Institute of Mathematics and Computer Science
European Bioinformatics Institute

AIMS
Installation Guide

Version 2.10

Pre-requirements

To install Assay Information Management System the following software needs to be preinstalled on your system:

- **Apache Tomcat** suitable for your OS. Tomcat version 6.0 or higher is recommended, although in principle Tomcat versions 5.x *might* work. Tomcat installation will also require a JRE compatible with a particular Tomcat version (JDK is required if Tomcat versions 5.x is used).

- **PostgreSQL** database. The current version of AIMS is tested with PostgreSQL version 8.4 and use of this version is recommended. Alternatively, you can also use another relational database of your choice, however in this case it is likely that you will need to make some modifications in database initialization scripts. In principle AIMS has been successfully tested on MySQL and Oracle databases.

Installation procedure

AIMS installation requires the following steps:

- **Creation of AIMS database schema**. Since several AIMS instances could be run on the same server, it is assumed that an empty PostgreSQL database has already been created and initialization script will create the required tables in already existing database. There are at least two simple ways how initialization script can be used:

  - if you use **pgAdmin** tool for PostgreSQL management, select the schema of the currently empty database where you wish to install AIMS (by default this will be called *public*), open *Execute arbitrary SQL queries* window, paste the contents of installation script here and press *Execute query*.

  - insert the line `\connect <your database name>` as the first line in installation script and run the command `psql –U postgres –f <name of initialization script file>`. You may need either to place script file in a directory visible to `psql`, or give the full path of initialization script file.

- **Deployment of AIMS software (standalone version)**. The simplest way to do this is to unzip Tomcat deployment file in directory $CATALINA_HOME/webapps ($CATALINA_HOME will be defined by your Tomcat installation). The name of deployment directory can be freely chosen and/or changed. The AIMS then most likely will be accessible via web addresses **http://your.server.name:8080/<your AIMS directory>** (if Tomcat is installed on default port 8080) or **http://your.server.name/<your AIMS directory>** (if Tomcat is installed on port 80). Other ports can also be used, but if the AIMS is intended to be accessed remotely the use of standard ports 80 or 443 (if you wish to use secure HTTPS protocol for accessing AIMS).

  The other alternative is to recompile the software (for this you will need both Tomcat deployment and Java sources archives).

- **Deployment of AIMS software for use together with SIMS**. To use AIMS together with SIMS you need a SIMS instance already installed *on the same Tomcat server*. Then you should install AIMS as described above. To link AIMS with SIMS you additionally will need to configure the access to SIMS database in WEB-INF/web.xml configuration file. In more detail this process is described in a separate section of this guide.

- **AIMS configuration**. Currently all configuration settings are accessible in configuration file WEB-INF/web.xml located in directory in which you deployed your AIMS application. The content of this file is described in details in a separate section of this guide.

  In addition you might want (or need) to change some settings in file conf/systemConst.xml or in WEB-INF/classes/hibernate.properties (this will need to be modified if you use other database engine instead of PostgreSQL) and/or other .properties files in the same directory.

- **Initialization of AIMS instance**. This includes two tasks:

  - To access AIMS you need a login name and password. If no users are defined as yet (table *Users* is empty) a new user will be created the first time when you attempt to log in the system. The *login name and password for this user will be those which you will provide for this first login* (thus it is important not to lose them!), the informal description of this user will be *First user* and the user will be given full administrator rights. Also one user group, called *default group* will be created and the first user will be made member of this user group. If for some reason you lose the login name and/or password to access your AIMS instance, a possible solution is to delete all entries from tables

Users and *User groups*, the *First user* then will be recreated when you will try to access the system.

- To use AIMS generally you will need to configure at least one technology. There are several technology configurations provided with this AIMS installation: *Bisulfate sequencing, CLINPROT MS, Genotyping, LC-MS, Microarray, NMR, Protein Array, Suspension Bead Array* and *Tissue Array*. Technology configurations are provided as XML files. To install any of these technologies in **AdminTables** page click on *import* link (next to table *Technologies*) and select the appropriate XML file. The same technology configuration may be imported several times – this will lead to several identical technology configurations being installed (derived technology names will be assigned for later installations).

  There is also an XML configuration file for *empty* technology. It is recommended to install it, if you don't wish to use any other of provided technology configurations, but just make your own configurations instead. The installation of *empty* technology will put some entries in tables: *Parameter groups, Common parameter types*, and *Default common parameters*; these could be useful for creating new technologies. (These tables will be filled with the same values also during the import of any other of provided technology configurations.) For more details on how to configure a new technology see a separate section of this guide.

  It is worth to note that whilst a technology import is an "easy" process requiring few mouse clicks, deleting of existing technology is more time consuming – you will need to delete manually a number entries from **Configuration** page first.

It is possible to install several AIMS databases on PostgreSQL and it is possible to install several AIMS instances on Tomcat.

AIMS configuration

The configurable parts of WEB-INF/web.xml file are described below:

Specify URL to your AIMS database. This should include database location (in example below it is assumed that PostgreSQL is being run on the same server as Tomcat) and your database name (it example below it is assumed that database name is *AIMS*).

```
<context-param>
  <param-name>assay.url</param-name>
  <param-value>jdbc:postgresql://localhost/AIMS</param-value>
</context-param>
```

Specify user name to access database. The example below assumes that it is *postgres*, but since this is usually the default PostgreSQL user with administrator rights, creation and configuration of a special user to access AIMS is recommended.

```
<context-param>
  <param-name>assay.username</param-name>
  <param-value>postgres</param-value>
</context-param>
```

Specify password for this database user. The example below assumes that it is *my_password*, but use of more complicated passwords is recommended.

```
<context-param>
  <param-name>assay.password</param-name>
   <param-value>my_password</param-value>
</context-param>
```

Uncomment the following three parameters and configure them **if and only if** you use AIMS linked together with SIMS.

Set the value to *yes*, if you use AIMS linked together with SIMS and to *no* otherwise.

```
<context-param>
  <param-name>usesamples</param-name>
  <param-value>no</param-value>
</context-param>
```

Specify directory of your SIMS application. This should include tomcat server location (in example below it is assumed that Tomcat 6.0 directory is "/opt") and your application name (in example below it is assumed that application name is *SIMS*).

```
<context-param>
    <param-name>sample.tomcat_directory</param-name>
    <param-value>/opt/Tomcat 6.0/webapps/SIMS</param-value>
</context-param>
```

Specify URL to your SIMS database. This should include database location (in example below it is assumed that PostgreSQL is being run on the same server as Tomcat) and your database name (it example below it is assumed that database name is *SIMS*).

```
<context-param>
  <param-name>sample.url</param-name>
```

```
    <param-value>jdbc:postgresql://localhost/SIMS</param-value>
  </context-param>
```

Specify user name to access database. The example below assumes that it is *postgres*, but since this is usually the default PostgreSQL user with administrator rights, creation and configuration of a special user to access SIMS is recommended.

```
  <context-param>
    <param-name>sample.username</param-name>
    <param-value>postgres</param-value>
  </context-param>
```

Specify password for this database user. The example below assumes that it is *my_password*, but use of more complicated passwords is recommended.

```
  <context-param>
    <param-name>sample.password</param-name>
     <param-value>my_password</param-value>
  </context-param>
```

Number of entries that will be shown in a single page of **List of <level>**. (The usual level names are "Views", "Studies" and "Assays".) Chose the value that best suits your needs.

```
  <context-param>
    <param-name>viewlimit</param-name>
    <param-value>1000</param-value>
  </context-param>
```

Specify the directory where data files will be stored. The example below assumes that directory is /aims/data_files. It is it is strongly recommended to use full path.
Don't place this directory within $CATALINA_HOME/webapps – you may lose all your data files, if by accident you ask Tomcat to undeploy your AIMS instance.

```
  <context-param>
    <param-name>data.directory</param-name>
    <param-value>/aims/data_files</param-value>
  </context-param>
```

Specify the directory where supplementary files will be stored. The example below assumes that directory is /aims/supplementary_files. It is it is strongly recommended to use full path.
Don't place this directory within $CATALINA_HOME/webapps – you may lose all your supplementary files, if by accident you ask Tomcat to undeploy your AIMS instance.
Directories for data and supplementary files have to be separate.

```
  <context-param>
    <param-name>protocol.directory</param-name>
    <param-value>/aims/supplementary_files</param-value>
  </context-param>
```

Configure path to directory for export metadata files. These are temporary XML files that are created if you use "**Trasnfer to repository**" function; after the scheduled amount of time these will be sent via fto to repository to which particular transfer is configured.

```
<context-param>
    <param-name>export_file.directory</param-name>
    <param-value>/local/usr/AIMS/export_metadata_files</param-value>
```

```
      </context-param>
```

Configure path to directory for configuration files for data transfers and exports. Don't place this directory within $CATALINA_HOME/webapps – you may lose all your exported files, if by accident you ask Tomcat to undeploy your AIMS instance.

```
  <context-param>
    <param-name>data_transfer_configuration.directory</param-name>
    <param-value>/local/usr/AIMS/config_files</param-value>
  </context-param>
```

If the application is configured to work with an ftp server to use for file import working directory must be provided. You can uncomment and configure the following two parameters, if option of file upload via FTP is enabled. In Examples below it is assumed that ftp server name is my.ftp.server.com.

```
  <context-param>
    <param-name>ftp.server</param-name>
    <param-value>my.ftp.server.com</param-value>
  </context-param>
  <context-param>
    <param-name>ftp.directory</param-name>
    <param-value>ftp/server/directory</param-value>
  </context-param>
```

For the users convenience it is also advisable to configure the informative popup information. These values are used only in pop-up tip **Batch upload** pages to remind user about the available FTP account for data upload. You are likely to benefit from this option if a single account for FTP upload is shared between all users.

```
  <context-param>
    <param-name>ftp.directory.popup</param-name>
    <param-value>ftp/server/directory</param-value>
  </context-param>
  <context-param>
    <param-name>ftp.server.popup</param-name>
    <param-value>my.ftp.server.com</param-value>
  </context-param>
```

Configuration of a new Technology

Although not strictly necessary, it is recommended that you import any of technology configurations provided with this installation first (use *empty* technology, if you don't need any of "real" configurations). This will put some entries in tables: *Parameter groups*, *Common parameter types*, and *Default common parameters*.

In general, to proceed you will need at least one entry in *Parameter groups* table. Tables *Common parameter types*, and *Default common parameters* are used of initialization technology-specific tables *Parameter types* and *Common parameters* in **Configuration** page – you will need at least some parameter types and entries in *Common parameters* table is a minimal (as well as recommended) set of entries for technology configuration to work correctly (but you are free to change *Parameter long name* and *Parameter short name* fields; *Parameter upload name* field is for reference purpose only (the values are hard-coded in software, because uploading process is very specific for most of "Common parameters") and it is recommended not to change it).

Assuming that you have the suitable values in above mentioned tables *Parameter groups*, *Common parameter types*, and *Default common parameters*, a new technology can be created and configured as follows:

- In **AdminTables** page click on *new* link next to Technologies table. An Edit dialog will open in which you will be asked to provide the following information:

    - *Technology* – Technology name, this should be unique for each technology, otherwise you are free to use any string of up to 200 characters;
    - *Id prefix* – a string which will be used as part of Id when creating new Experiments or Studies, a short string of few characters is recommended;
    - *FTP directory* – subdirectory from FTP root directory from which data and supplementary files will be listed (this applies if FTP option is enabled for this AIMS configuration);
    - *Is public* – if *true*, Technology is visible for all users with access to "all" technologies, otherwise only for users explicitly configured for this technology in *Technology access rights* table.
    - *Show Views* – if *true*, Views level is available for this Technology, otherwise Views level will not be shown;
    - *Transfer type* – if some kind of transfer options are configured for the application, they can be used for some or all technologies.
    - *Comment* – informal comment.

- Go to **Configuration** page and select newly created technology. Click on *new* link next to Technologies table. You should have some values already filled-in in *Parameter types* and *Common parameters* tables. To configure your newly created technology, the following steps should be completed:

    o In *Vocabularies* table add an entry for each controlled vocabulary (CV) you wish to use. Each entry is characterized by two fields: "Table" (the name which will be referenced in Parameter types table) and "Long name" (the vocabulary title in **Vocabularies** page). There is no particular reason to use different names in these two fields.

    o Configure each of CVs as a parameter type. For this, for each vocabulary create a new entry in *Parameter types* table and provide the following fields: "Parameter type" (the name with which it will be referenced in *Parameters* table; these names should be unique within a particular technology), "Base type" (you should chose *table* for vocabularies), "Table name" (corresponding to "Table" field for just created vocabularies), "Comment" (optional arbitrary

comment). The remaining fields "Attribute I", "Attribute II" and "Popup text" are not used for *table* base types.

- o You might wish to other customized parameter types. Generally customizable base types are currently limited with table (discussed above), *datatime* (from which all customizations are provided by default configuration), *array of float* and *array of number*. The last two currently are the only ones you might wish to customize:

  - *array of float* – an array from 1 to 10 numbers of type *float* that are displayed in a single row in edit dialogs and stored within a single field in database.
  - *array of number* – an array from 1 to 10 numbers of type *integer* that are displayed in a single row in edit dialogs and stored within a single field in database.

  To create a customized type of this kind add a new entry *Parameter types* table and provide the following fields: "Parameter type" (the name with which it will be referenced in *Parameters* table; these names should be unique within a particular technology), "Base type" (you should chose *array of float* or *array of number*), "Attribute I" (size of array), "Popup text" (a pop-up tip displayed in edit dialogs) and "Comment" (optional arbitrary comment). The remaining fields "Attribute I" and "Table name" are not used for these base types.

- o Create the parameters for your technology. At each level (*Views*, *Studies*, *Assays*) all parameters are kept in database fields named *Fxx* with *xx* ranging from 01 to 55 (this could be increased by editing and recompiling source files). Thus, the number of parameters at will also be limited to 55. Available parameter types are those contained in *Parameter types* table – by default this will include types: *string* (up to 200 characters, displayed in single line), *bigstring* (up to 200 characters, displayed in multiple lines), *integer, float, boolean* (displayed as checkbox) and several *datatime* types. In addition you will have just created types: *table*, *array of float* and *array of number*.

  To add a new parameter, click on *new* link next to *Parameters* table and provide the following information:

  - *Parameter short name* – name shown as column name in **List of Views/Studies/Assays** pages,
  - *Parameter group* – parameter group (see **Parameter groups** table in **Administrator tables** page) to which the parameter belongs. Parameter groups affect the upper level header in **List of Views/Studies/Assays** pages and parameter position in Add/Edit/View dialogs (all parameters from the same group are kept together),
  - *Parameter type* – parameter type from **Parameter types** table,
  - *Parameter long name* – name shown on the left side of the field in Add/Edit/View dialogs; although not strictly enforced, this should be unique,
  - *Parameter upload name* – name used in header of batch upload file, although not strictly enforced, this should be unique,
  - *Assay column name* – the name of field which is configured (field isn't used in Assays, if left empty),
  - *Experiment column name* – the name of field which is configured (field isn't used in Experiments, if left empty),
  - *Study column name* – the name of field which is configured (field isn't used in Studies, if left empty),
  - *Required* – if *true* the field can't be left empty,

- *Assay view column #* – column number in **List of Assays** page (not shown, if 0),
- *Sortable* – if *true*, the entries in **List of Assays** can be shorted by the values of this parameter,
- *Study view column #* – column number in **List of Studies** page (not shown, if 0),
- *Sortable* – if *true*, the entries in **List of Studies** can be shorted by the values of this parameter,
- *Views view column #* – column number in **List of Views** page (not shown, if 0),
- *Sortable* – if *true*, the entries in **List of Views** can be shorted by the values of this parameter,
- *Assay row* – order number of the field in Add/Edit/View dialog for Assays (within a parameter group),
- *Study row* – order number of the field in Add/Edit/View dialog for Studies (within a parameter group),
- *View row* – order number of the field in Add/Edit/View dialog for Views (within a parameter group),
- *Show in filter* – if *true* the parameter will be included in data filter,
- *Comment* – optional informal comment.

In principle a single entry in *Parameters* table may be used to configure the field with the same name(s) in several levels (e.g. in Experiments and Assays). A potential use (and side effect) of this is for prompting of default values – if a new Assay will be created under Experiment, the edit dialog by default will offer the default value of this field that is used in Experiment.

o   If you wish you may change "Parameter long name" and "Parameter short name" as well as "… view column #" fields in *Common parameters* table. Other changes generally will not be needed.

o   You can create entries in *Data types* and *Supplementary types* entries. These are memo-only fileds that can be selected/provided in edit dialogs and batch upload files to indicate the type of data/supplementary file that is submitted.

After a new technology is created it could be a good idea to backup it by using XML export option from **AdminTables** page.

Configuration of level names and upload names

There are several constants that need to be specified. These are kept in **conf/systemConst.xml** file. They are not server specific but are instead adjusted to each configuration. **If the default configuration is used with no alterations the following can be skipped** and the file used with no change.

The file begins with an opening tag:

```
<AIMS_Constants version="1.17">
```

There are upload names for static fields; column values are not to be changed, upload values may be changed:

```
<AssayUploadNames>

<UploadName column="visibleName" upload="ID"/>

<UploadName column="comment" upload="COMMENT"/>

<UploadName column="dataFiles" upload="DATA_F"/>

…

<UploadName column="modifier" upload=""/>

</AssayUploadNames>
```

This is where you would change the level names of the application. The top level may contain more than one entry, each of the top pages would have a many-to-many relation with middle level:

```
<AssayMainNames>

<Bottom plural="Assays" singular="Assay" prefix="Assay."/>

<Middle plural="Studies" singular="Study" prefix="Study."/>

<Top>

<View plural="Views" singular="View" prefix="View." name_prefix="VA-"/>

</Top>

</AssayMainNames>
```

The file ends with a closing tag.

```
</AIMS_Constants>
```

This file is read by the application only once at startup. For any changes thereafter to take place the application needs to be reloaded by Tomcat server.