

Parameter containers and their management

Gabriel García Jiménez, Jose Luis Rodríguez Sánchez

Héctor Alvarez Pol

Universidade de Santiago de Compostela

- Parameters in R3BRoot.
- Parameter repositories per experiment.
- Status S522 / S509.

R3BRoot parameters intro

Parameters are all-round information containers in R3BRoot, storing data. We should use parameters:

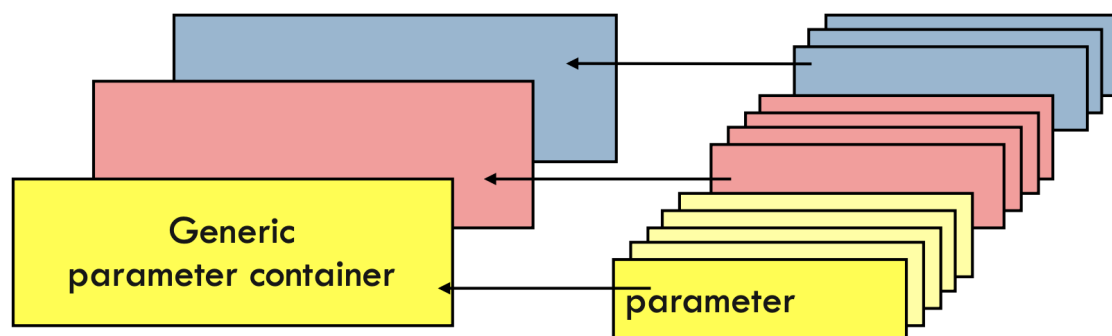
- In the mapping and calibration processes.
- In the geometrical definition and experiment setup alignment.
- Introducing and defining magnetic fields.
- Selectors and input values for reconstruction...
- ... **Anywhere** when some numbers or set of numbers that are not fixed forever are needed for the data processing.

The **Parameter classes manage all the information needed to process our data. It prevents data hardcoding!**

R3BRoot parameters intro

You need to know and distinguish among:

- **Parameter classes** are the C++ implementation of the parameters, with the data itself (integers, arrays, ...) and the interface (getParams(), putParams(), ...). → **in R3BROOT**
- **Parameter containers** are ASCII (.par) or ROOT (.root) files containing the data defined in the the parameter classes.
- **Parameter actual values** are included in the containers.
Different values for the same parameter could be required during the experiment, requiring versioning of the parameters.



in the parameter repositories

R3BRoot parameters: naming convention

- Parameter classes moving data from a Level 1 to another more advanced Level 2:

R3B[Detector][Level2]Par

Some tasks calculate parameters (mainly for calibrations):

- The naming convention for these parameter finder classes is:

R3B[Detector][Level1]2[Level2]Par

- Other parameters classes simply reflects the contents, as in

R3BGeoPassivePar,

R3BCalifaMappingPar,

R3BNeulandMultiplicityBayesPar, ...



R3BRoot parameters: ASCII values

The text/numerical values of the actual parameters are stored in the parameter containers, ASCII or root files:

```
#####
#####
# Class:      Mwpc0GeoPar
# Context: GeometryParameterContext
#####
[Mwpc0GeoPar]
//-----
PositionX: Double_t 0.
PositionY: Double_t 0.
PositionZ: Double_t -262.
RotationX: Double_t 0.
RotationY: Double_t 0.
RotationZ: Double_t 0.
DimensionX: Double_t 20.
DimensionY: Double_t 20.
DimensionZ: Double_t 4.5
SigmaX: Double_t 0.1
SigmaY: Double_t 1.
Z: Double_t 14.
A: Double_t 28.0855
Density: Double_t 1.0
IonisationEnergy: Double_t 57.4e-6
#####
```

Example: R3BParams_S455p2p/geometry/Setup.par

R3BRoot parameters: technical issues

The class **FairRuntimeDb** manages the parameter containers.

- Parameters can go through **several different versions**: changes in the detectors definition or operation, modification on the calibration parameters or settings, ...
- This makes necessary to have a **parameter repository with a well-defined versioning system**.
- The **runtime database** (**FairRuntimeDb** class) is used to store and control the parameters repository, loading the proper parameter set when needed, using a **runId** scheme: RuntimeDb provides always the right parameters! The update of parameters is triggered by the RunManager when unique **runId** changes.
- Different inputs are supported in different formats simultaneously.



R3BRoot parameters: repositories

Parameters are stored in experiment dependent repositories:

- Github repositories contain all the parameters, root geometries, macros and input files needed in simulation and data analysis.
- These git repositories are **private**. Anybody within the collaboration can obtain access to the repositories.
- New repositories could be created for any experiment from a common template (R3BParams_template repository).



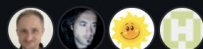
R3BRoot parameters: repositories

R3BParams_S455p2p Private Parameter repository for the S455 p2p fission experiment C ☆ 1 🔗 4 🔄 0 🔗 0 Updated 19 hours ago
R3BParams_S509 Private Parameter repository for the S509 experiment C ☆ 0 🔗 5 🔄 0 🔗 0 Updated yesterday
R3BParams_S515 Private Parameter repository for the S515 experiment C ☆ 2 🔗 9 🔄 0 🔗 0 Updated yesterday
R3BParams_S522 Private Parameter repository for the S522 experiment C ☆ 1 🔗 12 🔄 1 🔗 0 Updated 15 days ago
R3BParams_template Private template C ☆ 0 🔗 3 🔄 0 🔗 0 Updated on 20 Dec 2021

R3BRoot parameters: repositories

Join to the analysis team for your experiment:

- Please, mail to Dmytro Kresan to join GitHub R3B teams.
- Team members have “triage” role (read, clone, issues and PR).
- Each repository should be supervised by a release manager.

8 teams in the R3BRootGroup organization		Visibility ▾	Members ▾
NeuLAND		2 members	0 teams
R3B Developers		12 members	0 teams
R3BParams_S455p2p Team for the analysis of the S455 experiment		11 members	0 teams
R3BParams_S515 Team for the analysis of the S515 experiment		12 members	0 teams
R3BParams_S522 Experiment S522		24 members	0 teams
R3BRoot Team Secret Core developers		3 members	0 teams
R3BRoot Users		38 members	0 teams
Sofia Group SOFIA Experiment		4 members	0 teams



R3BRoot parameters: repositories

	ryotani and jose-luis-rs CalibPar for incoming ID updated for 21N beam tuning	71b2e3a yesterday	16 commits
	califa	Added mapping pars for Califa barrel full gamma and small macro for...	3 days ago
	fibers	Copied all the calibration files from s522	3 days ago
	foot	Copied all the calibration files from s522	3 days ago
	geometry	Copied all the calibration files from s522	3 days ago
	glad	Copied all the calibration files from s522	3 days ago
	los	Copied all the calibration files from s522	3 days ago
	macros	Updated main_online.C	2 days ago
	musli	Copied all the calibration files from s522	3 days ago
	mwpc	Copied all the calibration files from s522	3 days ago
	neuland	Copied all the calibration files from s522	3 days ago
	parameters	CalibPar for incoming ID updated for 21N beam tuning	yesterday
	rolu	Copied all the calibration files from s522	3 days ago
	rpc	Copied all the calibration files from s522	3 days ago
	sci2	Copied all the calibration files from s522	3 days ago
	target	Copied all the calibration files from s522	3 days ago
	tofd	Added new parameters for tofd	3 days ago
	.gitignore	Initial commit	3 days ago

Parameter repository for the S509 experiment

- Readme
- 0 stars
- 3 watching
- 5 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Contributors 3

- jose-luis-rs Jose Luis Rodriguez
- ryotani Ryo Taniuchi
- gabrigarjim

Languages

All parameters should be stored in the repository folders:

- There are a folder per detector (including incoming ID, passive, geometries, ...). Also a directory for macros (calibration, online, simulation, specific geometry, support, ...).
- Each folder contains a **README.md** file with general information about the detector and the parameters and ...
- ...a **History** file is included, complementing the git tracking system.
- Additionally a **METAFILE** keeps track of the parameters evolution along the experiment timeline...

R3BRoot parameters: repositories

Description of the detector folder

General information

A reverse chronological ordered list of modifications on the parameters for easy reference (complementing the github history)

```

1  -----
2
3  =====
4  Tofd - Parameter containers
5  =====
6
7          Category History file
8  -----
9  This file should be used by R3BRoot developers and category coordinators
10 to briefly summarize all major modifications introduced in the parameters
11 and keep track of all category-tags.
12
13 -----
14 * Reverse chronological order (last date on top), please *
15 -----
16 - May 10, 2022 Andrea Lagni
17 Added root file tofdcalpar_v2.root with the cal level parameters obtained
18 from the file runCR0605_Night.root CR run of 2 days.
19
20 - April 24, 2022 J.L. Rodriguez
21 Added root file tofdhitpar_v1.root with the hit level parameters obtained
22 as well from the run 70: main0070*.lmd
23
24 - April 15, 2022 J.L. Rodriguez
25 Added tofdmappingpar_v1.par with the trigger time mapping and the root file
26 tofdcalpar_v1.root with the tcal parameters obtained from the run 70:
27 main0070*.lmd
28
29 - July 13, 2021 J.L. Rodriguez
30 Created default parameter folder.

```

R3BRoot parameters: time evolution

The METAFILE allows the change of runId's and consequently tells the unpackers which parameter version has to be used.

METAFILE has to be changed after every new split in the runId.

Example:

#RunID	#ExpRun	#TimeStamp	#califaMapVersion	#califaCalVersion	#califaGeoVersion ...
1	1	166e4cf5d5305600	1	1	1

Let's say that for run 266 another set of calibration parameters is needed. The new RunID would be:
 $265 \times 2^{20} + 1 = 0x10900001$ (20 bites available per exp run: $2^{20} \sim 10^6$ possible params modification)

#RunID	#ExpRun	#TimeStamp	#califaMapVersion	#califaCalVersion	#califaGeoVersion ...
1	1	166e4cf5d5305600	1	1	1
0x10900001	266	1934d7e690445600	1	2	1



New parameter version will be loaded for the given timestamp!

TimeStamp marks the point where the RunID changes!

R3BRoot parameters: repositories

There are also relevant **macros** in every repository:

- **Unpacking macros** read metafile information and apply the correct set of parameters.
- **Geometry macros** generate geometry rootfiles for the detectors in the experiment.
- The **parameter merger** gathers all versions for all parameters and merge them into a single rootfile.
- **Online/offline macros** for the analysis of the data.
- **Helper macros** for many tasks: displaying parameters (for instance showing all params for a given crystal in CALIFA, ...).

R3BRoot parameters: S522/S509

Temporary parameters have been stored locally for S522/S509 at:

- `/u/land/r3broot/202205_s522/R3BParams_S522`
- `/u/land/r3broot/202205_s509/R3BParams_S509`

And regularly pushed to the GitHub repositories. The same scheme has worked out for modifications in R3BRoot.

- During the experiment, rules has been relaxed (regarding naming scheme, directories, ...). Please, follow them strictly from now on.

Thank you to all the participants for the great work supplying preliminary parameters for the online and near-line analysis!!

R3BRoot parameters: few remarks

A few important points for the parameters management:

- **Training and documentation:** R3BRoot school and training courses are strongly requested by the community and needed asap.
- **Parameters validation:** requires a procedure validated by the experiment community (presentation of results, performance comparative, ...).

And remember:

- Parameters are fruits of a collaborative effort. Share and document!
- Parameters and R3BRoot code versions need to be properly tracked (using unique git hash) to be accepted as a work of the collaboration (publications, conferences, ...) by the scientific board.