

Sedna Administration Guide

Contents

1	Introduction	1
2	Sedna Administration via Command Line	2
2.1	Sedna Directory Structure	2
2.1.1	Sedna Installed from Precompiled Binaries	2
2.1.2	Sedna Installed from RPM or DEB Packages	2
2.1.3	Sedna Installed from Source Code	3
2.2	Managing Sedna	3
2.2.1	Running Sedna	3
2.2.2	Stopping Sedna	4
2.2.3	Configuring Sedna	4
2.3	Managing Databases	5
2.3.1	Creating a Database	5
2.3.2	Configuring a Database	7
2.3.3	Deleting a Database	8
2.3.4	Running a Database	8
2.3.5	Stopping a Database	9
2.4	Running queries	9
2.4.1	se_trn utility	9
2.4.2	Sedna Terminal	10
2.5	The Examples of Using Command Line Utilities	12

1 Introduction

This guide describes administration of the Sedna XML Database Management System (Sedna for short). Sedna has the following components.

Governor serves as “control center” of the system. All other components register at the Governor. The Governor knows which other components (e.g. databases and sessions) are running in the system and controls them. Other

components cannot function properly if the Governor is not running so you must start it to use Sedna. Besides, Governor is responsible for handling remote client's requests. For each request it creates a session which keeps the direct connection with client. Then the client interacts with the system via session. You may run only one instance of the Governor.

Session keeps the client's settings and allows client to run a sequence of transactions. The transactions within a session are strictly serial, that is one transaction ends before next one starts, there is only one active transaction per session. Transaction provides functionality required for execution of the client's queries. The query execution consists of several steps: (1) parsing of the query and translation of the query into its logical representation, (2) processing of the logical representation by optimizer that produces the optimal query execution plan, (3) execution of the optimal query plan and passing results to the client.

Storage manager manages a database and provides memory management functionality to the executor. There is an instance of storage manager for each database run.

All the components described above are implemented as operating system processes.

One can run session locally via command line or as a remote client via remote request to Governor. In both cases, first of all the Governor and storage managers (one for each databases required) components should be run. For each client a session is created. In turn, within session client can run a sequence of transactions.

There are a set of command line utilities that allow you to administrate all aspects of Sedna. These utilities are described in detail in the following section.

2 Sedna Administration via Command Line

2.1 Sedna Directory Structure

The Sedna directory hierarchy is fundamental to obtaining an overall understanding of the system. This section describes how the Sedna directory structure is set up and what it contains for different kinds of Sedna packages.

2.1.1 Sedna Installed from Precompiled Binaries

If you install Sedna from precompiled binaries, the directory structures are identical on Windows and Linux platforms. Below `PREFIX` refers to the direc-

tory where Sedna is installed.

PREFIX/bin	command line utilities for managing Sedna and databases, running queries, etc.
PREFIX/cfg	database configuration files named <db_name>_cfg.xml, where <db_name> is the name of the corresponding database
PREFIX/data	database data stored in subdirectories named <db_name>_files, where <db_name> is the name of the corresponding database
PREFIX/doc	Sedna documentation
PREFIX/driver	API drivers for various programming languages
PREFIX/etc	Sedna config file sednaconf.xml (optional)
/etc	Sedna config file sednaconf.xml if it is not found in PREFIX/etc (also optional)
PREFIX/examples	example databases and applications for the Sedna APIs

2.1.2 Sedna Installed from RPM or DEB Packages

If you install Sedna from RPM or DEB packages on Linux, the directory structure is as follows. Below PREFIX refers to the directory where Sedna is installed. SHARE_PREFIX is defined as /usr/share if PREFIX is / and SHARE_PREFIX is defined as PREFIX/share otherwise.

PREFIX/bin/se_*	command line utilities for managing Sedna and databases, running queries
PREFIX/etc/sednaconf.xml	Sedna configuration file (optional)
/etc/sednaconf.xml	Sedna configuration file if it is not found in PREFIX/etc (also optional)
PREFIX/include/libsedna.h	C API header file
PREFIX/lib/libsedna.a	C API driver for Sedna
SHARE_PREFIX/doc/sedna-x.x.x	Sedna documentation
SHARE_PREFIX/java/sednadriver.jar	Java API Driver for Sedna
/var/lib/sedna/cfg	database configuration files named <db_name>_cfg.xml, where <db_name> is the name of the corresponding database
/var/lib/sedna/data	databases stored in subdirectories named <db_name>_files, where <db_name> is the name of the corresponding database
SHARE_PREFIX/sedna-x.x.x/examples	example databases and applications for

	the Sedna APIs
SHARE_PREFIX/sedna-x.x.x/driver	additonal API drivers

2.1.3 Sedna Installed from Source Code

If you install Sedna from source code using `make install`, the directory structure is the same as described in Section 2.1.1. If you install Sedna from source code using `make dispersal_install`, the directory structure is the same as described in Section 2.1.2. Note that the latter is available only on Linux.

2.2 Managing Sedna

2.2.1 Running Sedna

To start the Sedna server one should start *Governor* by executing the `se_gov` command. This starts the main server component. After it Administrator can run databases (as discussed in Section 2.3.4).

The usage of the `se_gov` command is as follows:

Usage: `se_gov [options]`

options:

<code>--help</code>	display this help and exit
<code>-help</code>	display this help and exit
<code>-version</code>	display product version and exit
<code>-background-mode on/off</code>	start the server in the background mode (default on)
<code>-port-number <int></code>	socket listening port (default 5050)

The first and second options are straightforward, typing these options Administrator can get the exhaustive information about the `se_gov` command. The third option allows to get the version of the *Governor*. The fourth option `-background-mode` allows Administrator to start *Governor* in the background mode. Finally, last option `-port-number` allows to set up the Sedna connection port number. The default value is 5050.

2.2.2 Stopping Sedna

To stop Sedna, Administrator should use the `se_stop` command. This command stops all Sedna components including all databases, which are run.

The command should be run without input parameters. But the command has several optional parameters.

The usage of the `se_stop` command is as follows:

Usage: `se_stop` [options]

options:

<code>--help</code>	display this help and exit
<code>-help</code>	display this help and exit
<code>-version</code>	display product version and exit

The first and second options are straightforward, typing these options Administrator can get the exhaustive information about the `se_stop` command. Option `-version` allows to get the version of the `se_stop`.

2.2.3 Configuring Sedna

Sedna configuration parameters are stored in the configuration file `sednaconf.xml` located in `[win:] PREFIX\etc` `[linux:] PREFIX/etc` or `[linux:]/etc`. On Linux, `sednaconf.xml` is searched in the following order: `verb!PREFIX/etc!`, `/etc`. The file is optional. If there is no such file, the default values are used. The file is an XML document, which satisfies the following DTD:

```
<!ELEMENT sednaconf (sedna_data)>
<!ELEMENT sedna_data (#PCDATA)>
```

The `sedna_data` element contains the path to the directory where database configuration files (in the `cfg` subdirectory) and databases (in the `data` subdirectory) are stored. The path by default is `[win:] PREFIX` `[linux:] /var/lib/sedna`.

To tune the Sedna configuration, Administrator should edit the proper elements of the Sedna configuration file.

2.3 Managing Databases

2.3.1 Creating a Database

To create a database, use the `se_cdb` command.

Each database consists of the following files, which are located as described in Section 2.1.

- The <db_name>.data file is used for storing persistent XML data, which are loaded to the database by user using LOAD/UPDATE expressions.
- The <db_name>.tmp file is used for storing temporary XML data. It appears in the database during query execution and contains intermediate results.
- The <db_name>.ph and <db_name>.ph.bu files are used for storing XML metadata (for example, the descriptive scheme of XML documents).
- The buf directory contains some intermediate data. It is created when the database is run and removed when the database is stopped.

Besides, for each database there are run-time configuration parameters, which are stored in the database configuration file (see Section 2.3.2).

The usage of the `se_cdb` command is as follows:

Usage: `se_cdb [options] db_name`

options:

<code>-help</code>	display this help and exit
<code>--help</code>	display this help and exit
<code>-version</code>	display product version and exit
<code>-data-file-max-size Mbs</code>	the max size of data file (in Mb), infinite size by default
<code>-tmp-file-max-size Mbs</code>	the max size of tmp file (in Mb), infinite size by default
<code>-data-file-ext-portion Mbs</code>	the data file extending portion size (in Mb), default 100Mb
<code>-tmp-file-ext-portion Mbs</code>	the tmp file extending portion size (in Mb), default 100Mb
<code>-data-file-init-size Mbs</code>	the data file initial size (in Mb), default 100Mb
<code>-tmp-file-init-size Mbs</code>	the tmp file initial size (in Mb), default 100Mb
<code>-persistent-heap-size Mbs</code>	the persistent heap size (in Mb), default 10Mb
<code>-bufs-num N</code>	the number of buffers in main memory, default 1600
<code>-max-trs-num N</code>	the number of concurrent micro transactions over database, default 10

`db_name` the name of the database to be created

The first and second options are straightforward, typing one of these options Administrator can get the exhaustive information about the `se_cdb` command.

The third option allows to get the version of the `se_cdb` command.

The `-data-file-max-size` option allows Administrator to set up the maximum size (in Mb) of the file where the persistent data are stored. The default value for this option is infinite.

The `-tmp-file-max-size` option allows Administrator to set up the maximum size (in Mb) of the file where the temporary data are stored. The default value for this option is infinite.

The `-data-file-ext-portion` option allows Administrator to set up the size (in Mb) of the portion to which the persistent data file will be extended as result of executing resize operation. Resize operation is applied to the persistent data file if its size is less than needed for storing XML documents. The default value is 100 Mb.

The `-tmp-file-ext-portion` option allows Administrator to set up the size of the portion to which the temporary data file will be extended as result of executing resize operation. The resize operation is applied to the temporary data file if its size is less than needed for storing temporary data (intermediate results of the query, for example). The default value is 100 Mb.

The `-data-file-init-size` option allows Administrator to set up the initial size (in Mb) of the persistent data file. It is obvious that the initial size can't be greater than the maximum size of the persistent data file. The default value is 100 Mb.

The `-tmp-file-init-size` option allows Administrator to set up the initial size (in Mb) of the temporary data file. It is obvious that the initial size can't be greater than the maximum size of the temporary data file. The default value is 100 Mb.

The `-persistent-heap-size` option allows Administrator to set up the size (in Mb) of the metadata file (for example the descriptive schema stores in this file) This file cannot be resized. The default value is 10 Mb.

The `-bufs-num` option allows Administrator to set up the number of buffers in main memory. This number is connected with the size of the main memory, but the performance of the database depends of this number very much. The default value is 1600 buffers, which is equal to 100Mb and we do not recommend to use less number of buffers. The size of one buffer is 64Kb and can't be customized by Administrator.

The `-max-trs-num` option is the parameter of the Sedna storage manager and we will not discuss the meaning of this parameter in this guide. We recommend to use default value.

The last parameter `db_name` is the name of the database to be created.

Administrator can use the arbitrary sequence of latin characters and digits for database name. There cannot be two databases with the same name.

2.3.2 Configuring a Database

For each database there is a set of run-time configuration parameters, which store in the database configuration file. The name of the database configuration file is `<db_name>_cfg.xml` where `<db_name>` is the name of the database. The location of configuration file is `[win:] PREFIX\cfg` `[linux:] PREFIX/cfg` or `[linux:] /var/lib/sedna/cfg` or as specified in the Sedna configuration file (see Section 2.2.3) for both Windows and Linux.

The configuration file is an XML document, which satisfies the following DTD:

```
<!ELEMENT db (name, bufs_num, max_trs_num)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT bufs_num (#PCDATA)>
<!ELEMENT max_trs_num (#PCDATA)>
```

The root of the XML document is the `db` element, which contains three elements: `name`, `bufs_num`, `max_trs_num`. The `name` element describes the name of the database. The `bufs_num` element describes the number of buffers in main memory for Storage Manager component. The semantic of `max_trs_num` element we will define in next version of Sedna.

To tune the database run-time configuration, Administrator should edit the proper elements of the database configuration file.

2.3.3 Deleting a Database

To delete the database with name `db_name` Administrator only should delete directory `<db_name>_files` located in `[win:] PREFIX\data` `[linux:] PREFIX/data` or `[linux:] /var/lib/sedna/data` or as specified in the Sedna configuration file for both Windows and Linux, and file `<db_name>_cfg.xml` located in `[win:] PREFIX\cfg` `[linux:] PREFIX/cfg` or `[linux:] /var/lib/sedna/cfg` or as specified in the Sedna configuration file for both Windows and Linux. See Section 2.2.3 to read about the Sedna configuration file.

2.3.4 Running a Database

To run the database with name `db_name` Administrator should use `se_sm` command. The run-time configuration parameters of the database can be set by means of `se_sm` command line options. If configuration parameter is not set

via command line option, than the database storage manager retrieves the default value of this parameter from configuration file (see section 2.3.2).

The usage of the `se_sm` command is as follows:

Usage: `se_sm [options] dbname`

options:

<code>-help</code>	display this help and exit
<code>--help</code>	display this help and exit
<code>-version</code>	display product version and exit
<code>-background-mode on/off</code>	start the server in the background mode (default on)
<code>-bufs-num N</code>	the number of buffers in main memory, (default value retrieved from config file)
<code>-max-trs-num N</code>	the number of concurrent micro transactions over database, (default value retrieved from config file)
<code>db-name</code>	The name of the database

The first and second options are straightforward, typing this option Administrator can get the exhaustive information about the `se_sm` command.

The third option allows to get the version of the `se_sm` command. The `-background-mode` option allows Administrator to start database in the background mode.

The `-bufs-num` option allows to set up the number of buffers in main memory. This number is connected with the size of the main memory, but the performance of the database depends of this number very much. The default value is retrieved from configuration file. The size of one buffer is 64Kb and can't be customized.

The `-max-trs-num` option is the parameter of the Sedna storage manager and we will not discuss the meaning of this parameter in the current version of Sedna. We recommend to use default value from configuration file.

The last parameter `dbname` defines the name of the database to be started up.

Note: before running any database be sure that Sedna Server is started up (see section 2.2.1).

2.3.5 Stopping a Database

To stop the database with name `db_name`, Administrator should use `se_smsd` command. This command takes one input parameter, which defines the name of the database to be shut down. The usage of the `se_smsd` command is as follows:

Usage: `se_smsd [options] dbname`

options:

<code>--help</code>	display this help and exit
<code>-help</code>	display this help and exit
<code>-version</code>	display product version and exit
<code>db-name</code>	The name of the database

The first and second options are straightforward, typing these options Administrator can get the exhaustive information about the `se_smsd` command. The third option allows to get the version of the `se_smsd`.

2.4 Running queries

2.4.1 `se_trn` utility

There are several ways to execute query over Sedna, but in this section we present the simplest way to do it. To execute the query over Sedna, the `se_trn` command should be used. This command requires two input parameters. The first one is the name of the database, to which the query is addressed and the second one is the file with `.xquery` extension, which contains the user query. Besides, there are several options, which semantics will be explained below. About supported QUERY/UPDATE language see in the Sedna Programmer's Guide. The result of the query is directed to standard output stream (stdout).

The usage of the `se_trn` is as follows:

Usage: `se_trn [options] dbname filename`

options:

<code>-help</code>	display this help and exit
<code>--help</code>	display this help and exit
<code>-version</code>	display product version and exit
<code>-show-time on/off</code>	show time of query execution (default off)
<code>-print-intermed on/off</code>	print intermediate results for debug purposes (default off)
<code>-query-type type</code>	type of the query to execute (XQuery, POR), (default XQuery)
<code>-name name</code>	user name (default SYSTEM)
<code>-pswd password</code>	user password (default MANAGER)
<code>db-name</code>	database name
<code>filename</code>	file with an XQuery query

The first and second options are straightforward, typing this option Administrator can get the exhaustive information about the `se_trn` command.

The third option allows to get the version of the `se_trn` command.

The `-show-time` option allows to know how much time the client's query were executed.

The `-print-intermed` and `-query-type` options are used for debug purposes and we don't recommend to use them.

The `-name` and `-pswd` options are reserved for further Sedna features.

2.4.2 Sedna Terminal

`se_term` is an interactive terminal to Sedna DBMS. It is a C application that uses Sedna C API (see "Sedna Programmer's Guide") to work with Sedna DBMS. It enables you to open a session to one of the Sedna database, type in queries interactively, issue them to the database and see the query results. Alternatively, input can be from a file, or a single query can be passed for execution as a command line parameter. In addition, it provides a number of meta-commands.

The usage of the `se_term` is as follows:

Usage: `se_term` [options] dbname

options:

<code>-help</code>	display this help and exit
<code>--help</code>	display this help and exit
<code>-version</code>	display product version and exit
<code>-file filename</code>	file with an XQuery query
<code>-query query</code>	XQuery query to execute
<code>-show-time on/off</code>	show time of query execution (default off)
<code>-host host</code>	hostname of the machine with Sedna running
<code>-port-number port</code>	socket listening port (default 5050)
<code>-name name</code>	user name (default SYSTEM)
<code>-pswd password</code>	user password (default MANAGER)
<code>db-name</code>	database name

Typing `-help` or `--help` Administrator provides the exhaustive information about the `se_term`.

`-version` option allows to get the version of the `se_term`.

`-file` option specifies the name of the file that contains any number of queries and meta-commands. `se_term` uses this file as the source of queries and meta-commands instead of reading them interactively. `se_term` executes the queries and meta-commands from file consequently and exit.

`-query` option specifies the query to execute.

The `-show-time` option allows to get the time of the latest query execution. If used with the `-query` option provides the time of the specified query execution; if used with the `-file` option provides the time of the execution of the last query in file.

`-host` option specifies the name of the machine with Sedna DBMS running. If not used, default value `localhost` is used.

`-port` option specifies the TCP/IP port on which Sedna server is listening to client applications. If omitted, port number 5050 is used by default.

`-name` and `-password` options specify the user name and the password. If omitted, connects as a predefined user `SYSTEM` with password `MANAGER`.

`dbname` is a non-option argument. It specifies the name of the database to connect to.

`se_term` returns 0 to the shell if it finished normally, 1 if there were errors.

When `se_term` is used in the interactive mode it takes in query/update statements and meta-commands. To execute a query/update statement type in a statement and use semicolon and line feed to terminate it (`;\n` terminates the query). An end of line does not terminate the query, thus queries can be spread over several lines for clarity. The first query/update statement automatically begins transaction. If the statement executed successfully, the results are displayed on the screen.

Meta-commands are commands for `se_term` that proccessed by the `se_term` itself. A meta-command begins with a backslash and that differs it from the query/update statements. Thus, the format of `se_term` meta-command is the backslash, followed immediately by a command, with no semicolon at the end. `se_term` takes in the following meta-commands:

```
\? - for help on internal slash commands
\commit - to commit transaction
\rollback - to rollback transaction
\showtime - to show the time of the latest query execution
\quit - to close session and quit the Sedna Terminal
```

2.5 The Examples of Using Command Line Utilities

In this section we present several examples of using the utilities described in the previous sections. These examples demonstrate how to create the database, run the Sedna server and the database, load documents into the database, run the query over the database, and finally stop Sedna.

For creating a database named `xmark` with default settings use the following command:

```
se_cdb xmark
```

To run Sedna, type the following command:

```
se_gov
```

After execution of these commands the created database can be started up. To do it, type the following command:

```
se_sm xmark
```

After it the xmark database is ready for serving user queries. Below we show how to load an XML document to the xmark database and how to write a queries over this document.

To load the XML document to the xmark database, pass the following steps:

1. Create the following file with name `load.xquery`:

```
LOAD "<path>" "<name>"
```

The `<path>` is the path to the XML document to be loaded to the xmark database. The `<name>` is the name of this document in the xmark database.

2. Run the command:

```
se_trn xmark load.xquery
```

3. Create the following file with name `query1.xquery`:

```
document("<name>")/*
```

Where `<name>` is the name of the loaded document in the xmark database.

4. Type the command and enter it:

```
se_trn xmark query1.xquery
```

To stop Sedna and all its components and databases, type and enter the following command:

```
se_stop
```