

# Status and proposal for NXCALS

# Status

Legacy application -> need of a drop-in replacement of CALS (no need to change codes) since it is very costly

Agreed with CO: C. Roderik statements Evian 2019

“PyTIMBER will continue to exist with NXCALS and users will not need to change their code.

The current CALS system will be phased out before the end of LS2, once the following criteria have been satisfied:

- NXCALS contains all of the required historical CALS data.
- A backwards compatible Java client API has been provided.
- A new version of TIMBER is available in production.
- The data extraction performance is at least comparable with CALS.”

These conditions are not yet there...

# Backport API status

## MetaService

`getVariablesOfDataTypeWithNameLikePattern`

`getVariablesWithNameInListofStrings`

## TimeseriesService

`getDataInTimeWindow`

`getDataAlignedToTimestamps`

`getDataInFixedIntervals`

`getLastDataPriorToTimestampWithinDefaultInterval`

`getNextDataAfterTimestampWithinDefaultInterval`

`getDataInTimeWindowFilteredByFundamentals`

`getVariableStatisticsOverMultipleVariablesInTimeWindow`

## createLHCFillService

`getLHCFillAndBeamModesByFillNumber`

`getLastCompletedLHCFillAndBeamModes`

`getLHCfillsAndBeamModesInTimeWindow`

`getLHCfillsAndBeamModesInTimeWindowContainingBeamModes`

## Summary of previous discussions

In January 2019 we provided a set of 13 methods required test the applications.

It was agreed they would be ready by June 2019.

As of 30 Jan 2020 the part handling LHC Fills and Beam modes is not available.

As of today:

- Fill service announced.
- Present porting stuck on different behaviors of the resulting data structures:

<https://issues.cern.ch/browse/NXCALS-3095>

## Request:

From the moment it becomes available we need 3 months to migrate pytimber and test applications relying on it.

Necessary before considering CALS decommissioning

# Status other tests

We tested ([CWG](#), Dec 13 2018) data availability, extraction speed, data integrity using native NXCALS API (requiring ad-hoc code development) and we gave the following feedback

- Extraction speed for small dataset is 20-10 times slower due to large initialization overhead
- Extraction speed for large raw dataset is 3 times slower (e.g. BBQ data)
- Found datasets missing data points in the few datasets tested
- Library requirements complicates integration in existing system
- Spark API efficient and fast to query for aggregated data

NXCALS team recently stated that the issue are being addressed or will be addressed.

Further tests are not planned for now -> We will make further tests after those points are addressed

# PyTimber

- After survey launched to LHC and Injector MD users
- PyTimber support for NXCal is blocking the following groups on April 2020 timescale:
  - OP (V. Kain& team)
  - ABP (linacs, injectors, lhc, omc, e-cloud, beam-beam, luminosity...)
  - RF (R. Calaga)
  - ABT (C. Bracco and team)
- Presently R. De Maria is the only maintainer with no time allocated.
- V. Kain asked CO to add another person. C. Roderik will try to find one, no time estimate yet. Next update at next Injector Performance meeting.
- Present porting stuck on different behaviors of the resulting data structures:  
<https://issues.cern.ch/browse/NXCALS-3095>

Request: Share follow-up issues and steering among the users groups and CO or allocate sufficient resources in ABP.

# Summary

- CO should define a new date in which:
  - Backport API is completed and fully compliant with CALS behaviors
  - Data integrity is guaranteed
- After these requirements are actually fulfilled we need in following 3 months we need to allocate resource for
  - Finalize PyTimber port
  - Test relevant applications
  - Provide new feedback on performance

Spare

# Report

Contact	Group	Machines	Additional people	pytimber	Time	Features
N. Mounet	ABP	LHC			end of year	CALS or NXCALS LHC instability monitoring activities and MD
V. Kain	OP	SPS	K. Lee, J. Dalla-Costa		April/2020	performance tracking with NXCALS
F. Velotti	ABT	PSB, LHC			Sep/2020	NXCALS
C. Bracco	ABT		10 users in the team		Sep/2020	
P. Skowronski	OP	Linac4	Gian piero?		May/2020	one api CALS, NXCALS is very very useful
T. Persson	ABP	LHC				is trying spark
R. Scrivens	ABP	Linac4			April/2020	NXCALS support of pytimber
G. Iadarola	ABP	LHC/SPS	7 people in team		April/2020	NXCALS support of pytimber in prospect CALS will be dismanteld in 2021 for ecloud studies
X. Buffat	ABP	LHC	6 people in the team		none	will try pyspark but not tried yet
S. Albright	RF	PSB			September	for CALS, will try pyspark for NXCALS
I. Efthymiopoulos	ABP	LHC	guido		summer	one api CALS, NXCALS is useful otherwise needs to reimplement functionalities. API stability is a concern. Include panda functions from guido
H. Damerau	RF	injectors			summer	is testing pyspark not aware before
R. Calaga	RF	SPS/LHC			no reply	general
A. Huschauer	ABP	injectors			September	need pytimber-like API, pyspark too cumbersome at the moment
F. Giordano	ABP	LHC?			no reply	yes
G. Sterbini	ABP	LHC & injectors			needed	NXCALS, tried pyspark too slow
K. Lee	ABP	Injectors			needed	many script in pytimber, if nxcals provides an easy enough interface could used
J. Dilly	ABP	LHC			needed	many script in pytimber, if nxcals provides an easy enough interface could used