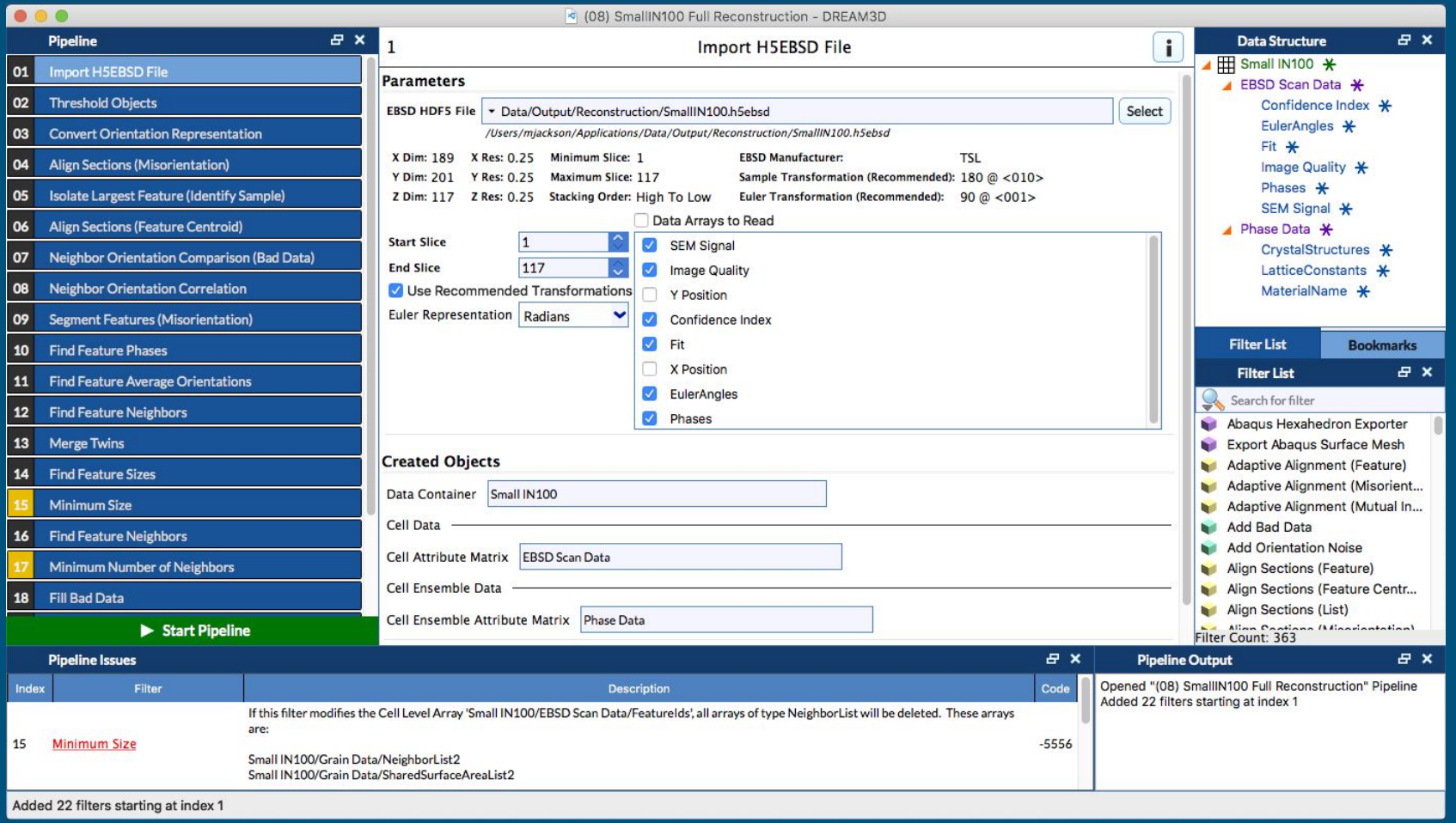
Experiences Integrating HDF5 into DREAM.3D

Mr. Michael A. Jackson
Owner/Developer
@ BlueQuartz Software

DREAM.3D

- Air Force Research Laboratory (AFRL) sponsored project
  - Continuously developed since 2009
  - Unified cross platform framework to allow disperate codes to work together
- Materials Science and Engineering (MSE) Data Analysis
  - Qt5 based GUI application
  - Extensible Framework
- Open Source hosted on GitHub
- Foster collaboration between MSE and engineering/design groups
  - Repeatable workflows
- Portable Data
  - Self describing, Fast I/O, available from desktop to HPC
- http://dream3d.bluequartz.net
- http://www.github.com/bluequartzsoftware/dream3d
- https://my.cdash.org/index.php?project=DREAM3D



DREAM.3D User Interface (6.5 version)

Selecting Open Binary File Format

- Spent a fair bit of time evaluating lots of different possibilities
- 2006 Time Frame
  - Images + sidecar file
  - Binary XML
  - Plain Binary Files + sidecar file
  - A few others which I can't remember
- HDF5 checked most of the boxes that we needed
  - Not a lot (at the time) of built up infrastructure for HDF5
  - Fast I/O, Self describing, Flexible
  - All the reasons that you might already be using HDF5
- In the end, HDF5 saves developer frustration, developer time, creates more consistent files that can be exchanged among research groups.

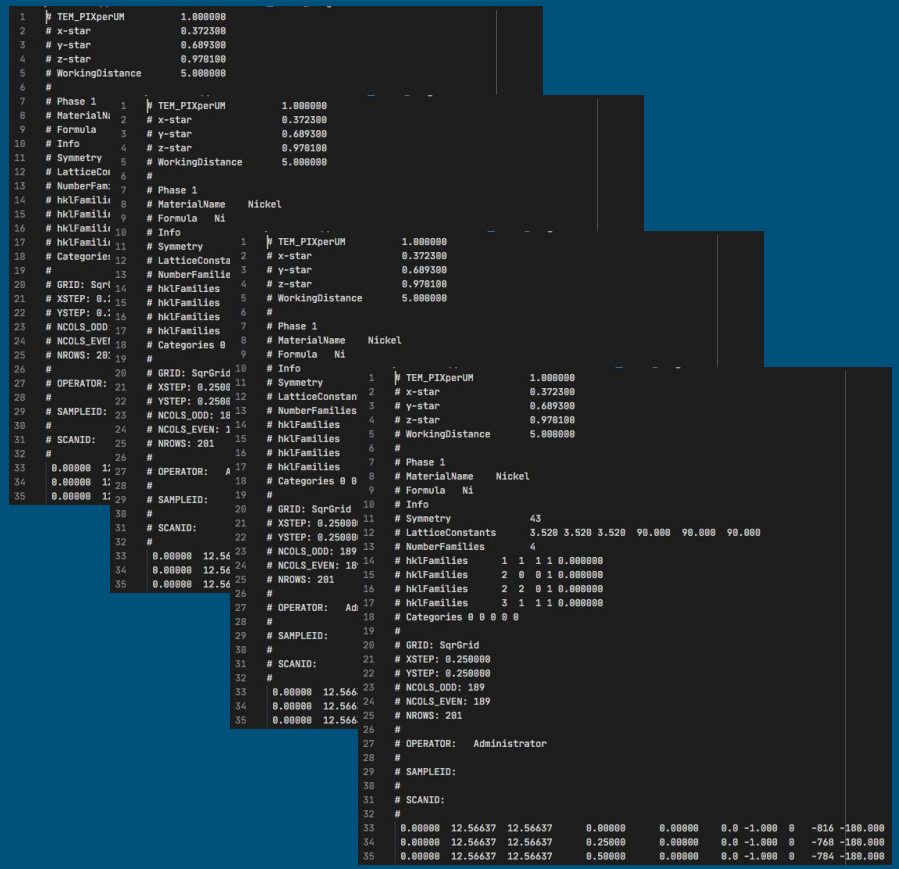
HDF5 & DREAM.3D History

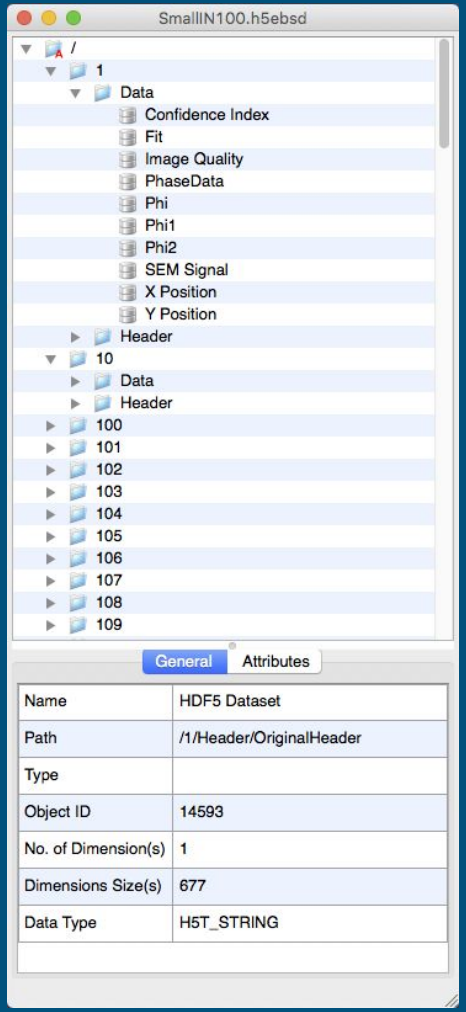
- Started with HDF5 1.6 (2008/2009)
  - Different build systems on each platform, difficult to get correct on Windows/MSVC
  - https://github.com/BlueQuartzSoftware/H5Support
- Moved to HDF5 1.8 (Late 2009)
  - Collaborated with Dr. John Biddiscombe to add CMake support to HDF5 1.8
  - Handed that code off to THG where it has been embraced and maintained since
  - Enables easier integration and use of HDF5 in DREAM3D, and other CMake based projects, from a configuration/compilation point of view
  - Changes to target naming inside of CMake throughout 1.8 series was problematic
- Moved to HDF5 1.10 (Late 2018)
  - Minor code updates in DREAM.3D
  - Generally smooth sailing

H5EBSD: Moving an Industry to HDF5

- EBSD: Electron BackScatter Diffraction
  - Use an electron beam to reveal internal structure of materials
- EBSD equipment vendors all have separate and incompatible file formats
  - Only readable format for external applications is ASCII text
  - Importing ASCII data is slow, prone to failures and has precision (float) issues
  - Advanced users need access to the raw data, not the processed data.
  - Raw data hidden away in proprietary file formats
  - Three main OEMs: EDAX, Oxford Instruments, Bruker
- DREAM.3D already generated HDF5 archives where those ASCII files were converted to a single HDF5 file (.h5ebsd)
  - Faster loads
  - More descriptive
  - Multiple ASCII files converted to single HDF5 file
- https://link.springer.com/article/10.1186/2193-9772-3-4

From Text to HDF5: EBSD Example





H5EBSD: Moving an Industry to HDF5

- Timeline
  - 2013 First conversations with EDAX, example code sent to EDAX
  - 2014 EDAX Updated release with support for writing HDF5 files
  - 2016 Bruker has tool to convert from .bcf (proprietary format) to HDF5
  - 2019 Oxford Instruments starts to use HDF5 as a supported file format

- Positive feedback from EBSD users regarding the changes
  - Easier to I/O data to/from their custom data analysis programs or control systems

Current Trends in Open Source

- Recent trends with some open source companies is to put the source/binaries behind a “wall”.
  - Paid wall, subscription wall, free account wall

- These walls stop automated scripts in their tracks
  - DREAM3D CI broke, DREAM3D SDK Build Scripts broke.
  - BQ mirrored the HDF5 sources/binaries for DREAM.3D’s use. All that tracking information was lost to The HDF Group
  - DREAM3D seriously considered moving away from HDF5
- We need better ways of financially supporting companies that produce open source software

HDF5: Open Discussions

- **The HDF Group moving to more open development**
  - Enlightening discussion on the HDF Forum.
  - Helped to understand the issues that The HDF Group were/are facing
- Repositories hosted on GitHub
  - https://github.com/HDFGroup/hdf5
  - Submit bugs/feature requests/code through the “Issues” area
  - Standard PR (Pull Release) workflow
- Binaries ***easily*** downloaded from www.hdfgroup.com
  - https://www.hdfgroup.org/downloads/hdf5/
- Downloads help HDF Group make a case for continued support from funding agencies

Contributing & Helping HDF5 Ecosystem

- If you are using HDF5 consider the value proposition that HDF5 has brought to your project.
- Consider getting paid support/consulting from The HDF Group
  - However small it might be
- If you have projects/proposals where data storage comes into play, talk to The HDF Group about being a partner/sub on your contract
  - This takes longer term planning and discussions with the groups for whom you work
  - Start those discussions now.
- Most people think that giving back is fixing a bug in code
- Any aspect of development can be a target for your efforts
  - Code, Tests, Documentation, build bots
  - All of these things can help

Takeaways

- HDF5 saves developer time
- Large passionate community of HDF5 users/developers
- Fosters collaboration between research groups
- Be advocates of HDF5 and introduce it into new industries
- HDF5 is not *_free_* to develop, although it is *_free_* to use
- You should consider using HDF5 Group on your next project
- The HDF Group are some of the best developers that I have had the pleasure of interacting with.

Setting the context...

- BlueQuartz Software
  - 3 Full Time, 3 part time
  - Primarily funding streams are from DoD sources
- **All** of our software has been open source
- Budgets are tight and funding is even tighter
- Finding funding is sometimes difficult for your own company, but consider the value proposition of HDF5 and your own project(s)
- Strong advocate for HDF5 since 2005'ish