

Table of Contents

Magnetic Field Interface.....	1
Contacts.....	1
Default configuration.....	1
Access to the map.....	1
Internals.....	1
How the field map works.....	1
Map sets and configuration in db.....	2
Summary of MF geometries currently loaded in the DB.....	3
Data tables.....	3
Development workflow.....	4
Preparing a new geometry.....	4
Table conversion.....	4
Publishing new tables for release.....	5
Conversion of the geometry to db.....	5
Preparing a db configuration.....	5
Debugging tools.....	6
Regression Testing.....	6
Visualization of the field map in CMSSW.....	6
Release history.....	7
Configuration in older releases.....	7
History of map versions.....	7
Version 851_030919 ("old 4T map"; obsolete, no longer supported).....	7
Version 11031_071212 (Pre-CRAFT-2009 default; 3.8T version deprecated since 3_1_X; still used for some nominal operating points other than 3.8T).....	7
Configuration of superseded table sets in DB.....	10
Fancy use cases.....	11
How to load two field maps in the same job.....	11
How to apply per-volume rescaling factors.....	11
Review status.....	12

Magnetic Field Interface

Contacts

- **TOSCA computation:** Vyacheslav Klyukhin
- **CMSSW interface:** Nicola Amapane

Default configuration

The python configuration to get the default field map is:

```
process.load("Configuration.StandardSequences.MagneticField_cff")
```

This sets the the map so that the Run I/Run II map is chosen automatically based on the GT, and the nominal working point is set automatically depending on the magnet current stored in the conditions DB (RunInfo). Nominal working points are:

- 0 T
- 2 T (9500 A)
- 3 T (14340 A)
- 3.5 T (16730 A)
- 3.8 T (18164 A)
- 4 T (19140 A)

For the MC, the magnet current in conditions is fixed, and corresponds to the 3.8 T working point; so the above always results in the 3.8 T map for MC GTs.

In order to configure CMSSW for **zero field (magnet off) in MC** one needs to replace the config with:

```
process.load("Configuration.StandardSequences.MagneticField_0T_cff")
```

Other nominal working points can be forced by replacing the "valueOverride" parameters that are set in the same file.

Access to the map

The magnetic field map is available in CMSSW from the EventSetup as an object of type CMS.MagneticField. An example of getting the magnetic field in CMSSW is available in

```
CMS.MagneticField/Engine/test/queryField.cc  
CMS.MagneticField/Engine/test/queryField.py
```

This is everything needed for the general user.

Internals

The following sections include developer documentation and are not relevant for users.

How the field map works

Concrete implementations ("engines") of a `MagneticField` are available to deliver different maps in either the whole CMS or in specific parts of it. The default map is called `VolumeBasedMagneticField`, and is based

on a geometrical description of the solenoid and yoke volumes, with a separate "field provider" for each volume. Normally a field provider is based on interpolation over a grid of points tabulated with a finite-element computation.

Providers can however also be based on e.g. parametrizations.

A special region is the inner tracker region. Here, `VolumeBasedMagneticField` allows to specify an ad-hoc parametrization to replace the normal volume/field provider mechanism, for either speed or accuracy purposes. Technically this is implemented by overlaying a "slave field engine" with a defined spatial validity region.

A "**static**" map, representing the MF for a given era (Run I or Run II) and current, can be created explicitly with e.g. `MagneticField/Engine/python/volumeBasedMagneticField_160812_cfi.py` ([link](#)). This cfi defaults to 3.8T, RunII map. It sets:

- MF geometry reader
- the parametrized map to be used in the tracker region
- the actual `VolumeBased` map, whose configuration includes specification of the field data tables to be used (`version`) and how tables are associated to volumes (`gridFiles`).

Static maps should **not** be used in production.

The **default map is dynamic** and built with `Configuration.StandardSequences.MagneticField_cff`, which retrieves the appropriate geometry and configuration from DB for the given GT and IOV. The internals of this builder is documented in the following section.

Map sets and configuration in db

A map set consists of six maps for the five nominal operation currents, corresponding to 2, 3, 3.5, 3.8 and 4T, plus a zero-field map. The configuration for each map is stored in the condition DB in a `MagneticFieldConfig` object `i`, which includes:

- a MF geometry version (int);
- a data table version (string);
- a map of table names to be associated to volumes;
- optional scaling factors for specific tables;
- name and parameters of a slave parametrized field for the tracker region.

In each GT, configuration for all six maps of the set must be present. They are stored as separate `MagneticFieldConfig` objects with the pre-defined labels "0T", "2T", "3T", "3.5T", "3.8T", "4T". The appropriate one is retrieved by label based on the coil current stored in the `RunInfoRecord`,

The geometry version of the required configuration is used to retrieve the correct geometry from the db by label. Each GT thus contains all required geometries, attached to the record `MFGGeometryFileRcd`, with labels like "180812", "90322", etc.

To find what maps are included in a given GT, search for the GT in `conddb`, eg for data and Run1/Run2 MC:

- data: `100X_dataRun2_HLT_v3`
- MC, run1: `90X_mcRun1_design_v2`
- MC, run2: `100X_mcRun2_design_v2`

Search for "`MagFieldConfigRcd`" in the Tags List table. You should get six entries with the labels "0T", "2T", "3T", "3.5T", "3.8T", "4T" for the six maps of the set.

Note that for data, the map should have IOVs for Run1/Run2 as needed. For MC, there should be only 1 valid IOV, since there are different GTs for Run1 or Run2.

Available geometries can instead be found searching for "MFGeometryFileRcd".

The following table summarizes the currently used map sets.

Current (A)	B(T)	MagFieldConfigRcd Tag Run I (from:1)	MagFieldConfigRcd tag Run II (from:220641)	Note
0-4779	0	MFConfig_71212_0T		requires tables from grid_11031_071212_4t even if zero field
4779-11987	2	MFConfig_71212_2T		Run I version, but we don't really care about 2T
11987-15617	3	MFConfig_160812_Run2_3T		Run II version; used also for Run conditions
15617-17543	3.5	MFConfig_160812_Run2_3_5T		ditto
17543-18765	3.8	MC tag: MFConfig_160812_Run1_3_8T	MC tag: MFConfig_160812_3_8T	
		Data tag: MFConfig_RI_RII_160812_3_8T		
>18765	4	MFConfig_71212_4T		Not really needed, but we keep it nevertheless.

cf TagsForMagneticField160812

Summary of MF geometries currently loaded in the DB

XML files for the different geometry versions are located under MagneticField /GeomBuilder/data:

- version 90322: MagneticFieldVolumes_11031.xml. This geometry is suitable for rarely used maps (currently, only 2T and 4T)
- version 120812: MagneticFieldVolumes_1_v7_large.xml, MagneticFieldVolumes_2_v7_large.xml - no longer in use, should be removed from future GTs
- version 130503: MagneticFieldVolumes_1_v9_large.xml, MagneticFieldVolumes_2_v9_large.xml - no longer in use, should be removed from future GTs
- version 160812: MagneticFieldVolumes_160812_1.xml, MagneticFieldVolumes_160812_2.xml
Current up-to-date geometries.

Data tables

The actual field data is stored in tables that are published under:
<https://github.com/cms-data/MagneticField-Interpolation>

The grids in use in current releases are:

```
grid_111031_071212_2t
grid_111031_071212_4t
grid_160812_3t
grid_160812_3_5t
grid_160812_3_8t
grid_160812_3_8t_Run1
```

The following grids are no longer used and can be removed:

```

grid_11031_071212_3t
grid_11031_071212_3_5t
grid_11031_071212_3_8t
grid_120812_3_8t_v7_large
grid_120812_3_8t_v7_small
grid_130503_3_8t_v9_large
grid_130503_3_8t_v9_small
grid_11031_090322_3_8t

```

Development workflow

The steps described below must be followed, in their order, to prepare, validate and deploy a new set of maps, assuming the starting point is a new xml geometry and a corresponding set of TOSCA grid tables in txt format. For new tables for an already existing geometry, the first part can be skipped.

Preparing a new geometry

Add the new xml file(s) in `Magfield/GeomBuilder/data`. Test geometry building by preparing a config file like `MagneticField/Engine/python/volumeBasedMagneticField_130503_cfi.py`, specifying:

- the new xml files in `magfield.geomXMLFiles`
- an empty `VolumeBasedMagneticFieldESProducer.gridFiles` list

Set this cfi in `MagneticField/Engine/test/testMagneticField_cfg.py` instead of the default one, and set `process.VolumeBasedMagneticFieldESProducer.debugBuilder = True`. Run this and search for errors (`ERROR`). Some `WARNINGS` are normal if mismatches are small (normally <0.04 cm; can get up to 0.2 cm).

The `debugBuilder` flag activates the test `MagGeoBuilderFromDDD::testInside`, which tests that the center of a volume is inside it and outside all other volumes. There must be no error.

Also run with `process.testVolumeGeometry = cms.EDAnalyzer("testMagGeometryAnalyzer")`. This calls `MagGeometryExerciser::testFindVolume`, which checks random points: the corresponding volume should be found, and the points must be `inside()` one and only one volume. There must be no error. Note: internally this uses `MagGeometry::findVolume`, that has a retry mechanism with increased tolerance to avoid tiny gaps.

Table conversion

To convert txt tables (extracted from TOSCA), use

`MagneticField/Interpolation/test/BinaryTablesGeneration/convertTables.py`. Check for `ERROR` s in the output. Note that files are provided both with cartesian (xyz) and cylindrical coordinates (rpz). They are equivalent, but it is better to use rpz for all volume types except than for Box and Trapezoid volumes.

Once completion is completed with no error, copy the new set of maps into

`MagneticField/Interpolation/data` following the existing directory structure. Edit the cfi file (eg. `MagneticField/Engine/python/volumeBasedMagneticField_130503_cfi.py`) and set `VolumeBasedMagneticFieldESProducer.version` to the name of the newly created directory in `MagneticField/Interpolation/data`, and specify the matching between grid files and volumes, eg:

```

gridFiles = cms.VPSet (
    cms.PSet (
        volumes = cms.string('1001-1464,2001-2464'),
        sectors = cms.string('0') ,

```

```

master    = cms.int32(=),
path      = cms.string('s[s]/grid.[v].bin'),
),

```

With this syntax, all volumes numbered with range 1001-1464 and 2001-2464 for all sectors (0=all) are associated to the file `s[sector]/grid.[volumeNumber].bin`. The `master` specification can be used to use the table for a given sector for other sector as well by rotation, in order to save memory; 0 means that each sector will use its own specific table.

Run `testMagneticField_cfg.py` with `debugBuilder = True` as explained above. It should now try to build interpolators for all volumes, reading the data tables. In addition a test is done, that each grid point should be inside() the corresponding volume. Search for ERRORS. They can be either due to errors in the tables (e.g. irregular grid spacing), or in the geometry (volume boundaries not matching the actual grid). To debug possible problems with the grid tables,

`MagneticField/Interpolation/test/BinaryTablesGeneration/GridFileReader.cpp` can be used to dump table grid and values. To dump the memory representation of the yable, open `MagneticField/Engine/test/testMagneticField.cc` and uncomment/edit the block containing `vol->provider()` according to specific needs.

The final check is to run a regression of the field map vs the data tables, using

`Interpolation/test/BinaryTablesGeneration/validateField_TOSCATables.py`. Set the `cfi.py` for the new geometry; the list of tables is taken from a file `tableList.txt`, that is written during conversion by `convertTables.py`. Numerical discrepancies can be present in a handful of cases up to ~ 0.00065 T, especially in rotated tables. This could be due to the appearance of spurious ~ 0 step parameters due to rounding in rotations. This level of accuracy is acceptable. Further study/development is needed to understand how to avoid these roundings.

Publishing new tables for release

Once data tables are ready, they should be published submitting a PR to <https://github.com/cms-data/MagneticField-Interpolation>.

Conversion of the geometry to db

Edit and run `MagneticField/GeomBuilder/test/mfxmlwriter.py` and `mfwriter.py`. The first writes the geometry in a single, self-contained xml ; the second converts it into a `.db` file and creates the corresponding metadata faile for upload to the condition db.

To load the geometry in the database:

```
uploadConditions.py <mfGeometryXXXXX.db>
```

Preparing a db configuration

See above for an explanation of how map configuration is stored in the condition DB.

A map configuration can be written into a `.db` file with

`/CondFormats/MFObjects/test/writeMagFieldConfigDB.py`. The the corresponding metadata faile for upload to the condition db is also created.

In general, before uploading to the condition db this `.db` file needs to be combined with others to get a single file with all IOVs for Run I and II maps.

Debugging tools

Regression Testing

Regression testing is available to validate changes that are not expected to change the field numerically in any point. It is done comparing the field returned by the map with a reference file created for each map. To run the regression testing, use [[

<https://github.com/cms-sw/cmssw/blob/master/MagneticField/Engine/test/regression.py>]]

MagneticField/Engine/test/regression.py]]

options:

- `producerType`: options are 'static_DDD', 'static_DD4Hep', 'fromDB', 'fromDB_DD4Hep'
- `era`: either 'RunI' or 'RunII'
- `current`: Magnet current (nominal values: 18164=3.8T; 16730=3.5T; 14340=3T; 9500=2T).

Failures are reported in the standard output.

Regression files are maintained in [cms-data/MagneticField-Engine/](#).

In case outliers are found, they can be visualized with gnuplot in this way (old recipe-to be updated)

```
cmsRun regression.py > ! output.txt
grep Disc output.txt | cut -f 7 -d' ' | tr '(,)' ' ' > ! outliers
cat <reference_table.txt> | cut -d' ' -f 1,2,3 > ! allpoints
```

and in gnuplot:

```
set terminal png
set xlabel "x coordinate (cm)"
set ylabel "y coordinate (cm)"
set zlabel "z coordinate (cm)"
set output "viewXY.png"
set view 0,90
splot 'allpoints','outliers'
set output "view3D.png"
set view 30,60
splot 'allpoints','outliers'
set output "viewXZ.png"
set view 90,0
splot 'allpoints','outliers'
```

Visualization of the field map in CMSSW

The magnetic field and MF geometry can be visualized in Eve. The field visualization is implemented in `Fireworks/Geometry/plugins/DisplayGeom.cc`. An example `.py` configuration is shown at:

`Fireworks/Geometry/plugins/cmsRun_displayProdMFGeom_cfg.py`.

Some notes:

- In the Eve window, click "Camera Home" in the bottom left panel (to return to this panel later, double-click on Viewers and select Viewer 1).
- To select/deselect individual volumes, double-click on `cms:World` and again on each new level of the hierarchy.

- To choose a pre-defined perspective or orthographic view, hover with the mouse on the bar below the "Viewer 1" tab. A new menu bar appears, with predefined views under Camera. Toggle auto-hide un-ticking "File->Hide Menus" in this bar.
- Help on visualization commands (zooming, panning, rotation etc can be found under Help in the menu bar mentioned above.
- Exit with .q from the root shell (closing the graphical window causes a crash).

Release history

The following section are kept for keeping track of past (obsolete) usage and development, and are not relevant for current users.

Configuration in older releases

In releases prior to CMSSW_7_4_3 and CMSSW_7_5_0_pre5, default configurations used to be:

- `Configuration.StandardSequences.MagneticField_AutoFromDBCurrent_cff`: Run I maps with automatic choice of nominal working point for data; fixed 3.8T for MC
- `Configuration.StandardSequences.MagneticField_38T_PostLS1_cff`: Run II map for 3.8T only.

When moving to CMSSW_7_4_3 and CMSSW_7_5_0_pre5, please replace these with `Configuration.StandardSequences.MagneticField_cff` as mentioned above. Configurations for fixed nominal working points, eg. 0 T, did not change (cf above).

History of map versions

Here is a brief description of the different development and production versions.

A collection of documents related to the field map computation, mapping, and comparison with measured data can be found here .

Version 85I_030919 ("old 4T map"; obsolete, no longer supported)

The 4T field map used in COBRA/ORCA and in CMSSW up to series 2.0.X included is based on a TOSCA finite-element computation performed in 2003. It is based on a geometric model ("magnetic field geometry") of 271 volumes representing one 30 degree-sector in $z < 0$ only (1/24 of CMS), thus the model is Z-symmetric and 12-fold phi-symmetric. For each volume, the field is computed on a regular (non-cartesian) grid defined appropriately. The CMSSW field interface returns the field at an arbitrary point using 3D linear interpolation within the grid cell containing the point inside each volume. The nominal B value in this map (4T) cannot be changed. A parametrization of the TOSCA computation (by V. Karimaki) can be optionally used in the region $r < 1.2$ m, $|z| < 3.0$ m in place of the interpolation; the main advantage is speed (and it also provides smoother derivatives than the interpolation).

This has been the only full field map available after 2003 and before CMSSW_2_1_0.

This version is no longer supported after CMSSW_2_2_10.

Version 1103I_071212 (Pre-CRAFT-2009 default; 3.8T version deprecated since 3_1_X; still used for some nominal operating points other than 3.8T)

This map is in use since CMSSW 2.2. and includes improvements based on the experience of the Magnet Test, which included a mapping campaign of the field within the solenoid, as well as measurements in the yoke (with flux loops during fast discharges, and with Hall probes). It is based on a TOSCA finite-element computation with a model that includes the Z-asymmetry of the coil spires and more accurate description of a

full half ($x>0$) of CMS including chimneys. However field data is still provided only for sector 1 (30 degrees, representing 1/12 of CMS with a total of 312 volumes), so the CMSSW map is 12-fold phi-symmetric (but not Z-symmetric).

The map with this model is available for several nominal field values: 2, 3, 3.5, 3.8 and 4T. The latter is expected to be different (more realistic) w.r.t. the previous 4T computation (851_030919).

A parametrization of the TOSCA computation within $r<1.15$ m, $|z|<2.8$ m is available for all field values (V. Karimaki). This parametrization matches very well with the computed data (difference of ~ 5 mT at the edges of the validity region) and **is used as a slave field in the default configuration** (cf section above), mainly to save CPU time in production and HLT.

A different parametrization of the **real mapping data** (V. Maroussov) is available **as an option**. It is defined in the region $r<1.9$ m, $|z|<3.5$ m, excluding the "corners" $|z|+2.5*r>6.7$ m. This parametrization reproduces the mapping data more precisely, but is slower; it is the suggested option for application where ultimate precision is necessary and CPU speed can be sacrificed. The agreement of this map with the mapping data is better than 0.4 mT (agreement to $\sim 10^{-4}$) in the tracker region and up to 0.8 mT at the edges of the validity region. To activate it specify the following:

```
process.ParametrizedMagneticFieldProducer.version = 'PolyFit2D'
process.ParametrizedMagneticFieldProducer.parameters = cms.PSet(
    BValue = cms.double(3.81143)
)
process.VolumeBasedMagneticFieldESProducer.useParametrizedTrackerField = True;
```

(the latter card is actually set to True by default in the standard config).

In 3.1.X releases and later, where this map version is superseded by 11031_090322_2pi_scaled, it can be activated using:

```
process.load("MagneticField.Engine.volumeBasedMagneticField_11031_cfi").
```

The **MF geometry** for this map was (a-posteriori) labelled version **90322** and is implemented in MagneticFieldVolumes_11031.xml.

Version 11031_090216 (Intermediate development; no longer supported)

Improved computation w.r.t. 11031_071212, with enlarged radius ($R=30$ m) of the total region where the field is computed, plus some additional details outside CMS (e.g. iron plate in the hall's floor). The same 312-volume 12-fold phi-symmetric geometry is used in the CMSSW interface. This map shows much better agreement with measurements with tracks in the iron yoke. This version was used for development only, and is no longer supported.

The map was still based on MF geometry version 90322.

Version 11031_090322 (deprecated)

Based on 11031_090216 this computation is done in volume extended also in $|Z|$ (up to 35 m). The same 312-volume 12-fold phi-symmetric geometry is used in the CMSSW interface. Changes w.r.t. 11031_090216 are visible basically only in the forward region.

This version was used for development only, and is deprecated in favor of later versions.

This version was available starting from recent 2_2_X versions with the configuration:

```
process.load("MagneticField.Engine.volumeBasedMagneticField_11031_090322_cfi")
```

Version 11031_071212 (Pre-CRAFT-2009 default; 3.8T version deprecated since 3_1_X; still used for some

The map was still based on MF geometry version 90322.

Version 1103l_090322_2pi (deprecated)

Based on 1103l_090322, uses different tables for the barrel yoke iron layers in the chimney sectors (S3, S4) and for S9, S10, S11 in the third barrel layer and in the endcaps. This version was used for development only, and is deprecated in favor of later versions.

Version 1103l_090322_2pi_scaled (Run I default for 3.8T, adopted starting from 3_1_0)

Same as 1103l_090322_2pi; in addition the field is scaled with factors measured from CRAFT data.

This version was used by default in CMSSW_3_1_0 and later releases for 3.8T (while 1103l_071212 remained the default for other nominal points) and was superseded by new map starting from version 120812.

The map was still based on MF geometry version 90322.

Version 1103l_090601 (Intermediate development; no longer supported)

Same as 1103l_090322_2pi; with small correction of the TOSCA geometry in L3 of barrel sectors 9,10,11. Since in S9,S11 the thicker L3 plates extend into air volumes 297-307, a finer grid was used for these volumes/sectors. Tables with this configuration were specified in MagneticFieldParameters_07_2pi_air.xml.

Two sets of scaling factors were prepared, based on CRAFT08 (ScalingFactors_090601_091022_cfi.py) and CRAFT08+09 (ScalingFactors_090601_091116_cfi.py). These were however never released.

Configuration (no longer supported)

```
process.load("MagneticField.Engine.volumeBasedMagneticField_090601_cfi")
```

or, to also apply scaling factors from CRAFT data:

```
process.load("MagneticField.Engine.volumeBasedMagneticField_090601_scaled_cfi")
```

The map was still based on MF geometry version 90322.

Version 090812 (Intermediate development; no longer supported)

Full TOSCA model made of two halves (_1 for X>0 and _2 for X<0). Has not been released into CMSSW.

Version 120812 (Default for Run II since CMSSW_6_2_0; replaced for data since 743/750pre5)

Full TOSCA model made of two halves (_1 for X>0 and _2 for X<0). Volume numbers are 1001-1360 and 2001-2360. Two versions exist, with the additional YE4 plate (post-LS1) and without (2010-2012 geometry.)

To use it in CMSSW_6_2_0, you had to use:

```
process.load("MagneticField.Engine.volumeBasedMagneticField_120812_largeYE4_cfi")
```

and for the 2010/2012 geometry:

```
process.load("MagneticField.Engine.volumeBasedMagneticField_120812_cfi")
```

The map was based on a new **MF geometry version 120812** (implemented in MagneticFieldVolumes_1_v7_large.xml, MagneticFieldVolumes_2_v7_large.xml), which is no longer in use.

Version 1103l_071212 (Pre-CRAFT-2009 default; 3.8T version deprecated since 3_1_X; still used for some

Version 130503 (Default for 3.8T data only, starting from 743/750pre5)

Full TOSCA model of the entire CMS; reduced memory footprint cf. this presentation[?]. 804 volumes per sector; volumes outside any detector are replicated starting from sector 1 (volumes extending from R~7.6 m to to R=9 m, ie 1011,

1028-1029,1034-1035,1042-1043,1050-1051,1058-1059,1064-1065,1072-1073,1078-1079; volumes beyond |Z|>10.86 ie 1084-1129,1136-1137; and the corresponding +1000). Volumes 1134-1135 (for the volume outside CASTOR) are replicated from sector 4 instead than 1, to avoid aliasing effects due to the plates in the cylinder gap.

The map was based on the new **MF geometry version 130503** (implemented in MagneticFieldVolumes_1_v9_large.xml, MagneticFieldVolumes_2_v9_large.xml), which is no longer in use.

Version 160812 (Current default for 3.8, 3.5 and 3T, both data and MC since later releases in the 93X series)

Updated TOSCA model, cf. this presentation[?]:

- Extend the description of the rotating shielding and TAS up to |Z|=22 m (was up to |Z|=17.86 m in the previous map)
- Add the 18 kA current leads in the chimney in YB-1 S3

The map was computed for 3.8T, 3.5T, 3T in the RunII configuration and for 3.8T only in the RunI configuration. It is currently used for all nominal points except for 2T for both RunI and II.

The map was based on the new **MF geometry version 160812** (implemented in MagneticFieldVolumes_160812_1.xml, MagneticFieldVolumes_160812_2.xml).

Configuration of superseded table sets in DB

The following tables report the DB configurations loaded in the DB **prior** to the present version (for historical purpose only).

Run 1 default ("090322_2pi_scaled") - data and MC

Current (A)	B(T)	geometryVersion	Table set (version)	Parametrization type (paramLabel)	paramData	scaling factors
0-4779	0	90322	grid_1103l_071212_4t	Uniform	0	-
4779-11987	2	90322	grid_1103l_071212_2t	OAE_1103l_071212	2	-
11987-15617	3	90322	grid_1103l_071212_3t	OAE_1103l_071212	3	-
15617-17543	3.5	90322	grid_1103l_071212_3_5t	OAE_1103l_071212	3.5	-
17543-18765	3.8	90322	grid_1103l_090322_3_8t	OAE_1103l_071212	3.8	ScalingFactors_090322_2pi_0905
>18765	4	90322	grid_1103l_071212_4t	OAE_1103l_071212	4	-

Run 2, MC

Current (A)	B(T)	geometryVersion	Table set (version)	Parametrization type (paramLabel)	paramData	scaling factors
0-4779	0	90322	grid_1103l_071212_4t	Uniform	0	-
4779-11987	2	90322	grid_1103l_071212_2t	OAE_1103l_071212	2	-
11987-15617	3	90322	grid_1103l_071212_3t	OAE_1103l_071212	3	-
15617-17543	3.5	130503	grid_130503_3_8t_v9_large	OAE_1103l_071212	3.5	-
17543-18765	3.8	120812	grid_120812_3_8t_v7_large	OAE_1103l_071212	3.8	-

Version 1103l_071212 (Pre-CRAFT-2009 default; 3.8T version deprecated since 3_1_X; still used for some

>18765	4	90322	grid_11031_071212_4t	OAE_11031_071212	4	-
--------	---	-------	----------------------	------------------	---	---

Run 2, data						
Current (A)	B(T)	geometryVersion	Table set (version)	Parametrization type (paramLabel)	paramData	scaling factors
0-4779	0	90322	grid_11031_071212_4t	Uniform	0	-
4779-11987	2	90322	grid_11031_071212_2t	OAE_11031_071212	2	-
11987-15617	3	90322	grid_11031_071212_3t	OAE_11031_071212	3	-
15617-17543	3.5	130503	grid_130503_3_5t_v9_large	OAE_11031_071212	3.5	-
17543-18765	3.8	130503	grid_130503_3_8t_v9_large	OAE_11031_071212	3.8	-
>18765	4	90322	grid_11031_071212_4t	OAE_11031_071212	4	-

Fancy use cases

How to load two field maps in the same job

Any number of different maps can be loaded in the same job and accessed in parallel, for e.g. comparison. One of them will be the "standard" one which is visible by all modules; it will be the one with a null label., the others are available in the job using their specific non-null label. In the following example we specify a second map with label '090216_3_8t' in addition to the standard one (default 3.8T map).

```
# The standard one
process.load("Configuration.StandardSequences.MagneticField_38T_cff")

# The new one with label "090216_3_8t"
# Note I use a different process name than what us used in the default.
process.VolumeBasedMagneticFieldESProducerNew =
cms.ESProducer("VolumeBasedMagneticFieldESProducer",
  timerOn = cms.untracked.bool(False),
  useParametrizedTrackerField = cms.bool(False),
  label = cms.untracked.string('090216_3_8t'),
  version = cms.string('grid_11031_090216_3_8t'),
  debugBuilder = cms.untracked.bool(False),
  cacheLastVolume = cms.untracked.bool(True),
  scalingVolumes = cms.vint32(),
  scalingFactors = cms.vdouble()
)
```

In the code, you will then be able to get both fields maps:

```
ESHandle<CMS.MagneticField> magfield;
setup.get<IdealMagneticFieldRecord>().get(magfield);

ESHandle<CMS.MagneticField> newMagfield;
setup.get<IdealMagneticFieldRecord>().get("090216_3_8t",newMagfield);
```

How to apply per-volume rescaling factors

A scaling factor for |B| can be specified independently for each of the 312 volumes in each sector.

The scaling is provided with two matching vectors. The first vector lists the volumes, encoded as (100*volume number + sector). Entries with sector=0 will match all sectors for which a specific sector entry is not set.

The second vector specifies the corresponding scaling factors.

The volume numbers for the iron yoke plates can be obtained from this schema [↗](#).

In version, 1103l_090322_2pi_scaled, the default scaling factors are implemented in MagneticField/Engine/python/ScalingFactors_090322_2pi_090520_cfi.py [↗](#). If you need to test modified scaling factor, please use this file as a starting point.

Annotated lists of volumes (actual factors are NOT significant):

- all barrel yoke iron volumes, including brackets.
- endcap iron disks volumes

Review status

Reviewer/Editor and Date (copy from screen)	Comments
KatiLassilaPerini - 23 Jan 2007	created template page
NicolaAmapane - 01 Jul 2008	first edit
NicolaAmapane - 10 Aug 2019	review

Responsible: Nicola Amapane

Last reviewed by: Never reviewed

CMS.MagneticField/GeomBuilder

This topic: CMSPublic > SWGideMagneticField

Topic revision: r43 - 2020-04-24 - NicolaAmapane



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback