

Sequoia & DODS

Sequoia & DODS

Together Teamlösungen EDV-Dienstleistungen GmbH

Elmargasse 2-4

A-1190

Vienna

Austria

+43 (0) 5 04 04 - 122

+43 (0) 5 04 04 - 11 122

<office@together.at>

<http://www.together.at/together/index.html>

Copyright © 2006 Together Teamlösungen EDV-Dienstleistungen GmbH

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the Together Teamlösungen EDV-Dienstleistungen GmbH.

Together Teamlösungen EDV-Dienstleistungen GmbH DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1. What is Sequoia?	1
2. Sequoia Driver	2
3. Sequoia JDBC URL	3
4. Additional Information	5

Chapter 1. What is Sequoia?

Sequoia is a database cluster middleware that allows any Java™ application (standalone application, servlet or EJB™ container, ...) to transparently access a cluster of databases through JDBC™. You do not have to modify client applications, application servers or database server software. You just have to ensure that all database accesses are performed through Sequoia.

Sequoia is a free, open source project that is the continuation of the C-JDBC project (<http://c-jdbc.objectweb.org>) hosted by the ObjectWeb Consortium (<http://www.objectweb.org/>). Sequoia is licensed under an Apache v2 license (<http://www.apache.org/licenses/LICENSE-2.0.html>) is licensed whereas C-JDBC is available under the GNU Lesser General Public License (<http://www.gnu.org/copyleft/lesser.html>) (LGPL).

Sequoia also provides driver for non-Java applications. These developments are hosted in the Carob project (<http://carob.continuent.org>). An Eclipse plug-in for Sequoia is also available in the Oak project (<http://oak.continuent.org>).

Chapter 2. Sequoia Driver

The Sequoia driver can be loaded as any standard JDBC driver from the client program. Full driver class name is:

```
org.continuent.sequoia.driver.Driver
```

Note: The sequoia-driver.jar file (deployed with DODS) must be in the client classpath else the driver will fail to load.

Chapter 3. Sequoia JDBC URL

The JDBC URL expected for the use with Sequoia is the following:

```
jdbc:sequoia://host1:port1,host2:port2/database
```

host is the machine name (or IP address) where the Sequoia controller is running, port is the port the controller is listening for client connections.

At least one host must be specified but a list of comma separated hosts can be specified. If several hosts are given, one is picked up randomly from the list. If the currently selected controller fails, another one is automatically picked up from the list.

The port is optional in the URL and the default port number is 25322 if it is omitted. Those two examples are equivalent:

```
jdbc:sequoia://localhost/tpcw  
jdbc:sequoia://localhost:25322/tpcw
```

Examples using two controllers for fault tolerance:

```
jdbc:sequoia://c1.continuent.org,c2.objectweb.org/tpcw  
jdbc:sequoia://localhost,remote.continuent.org:2048/tpcw  
jdbc:sequoia://smpnode.com:25322,smpnode.com:1098/tpcw
```

URL options

The Sequoia driver accepts additional options to override the default behavior of the driver. The options are appended at the end of the Sequoia URL after a question mark followed by a list of ampersands separated options. Here is an example:

```
jdbc:sequoia://host/db?user=me&password=secret
```

Another option is to use semicolons to delimit the start of options and options themselves. Example:

```
jdbc:sequoia://host/db;user=me;password=secret
```

The recognized options are:

- **connectionPooling:** By default the Sequoia driver does transparent connection pooling on your behalf meaning that when `connection.close()` is called, the connection is not physically closed but rather put in a pool for reuse within the next 5 seconds. Set this to false if you do not want the driver to perform transparent connection pooling.
- **debugLevel:** Debug level that can be set to 'debug', 'info' or 'off' to display driver related information on the standard output. Default is off.
- **escapeBackslash:** Set this to false if you don't want to escape backslashes when performing escape processing of PreparedStatements, default is true.
- **escapeSingleQuote:** Set this to false if you don't want to escape single quotes (') when performing escape processing of PreparedStatements, default is true.
- **escapeCharacter:** Character to prepend and append to the String values when performing escape processing of PreparedStatements, default is a single quote.
- **user:** user login

- password: user password
- preferredController: defines the strategy to use to choose a preferred controller to connect to.

```
jdbc:sequoia://node1,node2,node3/myDB?preferredController=ordered
```

Always connect to node1, and if not available then try to node2 and finally if none are available try node3.

```
jdbc:sequoia://node1,node2,node3/myDB?preferredController=random
```

Pickup a controller node randomly (default strategy)

```
jdbc:sequoia://node1,node2:25343,node3/myDB?preferredController=node2:25343,node3
```

Round-robin between node2 and node3, fallback to node1 if none of node2 and node3 is available.

```
jdbc:sequoia://node1,node2,node3/myDB?preferredController=roundRobin
```

Round robin starting with first node in URL.

- retryIntervalInMs: once a controller has died, the driver will try to reconnect to this controller every retryIntervalInMs to see if the backend is back online. The default is 5000 (5 seconds).

NOTE: For additional Sequoia configuration details, please consult Sequoia documentation.

Chapter 4. Additional Information

Please note that we didn't include complete Sequoia binary distribution in DODS. We only included Sequoia driver and offered some configuration support.

Concrete Sequoia controller configuration and instantiation are done independently from DODS!!!