Notes on the Gantry, OGP & Database Software

Russell Taylor
May 2004

1 Gantry Stuff

For background information on gantry topics, including the software and calibration, see the documents on the UCSB gantry web page and the CMS Note from CERN:
`http://cmsdoc.cern.ch/cgi-doc/Grep_Search_F/documents/allxx.html?STRING=NOTE2002005`

1.1 Adding New Assembly Plates

To add a new assembly plate to the program, do the following:

- Measure the fiducial and pin positions on the OGP (3 times, and take the average) and transpose these coordinates to give everything relative to the right-hand fiducial on the plate, and in the gantry coordinate system.
- Create a new file `plateXX_params.txt` (where `XX` is the number of the new plate) in `C:\U600\workspace\` by copying and renaming one of the existing parameter files.
- Fill in all the parameters, making measurements using the gantry where necessary. Initially, set all the calibration factors to zero.
- In `VARIABLE_DEFINITION.PGM`, find the line "`#define NUMASSEMBLYPLAT`" and increase the adjacent number to correspond to the new total number of plates.
- Apart from writing an OGP program (and the associated Excel macro) for the new plate, nothing more needs to be done with regards to the OGP data files. New files `PlateXX-PosY.txt` will automatically be created in the `data` directory the first time that real modules are measured on the new plate.

1.2 Adding a New Module Type

Coming....honest.

1.3 When Sensor Fiducial Positions Change

There are a few modifications that need to be made if the positions/separations of the fiducials on the sensors change. First of all, the "`nom_coord_`" files in `C:\U600\DBPROGRAM\local\` need to be edited. The changes are easy to work out for $r - \phi$ modules, rather more challenging for stereo modules. There are then two varibles in `MODULETYPE_PARAM.PGM` that need to be corrected. The first of these is "$NominalSiliMarkerDistance", which is the distance in Gantry Y/Frame X between the fiducials. In the case of TEC sensors this is four variables, since they are not rectangular. The other is "$XDistMarkerSiliCenter", which corresponds to half the fiducial separation in Gantry X/Frame Y. There are two elements here: [0] relates to the far sensor, [1] to the near sensor. In the case of TOB modules, the only difference is a minus sign.

1.4 Gantry Calibration

I would recommend avoiding this operation if at all possible! Nevertheless, here are some instructions on the process. First of all, the calibration plate needs to be measured on the gantry, with the calibration file switched off. To deactivate the calibration go into the Global Parameter page on the MMI and set the "Enable2DCalibration" parameter to 0 (1=enabled).

The program for measuring the plate is found in `C:\U600\qualif\CALIB.PGM`. Start by running this program and aborting after its first movement after warming up (if selected). The gantry should be at approximately X=30, Y=159. Place the calibration plate on the gantry and line up the back right corner fiducial with the centre of the camera, using the crosshairs on the monitor. Then attempt to line up the back row of fiducials with the gantry X-axis. Do this by moving the gantry in Y only (using the MMI manual mode) and lining up the opposite corner. This placement of the plate was chosen because the calibration grid then covers the entire active area used in the module assembly program.

With the plate in place, the program can be run again from the start. It will step over the entire grid measuring each point. All that the operator needs to do is adjust the focus every third point and at the start of each row. A text file will be produced containing the positions of the fiducials: `C:\U600\qualif\gantry-meas.txt`. The program should be run three times, taking care not to overwrite the results file each time.

The next step is for the plate to be measured on the OGP, and programs have been written for this purpose. Because the maximum OGP travel is insufficient to measure the entire plate in one go, the last two rows (which are the first two rows measured by the gantry) are done separately. The programs are called `manualmaddness[1,2].rtn`. There were however, some minor bugs in these programs, and I don't know if they've been fixed or not. One point is measured twice (80,0), so the duplicate entry should me manually deleted from the output file. Three points were missed out. One of these is (480,0) – the last point on the first row. This point is important because it's used for the alignment of the axis, so it should be measured separately on the OGP and inserted in the file. The other two points are (300,20) and (200,420). I got values for these by interpolating between the adjacent points

and inserting the values. A better alternative would be for the OGP programs to be fixed prior to measuring the plate!

The calibration file is created from these files using a FORTRAN program. This can be found in `/homes/rtaylor/gantry/calib/`. The code is called `calibrate.f` and the executable is `calibrate.exe`. The executable should work fine withough recompilation, so long as the input files are formatted and named correctly. This program takes as input the three files from the gantry measurements, which should be named `gantry-meas[1-3].txt`, and the three from the OGP, including those for the first two rows, which should be named `manual-stat[1-3].txt` and `manual2-stat[1-3].txt` respectively. The program handles the averaging of the measurements and the calculation of the corrections and then writes out a correctly formatted calibration file named `2d-calib.cal`.

The calibration file should be renamed and copied onto the gantry PC. To select the file go to the Setup page of the MMI and click on the "Browse" button to select the file. Then enable the 2D calibration once again. The final step is to re-measure the plate on the gantry. The resulting output file can be analysed using, for example, Excel.

The program for calibrating the θ-axis can be found in `C:\U600\qualif\QUALIF4_2.PGM`. This program will pick up a sensor placed in position 0 on the supply plate, rotate it a little, place it down and measure its position. It will repeat this (50 times) for angles between -500 mrad and $+500$ mrad. The supply plate should be removed whilst the sensor is being picked up the first time. For each step, the intended angle and the (actual - intended) angle are written to a file named `qual4_2.txt`.

1.5 Pattern Recognition

Instructions for using the pattern recognition software can be found in the following document: `http://hep.ucsb.edu/cms/gantrydocs/readme_patrec.doc`. Ignore all references to `socket.exe` and TCP/IP, since we use a file-polling method (denoted 'Filesystem' on the PR interface) which is not referred to in this document. A copy of the LabView application can be found in `/homes/rtaylor/gantry/software/PR/`. I never had to make any modifications to the original version received from CERN.

Should you ever wish to make use of the three-point circle routine for locating pins, this can be done, in the context of the module assembly program, by uncommenting the line "`CALL FIND_PIN`" in the routine `FIND_ASSMARKER.PGM`. The gantry will then go measure each pin in turn after the assembly plate fiducials have been measured.

2 Database Stuff

2.1 Gantry automatic file movements

The movement of files to and from the gantry PC is handled using the Windows 'Scheduled Task' utility (`Start -> Settings -> Control Panel -> Scheduled Tasks`). Setting up scheduled tasks is the only time you will need to know the password for the gantry PC (which is "elephant" for the username "babar"). Two scheduled tasks have been set up. The first of these backs up the module assembly software once a week on Saturday night. The files are copied to `/cms2/cms/gantry/u600/` on `sbottom` (this is the disk mounted on the gantry PC as drive H:). This directory also contains a static copy of the MMI software itself, along with configuration files. Copies of the module assembly software made at various times between the initial CERN release and May 2004 can be found in `/homes/rtaylor/gantry/software/`.

The second scheduled task deals with the movement of the module database files. Every night at 2am the MSDOS script `C:\U600\DBPROGRAMS\backup.bat` is executed. This looks like this:

```
copy C:\U600\DBPROGRAMS\LocalResults\302*.doc H:\cms\gantry\legible_results\
move C:\U600\DBPROGRAMS\LocalResults\302*.doc C:\U600\DBPROGRAMS\Real\
copy C:\U600\DBPROGRAMS\results\302*_gantry*.xml H:\cms\gantry\results\
move C:\U600\DBPROGRAMS\results\302*_gantry*.xml C:\U600\DBPROGRAMS\modules-xml\
copy H:\cms\gantry\indb\ass3020*.xml.indb C:\U600\DBPROGRAMS\assemblies\
move H:\cms\gantry\indb\ass3020*.xml.indb H:\cms\gantry\assemblies\
```

and it does the following:

- Copies the legible results files to `sbottom` and moves the originals to the directory 'Real'.
- Copies the XML files (before or after-cure) to `sbottom` and moves the originals to the 'modules-xml' directory.
- Copies the assembly file from `sbottom` to the gantry PC and moves the original to a different directory.

The use of "302*" in the backup file is intended to ensure that only files concerning real modules are moved/copied. Dummy module files should be moved or deleted by hand (and should never be given IDs starting with "302"). Incidentally, the files mentioned in the final point above are created by the Big Browser when a module is 'virtually assembled', and contain the ID numbers of the sensors and hybrids. The module assembly software (specifically `DB_MAKE_VIRTUAL_ASSEMBLING.PGM`) looks for this file to determine the sensor and hybrid numbers automatically (if the module number entered starts with "302" and contains 14 digits).

Movement of the files can be forced simply by running (double-clicking on) `backup.bat`. This can be useful if, for example, you want to upload files to the database immediately rather than wait for the overnight upload. The moved files can be uploaded in the old way (by using the 'Calibration' panel in the Big Browser) from either the gantry or data entry PC.

2.2 OGP database software

The software for converting the output from the OGP measurements to XML is located in `/cms2/cms/ogpdata/survey-text/`, which is the directory on `sbottom` in which the text output of the Excel macro is placed. It is mounted on drive E: on the OGP computer. The software consists of a number of files:

- `ogptoxml.exe` - the executable.
- `ogptoxml.run` - the driver script. This script looks for files containing the string '302000' in the filename. If any are found it creates a temporary file containing the filenames and their timestamp (along with some other irrelevant stuff) and then runs the executable. The generated XML files are placed in the `results` directory and the text files are moved to the `processed` directory.
- `last-calibration` - a text file containing the date of the most recent calibration of the OGP. Should be updated as appropriate.
- `input_parameters.param` - a text file containing the cuts on the placement precision and the database table ID numbers. It is identical to the file of the same name used by the gantry, so if that is changed this one should be too.

The FORTRAN code used to generate the executable can be found in `/homes/rtaylor/ogp/`. The program reads in the filenames one at a time from the temporary file created by the driver script and calls a number of subroutines:

- `FORMATDATE` - renders the date in the correct format for the XML file.
- `READOGP` - reads in the OGP data from file.
- `SETFLAGS` - sets the flags for the database according to the placement accuracy of the components as measured by the OGP.
- `WRITERESULTS` - appends the OGP data to a file of the form `PlateX-PosY.txt` in the `data` subdirectory.
- `GLUEMONITORING` - looks in the directory `/cms2/cms/gantry/legible_results/` to see whether an after-cure survey was performed on the gantry. If not, it calculates the curing time (looking in the `.doc` from the gantry to find the time of assembly) and

creates the GLUEMONITORING table in the XML file. Note that in this case it is assumed that no problems were encountered during curing (i.e. gluemonitoring_val is always :0:). If there were any problems, then an after-cure survey should be performed on the gantry.

- `COMPARETOGANTRY` - if a gantry after-cure survey is found, then this routine reads in the `_after.doc` created by the gantry software and calculates the differences of each fiducial with respect to the OGP measurement. The results are appended to a file of the form `Gantry_vs_OGP-PosX.txt` in the `data` subdirectory.
- `WRITEXML` - creates the XML file.

There should be no reason why this code ever needs to be modified. However, it is **VERY IMPORTANT** that no changes are made to the format or filenames of any of the input files, and that the directory structure stays the same.

2.3 Database (XML) file upload

The automatic upload of the XML files from both the gantry and the OGP, as well as the creation of the XML files from the OGP data, is performed each night under the cms account on `sbottom` using the UNIX `cron` command (basically equivalent to a Windows Scheduled Task). The file that controls this can be viewed by logging onto `sbottom` as cms and entering the command: `crontab -l`. It looks like this:

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (cronfile installed on Mon Feb  2 10:34:07 2004)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/cms2/cms/ogpdata/survey-text
:/cms2/cms/gantry/results
HOME=/cms2/cms/ogpdata/survey-text

 0 3 * * * Gantry-upload
 0 5 * * * ogptoxml.run
30 5 * * * OGP-upload
```

Without going into the details, what this means is that the program `Gantry-upload` is executed at 3am each day, `ogptoxml.run` at 5am and `OGP-upload` at 5:30am. This time spacing ensures that, in the case that after-cure measurements are made both on the gantry and the OGP, the OGP data is the last to be uploaded and becomes the 'reference' data. The output of each program is emailed to `cms@sbottom.physics.ucsb.edu`, so the mails can be read by logging onto `sbottom` as cms and running `pine`.

The uploading of files uses the CMS Tracker Database software (and in particular the 'Update' function) which is installed on `sbottom` in `/homes/cms/TrackerDB/`. The files from the

gantry are uploaded using the script `Gantry-upload`, which is located in `/cms2/cms/gantry/results/` (the same directory to which the XML files from the gantry are moved) and looks like this:

```
#!/bin/bash
# Set the installation directory
INST_DIR=/homes/cms/TrackerDB
export INST_DIR

# Set the working directory
WORK_DIR=/cms2/cms/gantry
export WORK_DIR

CLASSPATH=$INST_DIR/lib/xerces.jar:$INST_DIR/lib/classes12.zip:$INST_DIR/lib/jas
.jar:$INST_DIR/lib/hep.jar:$INST_DIR/lib/jbcl.jar:$INST_DIR/lib/mail.jar:$INST_D
IR/lib/Plot2D1.2.jar:$INST_DIR/lib/activation.jar:$INST_DIR/lib/cmstrk.jar
export CLASSPATH

LD_LIBRARY_PATH=$INST_DIR/jdk1.3/jre/lib/ext/

for x in `ls /cms2/cms/gantry/results/ | grep 302000`;
do

echo "  _____                _             ____  ____   "
echo " |_   _| __ __ _  ___| | _____ _ __|  _ \| __ )  "
echo "   | || '__/ _' |/ __| |/ / _ \ '__| | | |  _ \  "
echo "   | || | | (_| | (__|   <  __/ |  | |_| | |_) | "
echo "   |_||_|  \__,_|\___|_|\_\___|_|  |____/|____/  "
echo ""
echo "Starting ..."
date

#arg list
#  arg0 = center
#  arg1 = tool id
#  arg2 = working directory (place where output and log files will be)
#  arg3 = installation directory (place where you installed BB)
#  arg4 = test or pro depending on which database you want to access
#  arg5 = password for this database (obtain passwd : contardo@ipnl.in2p3.fr)
# last arg which will be given when running script is the file to be processed.

  $INST_DIR/jdk1.3/jre/bin/java Update SANTA-BARBARA 56 $WORK_DIR $INST_DIR pro
'UdBP!$p' /cms2/cms/gantry/results/$x

done
```

```
exit
```

This script is largely based on one provided by the database group in Lyon. It looks for files with titles containing the string '302000' and, if any are found, it loops over the files uploading them in turn. The uploaded files are moved into the `indb` directory. A file is also created in the `error` directory. If the upload was successful this file is empty (i.e. listing the directory contents using, e.g., `ls -l` would show that all the files have size 0), otherwise it contains the error message and the XML file is moved into this directory as well. The error message is very generic – a diagnosis of the reason for the failure is more easily made through looking at the emailed output. In the case that no results are found, the script does nothing and no email is sent. I've been deleting the files in the `error` directory from time to time.

The script for the OGP file upload, `OGP-upload`, is virtually identical. It can be found in `/cms2/cms/ogpdata/survey-text/`, which contains the same subdirectories discussed in the previous paragraph.

Contact Details

E-mail: rtaylor@hep.ucsb.edu – will continue to work.
Cellphone: 011-44-7905-139315 – No calls after 2pm!!