



BiBiWS

Administration Guide

Henning Mersch hmersch@techfak.uni-bielefeld.de

28. Juli 2005

Based on an excerpt of:
BiBiWS - Ein Framework für die Entwicklung asynchroner
WebServices auf dem BiBiServ [[HM04](#)]

Changes are documented within this guide or at
last chapter *ChangeLog*

Contents

1	BiBiWS Administration Guide	5
1.1	Setting up Apache http Server	5
1.1.1	Installation of the client side BiBiWS Library	6
1.2	Setting up Apache Jakarta Tomcat	7
1.2.1	Installation of the server side BiBiWS Library	8
1.3	Setting up database	10
1.3.1	Cleaning up database and spooldirectroy	11
1.4	Creating Webservice projects	11
1.5	Installation of Webservice projects on BiBiServ and BiBiServ Webservice Server	12
2	BiBiWS - FAQ	13
3	BiBiWS - ChangeLog	15
4	BiBiWS - Predefined statuscodes	17

1 BiBiWS Administration Guide

This guide documents configuration and installation of the BiBiWS framework.

Just for clarification:

A *developer* in this document is a person, who develops a bioinformatic tool and would like to provide a world-wide accessible Webservice for this tool. A *user* is a person, who uses such a Webservice. The BiBiServ Administrator Team's task is on the one hand to provide help for developing Webservices, and on the other hand answering questions coming from users. Users and developers can contact BiBiServ Administrator Team via email: bibi-help@techfak.uni-bielefeld.de

Overall there are three main programs that the project BiBiWS relies on:

Apache http Server Presenting HTML interface to Webservice server side (acting as Webservice client side) The servers present a user friendly HTML-form interface to the server side of Webservices.

Apache Tomcat Servlet container for the Webservice server side.

Postgresql Database For server side to manage the Webservice calls by ids.

1.1 Setting up Apache http Server

BiBiServ and BiBiTest are Apache http Servers accessible at [\[BIBISERV\]](#) and [\[BIBITEST\]](#). The Apache http Server was set up by the Support Group of *Technical Faculty of the University Bielefeld*, version 1.3.29. BiBiWS does not use any specific extensions of this version, so every (newer) version should work.

While BiBiTest is hosting a test environment system, it is only internally accessible at the *Technical Faculty of the University Bielefeld*. Assuming a running Apache http Server environment, this section will explain installation requirements for getting BiBiWS to run.

Basically the http server has to provide Perl-CGIs. CGIs of BiBiWS require the following packages to be installed, which are all accessible from CPAN [\[CPAN\]](#):

CGI for CGI support from Perl side

SOAP::Lite connecting to the Webservice server side

Data::Dumper stores and prints out Perl objects

Log::Log4perl proper logging facility

Net::DNS checking MX record of an email address

Net::SMTP sending mails

File::Path simple accessing of files

1.1.1 Installation of the client side BiBiWS Library

The supporting client side library of BiBiWS is the `BiBiServ::` module at `/vol/bibidev/bibi/wsc/vol/bibi/lib` directory.

It will be installed during the installation of the whole `bibi` project, which also does general installation like generating the layout of BiBiServ (not part of this document). The API documentation of the classes is located at [\[BIBIWSAPI-WSC\]](#). For generating these APIs, call special target `make api` at `/vol/bibidev/bibi/wsc/` directory.

There is a general properties file

`/vol/bibidev/bibi/wsc/vol/bibi/etc/<SERVER>.properties`, which will be installed on `<SERVER>`.¹ This properties file contains the following elements:

server the server name *bibitest* or *bibiserv*

mailfatal.email email address for log level *mailfatal*

mail.from sender email address of server sent emails

frame.bibiserv Location of general bibiserv environment layout

redirect.time time in seconds to wait before requesting results of WebService again (HTML redirect)

base.spooldir location of the spooldirectory

statuscode.cXXX default description of the statuscode generated on client side

statuscode.cXXX.internal default internal description of the statuscode generated on client side

statuscode.undef description of unknown statuscodes

statuscode.undef.internal internal description of unknown statuscodes

A complete list of default statuscodes can be found at Table [4.3](#).

¹<SERVER> is either BiBiTest or BiBiServ

1.2 Setting up Apache Jakarta Tomcat

BiBiServ WebService Server and BiBiTest WebService Server are Apache Jakarta Tomcat servers. The development of BiBiWS is based on Version 5.0.18, but higher versions should work, too.

Their only job is to provide the server side WebService access of the tools running on BiBiServ. Like the http BiBiTest, the BiBiTest WebService Server also is only internally accessible at the *Technical Faculty of the University Bielefeld*.

Again assuming a running Tomcat environment, this section will explain installation requirements for getting BiBiWS running.

Beside Tomcat's own libraries and requirements, BiBiWS needs some special libraries, which have to go to `$CATALINA_HOME/shared/lib/`²:

log4j-1.2.8.jar proper logging

mail.jar sending emails

postgresql.jar accessing database, where *ids* are managed (using JDBC Feature of Tomcat, this library must be located at `$CATALINA_HOME/common/lib/`)

The following libraries are also used by BiBiWS, but must be located at every WebService (`WEB-INF/lib/` directory) due to Tomcat restrictions on (un)deploy:

axis.jar AXIS library

commons-discovery.jar General Apache Library

Axis depends on the following libraries, which also have to be installed at the Tomcat server at `$CATALINA_HOME/shared/lib/`:

jaxrpc.jar Axis implements this interface API

saaj.jar Axis implements this interface API

wSDL4j.jar Generating WSDL and java source code

xercesImpl.jar Implementation of the Xerces XML parser

xml-apis.jar API of XML parser, belonging to Xerces

After installation of a new version of any library, the server has to be restarted for reloading the libraries! All this is done by installing the `bibi` project.

Tomcat supports a feature called *JDBC*, which gives all WebServices of a Tomcat Server connection to a database. This is less performance consuming than making own connections by every WebServices.

BiBiWS uses this for storing the statuscodes of a WebService call, so this database connection has to be configured. Just include the code of Figure 1.1 to the file `$CATALINA_HOME/conf/server.xml` at the `<Host>` element.

²`$CATALINA_HOME/` is basedirectory of the Tomcat Server

```

<DefaultContext>
  <Resource name="jdbc/postgres" type="javax.sql.DataSource"/>
  <ResourceParams name="jdbc/postgres">
    <parameter>
      <name>maxWait</name>
      <value>5000</value>
    </parameter>
    <parameter>
      <name>maxActive</name>
      <value>4</value>
    </parameter>
    <parameter>
      <name>password</name>
      <value>YOUR_PASSWORD</value>
    </parameter>
    <parameter>
      <name>url</name>
      <value>jdbc:postgresql://YOUR_HOST/bibiwstest</value>
    </parameter>
    <parameter>
      <name>driverClassName</name>
      <value>org.postgresql.Driver</value>
    </parameter>
    <parameter>
      <name>maxIdle</name>
      <value>2</value>
    </parameter>
    <parameter>
      <name>username</name>
      <value>YOUR_USER</value>
    </parameter>
  </ResourceParams>
</DefaultContext>

```

Figure 1.1: JDBC Connector of Tomcat to access statuscode database

1.2.1 Installation of the server side BiBiWS Library

BiBiWS server side library is the package `de.unibi.techfak.bibiserv`, which is located in the `/vol/bibidev/bibi/wss/` directory.

For installation, there is a `build.xml`, which has several targets:

The API documentation of the classes is located at [\[BIBIWSAPI-WSS\]](#).

As on the client side there is a properties file for the BiBiWS server side library at `/vol/bibidev/bibi/wss/config/<WSSERVER>.properties`³, which will be installed at the corresponding server.

This properties file contains the following elements:

³<WSSERVER> is either `bibiwstest` or `bibiwsserv`

<code>install_(bibiwstest bibiwsserv)</code>	(default: <code>install_bibiwstest</code>) installs the required libraries and the BiBiWS server library with properties file to the server
<code>clean_(bibiwstest bibiwsserv)</code>	removes the library from the server
<code>start_(bibiwstest bibiwsserv)</code>	starts the Apache Tomcat server (for use at the host machine)
<code>stop_(bibiwstest bibiwsserv)</code>	stops the Apache Tomcat server (for use at the host machine)
<code>api</code>	for generating the BiBiWS server side API

Table 1.1: Commands for installation of BiBiWS Serverside

server the server name either *bibiwstest* or *bibiwsserv*

spooldir.base base directory of spool dir

mail.smtp.host smtp host for sending mails

mailfatal.email email address for log level *mailfatal*

mail.from sender email address of server sent emails

sgе.root SGE_ROOT environment variable for all calls of SGE

sgе.cell SGE_CELL environment variable for all calls of SGE

sgе.qsub location of qsub

sgе.qstat location of qstat

sgе.batchfile.sh shell of the generated shell scripts for SGE job start

sgе.qsub.params.user parameters of qsub (can be changed by tool)

sgе.qsub.params.bibi parameters of qsub (can NOT be changed by tool)

sgе.qstat.finishedparams qstat parameters to return list of finished jobs

sgе.qstat.pendingparams qstat parameters to return list of pending jobs

sgе.waittime time to wait until looking next if SGE finished job

chmod location of chmod

chmodparams.spooldir parameters for chmod for spooldirectory creation

chmodparams.batchfile parameters for chmod for generated SGE shell scripts

statuscode.XXX default description of this statuscode XXX generated on the server side

statuscode.XXX.internal default internal description of statuscode XXX enered on the server side

statuscode.undef description of unknown statuscodes

statuscode.undef.internal internal description of unknown statuscodes

A complete list of default statuscodes can be found at Table 4.2.

For installing and reloading a BiBiWS WebService using the management interface of Apache Tomcat, some properties are required. They are stored in the file `/vol/bibidev/bibi/wss/config/<WSSERVER>-admin.properties`.

manager.url URI of manager interface of Apache Tomcat

manager.username username for log in

manager.passwd password for log in

server.url URL of server to access (used for creating WSDL files)

The corresponding production system file is readable only for BiBiServ Administrator Team, so only they can (re)install/reload and generate WSDL Definition of a WebService for the production system.

1.3 Setting up database

For managing the ids, BiBiWS uses a SQL database. This does not has to be a Postgresql, but this is recommended, because development of BiBiWS based on Postgresql Version 7.2.4. Every database wich includes a JDBC library should be fine. The library creates one row per WebService call and updates the statuscode, if required. So an incoming response() could look up the status of the WebService call, which is identified by its id.

Assuming the database server is installed and the tomcat is set up for the database connection, Figure 1.2 shows the installation commands in SQL.

A database with one table for managing the ids has to be created like described in

```

1 CREATE DATABASE wssstatus;
2 CREATE TABLE wssstatus (id varchar, toolname varchar, statuscode int,\
    description varchar, internaldescription varchar, drmaaaid varchar,\
    created timestamp, lastmod timestamp);
3 CREATE INDEX idindex ON wssstatus(id);
4 ANALYZE wssstatus;

```

Figure 1.2: Installing database for BiBiWS (Postgresql syntax)

Figure 1.2:

id id for identifying WebService

toolname name of tool

statuscode current statuscode of WebService call

description description of statuscode

internaldescription internal description of statuscode

drmaaaid if submitted to a grid, the Job Id of currently running job⁴

created timestamp of submitting WebService call

lastmod timestamp of last modification

1.3.1 Cleaning up database and spooldirectory

`cleanWSSSide` is a tool for cleaning up the database and spooldirectory to avoid overflows of the spooldirectory on testing and production WebService server side. It is located at `/vol/bibidev/bibi/wss/cleanWSSSide/` and has to get some properties from `/vol/bibidev/bibi/wss/cleanWSSSide/cleanWSSSide.properties`:

bibiwstest.properties location of `bibi.properties` of test WebService server

bibiwsserv.properties location of `bibi.properties` of production WebService server

cleantime delete spooldir of WebService call (days) and mark `id` as cleaned in database

deletetime delete `id` from database

It should be called regularly (possibly daily as cron job) and provides logging to `/vol/bibidev/bibi/wss/cleanWSSSide/cleanWSSSide.log`.

1.4 Creating WebService projects

`makeProjectWS <PN>` creates a new project with included example WebService. It is located at `/vol/bibiadm/bin/`. Overall this creates a running WebService from the project `templatews`.

While copying directory `/vol/bibidev/templatews/` including subdirectories and all files, this tool replaces any occurrence of the word `TEMPLATEWS` by `<PN>` in the filenames and content. `templatews` is replaced by lower case of `<PN>`.

For generating a two step WebService there is the program `makeProjectWS2step`, which does the same from `/vol/bibidev/templatews2step/` with keywords `TEMPLATEWS2step` and `templatews2step`.

The generated example WebService should be easy to expand to more than two steps upon request.

⁴former `sgeid`, changed from `int` to `varchar` due to DRMAA spec.

whole Webservice:	/vol/bibidev/<PN>/
make install_serv make clean_serv	install client and server side on production systems clean client and server side on production systems
client side:	wsc/ subdirectory
make install.bibiserv make clean.bibiserv	install client side clean client side
server side:	wss/ subdirectory
ant redeploy.bibiwsserv ant deploy.bibiwsserv ant reload.bibiwsserv ant undeploy.bibiwsserv	uninstall and install server side and release WSDL install server side and release WSDL (use for 1st install) install server side, reload and release WSDL undeploy server side and remove WSDL

Table 1.2: BiBiServ Administrator Team commands of BiBiWS projects

1.5 Installation of Webservice projects on BiBiServ and BiBiServ Webservice Server

Upon request of the developer, the BiBiServ Administrator Team will install the tool on the production system. The files `Makefile` (client side) and `build.xml` (server side) contain special targets to call for installing on production system (see Table 1.2). These targets perform the same like installation on the development system, so BiBiServ Administrator Team does not have to worry about tool specific file and directories extensions: If installation by the developer on development systems works, this will work on the production system, too.

2 BiBiWS - FAQ

While developing WebServices there are several traps. We try to document the phenomena here in an unordered list:

1. **Tomcat does not find** web.xml
Problem: While deploying Webservice, Tomcat logs INFO: Missing application web.xml, using defaults only StandardEngine to log/catalina.log
Reason: Tomcat sometimes has this problem... dont know why !
Solution: Shutdown Tomcat, delete WAR and directory from webapps/, start Tomcat.
2. soapenv:Server.userException **on Webservice Call**
Problem: After deploying, calling one Webservice method quits with SOAPFault 2.1.
Reason: Failed to include tool.properties
Solution: Check and fix build.xml WAR Target to include this.

```
<soapenv:Fault>
  <faultcode>soapenv:Server.userException</faultcode>
  <faultstring>java.lang.NullPointerException</faultstring>
  <detail>
    <ns1:hostname>padouk.TechFak.Uni-Bielefeld.DE</ns1:hostname>
  </detail>
</soapenv:Fault>
```

Figure 2.1: Soap Fault of Problem 2

3 BiBiWS - ChangeLog

BiBiWS is under constant development and many features are planned to implement. If changes are made, which are independent of the guide, they will be documented in this chapter. This makes it simpler to have a up-to-date documentation.

1. 05/02 - Tmp Dirs added

Upon request BiBiWS supports now temporary files, which should be deleted before finishing Processing. Some tools generate a huge amount of data, but only while processing. These temporary files do not have to be stored in SpoolDir, because they do not belong to the result.

The related methods are located at the `WSSTools` class and correspond to the `spoolDir` methods. (see [\[BIBIWSAPI-WSS\]](#))

2. 05/04 - SGECall array-execs and strict

Some Tools need to call several executables, so `SGECall.call()` now accepts an array of strings, which will be written to the SGE shell script.

These lines could be combined with logical AND (using shell „&&“ feature) setting the boolean "strict" within constructor, so a next line would only be called if the prior is finished successfully. This way `SGECall` could also determine if the execution was successful.

3. 05/04 - Log Level *info* as standard for TemplateWS

The `TemplateWS` Webservice (and created projects) now logs to level *info* instead of *debug*. *info* is now also the standard level defined in `wss/conf/log4j.properties`. This way the BiBiWS library (logging to *debug*) is separated from log messages of the Webservice.

4. 05/05 - BioDOM added

`BioDOM` is a package for handling non-xml formats to xml formats. could use. Have a look at `biodom.jar` – Homepage will come shortly.

5. 05/06 - Parameter parser added

We now have a parameter parser/checker `WSSPropertiesParser`, which checks for types, existence and max/min values. Configuration is done by the `tool.properties`. Have a look at the updated WSS-API.

6. 05/07 - getVersions()

All Webservices now include a `getVersions()` method, which provides information about used tools and databases. Have a look at the updated WSS-API. `WSSTools` Class.

7. **05/07 - Wiki as documentation**

BiBiServ internally documents now within a wiki: <http://bibitest.techfak.uni-bielefeld.de/wiki> This is NOT accessible from outside the TechFak.

8. **05/07 - build.xml changes**

Now, the build.xml compiles always against the BiBiWS Library of the Tomcat.

9. **05/07 - RequestResponseMapping**

Some Tools require a dependency between request and response methods. Status now could hold the request method for further processing on response call. Have a look at the updated WSS-API. Status Class.

4 BiBiWS - Predefined statuscodes

Like the (for now) underlying protocol http, BiBiWS returns statuscodes with descriptive messages to inform the user if the requested data isn't returned.

For not getting confused with http statuscodes, BiBiWS uses statuscodes beginning with 6 and 7, while http uses statuscodes beginning with 1 to 4 (see RFC 2616 [RFC2616] for details).

This chapter gives an overview of the predefined statuscodes by BiBiWS.

They can be overridden by the tool specific statuscodes, simply by giving same numbers at the tool specific properties file.

There are some general conventions to follow explained in Table 4.1, which are required, because BiBiWS decides on these rules whether an error occurred, the Web-Service finished or user has to wait.

Some statuscodes can be used for several reasons and it might be hard for the developer to decide, where it came from. So all statuscodes may have a internal description (`.internal` at the `properties` file), which never leaves the server. It is just for debugging and development of the tools on server side. It can be set individually to enhance the normal description, which is presented to the user.

Predefined statuscodes occurring on server side are explained in Table 4.2 and the client side ones are described in Table 4.3. Their symbolic names begin in general with a `c`, so the statuscodes can be as equal as possible on both sides.

The `Type` is defined by the *SOAP-specification* (see [SOAP11] Chapter 4.4.1). It represents the side, where the *Error* occurs. Type *Client* means the request should not be resend without change. So we indicate *Server* for all Statuscodes representing **not** a failure on client side.

Statuscode	Type	Description
600		WebService call is finished successfully, result ready.
6xx	Server	WebService call is not yet finished. xx gives more information.
700	Server	WebService call is in general error state. This is a fall-back, should be avoided by developers to give enhanced information.
70x	Client	User errors.
72x	Server	Submitted data is NOT processed by the WebService call. Errors relaying to BiBiServ Administrator Team or developer of the tool.

Table 4.1: General BiBiWS conventions for statuscodes

Statuscode	Description
600	Finished OK
601	Submitted
602	Preprocessing
603	Processing: Pending
604	Processing: Running
605	Postprocessing
700	General Error
701	Input Format Error (submitted data does not follow input format)
702	Input Size Error (submitted data too large)
703	Execution Error (executable gives enhanced errormsg)
704	RAM Size Error (Job consumed too much memory)
705	CPU Time Error (Job consumed too much CPU time)
706	ID unknown (or older than 30 days)
707	ID data deleted (older than 3 days, result data deleted)
708	Mail Check Failed (notification email is not valid)
720	General Temporary Error = please try again later.
721	Internal Resource Error - BiBiServ Team is informed, please try again later. <i>internal description: Internal Resource Error (Grid)</i>
722	Internal Resource Error - BiBiServ Team is informed, please try again later. <i>internal description: Internal Resource Error (DB)</i>
723	Internal Resource Error - BiBiServ Team is informed, please try again later. <i>internal description: Internal Resource Error (HDD)</i>
724	Internal Resource Error - BiBiServ Team is informed, please try again later. <i>internal description: Internal Resource Error (WS-Error)</i>
725	Internal Resource Error - BiBiServ Team is informed, please try again later. <i>internal description: Internal Resource Error (BiBiWSS-Lib Error)</i>
726	External Resource Error - An error occurred while calling an external Webservice - please try again later.
727	This feature is not yet implemented. Please contact author of tool.
730	Webservice Server busy - please try again later.
731	Resource busy - please try again later. <i>internal description: Resource Busy (Grid)</i>
732	Resource busy - please try again later. <i>internal description: Resource Busy (DB)</i>
733	Resource busy - please try again later. <i>internal description: Resource Busy (HDD)</i>

Table 4.2: Predefined BiBiWS statuscodes on server side

Statuscode	Description
c600	Ready - got result
c601	Submitted
c700	General unknown error
c701	Input Error
c702	Input size too large
c708	Mail Check Failed (notification email is not valid)
c723	Internal Error - BiBiServ Team is informed, please try again later. <i>internal description:</i> Internal Resource Error (HDDfull)
c724	Internal Error - BiBiServ Team is informed, please try again later. <i>internal description:</i> Internal Resource Error (WS-Error)

Table 4.3: Predefined BiBiWS statuscodes on client side

Literaturverzeichnis

- [HMO04] Mersch, Henning: (diploma thesis)
BiBiWS - Ein Framework für die Entwicklung asynchroner WebServices auf dem BiBiServ, 2004
- [ADA1] Adams, H.: *Asynchronous operations and WebService, Part 1: A primer on asynchronous transactions*.
12.10.2004 <<http://www-106.ibm.com/developerworks/library/wsasynch1/> >
- [ADA2] Adams, H.: *Asynchronous operations and WebService, Part 2: Programming patterns to build asynchronous WebService*.
12.10.2004 <<http://www-106.ibm.com/developerworks/library/wsasynch2/> >
- [APACHE] *The Apache Software Foundation*
12.10.2004 <<http://www.apache.org/> >
- [APACHEHTTP] *The Apache Software Foundation: Apache http Server*
12.10.2004 <<http://httpd.apache.org/http/> >
- [APALIZ] *The Apache Software Foundation - Licenses*
12.10.2004 <<http://www.apache.org/licenses/> >
- [AXIS] *The Apache Software Foundation: AXIS*
12.10.2004 <<http://ws.apache.org/axis/> >
- [AXISUM] *The Apache Software Foundation: AXIS User's Manual*
12.10.2004 <<http://ws.apache.org/axis/java/user-guide.html> >
- [BIBISERVSTATS] *Statistiken des BiBiServ*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/statistics/>>
- [BIBISERVPOLICIES] *Policies of BiBiServ* 12.10.2004
<http://bibiserv.techfak.uni-bielefeld.de/bibi/Administration_Policies.html >
- [BIBISERV] *Technische Fakultät der Universität Bielefeld: The Bielefeld University Bioinformatics Server (BiBiServ)*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/> >
- [BIBISERV] *Der BiBiServ (production system)*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/> >

- [BIBITEST] *Der BiBiTest (development system)*
12.10.2004 <<http://bibitest.techfak.uni-bielefeld.de/> >
- [BIBIWSAPI-WSS] *BiBiWS server side API*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/hobit/wss-api/> >
- [BIBIWSAPI-WSC] *BiBiWS client side API*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/hobit/wsc-api.html> >
- [BIBIWSDL] *WSDL Definitionen der auf BiBiServ angebotenen WebServices*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/wsdl/> >
- [CEBITEC] *Center for Biotechnology (CeBiTec)*
12.10.2004 <<http://www.cebitec.uni-bielefeld.de/> >
- [CHAP] Chappell D. and Jewell, T.:
Java Web Services O'Reilly, 2002
- [CORBA] Sun Microsystems: *Introduction to CORBA*
12.10.2004 <<http://java.sun.com/developer/onlineTraining/corba/corba.html> >
- [CPAN] *CPAN - Comprehensive Perl Archive Network*
12.10.2004 <<http://www.cpan.org/> >
- [DFG] *Deutsche Forschungsgemeinschaft*
12.10.2004 <<http://www.dfg.de/> >
- [DOM] W3C: *Document Object Model (DOM)*
12.10.2004 <<http://www.w3.org/DOM/> >
- [DRMAA] *Distributed Resource Management Application API Working Group*
12.10.2004 <<http://www.drmaa.org/> >
- [E2G] J. Krüger, A. Sczyrba, S. Kurtz, R. Giegerich : *e2g - An Interactive Web-Based Server for Efficiently Mapping large EST and cDNA sets to Genomic Sequences*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/e2g/> >
- [GSOAP] *gSOAP: C/C++ Web Services and Clients*
12.10.2004 <<http://gsoap2.sourceforge.net/> >
- [HELMHOLTZ] *Helmholtz Association of National Research Centres*
12.10.2004 <<http://www.helmholtz.de/> >
- [HERM] Hermjakob, Henning et. al.:
The HUPO PSI Molecular Interaction Format - A community standard for the representation of protein interaction data Nature Biotechnology, 2004
- [HOBIT] *HOBIT-Projekt*
12.10.2004 <<http://hobit.sf.net> >

- [HTTP] W3C: *Hypertext Transfer Protocol – HTTP/1.1*
12.10.2004 <<http://www.w3.org/Protocols/rfc2616/rfc2616.html> >
- [JAVA] Sun Microsystems: *Java*
12.10.2004 <<http://www.java.com/> >
- [JCP] *Java Community Process - Homepage*
12.10.2004 <<http://jcp.org/en/home/index/> >
- [JRMI] David Reilly: *Introduction to Java RMI*
12.10.2004 <<http://www.javacoffeebreak.com/articles/javarmi/javarmi.html> >
- [JSP] *JavaServer Pages Technology*
12.10.2004 <<http://java.sun.com/products/jsp/> >
- [JSC] *Java Servlet 2.4 Specification*
12.10.2004 <<http://www.jcp.org/aboutJava/communityprocess/final/jsr154/> >
- [LOG4P] *The log4perl project*
12.10.2004 <<http://log4perl.sourceforge.net/> >
- [LOG4J] Apache Software Foundation: *Logging Services*
12.10.2004 <<http://logging.apache.org/log4j/> >
- [MCL] McLaughlin, B.:
Java and XML O'Reilly, 2001
- [NETCRAFT] Netcraft: *Web Server Survey* 12.10.2004
<http://news.netcraft.com/archives/2003/08/01/august_2003_web_server_survey.html
>
- [OEST] Oestereich, B.:
Objektorientierte Softwareentwicklung: Analyse und Design mit der UML Oldenbourg, 2004
- [PERL] O'Reilly: *Perl - The Source for Perl*
12.10.2004 <<http://www.perl.com/> >
- [PSI] HUPO: *Proteomics Standards Initiative*
12.10.2004 <<http://psidev.sf.net/> >
- [RAY] Ray, R. J. and Kulchenko, P.:
Programming Web Services with Perl O'Reilly, 2002
- [REPUTER] S. Kurtz, C. Schleiermacher: *REPuter: Fast Computation of Maximal Repeats in Complete Genomes*
12.10.2004 <<http://bibiserv.techfak.uni-bielefeld.de/reputer/> >
- [RFC2616] *RFC 2616: Hypertext Transfer Protocol - HTTP/1.1*
12.10.2004 <<http://www.w3.org/Protocols/rfc2616/rfc2616.html> >

- [ROSE] Rose, Marshall T.:
Beep - The Definition Guide O'Reilly, 2002
- [RPCVSDOC] *RPC vs. Document WSDL encoding*
12.10.2004 <<http://www.rassoc.com/gregr/weblog/archive.aspx?post=465> >
- [SGE] Sun MircoSystems: *Grid Engine*
12.10.2004 <<http://gridengine.sunsource.net/> >
- [SGEUM] Sun MircoSystems: *Sun grid engine user manual*
12.10.2004 <<http://gridengine.sunsource.net/project/gridengine-download/SGE53AdminUserDoc.pdf?content-type=application/pdf>>
- [SOAP11] W3C: *SOAP Version 1.1*
01.02.2005 <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/> >
- [SOAP] W3C: *SOAP Version 1.2*
12.10.2004 <<http://www.w3.org/TR/soap12-part1/> >
- [TANE] Tanenbaum, A. S.:
Computer Networks Prentice Hall, 1996
- [TOMCAT] The Apache Software Foundation: *Apache Jakarta Tomcat*
12.10.2004 <<http://jakarta.apache.org/tomcat/> >
- [UDDI] *OASIS UDDI*
12.10.2004 <<http://www.uddi.org/> >
- [UDDIPROB] *UDDI Specifications*
12.10.2004 <<http://www.oasis-open.org/committees/uddi-spec/ipr.php> >
- [WSDEF] *Web Services Activity*
12.10.2004 <<http://www.w3.org/2002/ws/> >
- [WSDL] W3C: *Web Services Description Language (WSDL) 1.1*
12.10.2004 <<http://www.w3.org/TR/wsdl> >
- [XHTML] W3C: *XHTML 1.0 - The Extensible HyperText Markup Language*
12.10.2004 <<http://www.w3.org/TR/xhtml1/> >
- [XML] W3C: *Extensible Markup Language (XML)*
12.10.2004 <<http://www.w3c.org/XML/> >
- [XMLRPC] *XML-RPC Home Page*
12.10.2004 <<http://www.xmlrpc.com/> >
- [XMLSCHEMA] W3C: *XML Schema*
12.10.2004 <<http://www.w3.org/XML/Schema> >

Abbildungsverzeichnis

1.1	JDBC Connector of Tomcat to access statuscode database	8
1.2	Installing database for BiBiWS (Postgresql syntax)	10
2.1	Soap Fault of Problem 2	13

Index

.internal, 17

Apache http Server, 5

API, 6

CGI, 5

cleanWSSSide, 11

example, 2

http, 17

JDBC, 7

makeProjectWS, 11

makeProjectWS2step, 11

Postgresql, 10

Predefined statuscodes, 17

properties, 6

RFC 2616, 17

SQL, 10

Tomcat, 7

