CDAS WPS Request Specification

version 2.0

1.Request Specification:

A request to the CDAS analysis service is of the form:

http://host:port/cdas?arg1=value1&arg2=value2&arg3=value3…

Here is a list of the WPS request arguments that are utilized by cdas:

request	"Execute"	Specifies a WPS 'Execute' request.
dialect	"cdas"	Specifies the request dialect (optional- only 'cdas' is currently supported).
embedded	true/false	If embedded=true then return the result object json-embedded in the request response, otherwise embed a URL to access the result via a separate request.
datainputs	json	Input arguments to the cdas process (see below).

2. Datainputs Specification:

2.1 Format:

The *datainputs* value consists of a list of parameter, i.e. *'arg=value'* specifications, e.g.: *[arg1=value1, arg2=value2, …].* There are currently three supported parameters: *variable, domain*, and *operation*. The values of these parameters are json encoded as described below.

2.2 Variable input:

The data input argument consists of a dictionary (or a list of) with keys specifying a data collection, a variable within the collection, and the id of the domain associated with the variable. The collection is specified using either the 'dset' parameter, whose value is the name of a server-defined data collection (associated internally with the location -OpenDAP or local file system- of a dataset), or the 'url' parameter, whose value is an OpenDAP address of the collection. The variable within the collection is specified using the 'id' parameter, whose value is the name of the variable within the collection. Since variable names within collections are not necessarily globally unique, it is also possible to specify a unique id for the variable using the syntax *id="uniqueId:varname".* The unique id defaults to the variable name if not specified. Here is an example variable specification:

```
variable=[ {"dset":"MERRA/mon/atmos","id":"v0:hur","domain":"d0"},
        {"url":"http://dataserver.nccs.nasa.gov/thredds/dodsC/bypass/CREATE-
        IP/MERRA/mon/atmos","id":"v1:clt","domain":"d1"} ]
```

2.3 Domain input:

The domain input argument consists of a dictionary (or a list of) with keys specifying the domain id, the domain grid, and a list of axis names which map to slice values. A slice value can be either a single value, or a range of values. The range can be represented either as a two-member list of the form *[start_value, end_value]* or as a dictionary with keys *start, end,* and *system*. The *system* parameter value is either "*values*" or "*indices*". The axis names can be shortened to three letters. Here are some example domain specifications:

```
domain =[  {"id":"d0","longitude":[-104.7,-50.6],"latitude":[21.3,44.7],"level":100000 },
        {"id":"d1","lon":50.2,"lat":21.3,"time":"2010-01-16T12:00:00"} ]
domain =[ {"id":"d2", "lon":-104.,"lat":21.3,"lev":100,"tim":"2010-01-16T12:00:00"},
        {"id":"bottom","level":{"start":0,"end":1,"system":"indices"}},
        {"id":"NH","lat":{"start":0.0,"end":90.0,"system":"values"}} ]
```

2.4 Operation Input:

The operation input argument value is a list of function invocations. Each function invocation is of the form *"kernel1.function1(arg1,arg2,...)".* Here is an example:

```
operation = [    "CDTime.departures( v0, slice:t )",
                " CDTime.climatology( $1, slice:t, bounds:annualcycle )",
                " CDTime.value(v0)",
                "CWT.average(*,axis=exy) ]
```

The kernel value specifies a package of server-side analysis functions, the function value specifies which function within that package is being invoked. The function arguments are either references to variables or keyword arguments of the form *key:value*. Variable references can be either variable unique ids (as defined in section 2.2), positional arguments (of the form $**k**, denoting the **k**th variable in the variable input list starting from 0), or a wildcard (the value ***** represents all variables in the variable input list). Function invocation results can be given a unique id and then used as inputs to other functions, for example:

```
operation = [    "va: CDTime.average( v0, slice:t )",
                " CDTime.difference( v0, va, slice:t )" ]
```

The operation returns a list of the function invocation results (which in "embedded" mode is json encoded and embedded in the request response). Function invocations that are assigned varIds are considered internal and (by default) not included in the returned invocation results list.