# ABng administrator's guide

Daniel Mencnarowski

July 1999

## Contents

# 1 Introduction

The guide is purposed to help maintain *ABng* installation performing common administrative tasks, keeping the system running. It also identifies the directory structure and all components of the system, identifying all the source code location and procedures for compiling and installing the software.

# 2 ABng system

## 2.1 Directories structure

```
/home/iris/grant/abng/
  CVSROOT/
  bin/ -> /home/iris/abs/bin
  config/ -> /home/iris/abs/config
  data/ -> /home/iris/abs/data
  doc/
    admin_guide/
    arch/
    design/
    html/
    impl/
    orl_reviews/
    progress_report1/
    raport_au/
    skeleton/
  papers/
  etc/ -> /home/iris/abs/etc
  idl -> /home/iris/abs/idl
  lib -> /home/iris/abs/lib
  sources/
    common/ -> /home/iris/abs/sources/common
    common_java -> /home/iris/abs/sources/common_java
    servers/
      abng_cs_cpp/
      abng_location_description_rep/
```

```
  away_rep/
  badge_holder_rep/
  badge_rep/
  domain_cpp_rep/
  domain_java_rep/
  equipment_rep/
  location_description_rep/
  location_type_rep/
  meta_equipment_rep/
  scheduler/
  sensor_rep/
  server_manager/
  sighting_processors/
  user_rep
utils/
  NamingService_util/
  abng_location_description_rep_util/
  badge_holder_rep_util/
  badge_rep_util/
  commander_util/
  equipment_rep_util/
  event_log_util/
  location_description_rep_util/
  location_type_rep_util/
  logger/
  meta_equipment_rep_util/
  ping/
  ps_dev_util/
  sensor_rep_util/
  server_manager_util/
  sighting_processors_util/
  user_rep_util/
clients/
  Jabngaway/
  Jabngvisitor/
  RepManager/
  abngaway/
```

```
    abngvisitor/
/home/iris/abs/
  bin/
  config/
  data/ -> data_Orbix2.3
  data_Orbix2.3/
  data_omniORB/
  doc/
  etc/
  idl/
  lib/
  sources/
    Scheduler_Client/
    WRScheduler/
    abfinger/
    ablock/
    abtrace/
    common/
    common_java/
      all/
      OrbixWeb3.0/
      VBroker3.2/
    eq_scheduler/
    whereis/
```

## 2.2 Data and configuration files

When building or running ABng system it is essential to have certain environment variables set. It can be done by sourcing an appropriate script file: `/home/iris/abs/etc/abs_orbix` for Orbix version, or `/home/iris/abs/-etc/abs_omniorb` for OmniORB version.

All instructions contained in this document assume that the above scripts were run.

The main database directory for ABng system is `/home/iris/abs/data` for Orbix version and `/home/iris/abs/data_omniORB` for OmniORB version. They contain database files for repositories, for each of the view servers it contains a subdirectory named after the view name. There are two kinds of files: the repository databases (with an extension `.db`) and databases of observers (with extensions `.db.observer` and `.db.dynamic_observers`).

As far as the the structure of the `.db` files is explained in the files with BNF metalanguage, the observer files are not. Each of the observer files starts with a number denoting number of observers followed by a sequence of records consisting of an IOR of an observer and a number of repository records it observes, followed by repository item identifiers of objects. Most of the observer databases have an additional part consisting again of a number of observers followed by IORs of observers only. These are so called "add observers", that are interested in being informed about the fact that a new record has been added to the repository, as contrary to the other observers, that are interested in specific set of existing records.

The configuration files for Imake Makefile preparation system are put into directory /home/iris/grant/abng/config, they may require modification when an updated version of an ORB is installed.

## 2.3 Naming Service structure

The names in Naming Service are sequences of simple name components, that in turn consist of two string fields ("id" and "kind"). For ABng system it is assumed that these two fields are always equal, so all the names presented here are with "id" and "kind" fields equal.

The root naming context for the system is stored in a file. The environment variable `NS_IOR_FILE` contains the name of this file. All the names in this document will be relative to the above root naming context. As the Name Service implementation is shared with other systems, besides ABng, all the context names in this system starts from a context `ABng`.

Please note that when contacting the server manager, the objects with names ending with "@" sign, or residing inside a context with such a name are considered to be crucial for security and they are not visible using the object implementing ABng_Server_Manager::Register, but only with ABng_Server_Manager::Register_Manager object.

This feature was introduced in order to separate access to public objects, from the management ones.

Normal clients should have access only to the object implementing ABng_Server_Manager::Register object that permits access only to public objects, while managing client and core servers may have access to ABng_Server_-Manager::Register_Manager.

**ABng.EventLog** — the naming context containing references of event logs (interface EventLog), used only by MSP and view servers, used to measure times of message transmission, considered obsolete,

**ABng.Bootstrap@** — bootstrap object (interface ABng_Server_Manager::-Register_Manager),

**ABng.Bootstrap_Client.Client** — bootstrap object (interface ABng_Server_-Manager::Register),

**ABng.ps_dev** — the naming context containing references of server processes (interface ABng_Server_Manager::ProcessControl), allowing utility programs to query for status, kill processes, etc. The utility program `abngps` uses this feature to show a complete list of servers running (see section 2.7.2),

**ABng.Domain_Repository** — the naming context with objects of servers of Domain_Repository module,

**ABng.Domain_Repository.Management** — the naming context with management objects of Domain_Repository module, all the names in this context should end with "@" sign, currently it contains the following objects:

Location_Type_Repository_Manager@

User_Repository_Manager@

Meta_Equipment_Repository_Manager@

Location_Description_Repository_Manager@

Location_Description_Repository_Manager@

Equipment_Repository_Manager@

**ABng.Domain_Repository.Public** — contains public interfaces for using domain repositories, currently it has the following contents:

Meta_Equipment_Repository

Equipment_Repository

Location_Type_Repository

User_Repository

Location_Description_Repository

**ABng.ABng_Core**

**ABng.ABng_Core.Public** — public objects of the ABng core services:

**Away_Repository**

**Views** — the context containing references of public interface of view servers

**ABng.ABng_Core.Management** -

Sensor_Repository@

Badge_Holder_Repository@

ABng_Location_Description@

Away_Repository_Manager@

Badge_Repository@

View_Managers@ — a naming context with managing interfaces of view servers

**ABng.ABng_Core.System** -

Vsps

Pollers

Main_Sighting_Processor@

## 2.4 IDL files

The IDL files of ABng system are contained in directory `/home/iris/-grant/abng/idl`. They are separated into several modules: `Events`, `Naming`, `abng_core`, `abng_core_internals`, `abng_core_management`, `abng_core_system`, `abng_poller`, `abng_server_manager`, `domain_rep`, `domain_rep_management`, `event_log`. All modules are placed inside `._sk_.idl` files, they are transformed into corresponding `.idl` files with the help of some scripts, they should not be modified, as they are only intermediate files produced from `._sk_.idl` files. The dependencies of IDL modules are but on `._sk_.idl` files, so if you want to rebuild an IDL module, you should `touch` the appropriate `._sk_.idl` file. Inclusions of normal idl files are done with `#include` directives, but inclusions of other modules must be done with `INCLUDE_MODULE` instead.

## 2.5 Common libraries

The sources of common libraries are put in directories `/home/iris/grant/-abng/sources/common` (for C++ servers, clients and utilities) and `/home/-iris/grant/abng/sources/common_java` (for Java). The C++ version is a true static library called `ncommon`, in order to build and install it, one should go to /home/iris/grant/abng/sources/common and issue commands:
abmkmf /home/iris/grant/abng
make install

It should place a file libncommon.a into directory `/home/iris/grant/-abng/lib/ORBname`, where ORBname is either OrbixMT_23c03-01 or omniORB_2.6.0, or any other ORB name configured.

The Java common files are not put into a library, they are instead compiled each time with the rest of source files of a given Java server or client. It is composed of three subdirectories: all, OrbixWeb3.0, VBroker3.2. The subdirectory `all` should be compiled by all Java clients and servers, while `OrbixWeb3.0` should be compiled only by those using OrbixWeb. The last subdirectory is meant for those using VisiBroker, although it has not been tested for a while and may not be up to date.

## 2.6 Servers

All servers must be started inside a data directory, i.e. `/home/iris/abs/-data` (for Orbix version), or `/home/iris/abs/data_omniORB` (for OmniORB version), were they not run this way, most of the can be configured to use a other data directory using `-d` option.

The complete description of servers and their implementation details can be found in "ABng, Active Badge - next generation, Implementation Description".

### 2.6.1 Server manager

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/server_-manager` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngsm` *or*<br>`xdrun sm.ux abngsm` |
| DB files: | `server_mgr.db` |

### 2.6.2 User repository

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/user_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngurep` *or*<br>`xdrun urep.ux abngurep` |
| DB files: | `User_Rep.db`<br>`User_Rep.db.observers` |

### 2.6.3 Location type repository

| | |
|---|---|
| Author(s): | Małgorzata Steinder |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/-location_type_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngltrep` *or*<br>`xdrun ltrep.ux abngltrep` |
| DB files: | `Location_Type_Rep.db`<br>`Location_Type_Rep.db.observers` |

### 2.6.4 Location description repository

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | Java |
| Source files: | `/home/iris/grant/abng/sources/servers/-location_description_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make all install` |
| Run: | `abngldrep` *or*<br>`xdrun ldrep.ux abngldrep` |
| DB files: | `Location_Description_Rep.db`<br>`Location_Description_Rep.db.observers` |

### 2.6.5 Meta equipment repository

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/meta_-equipment_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngmerep` *or*<br>`xdrun merep.ux abngmerep` |
| DB files: | `Meta_Equipment_Rep.db`<br>`Meta_Equipment_Rep.db.observers` |

### 2.6.6 Equipment repository

| | |
|---|---|
| Author(s): | Małgorzata Steinder |
| Language: | Java |
| Source files: | `/home/iris/grant/abng/sources/servers/-equipment_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngerep` *or*<br>`xdrun erep.ux abngerep` |
| DB files: | `Equipment_Rep.db` |

### 2.6.7 Sensor repository

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/sensor_-rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngmerep` *or*<br>`xdrun merep.ux abngmerep` |
| DB files: | `Meta_Equipment_Rep.db`<br>`Meta_Equipment_Rep.db.observers` |

### 2.6.8 Badge repository

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/badge_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngbrep` *or*<br>`xdrun brep.ux abngbrep` |
| DB files: | `Badge_Rep.db`<br>`Badge_Rep.db.observers` |

### 2.6.9 Badge holder repository

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/badge_holder_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngbhrep` *or*<br>`xdrun bhrep.ux abngbhrep` |
| DB files: | `Badge_Holder_Rep.db`<br>`Badge_Holder_Rep.db.observers` |

### 2.6.10 ABng location description repository

| | |
|---|---|
| Author(s): | Małgorzata Steinder |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/abng_location_description_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngaldrep` *or*<br>`xdrun aldrep.ux abngaldrep` |
| DB files: | `ABng_Location_Description_Rep.db`<br>`ABng_Location_Description_Rep.db.observers` |

### 2.6.11 Away repository

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/away_rep` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngarep` *or*<br>`xdrun arep.ux abngarep` |
| DB files: | `Away_Rep.db`<br>`Away_Rep.db.observers` |

### 2.6.12 Scheduler

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/sched` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngsched` *or*<br>`xdrun sched.ux abngsched` |
| DB files: | none |

### 2.6.13 View and MSP

| | |
|---|---|
| Author(s): | Jakub Szymaszek, Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/servers/-`<br>`sighting_processors` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngmsp` *or*<br>`xdrun msp.ux abngmsp`<br><br>`abngview viewname` *or*<br>`xdrun viewname.ux abngview viewname` |
| DB files: | `viewname/Locatable_Rep.db`<br>`viewname/Locatable_Rep.db.dynamic_observers`<br>`viewname/Locatable_Rep.db.observers`<br>`viewname/Location_Rep.db`<br>`viewname/Location_Rep.db.dynamic_observers`<br>`viewname/Location_Rep.db.observers`<br>`View.db` |

## 2.7 Utilities

### 2.7.1 Naming service utility

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/Naming-Service_util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `nstools` |

The `nstools` command lists Naming Service contents, allows one to put an object reference, create a symbolic link to another naming context, etc.

Complex names in Naming Service are written in dotted notation; the components are separated with dots. It is assumed that "kind" and "name" fields of CosNaming::Name are equal, if they are not, they should be separated with a minus sign.

```
ctx1.ctx2.node
ctx1_name-ctx1_kind.ctx2.node_name-node_kind
```

The command line options of the programs are as follows:

```
-t                -list root context
-l context_name   -list context
-c context_name   -bind new context
-p name           -put an object in NamingService
                          (IOR read from standard input)
-s name           -symlink a context
                          (a source IOR read from standard input)
-g name           -get an object from NamingService
-i name           -ping an object
-r name           -remove (unbind) an object from NamingService
```

### 2.7.2 Process status utility

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/ps_dev_util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngps` |

The command utility is provided to check status of servers working in *ABng* system, whether ther are up, down, or deadlocked.

```
-l                    list all servers
-c server_name        check detailed server status
-k server_name        kill named server
```

### 2.7.3 Ping

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/ping` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `pingref` |

Tries to ping an IOR reference supplied to standard input.

### 2.7.4 User repository utility

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/user_rep_-`<br>`util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngurepcl` |

The command line tool for adding, removing and listing records in User repository.

```
Usage:  abngurepcl add [options] <"uid(identifier)
           forename(identifier) surname(identifier)
```

```
            position(identifier) email(identifier)
            phone(identifier)">
        abngurepcl remove [options] <item ID | item IOR>
        abngurepcl update [options] <item ID | item IOR>
            <"uid(identifier)
            forename(identifier) surname(identifier)
            position(identifier) email(identifier)
            phone(identifier)">
        abngurepcl find | find_values [options]

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

#### 2.7.5 Location type repository utility

| | |
|---|---|
| Author(s): | Małgorzata Steinder |
| Language: | C++ |
| Source files: | /home/iris/grant/abng/sources/utils/location_-type_rep_util |
| Build: | abmkmf /home/iris/grant/abng<br>make install |
| Run: | abngltrepcl |

The command line tool for adding, removing and listing records in Location Type repository.

```
Usage:  abngltrepcl add [options] <"name(identifier)
            number_of_properties(long) property_name(identifier)
            property_type(type_name) ...">
        abngltrepcl remove [options] <item ID | item IOR>
        abngltrepcl update [options] <item ID | item IOR>
            <"name(identifier) number_of_properties(long)
            property_name(identifier) property_type(type_name) ...">
        abngltrepcl find | find_values [options]

The following options may be used:
```

```
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.6 Location description repository utility

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | /home/iris/grant/abng/sources/utils/location_-description_rep_util/client_c |
| Build: | abmkmf /home/iris/grant/abng<br>make install |
| Run: | abngldrepcl |

The command line tool for adding, removing and listing records in Location Description repository.

```
Usage:  abngldrepcl add [options] <"property_name
           property_value(appropriate type)
           property_value(appropriate type) ...">
        abngldrepcl remove [options] <item ID | item IOR>
        abngldrepcl update [options] <item ID | item IOR>
           <"property_name
           property_value(appropriate type)
           property_value(appropriate type) ...">
        abngldrepcl find | find_values [options]

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.7 Meta equipment repository utility

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/meta_-equipment_rep_util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngmerepcl` |

The command line tool for adding, removing and listing records in Meta Equipment repository.

```
Usage:  abngmerepcl add [options] <"type(identifier)
           vendor(identifier) model(identifier) height(long)
           width(long) depth(long) number_of_properties(long)
           property_name(identifier)
           property_type(type_name) ...">
        abngmerepcl remove [options] <item ID | item IOR>
        abngmerepcl update [options] <item ID | item IOR>
           <"type(identifier) vendor(identifier)
           model(identifier) height(long)  width(long)
           depth(long) number_of_properties(long)
           property_name(identifier)
           property_type(type_name) ...">
        abngmerepcl find | find_values [options]

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.8 Equipment repository utility

| | |
|---|---|
| Author(s): | Małgorzata Steinder |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/equipment_-rep_util/client_c` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngerepcl` |

The command line tool for adding, removing and listing records in Location Description repository.

```
Usage:  abngerepcl add [options]
           <"meta_equipment_[ID(long)|IOR(string)]
           name(string) last_command(string)
           property_value(appropriate type)
           property_value(appropriate type) ...">
        abngerepcl remove [options] <item ID | item IOR>
        abngerepcl update [options] <item ID | item IOR>
           <"meta_equipment_[ID(long)|IOR(string)]
           name(string) last_command(string)
           property_value(appropriate type)
           property_value(appropriate type) ...">
        abngerepcl find | find_values [options]

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.9 Sensor repository utility

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/sensor_rep_util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngsrepcl` |

The command line tool for adding, removing and listing records in Sensor repository.

```
Usage:  abngsrepcl add [options] <"sensor_number(long)
           network_number(long) group_name(identifier)">
        abngsrepcl remove [options] <item ID | item IOR>
        abngsrepcl update [options] <item ID | item IOR>
           <"sensor_number(long)
           network_number(long) group_name(identifier)">
        abngsrepcl find | find_values [options]

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.10 Badge repository utility

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/badge_rep_util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngbrepcl` |

The command line tool for adding, removing and listing records in Badge repository.

```
Usage:  abngltrepcl add [options] <"name(identifier)
```

```
        number_of_properties(long) property_name(identifier)
        property_type(type_name) ...">
     abngltrepcl remove [options] <item ID | item IOR>
     abngltrepcl update [options] <item ID | item IOR>
        <"name(identifier) number_of_properties(long)
        property_name(identifier)
        property_type(type_name) ...">
     abngltrepcl find | find_values [options]

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

#### 2.7.11 Badge holder repository utility

| | |
|---|---|
| Author(s): | Andrzej Uszok |
| Language: | C++ |
| Source files: | /home/iris/grant/abng/sources/utils/badge_-holder_rep_util |
| Build: | abmkmf /home/iris/grant/abng<br>make install |
| Run: | abngbhrepcl |

The command line tool for adding, removing and listing records in Badge Holder repository.

```
Usage: abngbhrepcl add [options] <"my_badge_[ID|IOR]
           current_badge_[ID|IOR] home_sensor_[ID|IOR]
           kind_of_description_object_[U|E]
           description_object_[ID|IOR] is_assigned is_visitor">
       abngbhrepcl remove [options] <item ID | item IOR>
       abngbhrepcl update [options] <item ID | item IOR>
           <"my_badge_[ID|IOR] current_badge_[ID|IOR]
           home_sensor_[ID|IOR] kind_of_description_object_[U|E]
           description_object_[ID|IOR]
           is_assigned is_visitor">
       abngbhrepcl find | find_values [options]
```

```
The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.12 ABng location description repository utility

| | |
|---|---|
| Author(s): | Malgorzata Steinder |
| Language: | C++ |
| Source files: | /home/iris/grant/abng/sources/utils/abng_-<br>location_description_rep_util |
| Build: | abmkmf /home/iris/grant/abng<br>make install |
| Run: | abngaldrepcl |

The command line tool for adding, removing and listing records in ABng Location Description repository.

```
Usage:  abngaldrepcl add [options] <"location_descr_[ID|IOR]
           sensor_number sensor_[ID|IOR] sensor_[ID|IOR] ... ">
        abngaldrepcl remove [options] <item ID | item IOR>
        abngaldrepcl update [options] <item ID | item IOR>
           <"location_descr_[ID|IOR]
           sensor_number sensor_[ID|IOR]
           sensor_[ID|IOR] ... ">
        abngaldrepcl find | find_values [options]
        abngaldrepcl remove_sensor [options]
           <item ID | item IOR>
           <sensor ID | sensor IOR>
        abngaldrepcl add_sensor [options]
           <item ID | item IOR>
           <sensor ID | sensor IOR>

The following options may be used:
-i print RepositoryId instead of IOR
-r print IOR (default)
-h print this help info
```

### 2.7.13 Logger

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/logger` |
| Build: | `make install` |
| Run: | `uxlogger` |

### 2.7.14 View utility

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/utils/sighting_-processors_util` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngviewcl` |

A command line client for view servers, allows one to list locatables (users and equipment), locations of a given view, add and remove locations and locatables as well as display current whereabouts of locatables.

The burst time specifies the maximum time (in milliseconds) a sighting may be buffered inside a view server before it is sent to an observer (e.g. Jabba application), setting it to zero causes immediate forwarding of sightings. Setting this parameter may decrease system load when a lot of locatables and observers is involved.

```
-v view_name
-a abng_location_description_item_id -add location
-y location_item_id                  -remove location
-b badge_holder_item_id              -add locatable
-z locatable_item_id                 -remove locatable
-t locatable_item_id                 -get sighting
-r                                   -list users of a view
-u sighting_period                   -hours since last sighting
-s location_item_id                  -get sightings
-x                                   -get sightings (view method)
-l                                   -list view content
-o                                   -list view observers
```

```
-q burst_delay_time                     -set burst delay time
-w                                      -get burst delay time
-c                                      -get sighting channel
-e IOR                                  -set sighting channel
```

#### 2.7.15 Commander utility

## 2.8 Clients

### 2.8.1 Away repository client

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | Java |
| Source files: | `/home/iris/grant/abng/sources/clients/-Jabngaway` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `Jabngaway` |
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/clients/abngaway` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `abngaway` |

```
-u user_id             use this user id
-l                     list away messages
-a                     list all away messages
-v                     more verbose listing
-d away_id             remove an away message
-D                     delete current away message
-h -?                  display this help screen
```

Both clients are used to list and edit away messages in ABng system.

### 2.8.2 Visitor client

| | |
|---|---|
| Author(s): | Daniel Mencnarowski |
| Language: | Java |
| Source files: | `/home/iris/grant/abng/sources/clients/-` `Jabngvisitor` |
| Build: | `abmkmf /home/iris/grant/abng` `make install` |
| Run: | `Jabngvisitor` |
| Author(s): | Daniel Mencnarowski |
| Language: | C++ |
| Source files: | `/home/iris/grant/abng/sources/clients/-` `abngvisitor` |
| Build: | `abmkmf /home/iris/grant/abng` `make install` |
| Run: | `abngvisitor` |

```
-l               list current visitors (default)
-a               list all visitor badges
-b               list available visitor badges
-n visitor_id    add a new visitor
-r visitor_id    remove a visitor
-R               remove all visitors (must by accompanied by -s)
-s               must be specified to "remove all" could take effect
-h -?            display this help screen
```

These clients list available visitor badges and their assignments to respective guests' names, change these assignments when a new guest arrives or leaves the system.

### 2.8.3 New repository manager

### 2.8.4 Jabba

| | |
|---|---|
| Author(s): | Sławomir Zieliński, Joanna Chlebek, Grzegorz Wójtowicz |
| Language: | Java |
| Source files: | `/home/iris/grant/abng/sources/clients/jabba2.1` |
| Build: | `./makeall` |
| Run: | `run11` *or jabba* |

### 2.8.5 Finger

| | |
|---|---|
| Author(s): | Dariusz Żbik |
| Language: | C++ |
| Source files: | `/home/iris/abs/sources/abfinger` |
| Build: | Build the `abtrace` first |
| | `cd finger; make all` |
| | `cp finger /home/iris/abs/bin/ORBName/` |
| Run: | `finger` |

The Makefile for this client is not generated for an Imakefile, so it is necessary to edit it manually in order to accommodate it to a configuration change.

A standard UNIX finger client, that displays also information about users retrieved from *ABng* system.

### 2.8.6 X lock

| | |
|---|---|
| Author(s): | Dariusz Żbik |
| Language: | C++ |
| Source files: | `/home/iris/abs/sources/ablock` |
| Build: | Build the `abtrace` first |
| | `make all` |
| | `cp ablock /home/iris/abs/bin/ORBName/` |
| Run: | `ablock` |

The Makefile for this client is not generated for an Imakefile, so it is necessary to edit it manually in order to accommodate it to a configuration change.

### 2.8.7 Trace

| | |
|---|---|
| Author(s): | Dariusz Żbik |
| Language: | C++ |
| Source files: | `/home/iris/abs/sources/abtrace` |
| Build: | `abmkmf /home/iris/grant/abng` |
| | `make install` |
| Run: | `abtrace user_id` |

```
abtrace
        [-a] [-v view_name] username
```

### 2.8.8 Whereis

| | |
|---|---|
| Author(s): | Dariusz Żbik |
| Language: | C++ |
| Source files: | `/home/iris/abs/sources/whereis` |
| Build: | `abmkmf /home/iris/grant/abng`<br>`make install` |
| Run: | `whereis user_id` |

# 3 Running ABng system

Running from scratch (after machine reboot):

1. Start Orbix daemon: `orbixd -u -s`,
2. Start Naming service, the Naming service is properly configured in Implementation Repository and orbixd, so the easiest way to do it is to issue command `lsns`. The first time it is issued it may produce some error message, but the next `lsns` command should produce a clear listing of the Naming service contents. Were it still producing error messages, it means that the Naming service database is corrupted and needs to be cleared: `'rm' -f /home/iris/abs/etc/OrbixMT_23c03-01.cfg/-Names/*` and than one can try to start it again,
3. If the machine was rebooted it is necessary to remove UNIX pipes of logging servers, i.e. `'rm' -f /home/iris/abs/data/*.ux`. This step in only necessary after a reboot, or if the logger processes (`uxlogger`) were killed by some other mean,
4. Start ABng server manager: `xdrun sm.ux abngsm`,
5. now one can start subsequently all other ABng servers, the preferred sequence is: poller(s), abngsched, abngmsp, abngurep, abngsrep, abngbrep, abngltrep, abngldrep, abngmerep, abngerep, abngaldrep, abngbhrep, abngarep, abngview.

Running the OmniORB version is roughly the same with the exception of the first point:

1. Start omniNames, get its IOR reference and store it to a file that `NS_IOR_FILE` environment variable points to.

## 3.1 Server dependencies

The runtime dependencies of server in ABng system are shown in Figure 1.

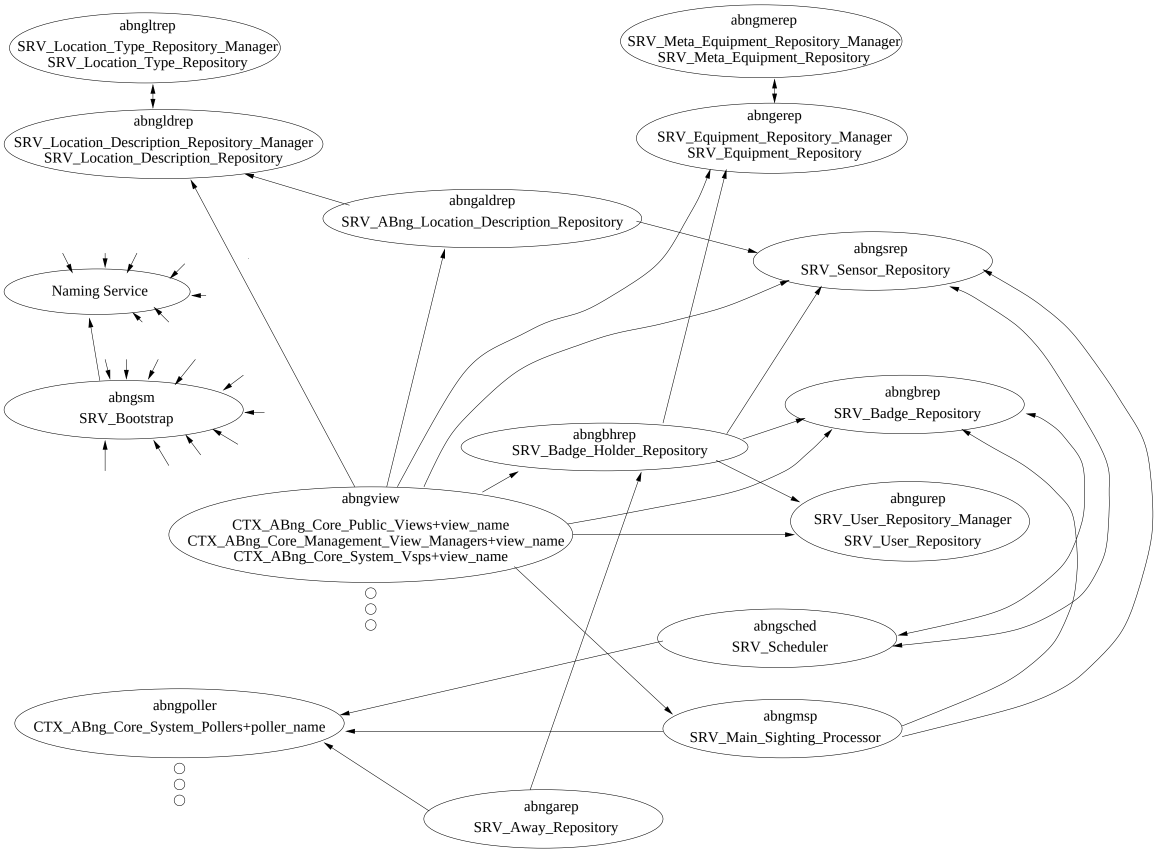


Figure 1: ABng servers dependencies

## 3.2 Database records dependencies

| Removing a record from | Side effects |
|---|---|
| User | removing badge holders, locatables (in view servers) and away messages that the user record is connected to |
| Away | none |
| Badge Holder | removing locatables (in view servers) and away messages that the badge holder record is connected to |
| Badge | removing badge holders, locatables (in view servers) and away messages that the badge record is connected to |
| Equipment | removing badge holders, locatables (in view servers) and away messages that the equipment record is connected to |
| Location Descr. | removing ABng location descriptions, locations (in view servers) that the record is connected to |
| Sensor | removing a sensor from records in ABng location descriptions |
| Location Type | removing location descriptions of a given type and then in turn ABng location descriptions, locations (in a view servers), removing a record from a location type repository should be considered a very dangerous step |
| Meta Equipment | removing equipment records of a given type, and badge holders, locatables (in view servers), away messages that the removed equipment records are connected to |

Also some updates may occur to be more dangerous than it seems, especially for meta equipment and location type repositories. Modifying a list of properties for these repositories requires special attention, because removing an existing property, or changing its type may cause removing or resetting to undefined values respective descriptions in location description and equipment repositories.

## 3.3 Crashes

If one of the servers crashes it may be observed with `abngps` client:

```
abngps -l
```

The usual procedure is to restart the crashed server. If for some reason it doesn't work and system becomes unstable, try to restart some of the servers that depend upon the crashed one.

### 3.4 Hang-ups

The server hang-ups usually can be discovered issuing the following command:

```
abngps -l
```

After identifying a server (or servers) that is (are) not responding select the one not dependent upon any of the hang-up servers and see its current log with `xdview` utility (if available) to verify if this server is not performing some long–running task (like notyfing clients), because the apparent deadlock may disappear after some time if left unattended. After verifying it is a true deadlock one should kill the offending server (with UNIX kill command). Now verify the other servers, if they are still deadlocked, they should be killed in turn. When all hang-up servers are killed or returned to normal operation one can start running servers preferably from those that do not have dependent servers (or have them least).

It may occur that killing a server and starting it again does not solve the problem, it may occur to be necessary to remove databases with observers (`.observers` and/or `.dynamic_observers`), doing it requires restart of all servers and clients depending upon the server.

## 4 Common administrative tasks

### 4.1 Adding a new user

1. Add a new user to user repository with `abngurepcl` client, not its repository item identifier, i.e.:

   ```
   abngurepcl add "jdoe John Doe student jdoe@domain.org
                    '617 3982'"
   ```

   The literals (strings) that contain spaces must be delimited with apostrophes,

2. Add a new badge (if a user is about to get a new unused badge) to badge repository with `abngbrepcl` client, again not its repository item identifier, i.e.:

   ```
   abngbrepcl -i "<13 b1> 0"
   ```

   The "0" as a parameter above is the badge holder repository item id, it should be updated to a proper value after creating the badge holder,

3. Add a badge holder to badge holder repository:

   ```
   abngbhrepcl add -i "badge_id badge_id 0 U user_id
                        true false"
   ```

   the `badge_id` and `user_id` are repository item identifiers noted in the previous steps, now note also repository item identifier of added badge holder,

4. Update the badge repository:

   ```
   abngbhrepcl update -i badge_id "<13 b1> badge_holder_id"
   ```

5. Add the badge holder to one or more views:

   ```
   abngviewcl -v viewroom -b badge_holder_id
   ```

## 4.2 Adding a new piece of equipment

Each kind of equipment has its own set of properties describing its individual parameters. The property types and names are stored in the meta equipment repository.

The following example will show how to add a new PC to the equipment repository:

1. First lets find the properties required for a PC:

   ```
   abngmerepcl find_values | grep PC | grep mini_tower
   35. 29 PC FarEast_kit mini_tower 40 60 43 ANYTHING ELSE 0
       processor_type 0 clockrate 1 RAM 1 total_disk_space 1
       graphics 0 monitor_size 1
   ```

This meta equipment description requires the following properties: `processor_type` (string), `clockrate` (long), `RAM` (long), `total_disk_space` (long), `grahpics` (string), `monitor_size` (long). One should note also the repository item identifier of this meta equipment description (29).

2. Now one can add a record to the equipment repository:

   ```
   abngerepcl add -i "29 netm '' PII 350 128 4
                       'ATI XPERT 98' 17"
   ```

   The first three parameters inside a quote are: repository item of the meta equipment record, the name (must be unique), and last command (should be empty), the following parameters are values of properties described above.

3. Add a badge for this piece of equipment to the badge repository, note its repository item id

4. Now add a record to badge holder repository:

   ```
   abngbhrepcl add -i "badge_id badge_id 0 E equipment_id
                       true false"
   ```

5. Add the badge holder to one or more views:

   ```
   abngviewcl -v viewroom -b badge_holder_id
   ```

## 4.3 Setting up a new view server

```
xdrun viewname.ux -s viewname viewtype
```

**viewname** — an arbitrary chosen unique view name,

**viewtype** — a name of the location type the view is created for.

After the first run it can (in fact it must) be started the other way:

```
xdrun viewname.ux viewname
```

When a new view server is created it is empty; it has no locations or locatables. To fill it, please follow steps for adding new users, equipment and locations.

### 4.4 Adding a new location

1. Add a location description to the database:

```
abngldrepcl add  "room 423a 'AL' 1
                  3982 0 0 yellow 16777088
                  200 0 200 200 6 0 50 40 2 150 40''
```

**room** — location type name (usually "room"),

**423a** — room number (a string),

**AL** — room name (a string),

**1 3982** — telephones (a sequence of strings), there is a number of phones followed by phone numbers separated by spaces,

**0** — people (a sequence of unsigned longs), there is a number of people in the room followed by repository item identifiers of user records in user repository (unused),

**0** — equipment (a sequence of unsigned longs), there is a number of pieces of equipment in the room followed by repository item identifiers of equipment records in equipment repository (unused),

**yellow** — a color of the room (a string), unused,

**16777088** — a 24-bit RGB color as displayed by Jabba (unsigned long),

**200,0** — the coordinates of the upper–left corner of the room as displayed by Jabba; x,y respectively,

**200,200** — the size of the room; x, y respectively,

**6 0 50 40 2 150 40** — doors in the room (a sequence of unsigned longs), the sequence length is 3*n (n is the number of doors), followed by descriptions of doors. Each 3 numbers represent a door in a room: the first specifies the placement of the doors (0 – left wall, 1 – upper wall, 2 – right wall, 3 – bottom wall), the second is the position of the center of the door, as a distance from the corner of the respective wall, the distance is measured clockwise, the third number is the width of the door. This sequence represents only a door for a specific room, in a room, or a corridor the door leads to, there should be a matching door description.

2. Add an ABng location description record:

   ```
   abngaldrepcl add -i "loc_descr_id 2 23 34"
   ```

   **loc_descr_id** — location description repository item identifier, as added in the previous step,

   **2 23 34** — the location will consist of 2 sensors with sensor repository item identifiers 23 and 34, one should add them to the sensor repository, if they are new. Please note that you cannot use a sensor that is already used by some other location with the same location type name (i.e. "room").

3. Add a location to view server(s)

   ```
   abngviewcl -v viewroom -a abng_loc_descr_item_id
   ```