Data Preservation and Long Term Analysis

Yves Kemp

GridKa School 2009
Karlsruhe 31.8.2009

HELMHOLTZ ASSOCIATION

DESY

My personal motivation and introduction

- Request of a retired professor in 2008 to DESY-IT:
- "Around 1975*, we had tapes from a bubble chamber experiment at the Computing Center. Are these still available, maybe copied to other media? I got a request from CERN concerning these tapes."
- We did not have them…
- Honestly, I thought, no one would need these data anymore
  - New experiments, higher energy, better resolution, …
  - No one able to read / understand the (scientific content of the) data
- I was proven wrong: Data preservation and long term analysis is needed!

*1975: Bill Gates founds Microsoft: An eternity in Computer Science!

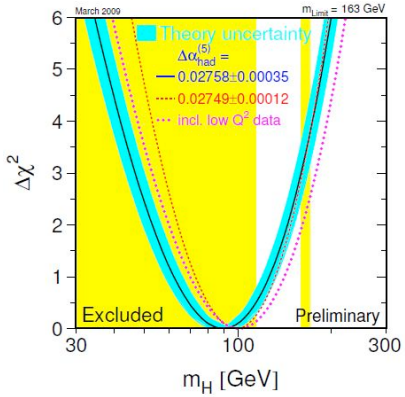
Questions:

- Why do/should we want to preserve data?
- Four models for preservation
- How to store data over a long period?
  - What does it cost? Which procedures and techniques?
- How to guarantee that one can read (I.e. analyze) it afterwards?

- I apologize for concentrating on HEP-centric examples
  - http://www.dphep.org/
  - I am sure though that most also applies for other science

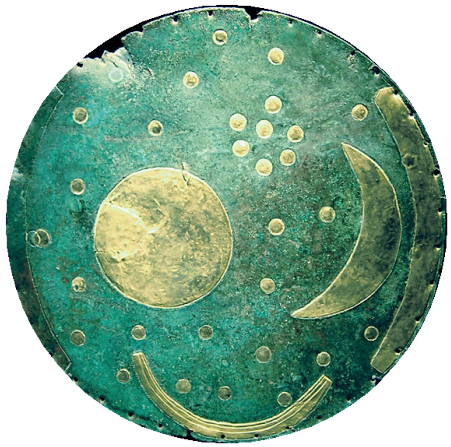
Why preserving experimental data?

- Long-term completion and extension of scientific programs
  - Allow “late analyses” to be done: +5-10% more publications
  - “Late analyses” benefit from full statistics, best understanding of systematics
- Cross-collaboration analyses
  - Often performed at the end of lifetime of collaborations
  - Even among generations of experiments
- Re-use of the data
  - Re-analyze the data with new theoretical models, new analysis techniques, …
- Education, training and outreach
  - E.g. analysis by students without restrictions (like collaboration membership…)



But with modern tools, old data is superseded?

- Obtaining scientific content of old data rapidly with new tools and experiments?
  - **Yes:** Obvious example: Precision increasing from generation to generation
  - **No:** Some experiments are unique: Hera probably the last electron-proton collider for long.
- More general arguments for data preservation:
  - Reproducibility of results is good practice in Science
  - Historical interest in past experiments and methods
  - Publicly funded research should/must be publicly accessible



Nebra sky disk
Source: Wikimedia Commons

Four preservation models: (dphep.org)

Listed by increasing complexity, higher ones includes lower ones.

1. **Providing additional documentation**
   - No data preservation per se. Examples:
   - Publication data (data tables, high-level analysis code)
   - Internal collaboration notes, (e)log-books, minutes, slides, news, blogs, wikis,…
   - Meta-data related to running conditions
   - …

> INSPIRE project replaces and enhances SPIRES
   - SPIRES database of particle physics literature (since late 1960's)
   - http://www.projecthepinspire.net/

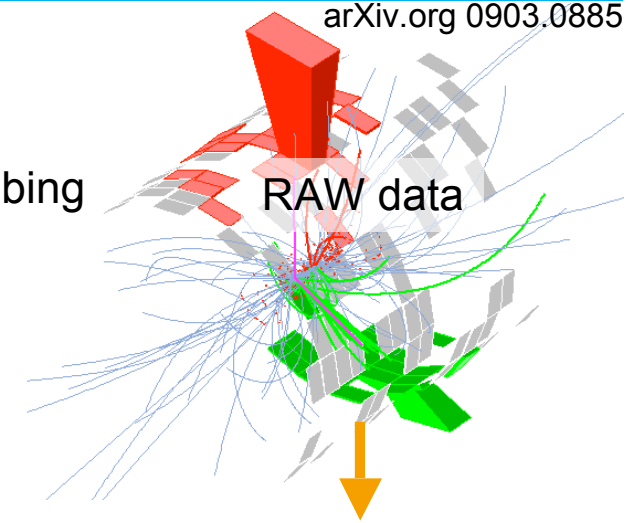
> Consulting with professional archivist helpful

> Minimal solution: Might show not sufficient in the end

> Minimal cost: Especially when planned from the beginning

Data preservation models 2 and 3

Image: CDF event
arXiv.org 0903.0885



2. Preserve data in simplified format

- Preserve reprocessed, basic, event-level, four-vectors describing detected particles
- Very simple structure, low data volume (~1 kB/event)
- Generally no full analysis, useful for outreach and education
- Moderate additional costs

3. Preserve the analysis level software and data format

- All analysis level software, including external software
- Existing detector and simulated data sets are sufficient, no reprocessing or new simulation
- More effort required. External software big "**?**"
- Software must run also on future computers

Ntuple with some advanced structure

(10-100 kByte/event)

Electron1 X Y Z M

Electron2 X Y Z M

Jet1 X Y Z M

Neutrino1 X Y Z 0

> Data formats can be agreed between experiments

DESY

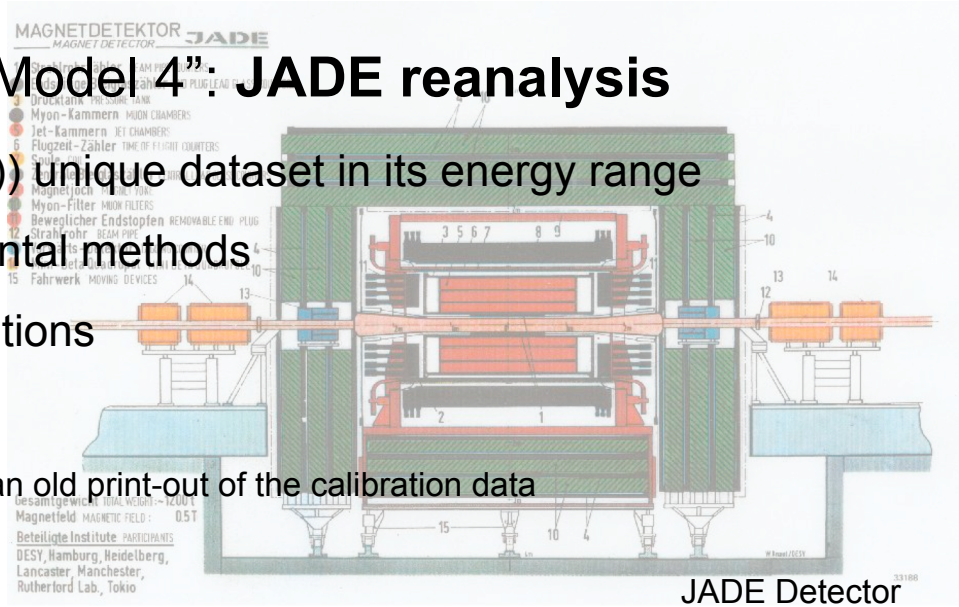
Model 4: All software and basic level data

4. **Some analyses need new simulated data or re-reconstruction**
   - Basic level data (raw or equivalent format), experiment specific
   - Special care: calibrations, simulation tunings,… (see model 1)
   - Common format not possible, rather look for a common standard

> Significant resources during preparation and maintenance phase

> Example of successful “unplanned Model 4”: **JADE reanalysis**
   - Data taken 1979-1986 (PETRA). (Now:-)) unique dataset in its energy range
   - New theoretical input and new experimental methods
   - Effort 1995-2003 resulted in new publications
   - Several anecdotes

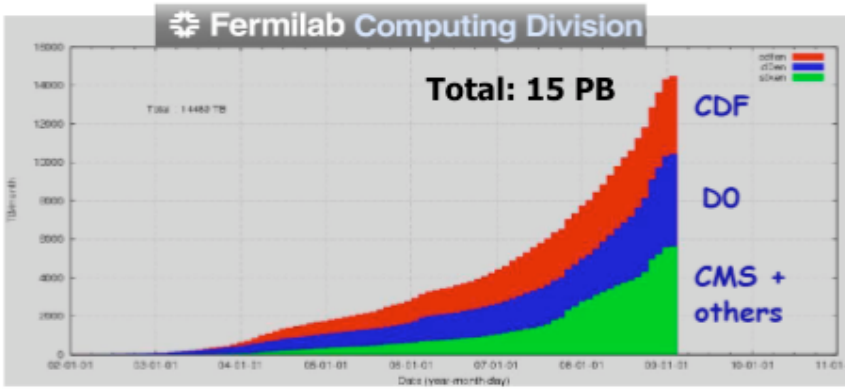
     My preferred one: Manually typing into computer an old print-out of the calibration data



JADE Detector

Storage technology: Simple model and costs

- **Just store everything, forever!**
- Assumptions:
  - Data can be copied to new media regularly (each generation)
  - Media is robot managed (no shelved tapes!)
  - Copy process is reasonably fast
  - Capacity/price doubles with each generation



- Storage costs and needed capacity **doubles**
  - $1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots=2$ (Geometric sum)
  - Theoretically for infinite time!
  - Even if acquired data grow exponentially!
- But needs constant work: Can't we store it "once and forever"?

Which technology for durable storage?

- Examples of successful systems:

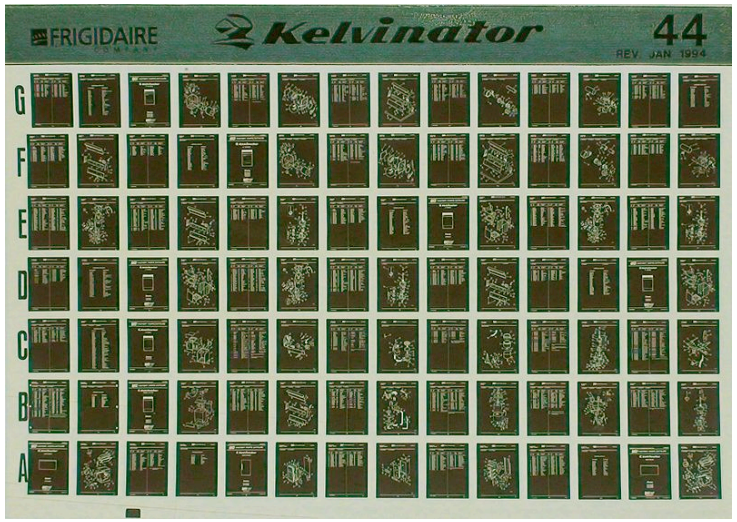


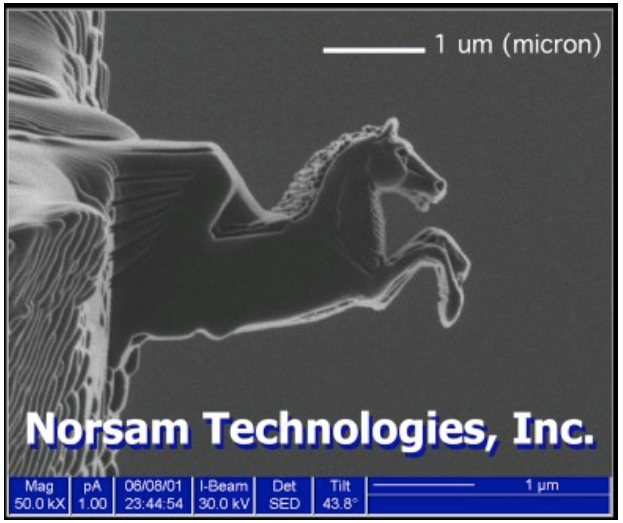
- Key ingredients for success:
  - Natural language
  - Readable with “build-in” tools (e.g. eyes)
  - Low abrasive / durable media (e.g. stone)
- Downside: Low information density
  - Can we repeat this “the modern way”?

Image source: Wikimedia Commons

Some examples of durable solutions:

- Microfiche
  - Reduce needed space by 95%
  - Up to 500 years (appropriate conditions)





- Ion beam based modelation of ‘very stable’ materials (i.e. gold plated silicon, stainless steel)
  - very high density (>20Gbit/sqin)
  - good for >1000years !

Image source: Wikimedia Commons
Norsam Technologies

Processing the data in NN years

- In NN years, someone has to make an interpretation of the data
  - Access the storage (e.g. through Grid / SRM / Cloud standards …)
  - Analysis
- Understand the content of the data
  - They might know what `Electron(1)->Px()` is
  - Will they know what `MidPointJet04(2)->Iso59()` is?
- They might need the whole analysis chain to reprocess your data and make new simulations
  - And understand every single step…
- Do this in many generations from now!

Some complex dependencies

Personal Analysis Code

↓

Experiment Framework

↓

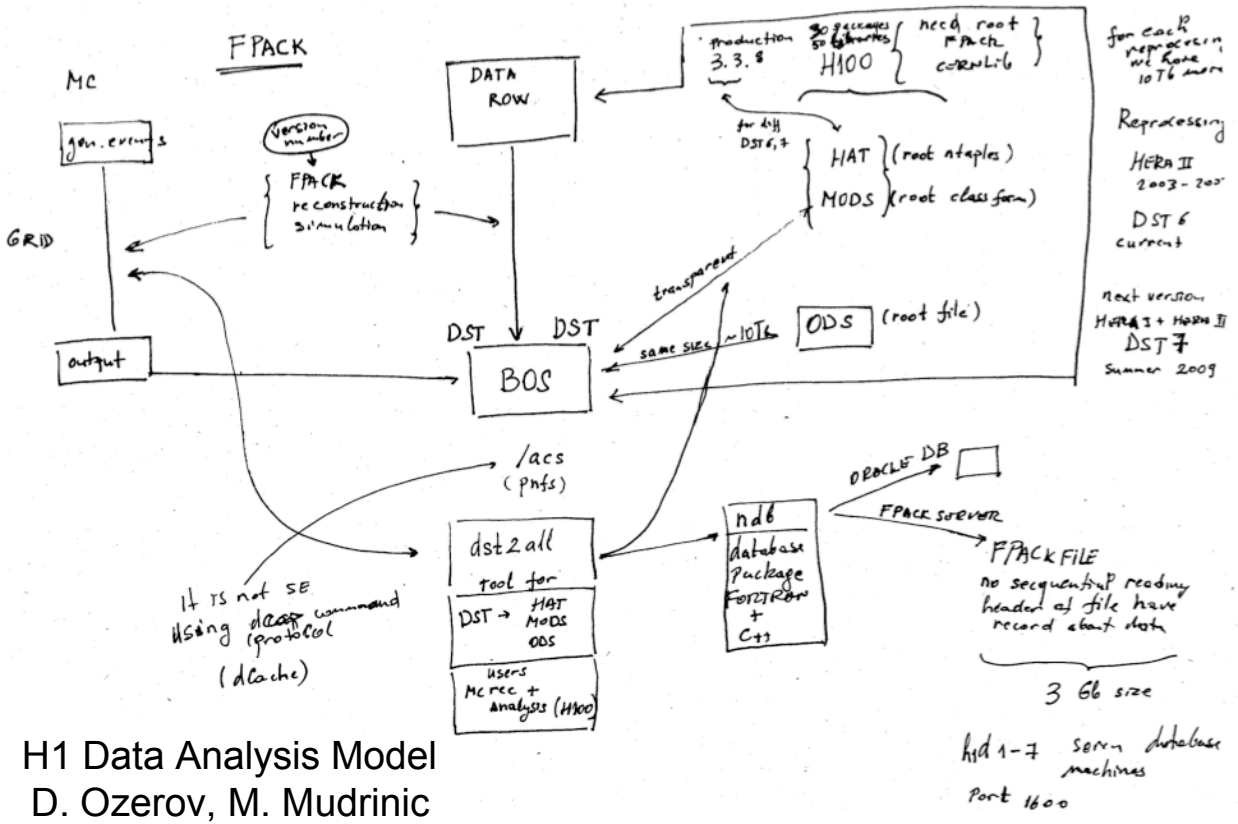
External Dependencies

- ROOT - Geant, Pythia, …
- Compiler - DB (Oracle)
- Grid / Storage Middleware
- OS & Libs ….

↓

Hardware Architecture

- x86 - x86-64
- Power
- Quantum Computers
- …



H1 Data Analysis Model
D. Ozerov, M. Mudrinic

> Will we be able to reproduce the same results after many years of not looking at the data?

- Two hypothetical antipodes involving Virtualization

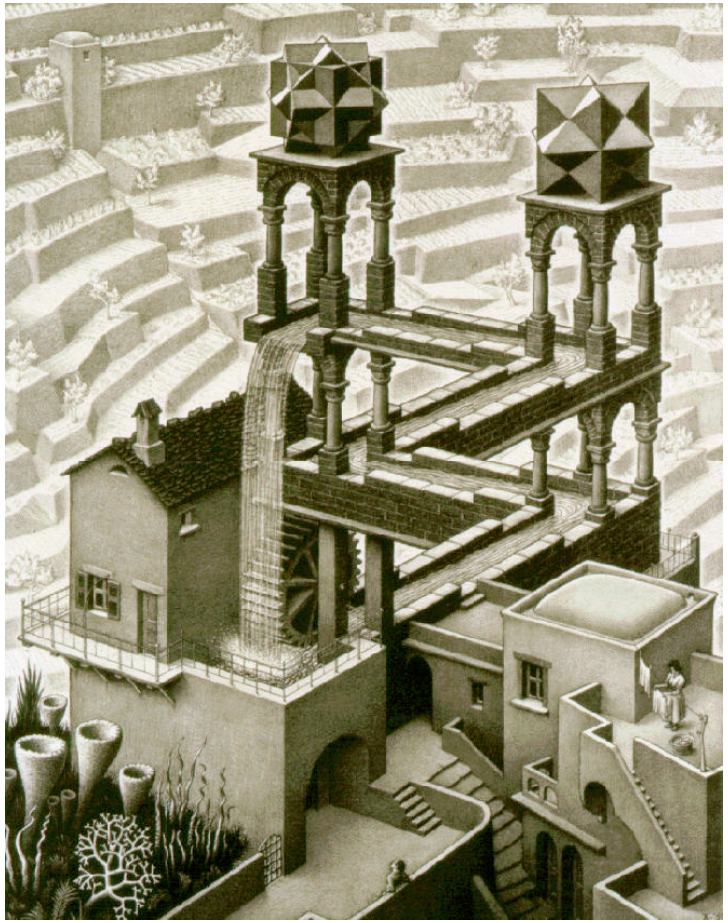
Scenario 1: ”Freezing”

- At the end of the experiment:
  - Datasets closed, final reprocessing done
  - Software framework stable
- Virtual image of the OS with software is done
  - Important: Use a standardized format, like OVF
- Necessary services like Cond DB.:
  - Either integrated into images
  - Or also frozen into another image
- Data access:
  - Either maintain the old protocol/interface
  - Or use high-level protocols
- Running analysis in 20NN (with NN >> 09):
  - Start the whole ensemble of VMs

Scenario 2: Continuous test-driven migration

- Start during running experiment
  - Or even before, when designing software framework
- Define tests
  - In the beginning on MC data, later real data
  - Certain code, running on certain data, yields certain result (e.g. M_{top}=172.4 GeV/c^2)
- Have an automated machinery, which regularly compiles code for different OS / architectures, and runs the tests
- If test fails (e.g. compilation or execution fails, or result divergent)
  - Manual intervention: understand (and fix) problem

→ Such automated tests are usually performed using virtualization techniques and workflows



M.C. Escher's "Waterfall"

Discussion “Freezing” / “test driven migration”

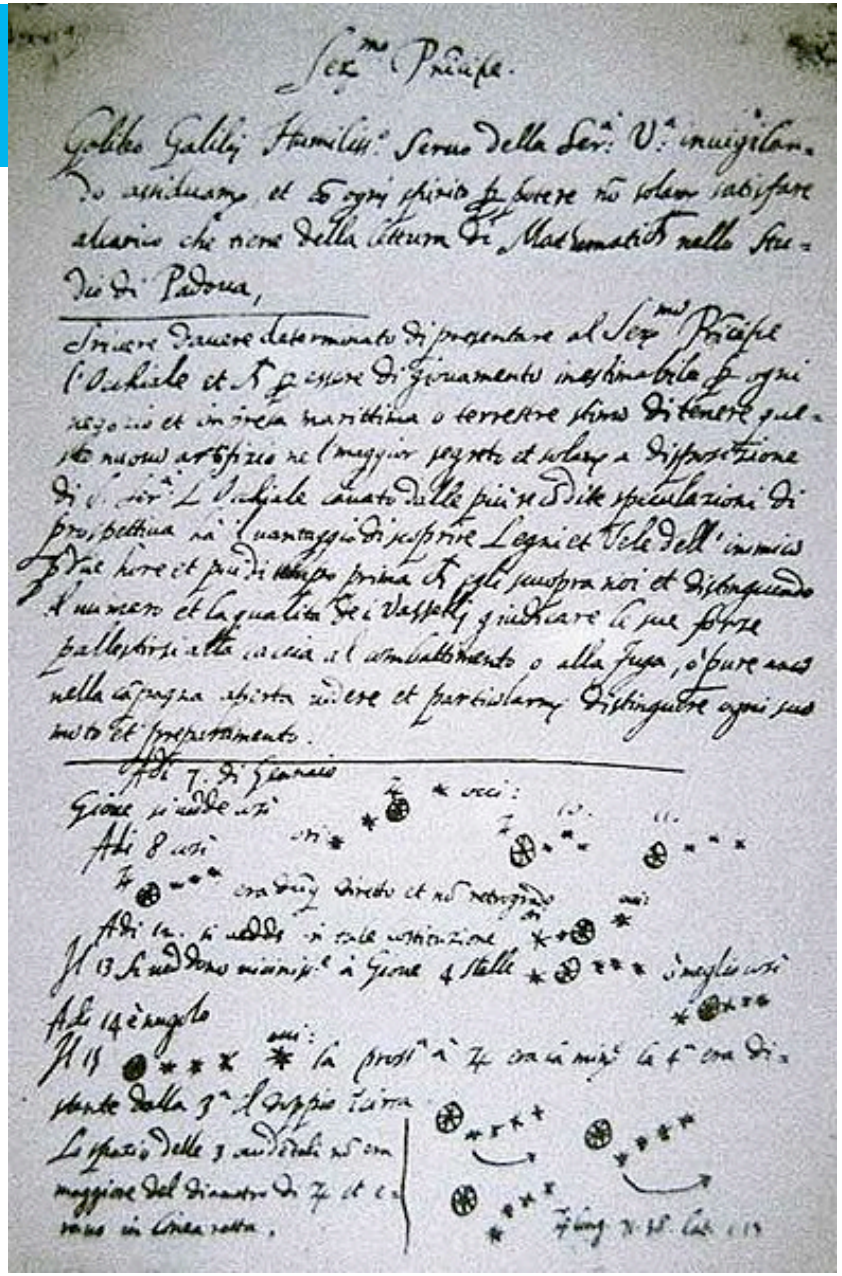
- Pro Freezing
  - One-time effort, very small maintenance outside of analysis phase
  - Also allows software w/o code (but might fail with DRM / licensing issues)
- Pro Test-driven migration
  - Usability and correctness of code is guaranteed at every moment
  - Data accessibility and integrity can be checked as well
  - Fast reaction to standard/protocol changes
  - General code quality can improve, as designed for portability and migration
- Cons Freezing
  - Rely on certain standards and protocols that may evolve
  - Potential performance problems
- Cons Test-driven migration
  - Needs long-time intervention, more man-power and resources needed
  - Some knowledge of the frameworks must be passed to maintainers

And many more aspects of Data preservation

- Who is the owner of the data?
  - This is clear for a running experiment, but afterwards?
  - Authentication / Authorization
- Authorship for after-time analyses
  - Include the original authors / creators of the datasets?
  - Strict internal review process no longer in place
- Coordination among different experiments helpful and needed
  - Proposal to HEP steering bodies

Summary and outlook

- Benefits from data preservation
- Plan early for data preservation
  - Use standards / (e.g. Grid)
- Understanding data in X years is much more than just preserving data
  - Preserve analysis know-how.
  - Make things simple right from the beginning
- Making your data and code understandable will help you
  - In X years
  - Already now:-)



Galileo, Sidereus Nuncius 1610