LogicMonitor

Alert Tuning for IT Monitoring

My Role

I worked as the lead UX Designer on the Alert Tuning tab at LogicMonitor. This part of the product allows users to customize and manage when they will be alerted for changes in the data they track.

This is a vital aspect of the product because alerting thresholds determine when potential issues are found, and thus how successful our users will be at their jobs.

I took on the project from a previous designer and saw it through to development, where I designed through technical limitations as they arose.

The Problem

The biggest issues with the Alert Tuning tab were that it was confusing to use, difficult to understand how alerting thresholds would behave once they were set, and easy for novice users to mess up alerting preferences by mistake. Some users did not even take advantage of the features at all because of how confusing it was.

Existing Design

Both the Static and Dynamic edit buttons led to the same threshold modal, which was redundant. The language for setting thresholds did not make sense to our users and they often were incorrect with their assumptions about how their thresholds would behave.
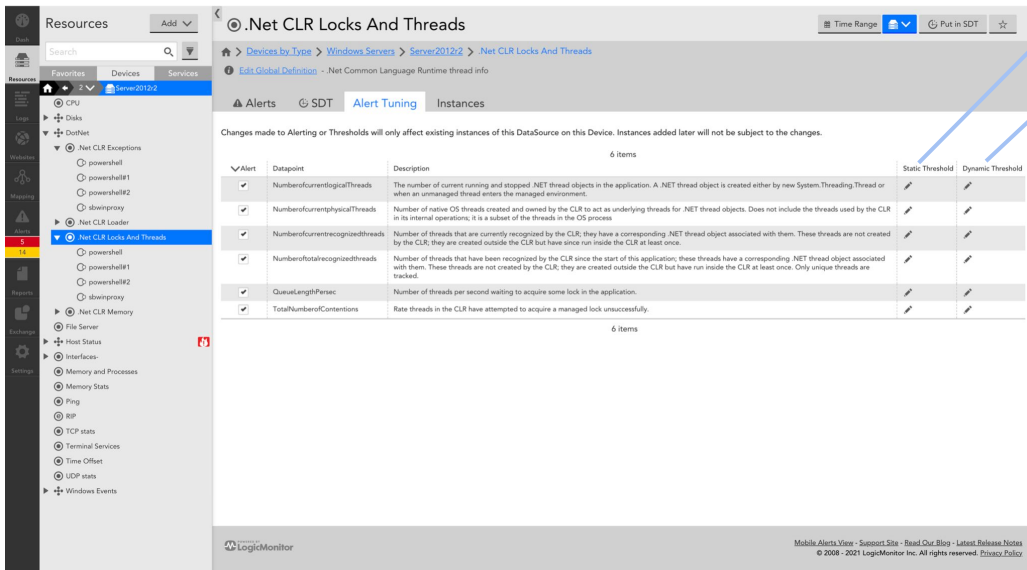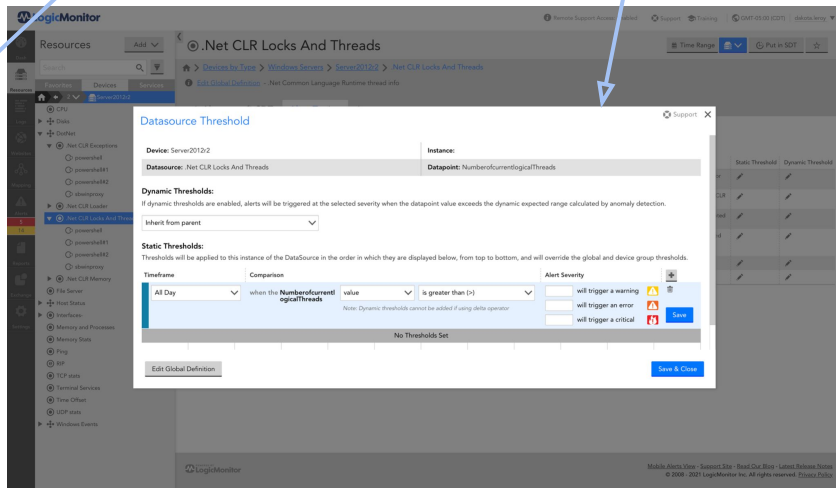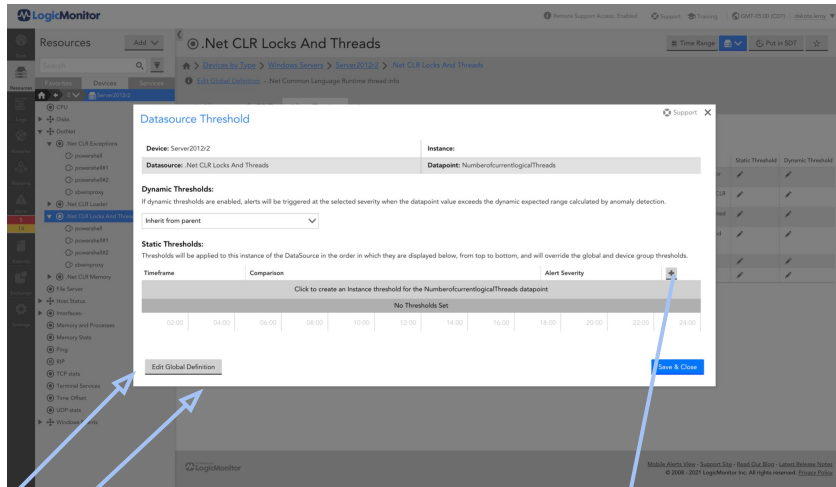
Starting Point

I was given some initial ideas from my team, shown below, but began by interviewing users about how they understood the feature. Their gaps in knowledge showed me that the way thresholds were shown needed to be much clearer. In initial testing, they responded well to visual indicators and more descriptive language.
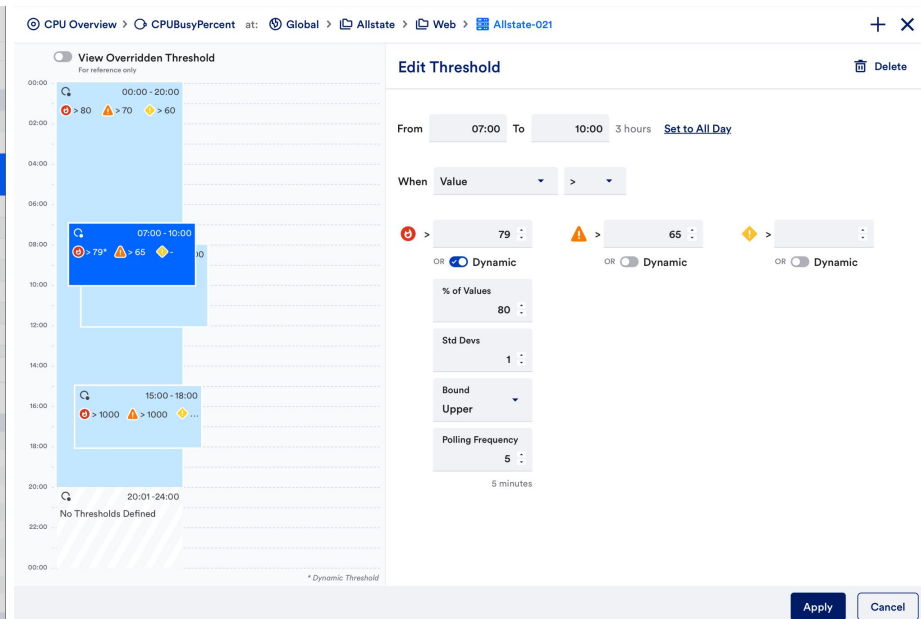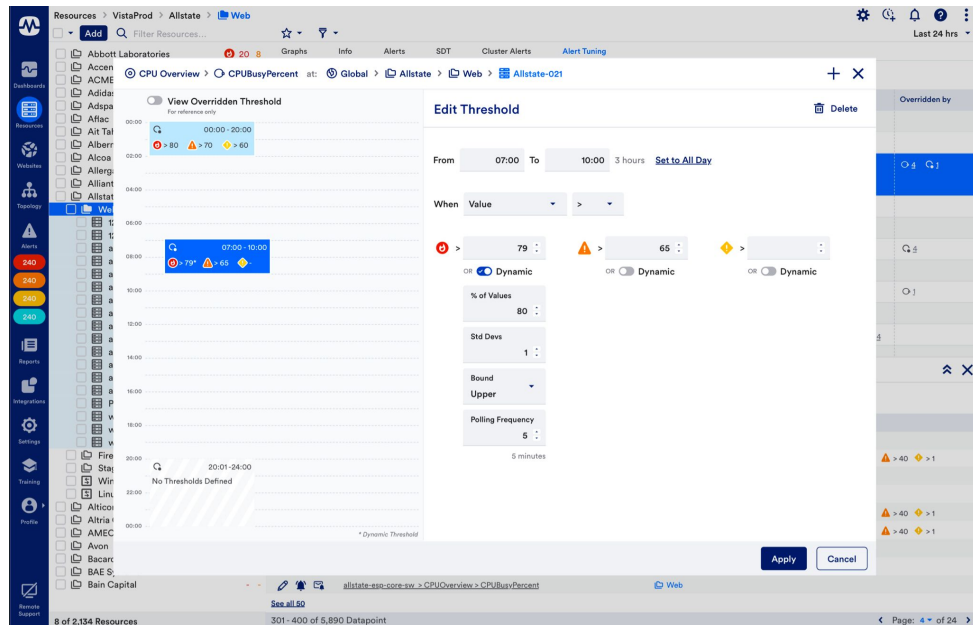
Iterating and Testing

I continued to iterate on my designs based on user feedback. I played around with combining or separating the two threshold types (Dynamic and Static), changing the order for when you can edit a threshold depending on where it is inherited, and many variations of language. The biggest takeaways from my research were that users wanted their alerting behaviors spelled out, visualizations of their alerting thresholds, and a clear representation of the inheritance hierarchies at play.
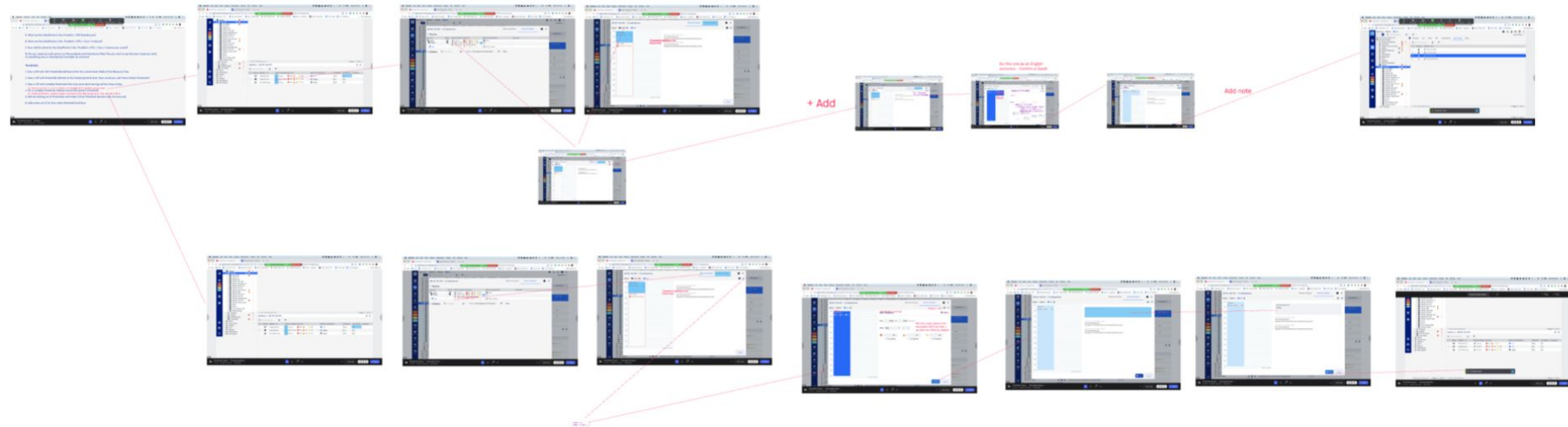
Final Designs

The new main page offers more insight into what is going on at the DataPoint and DataSource levels.

I kept the general table structure of listing DataSources, but removed all low-level information. Users can now see more information within a DataSource at a glance by selecting it and viewing the DataPoints details panel.
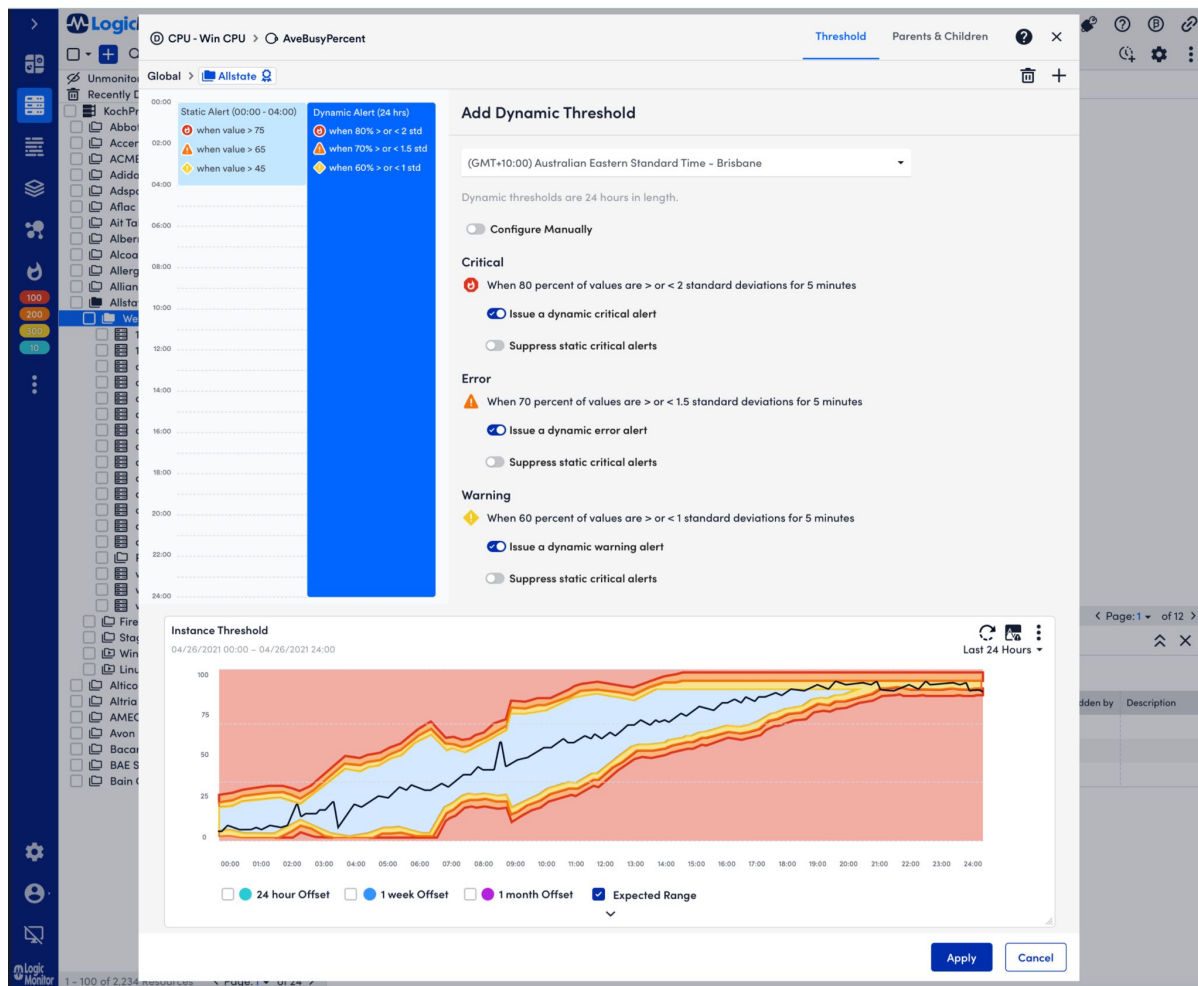
Final Designs

This modal tells users what will happen for each selection with clear descriptions and the blue rectangle visual guides. The image to the right shows a complicated use case of both a Static and Dynamic Threshold overlapping and users were able to understand it with ease given this structure.

The threshold graph shows users how their selection would have behaved during the selected time period.

Final Designs

The Parents & Children view gives more granular details about what thresholds are inherited and where as well as direct links to each of them. There are also brief overviews of how those thresholds are defined.

Users were able to parse through these complex architectures with much more ease by breaking down this information.