

# Running static analysis suite SAS on Gaudi

---

Gaudi developers meeting

September 28, 2016

**Joschka Lingemann**

Benedikt Hegner, David Ho

EP-SFT - CERN

# What is SAS: Static Analysis Suite?

---

Developed in SFT / CMS to hook into clang front-end

- Idea: Ease development of new clang-checkers, run them together with clang-modernize & -format
- Allows wide variety of checks to be implemented
  - Prototype with few checkers (arg-size, const casts, statics)
- One use case are coding conventions:
  - White-space etc. with clang-format
  - Naming conventions with dedicated checkers

# Status of SAS

---

Time spent over the summer to improve:

- Naming convention generator from regular expressions
- Extend python wrappers: Configurable + new reporting mechanism
- CMake macros to integrate into projects
  - Overrides compiler with python wrappers, adds arguments

```
option(SAS "Whether to run static code analysis with SAS." OFF)
if (SAS)
  find_package(sas)
  enable_sas(
    [FORMAT]
    [MODERNIZE]
    [DISABLE_CHECKERS sas.CodingConventions]
    COMPARISON_REPORT_DIR "${CMAKE_BINARY_DIR}/sas_report"
    IGNORE_DIRS ${CMAKE_SOURCE_DIR}/cmake ${CMAKE_SOURCE_DIR}/build
  )
endif(SAS)
```

# Demonstrator run with FCC software coding conventions

---

For FCC we decided to follow the Gaudi / LHCb conventions:

- LHCb conventions; FCCSW conventions
- Finalised over the summer:
  - ▶ clang-format configuration
  - ▶ naming convention configuration

These + performance checks were used to run SAS on Gaudi,  
see report

# Summary

---

## SAS can be easily integrated

- Small CMake configuration addition
- If there is interest: I can provide instructions / merge req.
- May need to further adopt configuration to Gaudi

Additional material

# How to deploy in nightlies...

---

```
source init.sh # setup the environment needed for Gaudi
# setup llvm and clang
source /afs/cern.ch/sw/lcg/external/llvm/3.6/x86_64-slc6-gcc49-opt/setup.sh
# setup environment needed for sas (currently deployed for FCC)
export SASDIR=/afs/cern.ch/exp/fcc/sw/snapshot/sas/snapshot/x86_64-slc6-gcc48-opt/
export CMAKE_PREFIX_PATH=$CMAKE_PREFIX_PATH:$SASDIR
export PATH=$PATH:$SASDIR/scripts
# get the clang-format configuration file (can also be put into the repo)
cp $SASDIR/config/FCCSW-clang-format.yaml .clang-format
mkdir build.static_analysis
cd build.static_analysis
cmake -DCMAKE_TOOLCHAIN_FILE=../toolchain.cmake -DLCG_SYSTEM=x86_64-slc6-gcc49 \
      -DSAS=ON \
      -DCMAKE_CXX_COMPILER=`which sas_check++` -DCMAKE_C_COMPILER=`which sas_check` ..
# run the tests
make
# create the report
make report_index

# ...deploy the report...
```