ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-THPD110

A CLIENT FOR THE ATLAS TIME MACHINE

A. Ramaswamy, K. Bunnell, M. Torres, C. Dickerson, D. Novak,
D. Stanton, B. Blomberg, G. Dunn
Argonne National Laboratory, Lemont, IL, USA

Abstract

The Argonne Tandem Linear Accelerating System (ATLAS) facility at Argonne National Laboratory is a National User Facility capable of delivering ion beams from hydrogen to uranium. The existing tune archiving system, which utilizes Corel's Paradox relational database management software, is responsible for retrieving and restoring machine parameters from previously optimized configurations. However, the Paradox platform suffers from outdated support, a proprietary programming language, and limited functionality, prompting the need for a modern replacement.

The new system is composed of a modular architecture featuring a client, a time-series storage database, and a backend that connects the two. The client was implemented using PySide6, with QML and Python being used together to implement the UI and backend logic, respectively.

INTRODUCTION

At ATLAS, new experiments begin every 3-5 days, with beams created at one of three ion sources continuously being delivered to one of nine target areas. Each startup requires operators to tune the accelerator for the chosen ion species by optimizing many beamline devices, which typically takes 24 hours. Operators start a tune by loading settings from a similar past run using a Paradox-based Tune Archiving System (TAS). However, TAS suffers from two key issues: it is based on outdated software, making maintenance challenging, and features a static user interface that makes basic operations tedious for ATLAS operators. We therefore present a Python-based replacement to modernize TAS, simplify maintenance, and streamline operator workflows.

The TAS replacement, which we call Atlas Time Machine (ATM) [1], is shown in Fig. 1. It enables operators to view and restore historical data as if they were using a time machine. ATM's client plays a critical role in the system as the interface between ATLAS operators and the backend server.

STRUCTURE OF THE ATM CLIENT

The ATM client can be broken up into two main components: the user interface, which is the part of the application that ATLAS operators directly interact with, and the backend interface, which is the part that connects to the ATM server to receive data about historical experiments.

User Interface

The user interface was written in QML rather than Python to allow for a clean separation between UI design and logic to interface with the backend. Operators were deeply involved throughout the development process, providing systematic, ongoing feedback that guided feature prioritization for the new ATM client and identified TAS workflows that were difficult to execute, which we subsequently simplified in the ATM client.

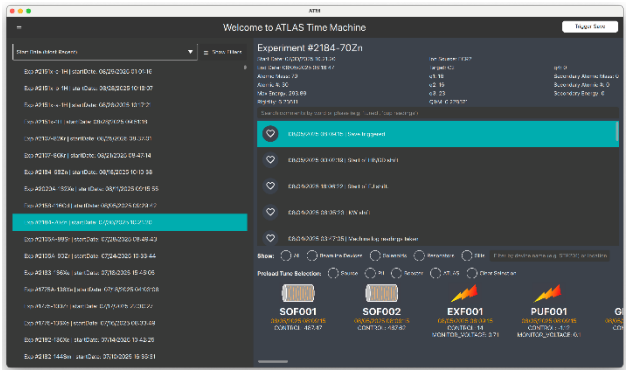


Figure 1: The ATM client displaying data for a particular experiment.

Backend Interface

The backend interface consists mostly of Python (with a small amount of JavaScript) used to interface with the ATM server [2]. This part of the client is crucial in making the UI dynamic, as connections between Python and QML are used to trigger in the display upon receiving new data from the server.

While most of the backend logic is written in Python, there is a small amount of JavaScript, which is the primary coding language for QML logic. The Experiment Filters menu (Fig. 2) has some QML JavaScript to handle date picking and updating slider ranges, while most of the rest of the components have smaller JavaScript code segments mostly to interface with Python functions.

COMPONENTS OF THE ATM CLIENT

Experiment List

One of the biggest improvements that ATM offers over TAS is how the list of experiments is displayed. Figure 1 shows how ATM has a hamburger menu that reveals the experiment selection UI on the left of the screen and allows it to be hidden away when not in use. This allows experiments to be navigated more conveniently than in TAS. Figure 2 shows the experiment filters in the ATM client. This is a pop-up that appears over the home page which contains tools for filtering historical experiments based on beamline parameters and date ranges. Previously in TAS, operators

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-THPD110

spent up to an hour combing through past runs for the closest match. Now, in the ATM client, this process is greatly accelerated through the use of the experiment filtering tool; it takes less than a minute to display experiments that match a given set of criteria (see Fig. 3).

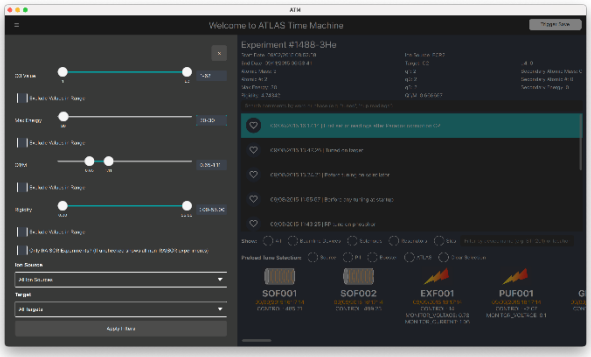


Figure 2: Using the Experiment Filter List to select experiments by specific beamline parameter values. In this case, the filters are set to only show experiments with a max energy reading between 20-30 MeV and a charge-to-mass ratio value between 0.65 and 1.11 (from 0 to 3). Clicking on the Apply Filters button closes the pop-up and updates the experiment list.

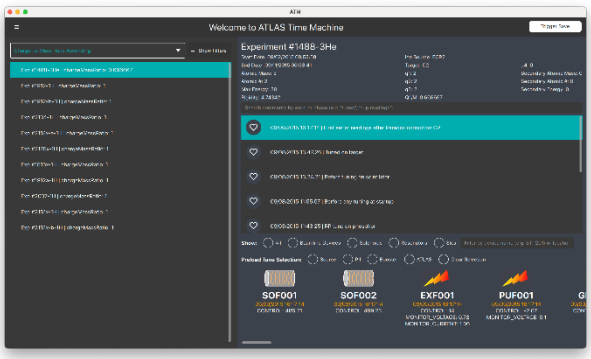


Figure 3: The results of applying the filters from Fig. 2. The Experiment List (here ordered by charge-to-mass ratio in ascending order) is updated to only show those experiments that fit both criteria. An experiment from the list is selected for further inspection.

Experiment Viewer

Once an experiment is selected from the list, an operator can view more information about it in the viewer to the right of the experiment list. This experiment viewer contains information about all beamline parameters, any comments made for the experiment, and a scrollable display of beamline devices which can be filtered by device type (see Fig. 4). Devices from the beamline device list can be selected for a preload tune.

Comment Editor

Once an experiment is selected, operators can select commented saves to make edits to them. ATM features a

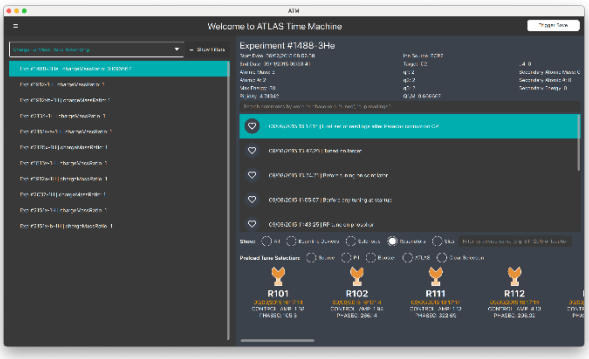


Figure 4: The result of applying beamline device filters to same experiment from Fig. 3, so only resonators and their readings are shown in the beamline.

new rich-text comment editor which is an improvement over TAS's simple text editor. This greatly increases the functionality of comment editing by allowing operators to bold, italicize, underline, and highlight important remarks (see Fig. 5). Support for numbered or bulleted lists also allows operators to record device readings in a more readable manner. Support for certain special characters commonly used in operator comments (the degree symbol and Greek letter mu) is also included. Good lab practices are enforced by only allowing operators to append to existing comments. A timestamp is placed before edits to serve as a record of when edits were made to a historical experiment comment (see Fig. 6).

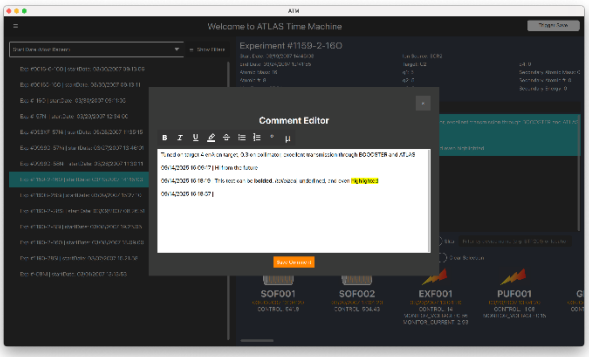


Figure 5: Functionality of the comment editor. While the edits shown here are not particularly useful, they highlight some of the important ways the comment editor can be used by an operator.

Preload Tune Builder

The final major feature in the beta version of the ATM client is the preload tune builder. ATLAS operators typically try to tune the beamline based on parameters from previous experiment data. There are four regions of the beamline that need to be appropriately tuned: Source, PII, Booster, and ATLAS. To do this, appropriate scale factors are applied to devices from the original experiment to calibrate their values for the isotope being used in the new experiment.

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-THPD110

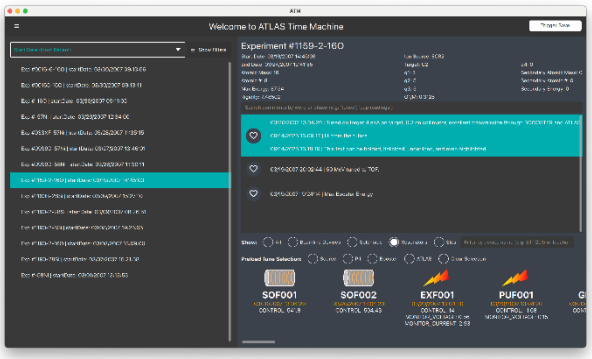


Figure 6: Once the comment is saved, the edits appear automatically. Rich text formatting is not shown in the comment list but rendered when the comment is opened in the editor.

From the "Preload Tune Selection" radio button bar above the beamline device section, an operator can select a region of a historical experiment to use in a preload tune (see Fig. 7). Unlike TAS, ATM's client allows operators to select beamline devices from multiple experiments by region. That is, the Source beamline data for a preload tune could come from an entirely different experiment than the PII beamline data, provided both experiments are applicable.

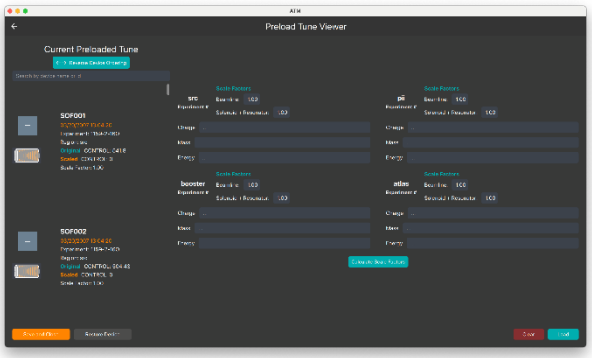
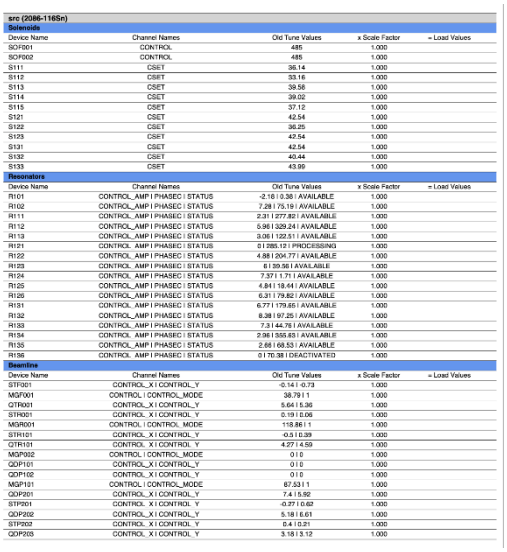


Figure 7: The preload tune viewer, with devices from the source region of an Oxygen experiment (#1159-2-16O) having already been added. Once devices from a region are added, an operator can input specific values for charge, mass, and energy to calculate a scale factor for that region, so the device values are correctly calibrated for the new experiment.

Once an operator is done adding devices, they can click Preload->Save Tune as PDF in the application menu to view a generated PDF report of preload data (see Fig. 8).



src (2086-116Sn)				
Solenoids				
Device Name	Channel Names	Old Tune Values	x Scale Factor	= Load Values
SOF001	CONTROL	485	1.000	
SOF002	CONTROL	485	1.000	
S111	CSET	36.14	1.000	
S112	CSET	33.16	1.000	
S113	CSET	39.56	1.000	
S114	CSET	39.02	1.000	
S115	CSET	37.12	1.000	
S121	CSET	42.54	1.000	
S122	CSET	36.25	1.000	
S123	CSET	42.54	1.000	
S131	CSET	42.54	1.000	
S132	CSET	40.44	1.000	
S133	CSET	43.99	1.000	
Resonators				
Device Name	Channel Names	Old Tune Values	x Scale Factor	= Load Values
R101	CONTROL_AMP1 PHASE0 | STATUS	-2.16 | 0.36 | AVAILABLE	1.000	
R102	CONTROL_AMP1 PHASE0 | STATUS	7.28 | 75.19 | AVAILABLE	1.000	
R111	CONTROL_AMP1 PHASE0 | STATUS	2.31 | 277.82 | AVAILABLE	1.000	
R112	CONTROL_AMP1 PHASE0 | STATUS	0.96 | 329.24 | AVAILABLE	1.000	
R113	CONTROL_AMP1 PHASE0 | STATUS	3.06 | 122.51 | AVAILABLE	1.000	
R121	CONTROL_AMP1 PHASE0 | STATUS	0 | 285.12 | PROCESSING	1.000	
R122	CONTROL_AMP1 PHASE0 | STATUS	4.88 | 204.77 | AVAILABLE	1.000	
R123	CONTROL_AMP1 PHASE0 | STATUS	6 | 39.98 | AVAILABLE	1.000	
R124	CONTROL_AMP1 PHASE0 | STATUS	7.37 | 1.71 | AVAILABLE	1.000	
R125	CONTROL_AMP1 PHASE0 | STATUS	4.84 | 18.44 | AVAILABLE	1.000	
R126	CONTROL_AMP1 PHASE0 | STATUS	6.31 | 79.82 | AVAILABLE	1.000	
R131	CONTROL_AMP1 PHASE0 | STATUS	6.77 | 179.65 | AVAILABLE	1.000	
R132	CONTROL_AMP1 PHASE0 | STATUS	8.38 | 97.25 | AVAILABLE	1.000	
R133	CONTROL_AMP1 PHASE0 | STATUS	7.31 | 44.76 | AVAILABLE	1.000	
R134	CONTROL_AMP1 PHASE0 | STATUS	2.96 | 355.83 | AVAILABLE	1.000	
R135	CONTROL_AMP1 PHASE0 | STATUS	2.66 | 68.53 | AVAILABLE	1.000	
R136	CONTROL_AMP1 PHASE0 | STATUS	0 | 76.38 | DEACTIVATED	1.000	
Beamline				
Device Name	Channel Names	Old Tune Values	x Scale Factor	= Load Values
STF001	CONTROL_X | CONTROL_Y	-0.14 | -0.73	1.000	
MSF001	CONTROL | CONTROL_MODE	36.79 | 1	1.000	
QTR001	CONTROL_X | CONTROL_Y	5.64 | 5.36	1.000	
STR001	CONTROL_X | CONTROL_Y	0.19 | 0.06	1.000	
MSR001	CONTROL | CONTROL_MODE	118.86 | 1	1.000	
STR101	CONTROL_X | CONTROL_Y	-0.5 | 0.39	1.000	
QTR101	CONTROL_X | CONTROL_Y	4.27 | 4.59	1.000	
MQP002	CONTROL | CONTROL_MODE	0 | 0	1.000	
QDP101	CONTROL_X | CONTROL_Y	0 | 0	1.000	
QDP102	CONTROL_X | CONTROL_Y	0 | 0	1.000	
MQP101	CONTROL | CONTROL_MODE	87.53 | 1	1.000	
QDP201	CONTROL_X | CONTROL_Y	7.4 | 5.92	1.000	
STP201	CONTROL_X | CONTROL_Y	-0.27 | 0.62	1.000	
QDP202	CONTROL_X | CONTROL_Y	5.18 | 6.61	1.000	
STP202	CONTROL_X | CONTROL_Y	0.4 | 0.21	1.000	
QDP203	CONTROL_X | CONTROL_Y	5.18 | 5.12	1.000	

Figure 8: Operators can also generate a full report for the preload tune in PDF form. Regions and device types are clearly delineated to allow for efficient lookup. This report is somewhat trivial, since no scale factors were applied to any of the devices, but in an actual report the load values would be displayed at right (load values = old tune values * scale factor)

CURRENT PROGRESS

A beta version of the ATM client was released in August 2025 and is currently being tested by ATLAS operators. Feedback has been overwhelmingly positive, though some unresolved bugs and feature updates preclude its widespread adoption in place of TAS.

Some of the most pressing updates for the client include building an installer for the client, implementing versioning, and adding access controls to prevent unwanted modification of experiment data.

CONCLUSION

The ATM client replaces the static and outmoded Paradox TAS with a modern, dynamic user interface for ATLAS operators. Positive feedback from beta testing shows that ATM represents a significant improvement over its predecessor and is well fit to serve as the future tune archiving system for ATLAS.

REFERENCES

[1] K. Bunnell *et al.*, "Upgrading the ATLAS tune archiving system", presented at ICALEPCS'25, Chicago, IL, USA, Sep. 2025, paper WEMG015, this conference.

[2] M. Torres *et al.*, "A Server for the ATLAS Time Machine", presented at ICALEPCS'25, Chicago, IL, USA, Sep. 2025, paper THPD097, this conference.