

Virtual DDBMS Resource Catalog Format

Draft v0.9.0.0

Copyright (C) 2008-2009 Aloysius Indrayanto & Chan Huah Yong
Grid Computing Lab - University Sains Malaysia

This document is licensed under the GNU Free Documentation License (GNU FDL) either version 1.3 of the license, or (at your option) any later version as published by the Free Software Foundation.

Note: In this draft (*major version is zero*), compatibility between versions are not guaranteed.

The VDDBMS engine needs the database **gcl_ddbms_system** to exist in the master node. This database will store all the configuration of the VDDBMS. This database must be accessible with full permission by user **gcl_ddbms_user_s** (can be changed) with password **gcl_ddbms_user_s** (can be changed).

SQL command for creating the table that lists all the data sources used by the VDDBMS:

```
CREATE TABLE IF NOT EXISTS resource_list
(
  ID          CHAR(4)   NOT NULL UNIQUE PRIMARY KEY
, Host       TEXT      NOT NULL
, Port       INTEGER   NOT NULL
, DBName     CHAR(255) NOT NULL
, Type       CHAR(16)  NOT NULL
, User       CHAR(128) NOT NULL
, Password   CHAR(128) NOT NULL
, ODBCOptions TEXT      NULL
, Description TEXT      NULL
, Flag       INTEGER   NULL
)
```

SQL command for creating the table that lists all the IDs of virtual databases and the number of virtual tables stored in each data source:

```
CREATE TABLE IF NOT EXISTS resource_usage_list
(
  ResourceID CHAR(4)   NOT NULL UNIQUE
, StoredDBs  TEXT      NOT NULL
, TotalTable INTEGER UNSIGNED NOT NULL
, Flag       INTEGER   NULL
)
```

SQL command for creating the table that lists all the specifications of the virtual databases managed by the VDDBMS:

```
CREATE TABLE IF NOT EXISTS user_database_list
(
  ID          CHAR(4)   NOT NULL UNIQUE PRIMARY KEY
, Name       CHAR(255) NOT NULL
, Description TEXT      NULL
, OwnerID    CHAR(4)   NOT NULL
, Flag       INTEGER   NULL
)
```

SQL command for creating the table that lists all the specifications of the virtual tables managed by the VDDDBMS:

```
CREATE TABLE IF NOT EXISTS user_table_list
(
    ID          CHAR(4) NOT NULL UNIQUE PRIMARY KEY
  , Name       CHAR(255) NOT NULL
  , DatabaseID CHAR(4) NOT NULL
  , Attributes TEXT      NULL
  , Description TEXT     NULL
  , AliasOf   CHAR(4)   NULL
  , ReadableBy TEXT     NULL
  , Flag      INTEGER   NULL
)
```

SQL command for creating the table that lists all the fragment-specifications of the virtual tables:

```
CREATE TABLE IF NOT EXISTS fragment_location_list
(
    TableID  CHAR(4) NOT NULL UNIQUE
  , FragMode CHAR(1) NOT NULL
  , VFragKey TEXT    NULL
  , HFragKey TEXT    NULL
  , Resources TEXT   NOT NULL
  , Flag     INTEGER NULL
)
```

SQL command for creating the table that lists all the name-mapping-rules of all aggregate virtual tables:

```
CREATE TABLE IF NOT EXISTS aggtable_mapping_list
(
    TableID  CHAR(4) NOT NULL
  , ResourceID CHAR(4) NOT NULL
  , VColName CHAR(255) NOT NULL
  , PTableName CHAR(255) NOT NULL
  , PColName  CHAR(255) NOT NULL
  , Flag      INTEGER   NULL
)
```

SQL command for creating the table that lists all the stored procedures:

```
CREATE TABLE IF NOT EXISTS stored_procedure_list
(
    Name          CHAR(255) NOT NULL
  , OwnerID      CHAR(4) NOT NULL
  , PubAccess    CHAR(1) NOT NULL
  , BindToDB    CHAR(255) NULL
  , Data        TEXT     NOT NULL
  , Description TEXT     NULL
  , OrgCode     TEXT     NULL
  , Flag        INTEGER   NULL
)
```

SQL command for creating the table that lists all the global-atomic-integer values:

```
CREATE TABLE IF NOT EXISTS giv_list
(
    Name          CHAR(255) NOT NULL UNIQUE PRIMARY KEY
  , LockedBy     TEXT      NULL
  , Value        BIGINT   NOT NULL
  , Flag         INTEGER   NULL
)
```