

SRS 3D

Administrator's Guide

LION bioscience AG

Juli 2002

Copyright 2002 LION bioscience AG. All rights reserved.

LION bioscience AG SRS 3D 1.1 documentation

This manual, as well as the software described in it, is furnished under license and may only be used or copied in accordance with the terms of such license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by LION bioscience AG. LION bioscience AG assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of LION bioscience AG.

Please remember that existing artwork or images that you may desire to scan as a template for a new image is protected under copyright law. The unauthorized incorporation of such artwork or images into new work could be a violation of the rights of the authors. Please be sure to obtain any permission required from the authors.

TABLE OF CONTENTS

1	SRS 3D Overview	1
	1.1. Introduction.....	3
2	SERVER INSTALLATION.....	5
	2.1. Requirements	7
	2.2. Software Installation	8
3	CLIENT INSTALLATION	13
	3.1. Introduction.....	15
	3.2. Requirements	15
	3.3. Customizing the Client Default Setup Directory	16
	3.4. Preparing Each Client Computer	16
	3.5. Running the SRS 3D Client Setup.....	16
	3.6. SRS 3D Client Setup Details	17
	3.7. Trouble-Shooting.....	19
4	LICENSE KEY FILE.....	21
	4.1. Introduction.....	23
	4.2. Obtaining a License Key File	23
	4.3. Problems with the License Key File	23
	4.4. Example	24
5	CUSTOMIZING SRS 3D	25
	5.1. Introduction.....	27
	5.2. SRS 3D Viewer Parameters.....	27
	5.3. Examples.....	31

CHAPTER

1

SRS 3D Overview

2 SRS 3D Overview

1.1. Introduction

SRS 3D is a highly customizable system for integrating sequence, structure, and annotation data. The principal components of SRS 3D 1.1 are:

- Icarus scripts for parsing, loading, viewing structure-related databases
- Core SRS 3D databases: PSSH, HSSPalign, and HSSPchain
- SRS 3D Client Setup applet
- The SRS 3D Viewer applet (build 226)
- Supporting Documentation:
 - SRS 3D Administrator's Guide (this document)
 - SRS 3D User Guide
 - SRS 3D Quick Reference Card

The Icarus scripts are included in the `srs3d.tar.gz` file on the SRS 3D distribution CD. These need to be installed into an SRS 7.0 server instance. These scripts enhance the integration of structure-related databases in SRS, enabling end-users to easily find relevant structures and view sequence features mapped onto structures. A complete guide to the server installation process is given in Chapter 2 (below). For convenience, the installation guide is also available as a plain ASCII text file `README.SRS3D-installation` on the distribution CD.

The `srs3d_data.tar.gz` file on the distribution CD contains cut-down versions of the core databases. The full database can be obtained from LION bioscience AG via `ftp` (see Section 2.2.7, below). The full databases contain all significant alignments between all known sequences and all known structures. These databases are described in detail in the SRS 3D User Guide. An optional customization to SRS 3D enables the core databases to be generated in-house, allowing proprietary sequences and structures to be incorporated.

The two Java applets (SRS 3D Client Setup and SRS 3D Viewer) are included in the `srs3d.tar.gz` file on the SRS 3D distribution CD. They will be installed into the SRS server instance. In addition to the server installation, each client computer needs to be prepared for using SRS 3D (see Chapter 3). The `JavaPlug-in` directory on the CD contains setup programs from Sun Microsystems that can be used as part of the client installation.

Supporting documents are included in the `srs3d.tar.gz` file; these will be installed into SRS and available from the SRS Help Page. For convenience, the supporting documents are also available in uncompressed PDF format on the distribution CD.

This manual describes how to install the SRS 3D server, how to install the SRS 3D Viewer on client machines, how to obtain and use the license key file from LION bioscience AG, and finally how to create customized views that integrate in-house sequence, structure, and annotation data with public data.

4 SRS 3D Overview

CHAPTER

2

SERVER INSTALLATION

6 Server Installation

2.1. Requirements

SRS 3D 1.1 requires SRS 7.0; server requirements are almost identical to SRS requirements. Below we mention only requirements that differ from SRS.

2.1.1. Hardware

- Hard disk space: 20+ GB in addition to standard SRS
- Memory: 1 GB RAM (for indexing)

Specific requirements depend on the user-required databases, applications, and usage.

2.1.2. Software

Before you can proceed with the installation, you will require:

- The SRS 7.0 distribution software.
- The SRS 3D 1.1 distribution software.
- An appropriate operating system.
- Supporting applications and databases.

2.1.3. Operating System

Supported server operating systems are described in the SRS 7.0 Administrator's Guide.

2.1.4. Supporting Applications and Databases

The following databases are required and need to be obtained and installed for SRS 3D to work properly:

- SwissProt, SwissNew (release + updates)
- SPTreMBL, TrEMBLnew
- Virtual databases SWISSPROT, TREMBL, and SWALL enabled
- PDB

The following are optional, but highly recommended:

- NCBI BLAST 2.1.2 or higher
- InterPro (EBI)
- InterPro matches (IPRmatches)

The following are optional databases:

- FSSP (downloaded from EBI)
- HSSP (downloaded from EBI)

2.2. Software Installation

The installation instructions in this section are also available as a plain ASCII text file `README.SRS3D-installation` on the distribution CD.

2.2.1. Installing SRS 7.0

See SRS installation instructions.

2.2.2. Install Public Databases

- Download and enable the following databases (see SRS Administrator's Guide for help):
 - PDB
 - SWISSPROTRELEASE
 - SWISSNEW
 - SPTREMBL
 - TREMBLNEW
- Check that the `$LibLoc` lines of SWISSPROT, SWALL and TREMBL are uncommented since they are virtual databases composed of the databases SWISSPROTRELEASE, SWISSNEW, SPTREMBL and TREMBLNEW.
- Run `srssection; srscheck; srsdo` to ensure that the index files are up-to-date.

2.2.3. Installing SRS 3D (Icarus files)

- If not already done:

```
source etc/prep_srs [if you are using csh or tcsh]
. etc/prep_srs.sh [if you are using sh or bash]
```

- Change into your SRS root directory and unpack `srs3d.tar.gz` (delivered on CD):

```
cd $SRROOT [make sure you are in the directory containing srsinstall]
gunzip -c srs3d.tar.gz | tar xvf -
```

- Run the SRS 3D installation script:

On IRIX, Linux, Tru64, AIX:

```
/bin/sh $SRSLION/srs3d/install
```

On Solaris:

```
/bin/ksh $SRSLION/srs3d/install
```

This will copy some files from `$SRSLION/srs3d` to other directories:

```
capture.jar to $SRWWW
install_Java3D to $SRWWW/bin
```

Furthermore, the following files will be modified but backups will be preserved (`.bak` attached):

```
$SRSSITE/srsdb.i
```

```
$SRSSITE/site.i
```

If necessary, the file `$SRSDB/swisssentry.il` gets renamed to `swisssentry.il.bak` and a modified version of `swisssentry.il` is copied from `$SRSLION/srs3d` to `$SRSDB`.

2.2.4. Installing Trimmed-Down Versions of SRS 3D Core Databases

- Change to the directory that should contain the SRS 3D core databases. We will refer to this directory as `$SRS3Droot`.
- Unpack `srs3d_data.tar.gz` from CD

```
gunzip -c srs3d_data.tar.gz | tar xvf -
```

This will create subdirectories for:

- PSSH, HSSPalign, HSSPchain (sharing the same directory)
- CustomAnnot
- PDBequiv

- Edit `$SRSSITE/srsdb.i`

For all databases listed below, uncomment the `$LibLoc` lines and change the `dir` attribute so that it holds the correct path to the data files (a subdirectory of `$SRS3Droot`).

- PSSH
- HSSPalign
- HSSPchain
- CustomAnnot
- FSSP
- FSSPmatches (same directory as FSSP)

Data files for FSSP (and FSSPmatches) are not included on the installation CD and have to be downloaded.

- Run `srssection; srscheck; srsdo`

2.2.5. Configuring the SRS 3D Web Server

The following description is for Apache httpd. As part of the SRS installation, you already inserted a few lines like these into `httpd.conf`:

```
Alias          /srs7/                /your/path/to/srs/www/
ScriptAlias    /srs7bin/cgi-bin/     /your/path/to/srs/www/bin/
```

For SRS 3D, another `Alias` line is needed to define the mapping from URLs to file names:

```
Alias /path/to/PDBfiles/ /path/to/PDBfiles/
```

Replace `/path/to/PDBfiles/` by the directory holding your uncompressed or gzipped PDB files. Note that the same string appears twice after `Alias` and please make sure to include the trailing slashes!

2.2.6. Installing the License file

- Obtain a `license.key` file from customer support at LION bioscience (see Chapter 4 for details). Copy this file to `$SRSSWWW/license.key`, then run

```
chmod a+r $SRSSWWW/license.key
```

The server software installation of SRS 3D is now complete.

2.2.7. Downloading and Updating the Core SRS 3D Databases

The SRS 3D distribution CD comes with cut-down versions of the core databases (PSSH, HSSPalign, and HSSPchain). This enables you to test the installation at this stage. However, to complete the installation, you need to copy the SRS 3D core databases from LION's ftp site ([ftp.lionbioscience.com](ftp://ftp.lionbioscience.com)) and index these databases with your local SRS. Contact the customer support team from LION bioscience for details of gaining access (support@lionbioscience.com). The SRS 3D core database are updated (at least) monthly. Each update involves downloading ~ 1 GByte of compressed data.

This site also provides the PDBseq and PDBequiv databases needed to enable BLASTing against the PDB (see Section 2.2.8, below).

2.2.8. Optional: Installing InterPro

- Download InterPro parsers from EBI

```
ftp://ftp.ebi.ac.uk/pub/databases/interpro/SRS/v4/InterPro4_SRS613.tar.gz
```

(size 35276 bytes, time stamp 15 Nov 2001)

- If not already done:

```
source $SRSSROOT/etc/prep_srs
cd $SRSSROOT
```

- Unpack the InterPro parsers

```
gunzip -c InterPro4_SRS613.tar.gz | tar xvf -
```

- Add the following line to `$SRSSITE/site.i`

```
file:"SRSLION:srs3d/interpro-support.i"
```

- In `$SRSSITE/srsdb.i`, change the line

```
file:"SRSDB:interpro.i"
```

to:

```
file:"SRSSITE:interpro.i"
file:"SRSSITE:iprmatches.i"
```

- In `$SRSSITE/srsdb.i`, change the line

```
# $LibLoc:[ $INTERPRO_DB dir:"$dataRoot2/interpro/" ]
```

to:

```
$LibLoc:[ $INTERPRO_DB      dir:"$dataRoot/interpro/" ]
$LibLoc:[ $IPRMATCHES_DB    dir:"$dataRoot/interpro/" ]
```

- In `$SRSSITE/srsdb.i`, just below the line

```
$LibId:[1051 lib:$INTERPRO_DB]
```

add the line:

```
$LibId:[1052 lib:$IPRMATCHES_DB]
```

- Edit `$SRSSITE/interpro.i` and change the line

```
$Link:[ $INTERPRO_DB to:$SWALL_DB fromField:$DF_Accession toField:$DF_dbref]
```

to:

```
# $Link:[ $INTERPRO_DB to:$SWALL_DB fromField:$DF_Accession toField:$DF_dbref]
$Link:[ $INTERPRO_DB to:$SWISSPROTREL_DB fromField:$DF_Accession toField:$DF_dbref]
$Link:[ $INTERPRO_DB to:$SWISSNEW_DB fromField:$DF_Accession toField:$DF_dbref]
$Link:[ $INTERPRO_DB to:$SPTREMBL_DB fromField:$DF_Accession toField:$DF_dbref]
$Link:[ $INTERPRO_DB to:$TREMBLNEW_DB fromField:$DF_Accession toField:$DF_dbref]
```

- Edit `$SRSSITE/iprmatches.i` and change the line

```
$Link:[ $IPRMATCHES_DB to:$SWALL_DB fromField:$DF_Accession toField:$DF_Accession]
```

to:

```
# $Link:[ $IPRMATCHES_DB to:$SWALL_DB fromField:$DF_Accession toField:$DF_Accession]
$Link:[ $IPRMATCHES_DB to:$SWISSPROTREL_DB fromField:$DF_Accession toField:$DF_Accession]
$Link:[ $IPRMATCHES_DB to:$SWISSNEW_DB fromField:$DF_Accession toField:$DF_Accession]
$Link:[ $IPRMATCHES_DB to:$SPTREMBL_DB fromField:$DF_Accession toField:$DF_Accession]
$Link:[ $IPRMATCHES_DB to:$TREMBLNEW_DB fromField:$DF_Accession toField:$DF_Accession]
```

- Run `srssection; srscheck; srsdo`

Expect some errors during indexing; usually they are harmless.

2.2.9. Optional: Enabling BLASTP

To BLAST against the PDB requires first to download and install the BLAST executables, and secondly to install two interdependent databases: PDBseq (a BLASTable sequence database with one entry for each protein structure chain in the PDB) and PDBequiv (needed to map PDBseq entries onto PDB chains). Normally, end-users will not need to access these databases directly; rather these databases are used behind the scenes, and only when running BLASTP against the PDB.

- Download the BLAST executable from NCBI:

```
ftp://ftp.ncbi.nih.gov/blast/executables/
```

- Edit `$SRSSITE/srsdb.i`:

- Set `$Application_Root` so that it points to the directory containing the "blastall" executable. If you have enabled other applications in SRS, all executables have to reside in this directory.
- Set `$MatrixRoot` to the directory of the BLAST distribution that contains the file "BLOSUM62"

12 Server Installation

- Uncomment these lines:

```
file:"SRSDb:fasta.i"  
file:"SRSLION:srs3d/blastp.i"  
$LibLoc:[$BLASTP_DB          dir:"$ApplResults/" indexDir:"$ApplIndex/"]
```

- Edit `$SRSSICA/appl.i`:

Depending on your server architecture, you will probably have to replace all occurrences of `"/usr/bin/csh"` by `"/bin/csh"`.

- Download PDBequiv and PDBseq from LION's ftp site (see Section 2.2.6, above). It is important that these two databases are always synchronized.
- Enable PDBequiv in `$SRSSITE/srsdb.i` by uncommenting the appropriate line.
- To enable BLAST to find the PDBseq database, `$FastaProtDB_Opt` has to be redefined. This can be accomplished by creating the following file `localApplicationOptions.i` in the `$SRSSITE` directory:

```
# fasta formatted databases  
$BdbRoot = '/data2/db'  
$FastaProtDB_Option=$AppOpt:[search  guiType:menu pos:next group:$PRIM_OPTS  
  prompt:'<!>Database<!> to search:'  
  comment:  
    |This option lets you to select a protein database to be searched.  
  values:{  
    $OptVal:['NRDB' valStr:"$BdbRoot/nrdb/nrdb"]  
    $OptVal:['SWALL' valStr:"$BdbRoot/swall/swall" on:y]  
    $OptVal:['SWISSNEW' valStr:"$BdbRoot/swissnew/swissnew"]  
    $OptVal:['SWISSPROT' valStr:"$BdbRoot/swissprot/swiss"]  
    $OptVal:['GENPEPT' valStr:"$BdbRoot/genpept/genpept"]  
    $OptVal:['PDBSEQ' valStr:"/projects/misc/SBI/pdbSeq/pdb"]  
  }  
]
```

- At the end of the file `$SRSDb/srsgen.i`, uncomment the line

```
file: "$SRSSITE: localApplicationOptions.i"
```

- Run `srssection`

CHAPTER

3

CLIENT INSTALLATION

14 Client Installation

3.1. Introduction

The document is designed to help system administrators with installing the SRS 3D Viewer on client computers. This document covers SRS 3D Version 1.1 and the SRS 3D Viewer, build 226.

3.2. Requirements

3.2.1. Operating System

SRS 3D is supported for use with MS Windows NT 4.0 (SP3 or later) as the client-side OS. MS Windows 2000 and XP are tested but not certified. Linux, Macintosh, and other clients are NOT supported in SRS 3D 1.1.

3.2.2. Hardware

SRS 3D has been designed to work on desktop PC systems, such as AMD Athlon, Intel Pentium III or Pentium 4, 366+ MHz Pentium (or higher). A minimum of 256 Mbytes RAM is required, and 512 Mbytes recommended.

A graphics card with hardware 3D acceleration supporting Open GL, such as nVIDIA GeForce 2 MX (or better), or ATI Radeon 7500 (or better), is highly recommended. The following graphics chips have been tested:

- nVIDIA GeForce2 GTS
- nVIDIA GeForce2 MX
- nVIDIA GeForce2 MX 400
- nVIDIA GeForce4 MX 440
- ATI Radeon 7500

Such a card is not strictly essential, but is highly recommended. For computers without such graphics cards, the SRS 3D Viewer will perform much slower. However, in such cases the 3D Viewer will automatically switch operation to use simple and fast rendering, and hence should still be useable on any Pentium III or better system.

3.2.3. Software

No matter which graphics card is used, it is essential that an up-to-date graphics driver is installed. Using an old graphics driver is likely to cause problems with Java 3D, and hence with the SRS 3D Viewer.

SRS 3D requires certain Java components to be installed, particular the Java Plug-in (see below for details).

SRS 3D is certified to work with Netscape 4.79 and Internet Explorer 5.5 SP 2 (Windows NT, 2000) and Internet Explorer 6 (Windows XP). Other browser versions should also work, but are not certified.

3.3. Customizing the Client Default Setup Directory

The administrator can customize the client default setup directory in the SRS3D_client_install/index.html file hosted by the SRS server. The parameter winSetupPath (and/or linuxSetupPath for linux operating systems) have to be adapted to the preferred path (For technical reasons this parameter occurs twice in the index.html file; both occurrences have to be modified appropriately). This path should be chosen with care. It should point to a directory where all users have write permissions, otherwise the SRS 3D Client Setup will fail installing to the default path. It's recommended that the folder is local to each client machine and not a temporary folder, which might get automatically deleted by the operating system.

The administrator can additionally control the virtual machine parameters from the SRS3D_client_install/index.html file by modifying the vmParameters parameter (see 3.6.1).

3.4. Preparing Each Client Computer

Before end-users start using the SRS 3D Viewer, the system administrator should ensure that the following steps have been performed for each client computer:

- *Install up-to-date graphics card driver*

It is essential to use an up-to-date graphics driver. The latest graphics card drivers can usually be downloaded from the web site of the graphics card manufacturer or vendor.

SRS 3D and Java 3D require a graphics driver that is compatible with OpenGL; most industry-standard drivers are OpenGL-compatible, however some are not. If your card does not support OpenGL, SRS 3D should still work however it will not take advantage of 3D hardware acceleration.

- *Install Java Plug-in version 1.3*

Installing the Java Plug-in from Sun is essential to using the SRS 3D Viewer. The Java Plug-in needs to be installed using the local administrator account. SRS 3D is supported with the Plug-in version 1.3.1_02; this version is included in the JavaPlug-in directory on the distribution CD (see also the README file in that directory). Installing the Java Plug-in will also install the Java Runtime Environment (JRE), usually in a location such as C:\Program Files\JavaSoft\Jre. The SRS 3D Viewer requires a JRE file that is extended to include the Java 3D binaries. However, for the SRS 3D Client Setup program to work correctly, it is important that the local administrator does *not* add Java 3D to the default JRE file mentioned above.

- *Include client computer name and IP in the SRS 3D license key file*

For a client computer to run the SRS 3D Viewer, it is necessary to obtain a valid license key from LION bioscience (see Chapter 4, below).

3.5. Running the SRS 3D Client Setup

Once all the above requirements are met, users can access SRS 3D on that client computer. The first time a user opens an SRS page that would normally launch the SRS 3D Viewer applet, he will be directed to SRS 3D Client Setup applet. If the default JRE does not contain the Java 3D

binaries. It is important that each user visits this page and completes the setup for his computer and his account. The client setup should not be run while logged in as local administrator, as this will add Java 3D to the default JRE. If Java 3D is added to the default JRE, then other users will not be able to correctly complete their individual client setup.

The SRS 3D Client Setup is a Java applet designed to streamline the client installation procedure. It is normally run only once by each user on each client computer. Assuming the installation was successful, the user will not be directed to this page again. The setup usually performs the following steps:

- Modifies the properties file of the Java Plug-in
- Installs a separate Java Runtime Environment (JRE)
- Installs the Java 3D binaries into the JRE

The recommended setup is that the user specifies the location to write a JRE file that is separate from the default JRE. The JRE file is about 30 Mbytes (uncompressed). It is recommended that the system administrator decides on a common default file location used by SRS 3D users, e.g. D:\Transfer\SRS3D\. All users will need to have write permission in this location. The SRS 3D Client Setup will default to the specified directory. Alternatively, he or she may choose to log in as each user and run the SRS 3D Client Setup, thus ensuring that the end-users can immediately start using SRS 3D.

If problems occur, you may wish to return to the page that launches the SRS 3D Client Setup program. You can always return to this page by going to the URL for the SRS Start page (with the LION paw), and appending /SRS3D_client_install/ to the URL. For example:

```
http://srserver.yourdomain.com/srs7/SRS3D_client_install/
```

3.6. SRS 3D Client Setup Details

This chapter describes in some detail the procedure followed by the SRS 3D Client Setup.

3.6.1. Modifying the Java Plug-in Properties File

The Java Plug-in properties file is owned by the user and can be found in the `.java` folder of the user's home directory (e.g. `c:\winnt\profiles\myname\.java\`). Note that the `.java` folder might not be visible by default. The file is normally called simply `properties`, however some version of the Java Plug-in use files with similar names, such as `properties131_02` or `properties140`. This file contains the path of the Java Runtime Environment (JRE) that will be used by default in a browser program (e.g. Netscape Navigator or Internet Explorer) started by the user. This file also contains parameter settings that are used by the Java Virtual Machine (VM) when it is started by the browser.

The SRS 3D Client Setup usually modifies the JRE path in the properties file to match that of the enhanced JRE created in the steps described below. In all cases, the setup modifies the Java VM parameters. Normally, the setup will set the following parameters: `-Xms20 -Xmx120 -Xincgc` (with Java 3D version 1.3 it is currently necessary to set the following parameter: `-j3d.displaylist=false`). The SRS 3D Client Setup handles the following cases: where a `properties` file exists, a `propertiesXXX` file exists (where `XXX` gives the Java Plug-in version number), or where no `properties` file exists. The VM properties file can be modified using the

18 Client Installation

Java Plug-in Panel accessible from the Control Panel. However this is only recommended if problems arise, and we recommend using the parameters set by the SRS 3D Client Setup. Note that if you change settings using the Java Plug-in Panel, the JRE path might be reset to the default.

Parameters and their mnemonics

- `-Xms20m` (sets the minimum heap size of the VM to 20 MByte)
- `-Xmx120m` (sets the maximum heap size of the VM to 120 MByte. This parameter should be adapted to the computers memory size (e.g. on a computer with 128 MByte RAM you should set `-Xmx80m`). This value limits the capabilities of the SRS 3D Viewer. In very special cases (e.g. looking at very large structures or applying resource intensive operations) the system administrator should increase this value accordingly.) The value chosen by the setup is 120 MByte.
- `-Xincgc` (enables incremental garbage collection; this setting helps the application to handle memory resources more carefully)
- `-Dj3d.displaylist=false` (disables a Java 3D optimization that causes a bug in the SRS 3D Viewer with Java 3D 1.3β2 (geometry appears to turn black or to disappear) this problem might be fixed in future releases of Java 3D)

It is particularly important to turn on incremental garbage collection (with `-Xincgc`); if this parameter is not set, the SRS 3D Viewer applet will accumulate memory, and may eventually cause the browser to stall.

3.6.2. Installing a Separate JRE

If the user has write permission in the default JRE folder (e.g. `C:\Program Files\JavaSoft\Jre`), this step is skipped. Normally, however, the user cannot write to this folder. The setup will copy a new JRE to the SRS 3D setup folder and add the Java 3D binaries as described below.

It is very important that the user chooses a folder for SRS 3D that will NOT be deleted (e.g. never choose a temporary folder)! It is recommended that the user chooses a folder that other users on the same machine also have write permission (e.g. `D:\transfer\SRS3D` or `D:\common\SRS3D`). In this way several users on the same machine can share a JRE with Java 3D. As mentioned above, it is recommend that the system administrator decides where the SRS 3D default directory is located.

If JRE already exists in the folder it will be simply overwritten by the setup. This makes it easy to update the setup and the respective binaries.

3.6.3. Installing Java 3D

In this step, the setup enhances the JRE to include the Java 3D binaries.

Different versions of Java 3D can be chosen. We recommend using Java 3D 1.2.1_03, however with some graphics cards, other versions may perform better.

If the user has write permission to the default JRE folder (e.g. `C:\Program Files\JavaSoft\Jre`), this default JRE will be enhanced. In this case, the path of the JRE path in the properties file is not changed. The parameters as described above will be modified in the properties file. However, note that the parameters are only modified for the current user. Each user still has to run the setup at least once for updating his properties file!

3.7. Trouble-Shooting

- 'Java Runtime Environment cannot be loaded from ...'

In case the browser starts a HTML page with a Java Applet and provides a dialog beginning with 'Java Runtime Environment cannot be loaded from ...', you have to modify the Java Plug-in to point to the default JRE:

- Go to `Start→Settings→Control Panel→Java Plug-in` (choose most recent version)
- Go to the tab `ADVANCED` and select the `Default JRE`
- Apply the changes and close the Java Plug-in window
- Restart the browser and go back to setup page
- If this does not work try another Java Plug-in in the Control Panel (the one that was installed most recently)

Note that also the next Chapter might apply in this case, too.

- Having installed Java Plug 1.4

The setup recognizes if you are using Java Plug-in 1.4. However it is not possible to install the supported JRE 1.3.1_02/Java 3D 1.2.1_03 combination with the 1.4 Plug-in. The recommended combination can only be installed after uninstalling the Java Plug-in 1.4 using `Start→Settings→Control Panel→Add/Remove Programs` and installing the Java Plug-in 1.3.1_02. Note that the recommended version shows better performance than installations with the 1.4 Java Plug-in.

- The SRS 3D Viewer starts and crashes the browser immediately after initialization

In case you are using a laptop with an ATI Rage 128 Mobility graphics card (or similar) we recommend installing Java 3D 1.3 (currently only a beta version is available). Simply rerun the SRS 3D Client Setup and choose the Java 3D 1.3β2 version.

The most frequent cause of such crashes is using an out-to-date driver for your graphics card.

- With linux operating systems

- The SRS 3D client setup may stall if there is no properties file available in the users `.java` folder. To create such a file simply start the Java Plug-in panel provided in your JRE folder (`ControlPanel.html`).

20 Client Installation

- The SRS 3D Viewer may still prompt you to install Java3D if no appropriate OpenGL (or Mesa) library was found. If you encounter this problem, you may need to upgrade linux distribution to RedHat 7.2 or a recent S.U.S.E distribution.

CHAPTER

4

LICENSE KEY FILE

4.1. Introduction

This chapter explains the license checking mechanism of the SRS 3D 1.1 and describes the steps involved in enabling computers in your intranet for the use of SRS 3D 1.1.

4.2. Obtaining a License Key File

You have purchased a license to SRS 3D based on a certain number of seats. To enable client computers within an intranet to use SRS 3D, the administrator has to collect the network names¹ and the IPs of the server and the computers that should be enabled. This list should be sent via email to support@lionbioscience.com with the subject “SRS 3D license key”. The customer support group at LION will process the request and send back a `license.key` file that has to be put into the `www` folder of the SRS installation. Ensure that the file permissions are set such that this file is accessible from each of the client computers.

Note that for enabling computers that obtain their IPs from a DHCP client you have to provide a range of IPs.

4.3. Problems with the License Key File

- The `license.key` file may be cached by a proxy and/or the browser used on the client computer. If you have updated the `license.key` file on the server you should always ensure that the version cached in the proxy is updated, too; you can do this by clicking SHIFT-Reload from Netscape on the url of the `license.key` file (e.g. <http://yourdomain/srs7/license.key>). To check whether a proxy is responsible for a not working client computer (that was enabled) you may temporarily disable the proxy settings in the browser.
- If a client is not included in the `license.key` file the applet provides information about computer name and IP as well as server name and IP. Note that both name/IP combinations have to be included in the `license.key` file.
- If a user requests to enable his computer, let him/her provide his/her computer name and IP and request a new `license.key` file from LION bioscience. Note that LION bioscience may only add name/IP combinations to the `license.key`. It is impossible to remove enabled computers from the list within the duration of the license contract.
- The `license.key` file may have expired according to your contract with LION bioscience. In this case you have to renew the contract and request a new `license.key` file.
- The combination `localhost/127.0.0.1` is not available for inclusion into the `license.key` file. If the user requests enabling a computer with this settings, you have to check the setup of this computer. If the SRS 3D Viewer shows this combination for the client computer, it is possible that the user did not accept the applet certificate to grant permissions to the application. In this case ask the user to restart his browser² and start a page containing the SRS 3D Viewer again. He or she has to click on `Grant` or `Grant always` in the Java Plug-in

¹ with domain information (e.g. 170.15.15.78/srsserver.mycompany.com)

² close all browser windows and restart the browser; using Netscape you should also ensure that all Netscape related applications are closed (i.e. Messenger, Composer and Navigator windows)

Security Warning dialog. This enables the license checking of the applet. Note that if the user presses `Grant` always the dialog will never be presented again. All granted certificates are listed in the Java Plug-in Panel (accessible via `Start` → `Control Panel` → `Java Plug-in`) and may be removed.

4.4. Example

Assuming you have a server named `srssserver` with the IP `170.15.15.70` and the server domain is `mycompany.com`. Moreover, assuming that you have two computers named `Mars` and `Earth` as well as a laptops with the names `LT001`, `LT008`, and `LT017`. `Mars` has the IP `170.15.1.48` and `Earth` has the IP `170.15.1.42`. The laptops obtains their IP dynamically via DHCP. The DHCP is configured to the IPs ranging from `170.15.255.1` to `170.15.255.255`.

Your request to LION should look like this:

```
#license file for mycompany (mycompany reference number)
#server
ip=170.15.15.70
name=srssserver.mycompany.com

#DHCP
ip=170.15.255.1.255
name=lt001.mycompany.com,lt008.mycompany.com,lt017.mycompany.com

#Mars
ip=170.15.1.48
name=mars.mycompany.com

#Earth
ip=170.15.1.42
name=earth.mycompany.com
```

For adding a computer named `Jupiter` with the IP `170.15.1.53` and another laptop `LT007` send the request :

```
#license file for mycompany (mycompany reference number)

#Jupiter
ip=170.15.1.53
name=jupiter.mycompany.com

#DHCP
ip=170.15.255.1.255
name=lt007.mycompany.com
```

4.4.1. Comments

- Note that the range is encoded in a five bytes IP address. Such range definitions are available for DHCP addresses, which only vary in the last byte of the IP.
- The names for one DHCP range have to be listed all in one line. If one line is not sufficient you can provide an additional name/IP combination.
- If the formatting of the request adheres to this example, the `license.key` file can be generated automatically. If problems occur they are hand-curated by customer support of LION bioscience, which might delay the processing of your request.

CHAPTER

5

CUSTOMIZING SRS 3D

5.1. Introduction

SRS 3D, like SRS, is designed to be highly customizable. You can modify or create SRS views that combine structures, sequences and feature data. To be able to customize SRS 3D, you will need to know how to customize SRS. This requires that you have completed an SRS system administrator's course. This is a prerequisite for using the material in this chapter.

Once you have learned how to customize SRS, customizing SRS 3D is rather straightforward. It is recommended that you begin by studying the Icarus view script for HSSPalign entries. This file is located at `$SRSLION/srs3d/hssp-align.il`. It is also instructive to study the view script for PDB entries.

These views invoke a Java archive file called `capture.jar`, located at `$SRSSWWW/capture.jar` which launches the SRS 3D Viewer applet. The views also pass invocation parameters to the applet. These invocation parameters include the file location of the structure(s) and information about sequences, alignments, and annotations extracted from fields of the database entry or of linked entries. Other invocation parameters determine the initial state of the SRS 3D Viewer. This Chapter describes in full the applet invocation parameters and their effects on the SRS 3D Viewer.

5.2. SRS 3D Viewer Parameters

5.2.1. Specifying a Parameter File

For stand-alone invocation of the viewer you can provide all parameters in a separate parameter file that can be specified in the following sections. Note that the parameters overwrite the parameters provided in the invocation line (except the `parameterFile` parameter itself).

In general:

```
parameterFile=<filename>
```

Example

```
parameterFile=d:/SRS3D/example.srs3d
```

5.2.2. Access Control

The `licenseKey` parameter described below enables you to specify the license key to use for the access control check that is mandatory for using the SRS 3D Viewer.

However, for stand-alone invocation of the SRS 3D Viewer you can explicitly point to a `license.key` file in a local folder or point to a server that provides the `license.key` file for the viewer. This can be done by using the parameter "codeBaseUrl". Note that the `licenseKey` defaults to `license.key`.

In general:

```
licenseKey=<url>
```

or

```
codeBaseUrl=<url>
```

28 Customizing SRS 3D

Example

```
licenseKey=http://computer01:8080/license.key
or
codeBaseUrl=http://computer01:8080/srs7/
```

Comments

The `license.key` file (or any file you specify) contains the IPs and names of the client and server computers in the network seen from the client. This way we enable access control checking and manage to monitor which computers cause problems. Make sure that the `license.key` file is not cached on the client or on some proxy on the way from the server to the client machine.

If no `license.key` file is specified the application will not properly start and present a “License conditions not met” message with additional information.

5.2.3. Specifying Structure Input

In general:

```
structures=structure-block
structure-block ::= `structures` `{` structure-attributes [{ `;` structure-
attributes }] `}`
structure-attributes ::= <identifier>,<url>, <structure-flags>, <superposition-
operator>
superposition-operator ::=
<rxx>:<rxy>:<rxz>:<tx>:<ryx>:<ryy>:<ryz>:<ty>:<rzx>:<rzy>:<rzz>:<tz>
```

Example

```
structures=structures {
  lbsxmodel,file:///C:/models/THA_HUMAN-lbsx.brk_model,,;
  lbsx,http://computer01:8080//bioSCOUT/PDB/lbsx.brk,,1:0:0:0:0:0:1:0:0:0:0:0:1:0
}
```

The URL may be one of the following:

```
http://computer01:8888/pdb/2mpr.brk
```

From local file system:

```
file:///c:/data/pdb/2mpr.brk
```

From working directory:

```
file:/2mpr.brk
```

Using http protocol from the document base (only for browser embedded applets):

```
http:/pdb/2mpr.brk
```

Comments

`identifier`: the identifier is used to differentiate between the different structures (and sequences) and thus should be unique. The identifiers are used when alignments and annotations are specified. If the specified PDB file contains a valid HEADER record and you provide structures with different PDB identifiers the `<identifier>` attribute can be left empty. Internally the identifier found in the HEADER record will be used. For example:


```
structures=structures { ,file:/latn.brk,, }.
```

superposition-operator: the superposition matrix is composed of 12 values; the 4 times 3 values represent the rows in the matrix; values differing from the default (identity plus no translation 1:0:0:0:0:0:1:0:0:0:0:0:1:0) can be omitted. If all values are default, the whole superposition-operator can be omitted.

Possible structure-flags:

- MASTER: the master structure needs to be classified in superposition views for enhancing the representation of the superposition.
- MERGEABLE: enables the merging of 100% aligned sequences in the annotation and sequence view.
- IMPOSED: enables the application to auto-compute alignments of two structures (physical distance criteria applied) and classifies the structure for enhancing the visualization (see styles in SRS 3D documentation).

5.2.4. Enabling the Different Views and Modes

In general:

```
annotationView=<activation>           defaults to `off`
sequenceView=<activation>             defaults to `off`
expertMode=<activation>               defaults to `off`
toolbar=<activation>                  defaults to `on`
defaultStyle=style                    defaults `First Impression`
style ::= `Superposition`|`Homology`|`First Impression`
activation ::= `on`|*
```

Comments:

First Impression is the default style if nothing or an unknown style is specified.

5.2.5. Specifying Annotations

Example:

```
annotations="
  2dnj{
    A01, 0.5:0:0, on, A01 Description, http://url-A01/,FLAG {
      StructureF01,0.3:0.3:0.7, F01 Description, http://url-F01/, A:30: :35:
    }
  }

  P00639{
    A02, 0.5:0:0, off, P0039 Description, http://url-P00639/,%20 {
      SequenceF01,0.1:0.9:0.1,Sequence feature no. 01,http://url-F01/,60:60
    }
  }
}"
```

In general:

```
annotations = layer-block [objectblock]
objectblock ::= <identifier> annotationblock
annotationblock ::= `{` { annotation } `}`
annotation ::= annotation-attributes `{` [feature [{`;` feature}] ] `}`
annotation-attributes ::= <identifier>, [color], [<activation>], <description>, [<url>],
[flags]
```

30 Customizing SRS 3D

```
feature ::= feature-attributes
feature-attributes ::= <identifier>, [color], <description>, [<url>] , residue-range
color ::= <red>:<green>:<blue>
residue-range ::= structure-range | sequence-range
structure-range ::= <chain-identifier>:<structure-residue>:<structure-residue>
structure-residue ::= <residue-number>:<residue-insertion-code>
sequence-range ::= <sequence-residue>:<sequence-residue>
sequence-residue ::= <residue-number>
```

Comments:

All special characters (i.e. “;:,{ }%<>”) have to be replaced by %XX, where XX is the hexadecimal code of the character.

Example:

“This character `%41` is an%20A!” à “This character `A` is an A!”

“Variant%3a %7bA%7d -%3e %7bG%7d” à “Variant: {A} -> {G}”

5.2.6. Specifying a Sequence Alignment

Example:

```
sequenceAlignments="
  sequences{
    P00639,DRN1_BOVIN,REFERENCE,
      MRGTRLMGLLLALAGLLQLGLSLKIAAFNIRTFGETKMSNATLASIYIVRIV
      RRYDIVLIQEVRDShLVAVGKLLDYLNQDDPNTYHYVVSSEPLGRNSYKERY
      LFLFRPNKVSVDLTYQYDDGCESCND SFSREPAVVKFSSHSTKVKEFAIV
      ALHSAPSDAVAEINSLYDVYLDVQQKWHLNDVMLMGDFNADCSYVTSSQWS
      SIRLRTSSTFQWLIPDSADTTATSTNCAYDRIVVAGSLLQSSVVPGSAAFP
      DFQAAAYGLSNEMALAI SDHYPVEVTLT;
    P26466,LAMB_SALTY, ,
      MMITLRKPLAVAVAAGVMSAQAMAVDFHGYARSGIGWTGSGGEQQCFQAT
      GAQSKYRLGNCEETYAELKLGQEVWKEGDKSFYFDTNVAYSVNQQNDWEST

      DPAFREANVQGKNLIEWLPGSTIWAGKRFYQRHVDHMFYWDISGPGAG
      IENIDLGFGLSLAATRSTEAGGSYTFSSQNIYDEVKDTANDVFDVRLAGL
      QTNPDGVLLELGDY GRANTTDGYKLADGASKDGWMFTAHTQSMLKGYNKF
      VVQYATDAMTTQKGQARGSDGSSSFTTEELSDGTKINYANKVINNNGNMWR
      ILDHGAISLGDKWDLMYVGM YQNIDWNNLGTWWTGVRPMPYKWTPIMST
      LLEVGYDNVKSQQTGDRNNQYK I TLAQQWQAGDSIWSRPAIRIFATYAKWD
      EKWGYIKDGDNISRYAAATNSGISTNSRSDSEWTFGAQMEIWW
  }
  matches{
    2dnj:A:1: :99: ,P00639:23:121;
    2dnj:A:107: :253: ,P00639:136:282;
    1mpq:A:254: :261: ,P26466:286:293
  }
"
```

In general:

sequenceAlignments = sequence-block match-block

```
sequence-block ::= `sequences` `{` sequence [{` ;` sequence}] `}`
sequence ::= <sequence-id>, <sequence-name>, <sequence-flags>, <sequence>
match-block ::= `matches` `{` match [ {` ;` match} ] `}`
match ::= <structure-id>, residue-range, <structure-id>, structure-range
```

Comments:

In the `<sequence>` string white spaces are ignored.

Possible `sequence-flags`:

REFERENCE: the sequence is the reference for the alignment.
 MERGEABLE: the aligned sequences should be merged in the annotation view for 100% alignments (compare to `structure-flags`).

Note that the current implementation does not check for multiple references. You must specify a reference (in the structure or the sequence parameters)

5.2.7. Specifying URL Templates**Example:**

```
links="PDB:http://computer01:8888/pdb/###.brk"
```

In general:

```
links= link-template{;link-template}
link-template ::= [<menu-entry>,<database-identifier>:<url-template>
url-template ::= <url-template-prefix>###<url-template-suffix>
```

Comments:

This template is used for composing URL that are associated with sequence or structure features. The user can then link to these URLs by clicking on features that are activated on the structure.

5.2.8. Specifying an Identifier for Saving/Restoring the Viewer's State**In general:**

```
instanceId=<identifier>
```

Example:

```
instanceId=PDB_1AOI
```

Comment:

You have to make sure that two different views do not provide the same `instanceId` to the viewer. The SRS 3D Viewer enabled for saving a file called `<identifier>.xml` to the user directory of the operation system.

5.3. Examples**5.3.1. SRS 3D Example (simple protein view)**

```
licenseKey="/license.key"
structures="
  structures{
    laoi,http://computer01:8080//bioSCOUT/PDB/laoi.brk,,
  }"
```

32 Customizing SRS 3D

```
defaultStyle="First Impression"  
annotationView=on  
sequenceView=on  
toolbar=on
```

5.3.2. SRS 3D Example (superposition)

```
licenseKey="/license.key"  
structures="  
  structures{  
    lqjd,http://computer01:8080//bioSCOUT/PDB/lqjd.brk,REFERENCE|MASTER|IMPOSED,;  
    ld4d,http://computer01:8080//bioSCOUT/PDB/ld4d.brk,IMPOSED,  
    0.189000 :0.752000 :0.632000:-14.334000:  
    0.077000:-0.653000 :0.753000:48.102000:  
    0.979000:-0.093000:-0.181000:67.974000  
  }"  
defaultStyle=Superposition  
annotationView=on  
sequenceView=on  
toolbar=on
```

5.3.3. SRS 3D Example (structure with aligned sequence and annotation)

```
structures="  
structures{  
  2src,http://computer01:8080/bioSCOUT/PDB/2src.brk.gz, REFERENCE,  
}"  
sequenceAlignments="  
sequences{  
  P12931, SRC_HUMAN, ,  
  GSNKSKPKDASQRRRSLEPAENVHAGGGGAFPPASQTPSKPASADGHRGPSAAFAPAAAEPKLFGGFNSSDVTVTSPQ  
  RAGPLAGGVTTFFVALYDYESRTEITDLSFKKGERLQIVNNTEGDWLHSLSTGQTGYIPSNYVAPSDSIQAEEWYF  
  GKITRRESERLLNAENPRGTFVRESETTKAYCLSVSDFDNAKGLNVKHYKIRKLDSSGGFYITSRTOFNSLQQL  
  VAYYSKHADGLCHRLTTVCPTSKPQTQGLAKDAWEIPRESLRLEVKLGQGCFCGEVVMGTWNGTTRVAIKTLKPGTM  
  SPEAFLQEAQVMKLRHEKLVQLYAVVSEPIYIVTEYMSKGSLLDFLKGETGKYLRPLQVDMAAQIASGMAYVE  
  RMNVVHRDLRAANILVGENLVCKVADFGFLARLIEDNEYTARQGAFFPIKWTAPEAALYGRFTIKSDVWSFGILLTE  
  LTTKGRVYPYPMVNRVLDQVERGYRMPCEPCEPSLHDLMCQCWRKEPEERPTFEYLAQFLEDYFTSTEPQYQPG  
  ENL  
}  
matches{  
  2src: :84: :533: ,P12931:86:535;  
}"  
annotations="  
P12931{  
  lipid,0.93:0.93:0,off,%20,%20,%20 {  
    lipid_1,,MYRISTATE (BY SIMILARITY). ,%20,1:1;  
  }  
  domain,0.5:0.5:0.5,off,%20,%20,color {  
    domain_1,,SH3. ,%20,83:144;  
    domain_2,,SH2. ,%20,150:247;  
    domain_3,,PROTEIN KINASE. ,%20,269:522;  
  }  
  np_bind,0.55:0.21:0.15,off,%20,%20,%20 {  
    np_bind_1,,ATP. ,%20,275:283;  
  }  
  binding,0.55:0.48:0.55,off,%20,%20,%20 {  
    binding_1,,ATP. ,%20,297:297;  
  }  
  act_site,1:0:0,off,%20,%20,%20 {  
    act_site_1,, ,%20,388:388;  
  }  
  mod_res,0.42:0.35:0.80,off,%20,%20,%20 {  
    mod_res_1,,PHOSPHORYLATION (AUTO-). ,%20,419:419;  
    mod_res_2,,PHOSPHORYLATION (BY CSK%3b NEGATIVE FTREGULATION). ,%20,529:529;
```

```
    }  
  }"  
  expertMode=on  
  defaultStyle=Homology  
  annotationView=on  
  sequenceView=on  
  toolbar=on
```