

## Object snapshot

It is possible to obtain a snapshot of an object that shows its data-items, with offsets in the holding object, its object descriptor, and its current value.

The library for obtaining at snapshot is located in

BETAworld/LIB/MetaSystem

And example may be found in

BETAworld/workspace/DumpObjEx

## Usage

The following code produces a snapshot of the object DumpObjEx and prints it:

```
SN: ref ObjSnap
SN := ObjSnap(DumpObjEx)
SN.print
```

## Format of snapshot

We have the following kind of objects

- PlainObject
- ValueArrayObject
- ReferenceArrayObject

For a plain object with n data-items the format of the snap is as follows:

ObjectKind	K
DI1	
...	
DI2	
...	
Din	

- objectKind:K has one of the values:  
plainSnap, valueArraySnap, refArraySnap
- D1, D2, ... Dn is the snap for each of the n data-items

The data-items may be one of the following kinds:

- Primitive value object
- Compound value object
- Object reference

## Format of snap of primitive value object data-item

Name:	C1
	C2
...	...

	Cn
	0
offset:	off

size:	sz
kind:	K
descNo:	dn
val1:	V1
val2	V2

- Name:c1c2...c2n is the name of the data-item in the form of a string terminated by zero.
- offset : off is the offset of this data-item in the enclosing object
- size: sz is the size of the this data-item
- kind:K is the kind of the data-item
- descNo:dn is the internal number of the ObjectDescriptor for the object.
- val1: v1 is the value at the time when the snap was made
- val2: v2 – if K = float, then size = 2, and V2 is an additional value defining the float = (V1,V2). If size = 1, then V2 is not present.

### Format of snap of object reference data-item

Such a snap is similar to a snap of a primitive value object data-item. The only difference is that val1:V1 is a reference to an object or none. It is shown as @objId where objId is a unique number identifying the object being referred to