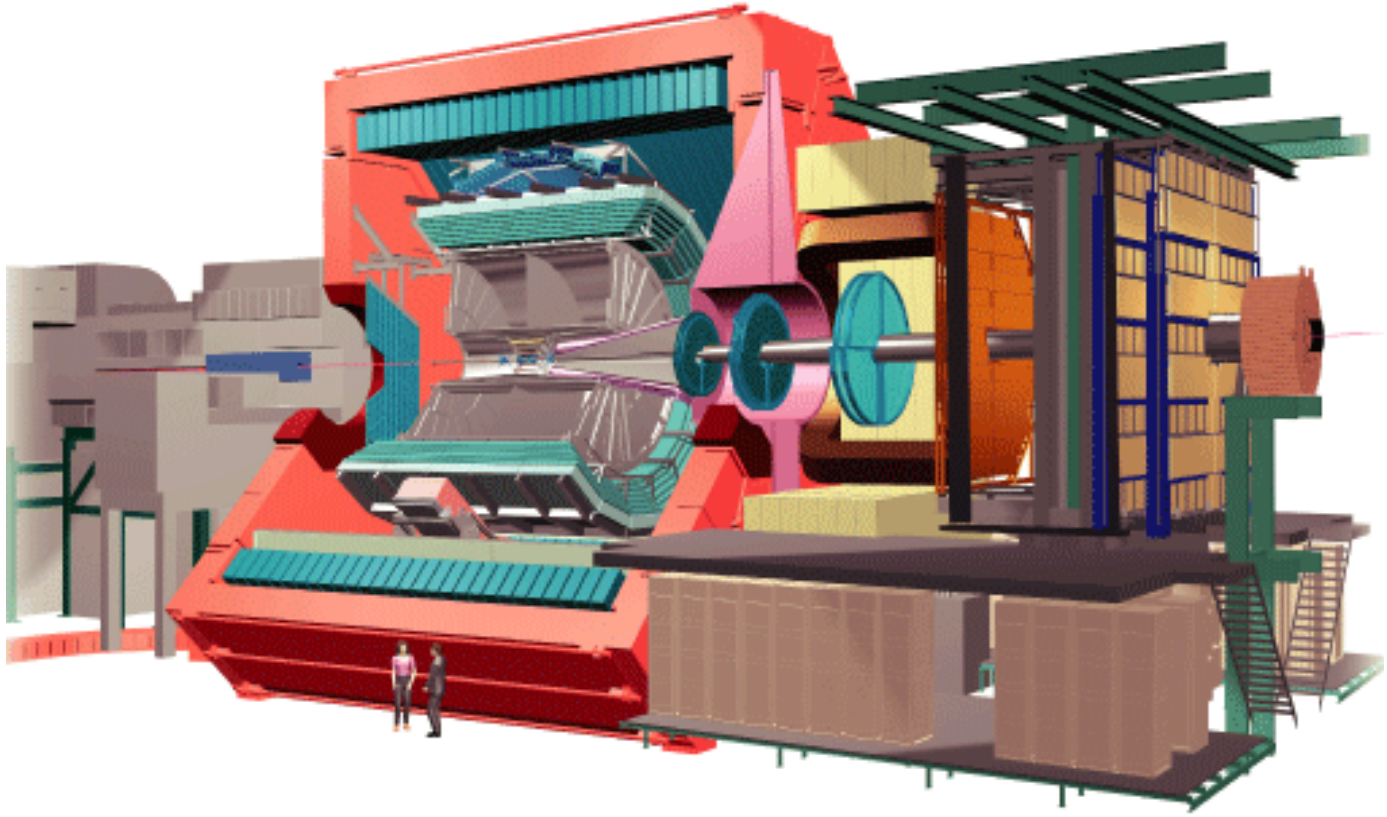
AliEn How-To

Alberto Colla

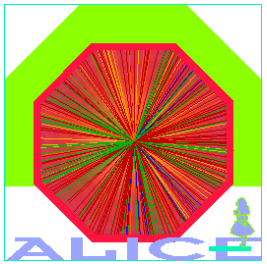
(Alice off-line Calibration and Alignment grup)



Alice off-line meeting

Cern, October 3, 2005

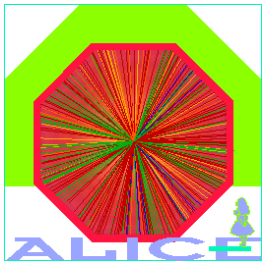
Disclaimer



This talk is intended to give the minimal set of instructions required to use the CDB access classes in a Grid environment

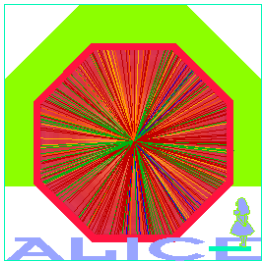
The use of AliCDBGrid and AliEn is reserved to the users who have already implementation of calibration procedures using AliCDBLocal

Summary



- Disclaimer
- gShell client installation
- Grid file management with gShell
- Building Root with AliEn support
- Grid access with Root

gShell client installation



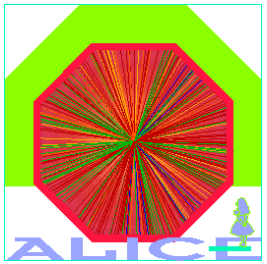
- User has to be registered CERN and AliEn organisation (AFS password will be required)
- Current gShell client version: **1.0.6-noglobus**, installed on /afs
- If /afs is mounted, do:

```
# Export
   GSHELL_ROOT=/afs/cern.ch/alice/library/alien1/alien-2/
                              gShell/1.0.6-noglobus/i386/
```

- If /afs is not mounted, copy gShell 1.0.6-noglobus folder locally, export **GSHELL_ROOT** with the local gShell path, and run the configuration script:

```
# . $GSHELL_ROOT/bin/alien_apiservice-bootstrap
```

Loading the gShell environment



- Run the **alien_VO** script:

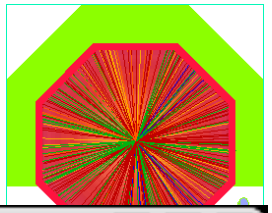
```
# . $GSHELL_ROOT/bin/alien_VO <username>
<afs password required>
```

- alien_VO does (among other things):
  - ➔ `source $GSHELL_ROOT/etc/env_gbbox.sh` (required for Grid access with Root)
  - ➔ `export LD_LIBRARY_PATH=$GSHELL_ROOT/lib` (where gliteUI library is)
  - ➔ connect to aliendb4.cern.ch CE
  - ➔ `alias grid='. $GSHELL_ROOT/bin/alien_Switch'`

    `grid` switches the shell from **local** to **grid mode** (and vice versa)!

- When in grid-mode, **unix** executables **ls, cp, mkdir** etc… are set as **alias** of the corresponding **AliEn** commands: **alien_ls, alien_cp, alien_mkdir** …
- AliEn executables are in $GSHELL_ROOT/bin

Loading the gShell environment (2)

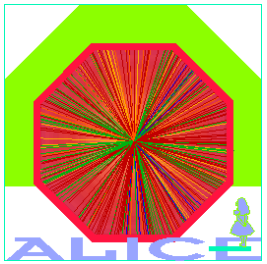


colla@PCALBE1:~

File Edit View Terminal Tabs Help

```
colla@PCALBE1:[~]
# source $GSHELL_ROOT/bin/alien_V0
Password:
[PCALBE1] ~ >
[PCALBE1] ~ >pwd
/home/colla
[PCALBE1] ~ >
[PCALBE1] ~ >grid
 [ gShell 1.0.5 (C) ARDA/Alice: Andreas.Joachim.Peters@cern.ch/Derek.Feichtinger@cern.ch]
[alien://colla@aliendb4.cern.ch:9000/ /alice/cern.ch/user/c/colla/ >
[alien://colla@aliendb4.cern.ch:9000/ /alice/cern.ch/user/c/colla/ >pwd
/alice/cern.ch/user/c/colla/
[alien://colla@aliendb4.cern.ch:9000/ /alice/cern.ch/user/c/colla/ >ls
DB
DBGrid
dummyHisto.root
gcc323.bash
prova
[alien://colla@aliendb4.cern.ch:9000/ /alice/cern.ch/user/c/colla/ >grid
[PCALBE1] ~ >
[PCALBE1] ~ >
[PCALBE1] ~ >ls
alice  cern     Desktop  gcc-3.2.3         mail    root   wireless.log
alien  cvs.csh  diamine  higz_windows.dat  MyData  Stuff
[PCALBE1] ~ >
[PCALBE1] ~ >
```

File management on Grid



- In Grid mode:

➔ **`ls file:<TAB>`** or **`ls alien:<TAB>`** will redirect to local or AliEn paths respectively (by default, **`ls`** lists AliEn folder content).

➔ **`whereis <file>`** prints out the file's GUID

➔ most of the AliEn commands have the same behaviour as the corresponding Unix ones (**`cd`**, **`pwd`**, **`mkdir`**, etc…); use **`rmdir`** to remove directories (also if they aren't empty)

➔ edit files with **`edit <file>`** (set your favourite editor in $GSHELL_ROOT/bin/alien_edit!)

- Copying files from local to grid (and vice versa):

➔ **`cp file:<localPath/localfileName> gridFileName@SE`**

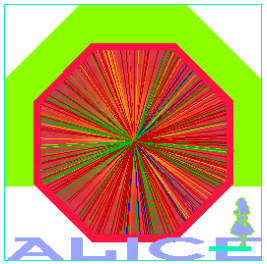
```
# cp file:/home/colla/file.root file.root@ALICE::CERN::se01
```

➔ For the moment use storage element ALICE::CERN::se01

➔ S.E. specification can be skipped if alien_CLOSE_SE variable is defined:

```
# export alien_CLOSE_SE="ALICE::CERN::se01"
```

Building Root with AliEn support



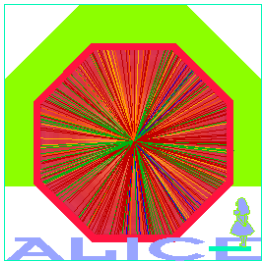
- Check out Root v5-04-00
- Configure Root enabling AliEn:

```
# cd $ROOTSYS
# ./configure --enable-alien
          --with-alien-incdir=$GSHELL_ROOT/include
          --with-alien-libdir=$GSHELL_ROOT/lib/
# make
```

- Compile with gcc 3.2.3!
- To check if AliEn support is enabled in Root:

```
# root-config --has-alien
yes
```

Connection to the Grid with Root



- Run **`$GSHELL_ROOT/bin/alien_VO`** before entering Root (it sets the environment variables needed by Root)
- In Root: login to **`alien://aliendb4.cern.ch:9000`** with **`TGrid::Connect()`**:

```
root[] TGrid::Connect("alien://aliendb4.cern.ch:9000","colla");
<password>
```

Url ("gridType://host:port") User name

- When **`TGrid::Connect()`** is called:
  - ➔ The library **`$ROOTSYS/lib/libRAliEn.so`** is loaded via **`TPluginManager`** class
  - ➔ The constructor of **`TAlien`** class is invoked
  - ➔ The class **`$GSHELL_ROOT/include/gliteUI.h`** is instantiated
  - ➔ **`GLiteUI::Connect(Host, Port, User)`** is called
- If connection works a **`TGrid* gGrid`** global pointer is returned!

TGrid methods

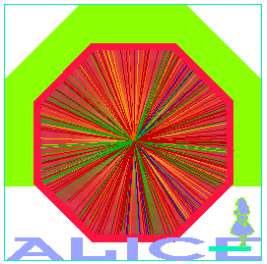
- Grid file catalogue is handled with TGrid methods called by **gGrid**:

```
char *currentFolder = gGrid->Pwd();
TGridResult *res = gGrid->Ls("folderName");
for(int i=0; i < res->GetEntries(); i++){
    char *dirEntry = res->GetFileName(i);
    cout << "file = " << dirEntry << endl;
}
gGrid->Cd("folderName"); //Bool_t
gGrid->Mkdir("folderName"); //Bool_t
gGrid->Rmdir("folderName"); //Bool_t
gGrid->Rm("fileName"); //Bool_t
...
```

Returns TList of the directory entries

Prints directory entries

Creating/accessing Root files in AliEn




```
TFile *f = TFile::Open("/alien/path/fileName?SE","option");
```

- example:

```
TFile *f =TFile::Open
  ("/alien/alice/cern.ch/user/c/colla/file.root?ALICE::CERN::se01",
  "CREATE");
```

- Note:
  - "/alien/" ≡ "alien://"
  - `TFile *f = TFile::Open("/alien/file.root") ≡`
    `TAlienFile *f = new TAlienFile("file.root")`
  - s.e. field can be avoided if the variable alien_CLOSE_SE is defined
  - currently it is not possible to list the content of a file with `f->ls()`!
  - It is however possible to get an object storeg in the file with `f->Get("objName")`