New CAD / Mirrors Import Mechanisms in GEMC

M. Ungaro

*Jefferson Lab, 12000 Jefferson Avenue, 23606 Newport News, VA, USA*

**Abstract**

The CAD import mechanism in GEMC [1] has been enhanced to support geometry variations. A new CADSQLITE factory has been introduced, enabling the indexing of CAD-imported geometry by Run Number, with definitions stored in an SQLITE database. Furthermore, the SQLITE factory has been extended to include the definition and storage of mirror properties.

**Overview**

Prior to GEMC version 5.12, the process for importing volumes from CAD into simulations relied on variation names embedded in the CAD file paths. Additional attributes for the volumes, such as displacements, materials, and mother volumes, could be specified using an optional XML file. However, this approach had several limitations:

* **Lack of Proper Geometry Variation Handling**: Geometry variations were not adequately supported. To address this limitation, a workaround was implemented for CLAS12, which involved duplicating directories under different path names, each with its own XML file specifying variation-specific changes. For instance:

<detector name="ctof/javacad\_rga\_spring2018/" factory="CAD"/>

<detector name="ctof/javacad\_rga\_fall2018/" factory="CAD"/>

This approach resulted in file duplication, which has now been eliminated in the updated system.

* **Loading Unwanted Files**: All files within a CAD directory were loaded indiscriminately, including those irrelevant to certain variations. To manage this, additional directories were created containing only the desired files, even if shared with other variations.

This approach also resulted in file duplication, which has now been eliminated in the updated system.

To provide a simpler and more general mechanism, the following improvements have been implemented:

* **Geometry Variations with XML Filenames**: Geometry variations are now managed through XML filenames, eliminating the need for embedding variation names in file paths.
* **Selective Volume Loading**: Only the volumes explicitly specified in the XML files are loaded, preventing the inclusion of unnecessary files.
* **Enhanced SQLITE Support**: SQLITE functionality has been added to streamline the management of CAD-imported geometry.

These changes eliminate the need to duplicate directories or rely on additional attributes in steering cards.

Furthermore, SQLITE support has been extended to include the definition and storage of mirror properties.

**API Changes: CAD factory in the steering card**

The previous CAD factory mechanism has been updated to incorporate geometry variations directly through XML filenames. For instance, steering card entries such as:

<detector name="htcc/cad\_rga\_fall2018/" factory="CAD"/>

Have been replaced with:

<detector name="htcc/cad/" factory="CAD" variation="rga\_fall2018"/>

This new approach loads the XML file htcc/cad/cad\_rga\_fall2018.gxml, which specifies the list of volumes to be loaded and their associated properties. If a volume is not required for a specific variation, its absence in the XML file will indicate to GEMC that it should not be loaded.

Importantly, the same htcc/cad/ subdirectory is now shared across all variations, eliminating the need for duplicated directories.

**SQLITE factory for CAD volumes**

New functionality has been added to the api/perl/sqlite.py utility to create a SQLITE database table named cad, incorporating all modifiers supported in the XML. Additionally, a new PERL API has been introduced to populate the cad table with the required entries.

An example code to define an HTCC CAD volume and store in the SQLITE DB:

my %cad = init\_cad();

$cad{"name"} = "htccMollerConeExt";

$cad{"color"} = "888888";

$cad{"material"} = "rohacell31";

$cad{"position"} = "0\*cm 0\*cm $zpos1\*cm";

$cad{"rotation"} = "0\*deg 180\*deg 0\*deg";

print\_cad(\%configuration, \%cad);

The sqlite filepath is specified in the dbhost entry in the configuration file.

The API populates the database table with the relevant entries, as illustrated in Figure 1. To ensure unique naming in the GCARDS, the system name is prefixed with “\_cad”.

Consequentially entries such as:

<detector name="htcc" factory="TEXT" variation="rga\_spring2018"/><detector name="cad/" factory="CAD" variation="rga\_spring2018"/>

are written using the SQLITE factories as:

<detector name="htcc" factory="SQLITE"/>  
<detector name="htcc\_cad" factory="SQLITECAD"/>

By default, the variation is set to “default” and does not need to be explicitly mentioned in the SQLITE factories. Instead, the run number is used to distinguish configurations.

For instance, Figure 1 illustrates the HTCC CAD definitions for the original (run=11), rga\_spring2018 (run=3029), and rga\_fall2018 (run=4763) configurations.

A picture containing graphical user interface

Description automatically generated

*Figure 1: the HTCC CAD geometry definitions in the SQLITE database for three run numbers.*

**API Changes: API PERL SQLITE factory for Mirrors**

New functionality has been added to the api/perl/sqlite.py utility to create a SQLITE database table named mirrors, which stores properties required to define reflectivity, refractive index, finish type, and more. The PERL API for mirrors has been extended to populate the mirrors table with the same entries as the TEXT API.

To ensure a seamless transition for users, the same code used to define mirror surfaces is employed to populate the SQLITE database. When the SQLITE factory is selected, the output automatically generates entries in the mirrors table. An example of this process is illustrated in Figure 2 for the HTCC system.

**Graphical user interface, application

Description automatically generated**

*Figure 2: the HTCC mirrors definitions in the SQLITE database for three run numbers.*

**CLAS12 Simulation CAD imports changes**

Below is the list of changes to the CLAS12 simulation [2] in the GitHub gemc/detectors repository that use the new CAD import mechanism:

* **HTCC**:
* cad\_spring18 and cad\_fall18 files condensed to cad
* cad/cad.gxml 🡪 cad/cad\_original.gxml
* cad\_spring18/cad.gxml 🡪 cad/cad\_rga\_spring2018.gxml
* cad\_fall/cad.gxml 🡪 cad/cad\_rga\_fall2018.gxml
* **LTCC**:
* cad/cad.gxml 🡪 cad/cad\_default.gxml
* cad\_cones files moved to cad
* variation “rgb\_winter2019” renamed “rgb\_winter2020” to follow CALCOM conventions. Variation “rgm” renamed “rgm\_winter202”
* Added gcards:
* ltcc\_text\_default.gcard
* ltcc\_text\_rga\_spring2018.gcard
* ltcc\_text\_rgb\_winter2020.gcard
* ltcc\_text\_rga\_fall2018.gcard
* ltcc\_text\_rgb\_spring2019.gcard
* ltcc\_text\_rgm\_winter2021.gcard
* **DDVCS**:
* cad/cad.gxml 🡪 cad/cad\_original.gxml
* **Targets**:
* hdIce/cad.gxml 🡪 cad\_hdIce/cad\_default.gxml
* rge-dt/Pb/cad.gxml 🡪 rge-dt/Pb/cad\_default.gxml
* rge-dt/Empty/cad.gxml 🡪 rge-dt/Empty/cad\_default.gxml
* rge-dt/Al/cad.gxml 🡪 rge-dt/Al/cad\_default.gxml
* rge-dt/Cu/cad.gxml 🡪 rge-dt/Cu/cad\_default.gxml
* rge-dt/common/cad.gxml 🡪 rge-dt/common/cad\_default.gxml
* rge-dt/Sn/cad.gxml 🡪 rge-dt/Sn/cad\_default.gxml
* rge-dt/C/cad.gxml 🡪 rge-dt/C/cad\_default.gxml
* cad/cad.gxml 🡪 cad/cad\_default.gxml
* ltarget/cad.gxml 🡪 cad\_long/cad\_default.gxml
* transverseTargetCad/cad.gxml 🡪 cad\_transverse/cad\_default.gxml
* cade/cad.gxml 🡪 cad\_extended/cad\_default.gxml
* cadrgm/cad.gxml 🡪 cad\_rgm/cad\_default.gxml
* flagCad/cad.gxml 🡪 cad\_rgd/cad\_default.gxml
* PolTarg/cad.gxml 🡪cad\_poltarg/cad\_default.gxml
* **Magnets**:
  + magnets/cad/cad.gxml 🡪 magnets /cad/cad\_default.gxml
* **CTOF**:
  + fixed lightguides overlaps and upstream positions
  + javacad\_default, javacad\_rga\_spring2018, javacad\_rga\_fall2018 condensed to cad
  + corresponding \_upstream directories condensed to cad\_upstream
  + javacad\_default/cad.gxml 🡪cad/cad\_default.gxml
  + javacad\_rga\_spring2018\_upstream/cad.gxml 🡪 cad/cad\_rga\_spring2-18.gxml
  + javacad\_rga\_fall2018/cad.gxml 🡪 cad/cad\_default.gxml
  + javacad\_default\_upstream/cad.gxml 🡪 cad\_upstream/cad\_default.gxml
  + javacad\_rga\_spring2018/cad.gxml 🡪 cad\_upstream/cad\_default.gxml
  + javacad\_rga\_fall2018\_upstream/cad.gxml 🡪 cad\_upstream/cad\_default.gxml
* **RICH**:
  + cad\_default, cad\_rgc\_summer2022, cad\_rga\_fall2018 condensed to cad
  + cad\_default/cad.gxml 🡪cad/cad\_default.gxml
  + cad\_rgc\_summer2022/cad.gxml 🡪 cad/cad\_rgc\_summer2022.gxml
  + cad\_rga\_fall2018/cad.gxml 🡪 cad/cad\_rga\_fall2018.gxml
* **Beamline**:
* cadBeamline, cadBeamlineFTOFF, cad\_downstream\_beamline, cadBeamlineELMO condensed to cad
* cadBeamline/cad.gxml 🡪cad/cad\_FTOn.gxml
* cadBeamlineFTOFF/cad.gxml 🡪 cad/cad\_ FTOff.gxml
* cadBeamlineELMO/cad.gxml 🡪 cad/cad\_ELMO.gxml
* transverseTargetMagnets/forwardWedges/cad.gxml 🡪 cad\_transverse/ forwardWedges/cad\_default.gxml
* transverseTargetMagnets/midpointWedges/cad.gxml 🡪 cad\_transverse/ midpointWedges /cad\_default.gxml
* transverseTargetMagnets/noWedges/cad.gxml🡪 cad\_transverse/ noWedges /cad\_default.gxml

**Gcards changes in clas12-config**

The following changes have been made to the steering cards in the GitHub jeffersonlab/clas12-config repository:

* Proper variations now replace the CAD subdirectory path directives.
* Many variation definitions have been updated to replace GCARDS entries for displacing or removing detector elements.

Additionally, the use of the string default to specify the detector variation has been removed, as this is already the default behavior in GEMC.

**HTCC**

* The text “original” variation has been replaced with rga\_spring2018 and rga\_fall2018 when applicable.
* The entries:

<detector name="htcc/javacad\_rga\_spring2018/" factory="CAD"/>

<detector name="htcc/javacad\_rga\_fall2018/" factory="CAD"/>

Have been replaced with, respectively:

<detector name="htcc/cad/" factory="CAD" variation="rga\_spring2018"/>

<detector name="htcc/cad/" factory="CAD" variation="rga\_fall2018"/>

The HTCC displacements in all gcards have been removed as they are incorporated in the variations.

**LTCC**

* The “cad\_cone” detectors entries have been removed.
* Variation “rgb\_winter2019” is renamed “rgb\_winter2020” to follow CALCOM conventions. Variation “rgm” renamed “rgm\_winter2021”

**Targets**

* cadrgm replaced with cad\_rgm in the rgm gcards

**CTOF**

* The entries:

<detector name="ctof/javacad\_rga\_spring2018/" factory="CAD"/>

<detector name="ctof/javacad\_rga\_fall2018/" factory="CAD"/>

Have been replaced with, respectively:

<detector name=" ctof/cad/" factory="CAD" variation="rga\_spring2018"/>

<detector name=" ctof/cad/" factory="CAD" variation="rga\_fall2018"/>

**RICH**

* Removed cad\_downstream\_beamline entries in the gcard as they’re now condensed in cad
* The entries:

<detector name="rich/cad\_rga\_spring2018/" factory="CAD"/>

<detector name="rich/cad\_rgc\_summer2022/" factory="CAD"/>

Have been replaced with, respectively:

<detector name="rich/cad/" factory="CAD" variation="rga\_spring2018"/>

<detector name="rich/cad/" factory="CAD" variation="rga\_fall2018"/>

**BEAMLINE**

<detector name="beamline/cadBeamline/" factory="CAD"/>

<detector name="beamline/cadBeamlineFTOFF/" factory="CAD"/>

<detector name="beamline/cadBeamlineELMO/" factory="CAD"/>

Have been replaced with, respectively:

<detector name="beamline/cad/" factory="CAD" variation="FTOn"/>

<detector name="beamline/cad/" factory="CAD" variation="FTOff"/>

<detector name="beamline/cad/" factory="CAD" variation="ELMO"/>

**Summary**

Recent updates to GEMC and its supporting utilities introduce significant improvements to the CAD import mechanism, addressing limitations in geometry variation handling, volume selection, and mirror property definition. Key enhancements include:

1. **CAD Import Mechanism**: Geometry variations are now managed using XML filenames, eliminating the need for directory duplication and steering card workarounds.
2. **SQLITE Integration**: The introduction of the CADSQLITE factory allows CAD-imported geometry to be indexed by Run Number and stored in an SQLITE database. Additional SQLITE support enables defining and storing mirror properties, making the transition seamless for users.
3. **Updated APIs**: Enhancements to api/perl/sqlite.py provide robust functionality for creating and populating database tables (cad and mirrors) and managing experiment configurations based on variations or run numbers.
4. **GCARDS Adjustments**: Replacing CAD subdirectory path directives with proper variations, alongside removing the redundant default string, simplifies configuration in the clas12-config repository

**References**

[1] *M. Ungaro*, GEMC: A database-driven simulation program, EPJ Web of Conf.

Volume 295, 2024

[2] *M. Ungaro et al*, The CLAS12 Geant4 simulation, Nucl.Instrum.Meth.A 959 (2020) 163422