STEP #0 connecting to a server

```
Connector connector = new Connector();
connector.setIp("127.0.0.1");
connector.setPort("1254");
```

STEP #1 Creating a Master Table (cf p 72 of the API reference document)

First step is to create a “master” table to store locally the data you provide:

```
connector. directExecute = true; // line by line mode
String structure = “title CHAR 254, text STRING”;
connector.TABLE_Create(“myName”,”NULL”,”100000”,”MASTER”, structure);
```

STEP #2 Inserting, learning and indexing your data

Once the master table instantiated, you may insert and/or learn and/or index your data.

In the previous step, we gave the table a simple structure made up of two fields named “title” which is defined to be a character array string containing 254 characters max and a field named “text” of STRING type (up to 2 GB character array).

Inserting data, learning it and indexing it:

```
connector. directExecute = true; // line by line mode
String[] colnames = new String[]{“title”, “text”}; // the table fields where to insert
String[] values = new String[]{“your title”, “your text”}; // the data to insert
connector.TABLE_Insert((“myName”,colnames,values); // inserting in table the data inside defined fields
String [] rowid = connector. getDataByName(“rowid”,-1); // we retrieve  the line number where data was inserted
// rowid[0] contains the line number where data was inserted
String signal = values[0]+”. “+values[1];
connector.directExecute = false; // script mode
connector.openScript(null); // script starts after this line
connector.SESSION_Store(signal,”ranked”,rowid[0]); // learning and indexing data
connector.CONTEXTS_Fetch(“-1’,”1”,”1”);
connector.CONTEXTS_Drop(“1”);
connector.executeScript(); // execute script on mARC server
String[] shapes = connector. getDataByName(“shapes”,1); // the indexation data
```

```
String[] activities = connector. getDataByName(“activity”,1); // corresponding activity
```

Note : Pure learning

You may also learn data without indexing it:

```
String signal = “the data to learn”;

connector.directExecute = false; // script mode

connector.openScript(null); // script starts after this line

connector.SESSION_Store(signal,”ranked”,”-1”); // learning but NO indexation

connector.CONTEXTS_Fetch(“-1’,”1”,”1”);

connector.CONTEXTS_Drop(“1”);

connector.executeScript(); // execute script on mARC server side

String[] shapes = connector. getDataByName(“shapes”,1); // the indexation data

String[] activities = connector. getDataByName(“activity”,1); // corresponding activity
```

The indexation data ready for your indexation engine is stored in the string array shapes with their associated activities

STEP #3 : basic querying

To query mARC server, the basic method is SESSION.stringToContext() (p 44 of API reference document).

```
String[] format = new String[]{“format = title text”}; // the format to show the results
String query = “orange agent Vietnam”;
connector.directExecute = false; // script mode
connector.openScript(null); // script starts after this line
connector.SESSION_Clear(“contexts”); // we empty the contexts stack
connector.SESSION_StringToContext(query,”false”); // the query
connector.SESSION_ContextToDoc(); // retrieve documents associated with the query
connector.RESULTS_GetProperties(“count”,”1”); // retrieve the number of returned documents
connector.CONTEXTS_SetProperties(“1”,format); // set the retrieve format of data
connector.CONTEXTS_Fetch(“-1”,”1”,”1”); // retrieve the documents from the query
connector.CONTEXTS_Drop(“1”);
connector.executeScript(); // execute script on mARC server side
String[] doc_numbers = connector. getDataByName(“count”,3);
int doc_number = Integer.parseInt(doc_numbers[0]); // the number of retrieved docs
String[] fields = new String[]{“title”,”text”}; // the fields to retrieve
String[][] results = new String[2][doc_number]; // 2D array containing the results for each field format
// we retrieve the data for each field format inside the 2D array
int i = 0;
        for (String s : fields)
        {
            results[i++] = connector.getDataByName(s, 5);
        }
```