Abaqus Python for post-processing Cheat Sheet

by louislec via cheatography.com/38404/cs/11972/

Open the output database (.odb)

Import the post-processing library	`>>> from odbAccess import *`
open the output database	`>>> myOdb = openOd b(p ath ='p ath \my fil e.odb')`

Once you've created your odb object, you can access all the variables throught the different attributes of the class. In the following cheatsheet, we only point out the way to access the results that mostly interest us such as **Field Outputs** and **History Outputs**

Accessing to Field Outputs

In the **odb object** we will access the **steps** attribute, then the **frames** attribute to get our field output.

```
>>> steps = myOdb.s teps
>>> frames = steps[ 'my _st ep_ nam e'].frames
```

List of the main attributes in a step

- description
- domain
- frames
- historyRegion
- inertiaAboutCenter
- inertiaAboutOrigin
- loadCases
- mass
- massCenter
- name
- number
- previousStepName
- procedure
- timePeriod
- totalTime

List of the main attributes in a frame

frames.descr-iption	Return descrition if given
frames.domain	TIME or FREQUENCY
frames.field-Outputs	**return all the different outputs you specified in the simulation for each node and element**
frames.frameId	
frames.frame-Value	return the simulation time as a float
frames.incre-mentNumber	return the increment number as an integer

Which field outup do we usualy encounter

NODAL OUTPUTS	
U	displacement
UR	rotational displacement
COORD	coordinates
RF	Reaction force
ELEMENT OUTPUT	
S	
SMISES	

Exemple to access the displacement along the x axis (U1), for a given node:

```
>>> myOdb = openOd b(p ath ='p a th \my odb.odb')
>>> frame1 = myOdb.s te ps[ 'St - ep- 1'].fr ame s[-1]
>>>t = frame1.fr ame Value
>>>u1 = frame1.fi eld Out put [' U '].v al ues [27 ].d ata[0]
```

For all possible outputs, see the official Abaqus documentation

Main attributes in a fieldOutput object

- componantLabels
- description
- name
- type
- values — all the main values are stored in `v alues [my Nod e].data`

What about the instance

the **Root Assembly**	The odb object has an attribute rootAssembly, which gives informations about assembly such as:
elementSets	
elements	
instances	
name	
nodeSets	
nodes	
rigidBodies	
sectionAssignement	
surfaces	

Access History Output results

In your simulation, you may have specify **Histo** an attribute named **historyRegion** in which you historyOuptut Objects. In such objets are store

```
region = myOdb.s te ps[ 'st epN am -1.15']
```

Here is a quick exemple to access to time and

```
>>> myS tep.hi sto ryR egi ons ['N uts [CV OL].data
```

returns a tuple organized as:

```
((t1,C VOL 1), (t2 ,CV OL2 )...)
```