Quick Reference to mod_ndb

Request handlers

ndb-cluster Direct apache requests to mod_ndb.

ndb-dump-format Dump a user-defined output format in readable compiled form.

ndb-exec-batch Execute a set of scripted subrequests in mod_ndb.

Per-server configuration

ndb-connectstring Connection string to NDB management server. The default connects to a management server on the local host.

ndb-max-read-subrequests Limit to the number of read-query subrequests when mod_ndb is scripted in PHP, mod_perl, etc. Default is 20.

Scripting

Call mod_ndb endpoints as Apache subrequests (*i.e.* with `virtual()` in PHP), then call into the ndb-exec-batch handler, then retrieve query results from the Apache notes table.

Apache Notes

`ndb_request_method`	set to POST or DELETE for write requests
`ndb_request_data`	set POST data for write requests
`ndb_send_result`	set to send results directly to client
`ndb_result_0` *etc.*	will contain results of read queries

Example:

```
<?php
  virtual("/ndb/ex/1/session/3");
  virtual("/ndb/ex/1/session/1");
  virtual("/ndb-commit-all");  #ndb-exec-batch handler

  echo "ndb_result_0: " . apache_note("ndb_result_0");
  echo "ndb_result_1: " . apache_note("ndb_result_1");

  $x = json_decode(apache_note("ndb_result_1",true));
  var_dump($x);
?>
```

<Location> sections

Database MySQL schema *(required)*.

Table Specify table *(required)*. Optionally, can denote that an endpoint returns a full-table scan or a full ordered index scan.

Syntax Table table_name [scan] *[ordered_index_name]*

Examples
```
Table users
Table users scan
Table users scan PRIMARY
```

Columns White space-separated list of columns to include in the result set.

AllowUpdate White space-separated list of columns that may be updated by a POST request.

Format The output format used for the response. The **JSON** internal format is the default. The **raw** internal format may be used if the output contains only one column (*e.g.* a BLOB). Otherwise this can name a user-defined output format.

PrimaryKey, UniqueIndex, OrderedIndex
Data access plans map *key column aliases* that can be used in HTTP requests to specific indexes in NDB. Key column aliases do not need to match actual column names, but index names must correspond to the internal index names as shown by the *ndb_desc* utility.

Syntax PrimaryKey column_alias *[column_alias ...]*

Examples
```
PrimaryKey car_id
PrimaryKey key_part_1 key_part_2
```

Syntax UniqueIndex index_name column_alias *[...]*

Example
```
UniqueIndex name$unique name
```

Syntax OrderedIndex index_name column_alias *[...] [sort_flag]*

Examples
```
OrderedIndex PRIMARY license
OrderedIndex year_month_idx year month
OrderedIndex PRIMARY license [ASC]
OrderedIndex PRIMARY license [DESC]
```

Filter defines a filter in an ordered index scan (*see wiki documentation*).

<Location> *(continued)*

Etags (On or Off). Whether to generate MD5 entity tags for each response, for use by proxy servers. Default is On.

Deletes (On or Off). Whether to accept HTTP DELETE requests. Default is Off.

PathInfo Associate the rightmost parts of the URL with key column aliases (which must be defined elsewhere in a PrimaryKey, UniqueIndex, or OrderedIndex directive).

Syntax PathInfo *column_alias/column_alias...*

Example
```
PathInfo user_id/icon_id
```

<ResultFormat> sections

Format syntax

`$name$`	column name
`$value/qj$`	column value, quoted, and escaped for JSON
`$1$`	value of first output column
`$row$`	expand first inner Row or Record named "row"
`...`	expand subsequent inner Rows or Records
	any other text is copied verbatim into the output.

Column Modifiers

`/q`	quote string values
`/Q`	quote both string and numeric values
`/x`	encode escapes for XML output
`/j`	encode escapes for JSON output

All formats must define a **Scan** loop and a **Row** loop. An optional **Record** format defines both a "non-null" and an (optional) "null" records format. Scan and Row formats have the syntax:

'start_text $inner_loop$ separator_text ... end_text'

Example:

```
<ResultFormat "XML">
 Scan scan '<NDBScan>\n$row$\n...\n</NDBScan>\n'
 Row  row  ' <NDBTuple> $attr$ \n ... </NDBTuple>'
 Record attr '<Attr name=$name/Q$ value=$value/Qx$ />'
</ResultFormat>
```

http://code.google.com/p/mod-ndb
http://forge.mysql.com/wiki/Mod_ndb