Virtual DDBMS C++ Network-Library API Documentation
Draft v0.8.0.0

The public API consist of three classes:

- **`Exception`**
- **`DBRow`**
- **`DDBMSClient`**

Please refer to the **Direct-Engine-Call API Documentation** for the API of the **`Exception`** and **`DBRow`** classes.

The DDBMSClient Class

Below are the public interface of the class :

```
class DDBMSClient {
    public:
        // Construct a new DDBMS client object
        DDBMSClient(const string& host, int port, const string& user, const string& password,
                    bool secure, bool tryUseCompression = true, int queryTimeoutInSecond = 30);

        // Standard dtor
        ~DDBMSClient();

        // Evaluate the given SQL statements; use the function fetchNextRow() to get the
        // the evaluation result
        void evalSQL(const string& sqlStatement);

        // Execute the given SQL statements; in case the statements produce row sets, use
        // the function fetchNextRow() to collect them
        void execSQL(const string& sqlStatements);

        // Prepare the given SQL statement
        void prepSQL(const string& sqlStatement);

        // Set values of the prepared SQL
        void prepSetVal(char id, const string& value);

        // Execute the prepared SQL
        void prepExec();

        // Fetch the next row; returns 'false' if no more row can be fetched
        bool fetchNextRow(DBRow& row, uint8_t maxCacheSize = 10);

        // Discard some rows
        void discardRow(uint32_t count);

        // Discard all pending rows
        void discardAll();

        // Add a new resource definition
        const string addResource(const string& type,        const string& host,
                                       int     port,        const string& dbName,
                                 const string& user,        const string& password,
                                 const string& odbcOptions, const string& description);

        // Delete a resource definition (it maybe not something you really want to do!)
        void removeResource(const string& resID);

        // Get the client library version
        void clientLibraryVersion(uint8_t& major, uint8_t& minor, uint8_t& revision,
                                  uint8_t& bugFix) const;

        // Get the client protocol version
        void clientProtocolVersion(uint8_t& major, uint8_t& minor, uint8_t& revision,
                                   uint8_t& bugFix) const;

        // Get the server protocol version
        void serverProtocolVersion(uint8_t& major, uint8_t& minor, uint8_t& revision,
                                   uint8_t& bugFix) const;

        // Ping the server
        bool ping() const;
};
```

The class interface should be self-explanatory.