstName + LastName),
  FirstName : PersonNameType   := firstName,
  LastName  : PersonNameType   := lastName
);

--- Setup some domain specific constants to use when creating objects
DEFINE CONSTANT DEFAULT_CURRENCY_SYMBOL AS '$';
DEFINE CONSTANT DEFAULT_CURRENCY_NAME  AS "StudentCoin";
```

```

--- Create a type that describes the Student Coin
DEFINE TYPE SudentCoinType
  ( initialBalance : IN FIXED NOT NULL           := 0.00,
    currencyName   : IN Text NOT NULL NOT EMPTY :=
DEFAULT_CURRENCY_NAME,
    currencySymbol : IN Text NOT NULL NOT EMPTY :=
DEFAULT_CURRENCY_SYMBOL,
    PRECISION      : IN Text NOT NULL NOT EMPTY := "5,2",
    MIN            : IN FIXED NOT NULL           := -50.00,
    MAX            : IN FIXED NOT NULL           := 1000.00
  ) AS
FIXED
ALLOWING NONULL
WITH CONSTRAINTS
  ( MIN      = MIN,
    MAX      = MAX,
    VALUE    = initialBalance,
    UNITS    = currencyName,
    SYMBOL   = currencySymbol,
    PRECISION = PRECISION
  );

--- Create a Student Account that uses the StudentCoinType to track a
balance.
--- The Balance is also a TOKEN which marks the value as requiring a
Transaction
--- to prevent "Double Spend"
CREATE OBJECT OurDido.StudentAccount
  ( studentId : IdType NOT NULL,
    balance   : SudentCoinType NOT NULL
  ) RETURN AS IdType AS
  ( AccountId : IdType RETURN := NEW IdType ( message =
studentId + balance),
    StudentId : IdType := studentId,
    Balance   : SudentCoinType TOKEN := balance
  );

```

From:
<https://www.omgwiki.org/dido/> - DIDO Wiki

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:s_cli:05_contents:01_prt:06_dddl:objects&rev=1623769430

Last update: 2021/06/15 11:03

