National Center for Atmospheric Research (NCAR).

High Altitude Observatory (HAO).

Boulder, CO.

IDL_CEDAR_catalog: An IDL interface to the CEDAR catalog on my MYSQL

Prepared By: Jose H. Garcia

This document describes the IDL interface to access the CEDAR catalog running on MYSQL.

Currently the latest schema for the catalog is 1.3.2.0. The IDL access is independent of the schema therefore the software can still be used with future versions of the catalog.

Conceptual design:

The IDL interface to access the CEDAR catalog (which we will refer from now on as ICD) is designed as a finite state machine. Once loaded into the address space of a IDL session, it will behave in ways that depend on previous states. Going from state to state is determined by invocations of the methods it exposes to the IDL interface.

ICD depends on the MySQL C++ application programming interface (API) called MySQL++. The current version in use is 1.7.

The credentials that ICD uses to log with the MySQL server are hard coded. Currently, with this set of credentials it is possible only to execute 'SELECT' statements from 'cedar' as the ICD host.

Methods in ICD:

1. LOAD_CATALOG_QUERY (query, rows, columns, results)

Table 1: Parameters

Name	Type	Usage	Meaning
query	an IDL string	INPUT	Query to be executed
rows	an IDL unsigned long variable	OUTPUT	Number of rows returned
columns	an IDL unsigned long variable	OUTPUT	Number of columns returned
results	an IDL unsigned long variable	OUTPUT	Address of the result set.

Return values:
0 = OK
1 = Bad Query (syntax wise)
2 = Bad Conversion.
3 = Undefined exception occurred.
4 = Wrong number of parameters.
5 = Bad parameter (incorrect data type of IDL parameter)

Example (IDL syntax):

```
query='select * from tbl_instrument'
rows=0ul
columns=0ul
results=0ul
val0=LOAD_CATALOG_QUERY (query, rows, columns, results)
```

Summary:
Executes a query and loads data into the results set.

2. GET_CELL(results,access_row,access_column,cell)

Table 2: Parameters

Name	Type	Usage	Meaning
results	an IDL unsigned long variable	INPUT	The address of a results set obtained via LOAD_CATALOG_QUERY
access_row	an IDL unsigned long variable	INPUT	Which row to access
access_column	an IDL unsigned long variable	INPUT	Which column to access

Table 2: Parameters

Name	Type	Usage	Meaning
cell	an IDL string	OUTPUT	The contents of the cell casted as an string

Return values:
0 = OK
1 = access row out of bounds
2 = Access column out of bounds
3 = Undefined exception occurred.
4 = Wrong number of parameters.
5 = Bad parameter (incorrect data type of IDL parameter)

Example (IDL syntax):

```
FOR i=0ul, ROWS DO BEGIN
  val1=GET_CELL(results,i,0ul,cell)
  if (val1 EQ 0UL) THEN BEGIN
     help, cell
  ENDIF
ENDFOR
```

Summary:
Get the contents (as an IDL string) of the cell in position access_row, access_column of a results set

3. DESTROY_RESULT_SET(results)

Table 3: Parameters

Name	Type	Usage	Meaning
results	an IDL unsigned long variable	INPUT	The address of a results set obtained via LOAD_CATALOG_QUERY

Return values:
0 = OK
1 = Wrong number of parameters.
2 = Bad parameter (incorrect data type of IDL parameter)
3 = The address indicated by results is set to NULL

Example (IDL syntax):

val3=DESTROY_RESULT_SET(results)

Summary:
Destroys a result set

Putting it all together:

This example shows how all methods can be used to get data from the catalog:

```
query='select * from tbl_instrument'
rows=0ul
columns=0ul
results=0ul
cell=''
access_row=0ul
access_column=1ul

val0=LOAD_CATALOG_QUERY (query, rows, columns, results)

if (val0 EQ 0UL)THEN BEGIN
   FOR access_row=0ul, rows DO BEGIN
     val1=GET_CELL(results,access_row,access_column,cell)
     if (val1 EQ 0UL) THEN BEGIN
        help, cell
     ENDIF
   ENDFOR
ENDIF

val3=DESTROY_RESULT_SET(results)
```

Architecture: