NIRVA Database service

Document version 1.29

Table of contents

Overview

The database service is a NIRVA external service that provides access to any kind of data source providing a driver for ODBC.

The database service allows access to any ODBC system data source configured on the server.

With this service, a user can search, add, modify and remove records from ODBC data sources. He can also send any kind of SQL statement or call stored procedures.

NIRVA SYSTEMS only certifies the following ODBC products and drivers:

- MICROSOFT Window ODBC system with drivers for SQL server and Access (Windows platform).
- DataDirect CONNECT FOR ODBC product with any DataDirect (ex Merant) associated driver (Windows and UNIX platforms).
- MYODBC version 3.51 with MYSQL. This has been tested successfully on Linux and Windows. For Linux, the driver must have been compiled with the thread safe option. The tested configuration on Linux was MYODBC 3.51.11 with unixODBC 2.2.11.

Any other ODBC product and/or driver can be used but is not supported by NIRVA SYSTEMS. Especially, for Oracle connection, the ODBC drivers delivered from Oracle software or Microsoft are not usable with the database service because they are not stable enough. For oracle, please use the DataDirect (ex Merant) drivers.

Login and logout

Users can login and logout to ODBC data sources.

Working on data sources

Users can add, modify remove and search into ODBC data sources.

Security

The service uses the security functions of the ODBC data source allowing protection for users, databases, records and functionality.

Performance

The performance of the database accesses depends of the ODBC driver used. This performance is generally nearly the same than for a native access.

For loading data, the DATABASE service must create a SELECT order that returns the minimum of records (no record is the best) but is fast and valid. Here is the algorithm used by the service to build this query:

- The service first searches for a column named “DOCN” that is not nullable and issues a query for “DOCN=NULL”.
- Otherwise, the service searches for an automatic column that is not nullable and issues the query “*FoundColumn*=NULL”.
- Otherwise, the service searches for a column that is not nullable and issues the query “*FoundColumn* =NULL”.

- Otherwise, the service issues a blank query retrieving all records of the table. This query may be long.

Considering this algorithm, the best performance will be obtained if the table contains a column named “DOCN” that is indexed and not nullable, or if the first column (whatever its name) is indexed and not nullable.

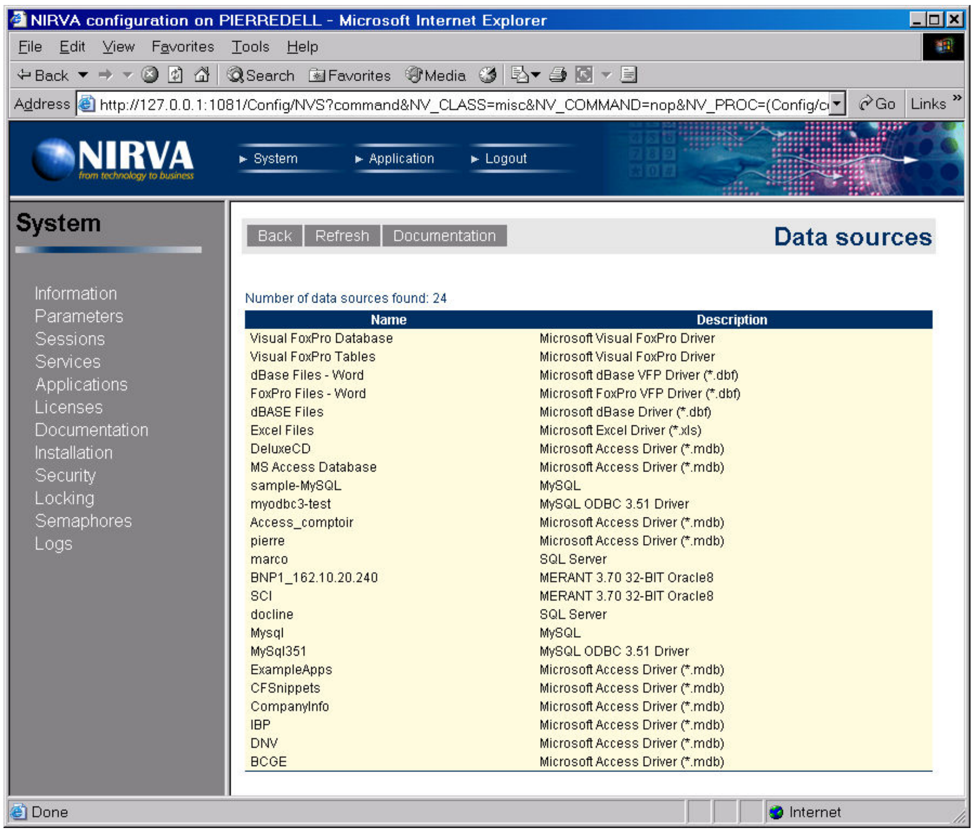
Installation

The database service is delivered as a NIRVA package and can be installed like any NIRVA service directly from the NIRVA configuration web site. Please see the NIRVA configuration chapter in the NIRVA user's guide for further information.

The service doesn't install the ODBC layer. The ODBC layer is native under windows. For UNIX platform, the DataDirect CONNECT FOR ODBC product must be installed (works also with MYODBC and unixODBC under Linux).

Configuration

The database service configuration is entirely dynamic and available from a web browser.
In fact, the service doesn't require any specific configuration so the configuration screen just displays the list of available system data sources.
The configuration of the database service is accessible directly from the main list of services of the NIRVA configuration web site:


NIRVA configuration on PIERREDELL - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://127.0.0.1:1081/Config/NVS?command&NV_CLASS=misc&NV_COMMAND=nop&NV_PROC=(Config/c
Go Links
NIRVA from technology to business
System Application Logout
System
Information
Parameters
Sessions
Services
Applications
Licenses
Documentation
Installation
Security
Locking
Semaphores
Logs
Back Refresh Documentation
Data sources
Done
Internet

Number of data sources found: 24

Name	Description
Visual FoxPro Database	Microsoft Visual FoxPro Driver
Visual FoxPro Tables	Microsoft Visual FoxPro Driver
dBase Files - Word	Microsoft dBase VFP Driver (*.dbf)
FoxPro Files - Word	Microsoft FoxPro VFP Driver (*.dbf)
dBASE Files	Microsoft dBase Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
DeluxeCD	Microsoft Access Driver (*.mdb)
MS Access Database	Microsoft Access Driver (*.mdb)
sample-MySQL	MySQL
myodbc3-test	MySQL ODBC 3.51 Driver
Access_comptoir	Microsoft Access Driver (*.mdb)
pierre	Microsoft Access Driver (*.mdb)
marco	SQL Server
BNP1_162.10.20.240	MERANT 3.70 32-BIT Oracle8
SCI	MERANT 3.70 32-BIT Oracle8
docline	SQL Server
Mysql	MySQL
MySql351	MySQL ODBC 3.51 Driver
ExampleApps	Microsoft Access Driver (*.mdb)
CFSnippets	Microsoft Access Driver (*.mdb)
CompanyInfo	Microsoft Access Driver (*.mdb)
IBP	Microsoft Access Driver (*.mdb)
DNV	Microsoft Access Driver (*.mdb)
BCGE	Microsoft Access Driver (*.mdb)

Reference

This chapter gives the complete reference of all the database service commands.

Classes

Here are the available DATABASE service classes:

-	DATABASE	Main class.
-	SOURCE	Data source information and requests.
-	TABLE	Table information.
-	QUERY	Result set management.

Error codes

DATABASE Class

Value	Description
101	Invalid file
102	Cannot copy file
103	Time out
104	Invalid command
105	Cannot connect ODBC layer
106	SQL Error
107	Invalid SQL Handle
108	Cannot open file
109	Cannot write file
110	Memory error (not enough)
111	Invalid parameter
112	Cannot open ODBC data source
113	Invalid ODBC data source name
114	ODBC data source not open
115	Cannot open query
116	No table available
117	Cannot get column
118	Invalid table
119	No current source selected
120	Invalid query
121	No record
122	No current query selected
123	Cannot get column information
124	Cannot get records
125	Last record
126	First record
127	The data source is read only
128	The table is read only
129	Cannot add records
130	Unsupported operation
131	Cannot execute SQL request
132	Cannot re-query
133	The table is not updatable
134	Cannot edit record
135	Cannot modify record

136	Cannot get the number of records
137	Cannot remove records
138	Cannot read file
139	No input records

Permissions

Name	Description

Commands

For each command, the reference gives the command name, the sources for which the command may be used, the command description, the eventual command permissions, the parameter list and the eventual list of objects created by the command.

The parameters described in this chapter are command specific parameters. For general parameters, please refer to the Nirva command syntax chapter.

The available sources are:

- Client for all Nirva client interfaces including Nirva client library (nvc).
- Web for commands from a web browser.
- Procedure for commands from a Nirva procedure.
- Service for commands from service to service.

DATABASE class

This is the standard service class that provides service scope commands.

NOP

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command does nothing but allows to test that the database service is on line and answers correctly. If the service is not on line, this command returns an error.

Parameters

None

SET_MAX_COLUMN_SIZE

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command allows to change the maximum size of the column buffer used for data binding.
The database service allocates buffers according the column sizes defined in the table. This column size cannot exceed the maximum value defined by this command. If a column size is greater than the maximum column size value, the data for this column will be truncated to the maximum column size value.
The default maximum column size buffer is 65536 bytes.

Parameters

SIZE	New value of the maximum column size. If the SIZE parameter is given and is less than 1000, the service set it to 1000. The default value is 65536.

SOURCE class

The SOURCE class provides commands for connecting data sources, for retrieving information about data sources and for working with them.

For one session, several data sources can be opened by the service. This number of data sources can be limited by the license.
After opening a data source, this one becomes the current one.

CALL

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command executes a stored procedure. The requested stored procedure must exist on the DBMS system. The result set (query) generated by this function is always a forward only result set.
The command creates a NIRVA indexed string list object that gives some information about the query and a NIRVA string list object that gives the list of columns.
This command is deprecated, please use the SOURCE:SQL command instead.

Parameters

PROCEDURE	Name of the stored procedure to execute.

Objects created

QUERY_RESULT This is a Nirva indexed string list object containing following keys:

- "IDENTIFIER" is the query identifier used in other QUERY class commands.
- "NUM_COLUMNS" is the number of columns returned by this statement. The column names are then given by the QUERY_COLUMNS return object.
- "SQL" is the SQL statement.
- "TABLE" is the table name.
- "NUM_RECORDS" is the number of records found. This value is generally correct but some drivers may be not able to return it. At this time, it's set to 0 and the command QUERY:NUM_RECORDS may be used to retrieve the exact number of records. In fact, the command returns the number of records via a call to the SQL COUNT() function.
- "FETCH" has always the value "FORWARDONLY" for a stored procedure call.

QUERY_COLUMNS This is a Nirva string list object containing the list of columns retrieved by the CALL statement.

CLOSE

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command closes a previously opened ODBC data source. All queries eventually associated to the opened data source are automatically closed by this command.

Parameters

NAME	Name of the data source to close. If this parameter is provided, the given source is closed. If it's not provided, the current source is closed.

COMMIT

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command commits any modification made from last TRANSAC command to the database..

Parameters

None

DELETE

Source	Use Input container	Use output container
Client Web Procedure Service	Yes	No

Description

This command removes the selected records of the current query from the table. The current query must exist. After removing records, the service issues a requery on the current query and the cursor is positioned to the first document of the query.

The command fails if the table doesn't have at least one primary key.

The indexes of the records to remove may come from a NIRVA string list object or from a NIRVA file object.

If the record comes from a string list object, each item of the list contains the index of the record to remove. This index is a numeric index relative to the query. The first record has the index 1. If one of the indexes is set to "ALL", then all the records of the current query are deleted.

If the record comes from a file object, the file must contain the indexes of the records to remove. Each index must be on a separated line. If one of the indexes is set to "ALL", then all the records of the current query are deleted.

Here is an example of input file that removes records 3, 8 and 9 of the current query:

```
3
8
9
```

If the source accept transaction, the database service automatically removes the records in a transaction context and rolls back in case of error.

Parameters

RECORDS	Name of the NIRVA object containing the records to remove. The NIRVA object must be in the input container. It can be a string list object or a file object. Please see the description of the command for information about these objects.

INFO

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns information about the given source. The source must have been opened before with the SOURCE:OPEN command.
The command creates a NIRVA indexed string list object returning following data source information:

- Name.
- Data source provider.
- Multiquery flag.
- Read only flag.
- Forward only flag.
- Quote character.
- Conversion functions flag.

Parameters

NAME	Name of the data source for which to get information. If this parameter is not provided, the current source is used.

Objects created

SOURCE_INFO This is a Nirva index string list object containing following keys:

- “NAME” is the data source name.
- “DBMS” is the name of the data source provider.
- "MULTIQUERY" is set to “YES” if the data source accepts several queries simultaneously (otherwise it’s set to “NO”).
- "READONLY" is set to “YES” if the data source is in read only mode (otherwise it’s set to “NO”).
- "FORWARDONLY" is set to “YES” if the data source has a forward only cursor (otherwise it’s set to “NO”).
- "QUOTECHAR" is the character used for delimiting table names in the SELECT or INSERT statements.
- "CONVERSION" is set to “YES” if the data source accepts SQL conversion functions (otherwise it’s set to “NO”).

INSERT

Source	Use Input container	Use output container
Client Web Procedure Service	Yes	No

Description

This command adds new records to the given table of the current opened source.

The new records to add may come from a NIRVA table object or from a NIRVA file object. The loading performance is better with a file object so this solution may be preferable when an important block of records has to be added into a table.

If the records come from a table object, the table column names of the NIRVA object must correspond to the table column names of the data source.

If the records come from a file object, the file object must have the following format:

- Each record is separated by the string "[CN=%DOC_SEPARATOR%]".
- Each column is separated by the string "[CN=*ColumnName*]" where ColumnName is the name of the column.
- The content of the columns must correspond to their definition in the table (for example, if the column is integer, an integer must be given and not an alphanumeric string). Otherwise, the function will fail.
- The date columns must have the following format: YYYY-MM-DD.
- The time columns must have the following format: HH:MM:SS.
- The time stamp columns must have the following format: YYYY-MM-DD HH:MM:SS:mmm (mmm is the number of milliseconds).
- If the column has been defined as automatic, it is not necessary to deliver it (The database system will set its value automatically).

In fact, the file format is the same that the format delivered with the QUERY: GET_RECORDS command. Here is an example of input file:

```
[CN=%DOC_SEPARATOR%]
[CN=Name]
Toto
[CN=Age]
21
[CN=Photo]
V1-H1\L6\124
[CN=%DOC_SEPARATOR%]
[CN=Name]
Titi
[CN=Age]
45
[CN=Photo]
V1-H1\L6\125
```

For loading data, the DATABASE service must create a SELECT order that returns the minimum of records (no record is the best) but is fast and valid. Here is the algorithm used by the service to build this query:

- The service first see if a WHERE parameter has been given to the command. If yes, it uses the value of this parameter as the WHERE clause of the search: “SELECT * from *tablename* WHERE *whereclause*". Where *tablename* is the content of the TABLE parameter and *whereclause* is the content of the WHERE parameter.

- The service first searches for a column named "DOCN" that is not nullable and issues a query for "DOCN=NULL".
- Otherwise, the service searches for an automatic column that is not nullable and issues the query "*FoundColumn*=NULL".
- Otherwise, the service searches for a column that is not nullable and issues the query "*FoundColumn* =NULL".
- Otherwise, the service issues a blank query retrieving all records of the table. This query may be long.

Considering this algorithm, the best performance will be obtained if the table contains a column named "DOCN" that is indexed and not nullable, or if the first column (whatever its name) is indexed and not nullable.

If the source accept transaction, the database service automatically adds the records in a transaction context and rolls back in case of error.

Parameters

TABLE	Name of the table for new records. The table name is generally composed of 2 parts: *owner.table* where *owner* is the owner of the table and *table* is the name of the table. For some drivers owner is not necessary. The owner of a table can be retrieved with the TABLE:LIST command. If the owner or the table name contains a blank character, both must be enquoted with the quote character returned by the SOURCE:INFO or TABLE:LIST commands.
WHERE	Optional WHERE clause use to make an empty result set for adding documents. If this parameter is not provided, the system automatically creates a WHERE clause following an algorithm given above.
RECORDS	Name of the NIRVA object containing new records to insert into the table. The NIRVA object must be in the input container. It can be a table object or a file object. Please see the description of the command for information about the file format if a file object is to be used.

LIST

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns the list of available ODBC data sources defined on the server. Under windows, only the system sources are taken in care.
The command creates a table object returning following data source information:

- Name.
- Description.

Parameters

None

Objects created

SOURCE_LIST	This is a Nirva table object containing following columns:

- "NAME" is the data source name.
- "DESCRIPTION is the data source description.

OPEN

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command opens an ODBC data source. This source becomes the current source (it's not necessary to issue a SOURCE:SET_CURRENT command on it). Several data sources can be opened simultaneously for a single NIRVA session. The maximum number of data sources by session is controlled by the license of the database service.

Parameters

NAME	Name of the ODBC data source to open. This name must be a valid ODBC data source defined on the server. It's possible to get the list of valid sources by issuing a SOURCE:LIST command. The NAME parameter is mandatory.
USER	Optional user name for connecting the source. This is the specific data source user name that can be different than the user named of the NIRVA session.
PASSWORD	Eventual password for user identification.
CURSOR	Specifies how the Driver Manager uses the ODBC cursor library. Values can be "ODBC" or "DRIVER". If set to "ODBC", the Driver Manager uses the ODBC cursor library. If set to "DRIVER", the Driver Manager uses the scrolling capabilities of the driver. This is the default setting. It's not recommended to use "ODBC" for big size result sets.
STRIP_TRAILING_SPACES	By default, the database service strips the trailing spaces of columns when retrieving records. If this parameter is set to "NO", the trailing spaces are preserved.
QUERY_TIME_OUT	Sets the time out value for all queries issued for the current data source. The default time out is 60 seconds. This value can be changed later using the SET_QUERY_TIME_OUT command.

ROLLBACK

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command rools back any modification made from last TRANSAC command.

Parameters

None

SELECT

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command executes an SQL SELECT statement. Please read the SQL reference for the syntax of SQL statements. The returned query becomes the current query. The name of the table eventually used in the select statement is generally composed of 2 parts: *owner.table* where *owner* is the owner of the table and *table* is the name of the table. For some drivers owner is not necessary. The owner of a table can be retrieved with the TABLE:LIST command. If the owner or the table name contains a blank character, both must be enquoted with the quote character returned by the SOURCE:INFO or TABLE:LIST commands.
The command creates a NIRVA indexed string list object that gives some information about the query and a NIRVA string list object that gives the list of columns.

Parameters

SELECT	complete SELECT SQL statement to proceed including the SELECT keyword at the beginning.
ERROR_NO_DOC	When set to “NO”, the command doesn’t produce an error if no documents has been found on the database. At this time, the resulting object QUERY_RESULT is created but there is no query identifier and the NUM_RECORDS key is set to 0. When ERROR_NO_DOC is set to “YES”, the command produces an error if no documents has been found. The resulting objects are then not created. The default is “YES”.

CLEAN_QUERY	When set to “YES”, the command does some cleaning on the SQL string: it replaces tabs with space, it replaces line feeds with space, it removes carriage returns and concatenates consecutive spaces. The default is “YES”.
NUM_DOCS_METHOD	This parameter defines which method to use for counting documents. It can take values “LOOP” or “SQL”. If “LOOP”, Nirva is directly counting by moving from first to last record. If “SQL”, Nirva is adding some SQL code to the given query in order to retrieve the number of documents. “LOOP” is faster in terms of database server resource but is slower on Nirva server CPU because it has to iterate threw all records. SQL is fast at Nirva level but slower on database server because it has to execute 2 queries. You can use generally the “SQL” method except if you know by advance that you will get a small number of records. At this time use the “LOOP” method. The default is SQL.

Objects created

QUERY_RESULT

This is a Nirva indexed string list object containing following keys:

- "IDENTIFIER" is the query identifier used in other QUERY class commands.
- "NUM_COLUMNS" is the number of columns returned by this statement. The column names are then given by the QUERY_COLUMNS return object.
- "SQL" is the SQL statement.
- "TABLE" is the table name.
- "NUM_RECORDS" is the number of records found. This value is generally correct but some drivers may be not able to return it. At this time, it’s set to 0 and the command QUERY:NUM_RECORDS may be used to retrieve the exact number of records. In fact, the SELECT command returns the number of records via a call to the SQL COUNT() function.
- "FETCH" can take the value "FORWARDONLY" if the data source accepts only forward cursors or "ANY" otherwise.

QUERY_COLUMNS

This is a Nirva string list object containing the list of columns retrieved by the SELECT statement.

SET_CURRENT

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command sets the current ODBC data source. Many service commands works only on the current source. The source must have been opened before by the SOURCE:OPEN command.

Parameters

NAME	Name of the data source to become the current one. This parameter is mandatory.

SET_QUERY_TIME_OUT

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command sets the time out value for all queries issued for the current data source. The default time out is 60 seconds.

Parameters

QUERY_TIME_OUT	New time out value in seconds. The minimum value is 1 second. The default value is 60 seconds.

SQL

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command executes any SQL statement.
It works in 2 different modes following the SELECT option.
If the SELECT option is set, the command returns information exactly in the same way than a SELECT command. So the result set stay opened and the QUERY class commands can be used to access data. This must be used to send SGBD specific SQL orders that are able to return a table (like a SELECT order). For example, this command is used internally in order to get the "auto_increment" flag of the MYSQL columns (the order is "SHOW COLUMNS FROM *TableName*").
If the SELECT command is not set, the SQL command just sends the order to the SGBD without requiring any result.

When the SELECT option is set, the result set is available only in read mode and the REQUERY command fails.

The SQL command can also be used to execute a stored procedure that return parameters or not. If the stored procedure doesn't return parameter, the SQL command must be used without the SELECT option. If the stored procedure return parameters, the command must be used with both the SELECT and STORED_PROCEDURE parameters set to "YES". At this time, the resulting result set is a forward only result set so in order to get the documents, one can issue the QUERY:GET_RECORDS command with

WHAT parameter set to “ALL” or loop with QUERY:GET_RECORDS with WHAT parameter set to “FIRST” and then WHAT parameter set to “NEXT”.

Here is an example of calling a stored procedure returning a value with SQL server:

Definition of the stored procedure on SQL server side that searches into a table and returns the number of records found:

```
CREATE PROCEDURE GetStateCount ( @State char(2), @StateCount int OUTPUT )
AS
SELECT @StateCount = Count(*)
FROM authors
WHERE State = @State
go
```

For searching from Nirva, here is the necessary commands (we suppose that the data source is opened):

```
NV_CMD=|DATABASE:SOURCE:SQL| SQL=|DECLARE @TheCount int EXEC GetStateCount
@State = 'UT', @StateCount = @TheCount OUTPUT Select TheCount = @TheCount|
SELECT=|YES| STORED_PROCEDURE=|YES|
NV_CMD=|DATABASE:QUERY:GET_RECORDS| WHAT=|ALL|
```

This returns a result set containing one column named “TheCount”.
When STORED_PARAMETER is set to “YES”, all other queries must be have been purged before and the resulting query must also be purged before issuing any other queries.

The possibility to return values from stored procedures is driver dependant and some drivers doesn’t accept it (For example, some versions of MyOdbc doesn’t work with stored procedure returning a value).

Parameters

SQL	SQL statement to execute.
SELECT	SELECT option. If this parameter is set to “YES”, the SQL order is processed like a SELECT command (the result set stay opened and resulting data can be get by using the QUERY class commands). If the SELECT parameter is set to “NO”, the SQL order is just sent to the SGBD without requiring any result. There is no result set. The default is “NO”.
STORED_PROCEDURE	This parameter has meaning only when the SELECT parameter has been set to “YES”. It tells Nirva that the query contains a call to a stored procedure that returns parameters as a result set. The default is “NO”.
ERROR_NO_DOC	This parameter has measning only when the SELECT parameter has been set to “YES”. When ERROR_NO_DOC is set to “NO”, the command doesn’t produce an error if no documents has been found on the database. At this time, the resulting object QUERY_RESULT is created but there is no query identifier and the NUM_RECORDS key is set to 0. When ERROR_NO_DOC is set to “YES”, the command produces an error if no documents has been found. The resulting objects are then not created. The default is “YES”.
CLEAN_QUERY	When set to “YES” and when the select option has been set to “YES”, the command does some cleaning on the SQL string: it replaces tabs with space, it replaces line feeds with space, it removes carriage returns and concatenates consecutive spaces. The default is “YES” (no impact if SELECT is set to”NO”).

NUM_DOCS_METHOD	This parameter has measning only when the SELECT parameter has been set to “YES”. This parameter defines which method to use for counting documents. It can take values “LOOP” or “SQL”. If “LOOP”, Nirva is directly counting by moving from first to last record. If “SQL”, Nirva is adding some SQL code to the given query in order to retreive the number of documents. “LOOP” is faster in terms of database server resource but is slower on Nirva server CPU because it has to iterate threw all records. SQL is fast at Nirva level but slower on database server because it has to execute 2 queries. You can use generally the “SQL” method except if you know by advance that you will get a small number of records. At this time use the “LOOP” method. If STORED_PRECEDURE parameters is set to “YES”, the NUM_DOCS_METHOD parameter has ne meaning. The default is “SQL”.

<u>Objects created</u>

SQL_RESULT	This object is created only when the SELECT option is not set. This is a Nirva indexed string list object containing following keys: - "COMMAND" contains the SQL statement. - "RESULT" gives the status of the statement. For now, this status is always set to "OK". - "AFFECTED" gives the number of rows affected by the SQL order when this one is an UPDATE, DELETE or INSERT statement. This information is not implemented in all ODBC drivers. If not available, the value is 0.
QUERY_RESULT	This object is created only when the SELECT option is set. This is a Nirva indexed string list object containing following keys: - "IDENTIFIER" is the query identifier used in other QUERY class commands. - "NUM_COLUMNS" is the number of columns returned by this statement. The column names are then given by the QUERY_COLUMNS return object. - "SQL" is the SQL statement. - "TABLE" is the table name. - "NUM_RECORDS" is the number of records found. This value is generally correct but some drivers may be not able to return it. At this time, it’s set to 0 and the command QUERY:NUM_RECORDS may be used to retrieve the exact number of records. In fact, the command returns the number of records via a call to the SQL COUNT() function. - "FETCH" can take the value "FORWARDONLY" if the data source accepts only forward cursors or "ANY" otherwise.
QUERY_COLUMNS	This object is created only when the SELECT option is set. This is a Nirva string list object containing the list of columns retrieved by the SQL statement.

TRANSAC

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command enters transaction mode.
Any source modification made in transaction mode can be rolled back or committed by the way of the corresponding commands.

Parameters

None

UPDATE

Source	Use Input container	Use output container
Client Web Procedure Service	Yes	No

Description

This command modifies the current record of the current query of the current source. The current record must have been set before by the QUERY:GET_RECORDS command.
The command fails if the table doesn't have at least one primary key.

The new record content may come from a NIRVA table object or from a NIRVA file object.

If the record comes from a table object, the table column names of the NIRVA object must correspond to the table column names of the data source.

If the record comes from a file object, the file object must have the following format:

- Each column is separated by the string "[CN=*ColumnName*]" where ColumnName is the name of the column.
- The content of the columns must correspond to their definition in the table (for example, if the column is integer, an integer must be given and not an alphanumeric string). Otherwise, the function will fail.
- The date columns must have the following format: YYYY-MM-DD.
- The time columns must have the following format: HH:MM:SS.
- The time stamp columns must have the following format: YYYY-MM-DD HH:MM:SS:mmm (mmm is the number of milliseconds).

Here is an example of input file:

```
[CN=Name]
```

```
Titi
[CN=Age]
22
```

Only the columns found in the input object will be changed. The record is physically modified into the database. The current query must exist.

Parameters

RECORD	Name of the NIRVA object containing the record columns to update. The NIRVA object must be in the input container. It can be a table object or a file object. Please see the description of the command for information about the file format if a file object is to be used.

TABLE class

The SOURCE class provides commands for retrieving information about tables of the current data source.

COLUMN_LIST

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns the list of columns of a single table of the current data source. The command creates a table object returning following column information:

- Name.
- Description.
- Data type.
- Type name.
- Precision.
- Length.
- Scale.
- Radix.
- Nullable flag.
- Automatic flag.
- Primary flag.

Parameters

TABLE	This parameter is the real name of the table to get columns without quote characters or owner name. This parameter is mandatory.

Objects created

COLUMN_LIST	This is a Nirva table object containing following columns: - "NAME" is the column name. The driver returns an empty string for a column that does not have a name. - "DESCRIPTION" is a a description of the column. - "DATATYPE" is the SQL data type. This can be an ODBC SQL data type or a driver-specific SQL data type. Please consult the ODBC or SQL references for available data types. - "TYPENAME" is the data source – dependent data type name; for example, “CHAR”, “VARCHAR”, “MONEY”, “LONG VARBINAR”, or “CHAR ( ) FOR BIT DATA”. - "PRECISION" is the column size. - "LENGTH" is the length of the column. - "SCALE" is the number of decimal digits of the column. - "RADIX" is for numeric data types, either 10 or 2. If it is 10, the values in PRECISION and SCALE give the number of decimal digits allowed for the column. For example, a DECIMAL(12,5) column would return a RADIX of 10, a PRECISION of 12, and a SCALE of 5; a FLOAT column could return a RADIX of 10, a PRECISION of 15 and a SCALE of NULL. If it is 2, the values in PRECISION and SCALE give the number of bits allowed in the column. For example, a FLOAT column could return a RADIX of 2, a PRECISION of 53, and a SCALE of NULL. NULL is returned for data types where RADIX is not applicable. - "NULLABLE" is set to "Yes" if the column can be empty. Otherwise, it is set to No. - "AUTO" is set to "Yes" if the column is automatically calculated by the database system. Otherwise, it is set to No. - "PRIMARY" is set to "Yes" if the column is a primary key. Otherwise, it is set to No.

LIST

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns information about all or a single table of the current data source.
The command creates a table object returning following table information:

- Name.
- Description.
- Owner.
- Qualifier.
- Type.
- Quote character.
- Conversion flag.

Parameters

TABLE	If this parameter is provided, the service only returns information about the given table. This parameter is the real name of the table without quote characters or owner name.

Objects created

TABLE_LIST This is a Nirva table object containing following columns:

- "NAME" is the table name.
- “DESCRIPTION” is the table description.
- "OWNER" is the owner of the table.
- "QUALIFIER" is the name of the database containing the table.
- "TYPE" is set to “TABLE”.
- "QUOTECHAR" is the character used for delimiting table names in the SELECT or INSERT statements.
- "CONVERSION" is set to “YES” if the table accepts SQL conversion functions (otherwise it is set to “NO”).

PRIVILEGES

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns read, add, update and delete privileges of the given table of the current source. The command creates a table object returning following column information:

- SELECT privilege.
- INSERT privilege.
- UPDATE privilege.
- DELETE privilege.

Parameters

TABLE	This parameter is the real name of the table to get privileges. This parameter is mandatory.

Objects created

TABLE_PRIVILEGE This is a Nirva table object containing following columns:

- "SELECT" is set to “YES” if the user has the right to send SELECT or CALL orders to the table. Otherwise, the value is “NO”.
- "INSERT" is set to “YES” if the user has the right to send INSERT orders to the table. Otherwise, the value is “NO”.
- "UPDATE" is set to “YES” if the user has the right to send UPDATE orders to the table. Otherwise, the value is “NO”.
- "DELETE" is set to “YES” if the user has the right to send DELETE orders to the table. Otherwise, the value is “NO”.

QUERY class

The QUERY class provides commands for queries. A query is the result set generated by a SOURCE:SELECT or SOURCE:CALL command. Each data source may maintain several queries (if the data source accepts it).

GET_CURRENT

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command retrieves the current query identifier if there is one.

Parameters

None

Objects created

QUERY	This is a Nirva string object containing the current query identifier.

GET_RECORDS

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command gets the requested records from the current query. The current query must exist.
The last retreived record becomes the current record. The current record can be used in the UPDATE command.
The records can be put into a NIRVA table object or into a NIRVA file object.

If the records go to a table object, the table column names of the NIRVA object correspond to the table column names of the data source.

If the records go to a file object, the file object have the following format:

- Each record is separated by the string "[CN=%DOC_SEPARATOR%]".
- Each column is separated by the string "[CN=*ColumnName*]" where ColumnName is the name of the column.
- The date columns have the following format: YYYY-MM-DD.
- The time columns have the following format: HH:MM:SS.
- The time stamp columns have the following format: YYYY-MM-DD HH:MM:SS:mmm (mmm is the number of milliseconds).

Here is an example of output file:

```
[CN=%DOC_SEPARATOR%]
[CN=stor_id]
6380
[CN=ord_num]
6871
[CN=ord_date]
1994-09-14 00:00:00.000
[CN=qty]
5
[CN=payterms]
Net 60
[CN=title_id]
BU1032
[CN=%DOC_SEPARATOR%]
[CN=stor_id]
8042
[CN=ord_num]
423LL930
[CN=ord_date]
1994-09-14 00:00:00.000
[CN=qty]
10
[CN=payterms]
ON invoice
[CN=title_id]
BU1032
[CN=%DOC_SEPARATOR%]
[CN=stor_id]
8042
[CN=ord_num]
P723
[CN=ord_date]
1993-03-11 00:00:00.000
[CN=qty]
25
[CN=payterms]
Net 30
[CN=title_id]
BU1111
```

Parameters

FILE	Flag telling if the output of the command must go to a NIRVA file or table object. If it’s set to “YES”, the selected records go to a file object. The object name itself is pointed by the RECORDS parameter. The default is “NO” (table output).
RECORDS	Name of the NIRVA object that will contain the selected documents. This can be a file or table object following the value of the FILE parameter. The default is “RECORDS”.

WHAT	Records to get. This parameter can take the following values: - ALL : retrieves all the records of the query. This is the default. - FIRST: retrieves the first record. - LAST: retrieves the last record. - NEXT: retrieves the next record. - PREV: retrieves the previous record. - CURRENT: retrieves the current record. - n: retrieves nth record. The first record has number 1. - n-m: retrieves records number n to m. The first record has number 1.
ROWBUFSIZE	Size of the row buffer. This is only available with Nirva version at least 2.5.011. It allows to group records into a buffer and to send a unique command to Nirva to populate the resulting table object. This can save time when retreiving an important amount of records. The default value is 20. This parameter has meaning only when table output is requested.

Objects created

RECORDS	Selected records. The name of this object can be changed by using the RECORDS parameter. This is a Nirva file or table object, depending of the value of the FILE parameter. Please see the command description for information about the RECORDS object.

INFO

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command retrieves information about the current query.
The command creates a NIRVA indexed string list object that gives some information about the query and a NIRVA string list object that gives the list of columns.

Parameters

None

Objects created

QUERY_RESULT	This is a Nirva indexed string list object containing following keys: - "IDENTIFIER" is the query identifier used in other QUERY class commands. - "NUM_COLUMNS" is the number of columns returned by this statement. The column names are then given by the QUERY_COLUMNS return object. - "SQL" is the SQL statement. - "TABLE" is the table name.

- "NUM_RECORDS" is the number of records found. This value is generally correct but some drivers may be not able to return it. At this time, it's set to 0 and the command QUERY:NUM_RECORDS may be used to retrieve the exact number of records. In fact, the SELECT command returns the number of records via a call to the SQL COUNT() function.
- "FETCH" can take the value "FORWARDONLY" if the data source accepts only forward cursors or "ANY" otherwise.
- "POSITION" is the actual cursor position into the list of found records. It is a number between 1 and the number of found documents. If the cursor is below the first document, POSITION takes the value "HOME". If the cursor is after the last document, POSITION takes the value "END". If the cursor position is unknown, POSITION takes the value "UNDEFINED".

QUERY_COLUMNS This is a Nirva string list object containing the list of columns retrieved by the SELECT or CALL statement.

LIST

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns information about all queries of the current data source.
The command creates a table object returning following query information:

- Identifier.
- Number of columns.
- SQL statement.
- Table name.

Parameters

None

Objects created

QUERY_LIST This is a Nirva table object containing following columns:

- "IDENTIFIER" is the query identifier.
- "NUM_COLUMNS " is the number of columns returned by the SQL statement of the query statement.
- "SQL" is the SQL statement.
- "TABLE" is the table name.

NUM_RECORDS

Source	Use Input container	Use output container
Client Web Procedure Service	No	Yes

Description

This command returns the number of records of the current query. This command may be used if the SOURCE:SELECT command did not returned a value for the number of found records. Depending of the driver used, this function can be slow. With the drivers recommended for the NIRVA database service, it's not necessary to call this command because the SOURCE:SELECT command correctly returns the number of records.

Parameters

EXACT	If this parameter is set to "YES", the database service will use cursor functions in order to retrieve the number of documents of the current query. Otherwise, it uses the result of an SQL COUNT() function call. The default value is "NO".

Objects created

NUM_RECORDS	This is a Nirva string object containing the number of records of the current query.

PURGE

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command removes the given query. After being removed, a query is not usable any more. All queries maintained by a given data source are automatically removed when the data source is closed. If the removed query is the current one, then the current query is undefined.

Parameters

QUERY	Query identifier to remove. The query identifier is returned by the SOURCE:SELECT command. If this parameter is not provided, the current query is removed. Then, the current query is undefined. If this parameter is set to "ALL", all the queries are removed.

REQUERY

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command restarts the SQL statements of the current query. This is useful after removing documents or if the database content has changed. The current query must exist.

Parameters

None

SET_CURRENT

Source	Use Input container	Use output container
Client Web Procedure Service	No	No

Description

This command sets the current query. This command is only useful if several queries have to be used (and if the source accepts it) because the SOURCE:SELECT command always sets the result query as the current one.

Parameters

QUERY	Query identifier to set as the current one. The query identifier is returned by the SOURCE:SELECT command.