



Unlock greater control, visibility & flexibility with self-service tools

Equip your team with the tools they need to move faster, troubleshoot issues in real time, and take control of their integrations

Redox self-service tools put more power in your team's hands

Our suite of **user-controlled features** gives teams the power to troubleshoot faster, fix issues in real time, and manage their integrations independently — all without relying on a Redoxer for support.

With self-service tooling, customers unlock greater



Visibility

Know what's happening with your data in real time

Enabling self-service features:

- Log Inspector
- Alert Rules



Control

Troubleshoot issues independently without filing a support ticket

Enabling self-service features:

- Config Modifiers
- Promotion



Flexibility

Modify how data flows in and out of your organization

Enabling self-service features:

- Translation Sets
- Filters

Benefits



Speed up implementation timelines



Resolve issues faster



Reduce manual, error-prone work



Build greater trust with your customers

What Redox customers say about our self-service tools

"This is a big step forward. Giving us the tools to let our team do this reduces the burden on our engineering team."

— Virtual care provider and platform

"We have a better understanding of the flow of messages and what things are being mapped to. Much easier and quicker to just go look things up and save everyone time."

— Mental health care solution

"This is something we've been hoping for for a long time and gives us enough customization to solve like 90% of use cases."

— Patient engagement platform

Self-service tools

Interested in how our self-service tools work? Here are a few examples of how customers can use these features to move faster, troubleshoot issues in real time, and manage configurations on their own in the Redox dashboard:

Log Inspector	Alert Rules	Config Modifiers
<i>"I want to view what happened to my data."</i>	<i>"I want to set custom alerting rules based on data traffic."</i>	<i>"I want to modify how my data is mapped to fields."</i>
How Log Inspector Works	How Alert Rules Work	How Config Modifiers Work
SCENARIO: I expect to see a value in the [Insurance PolicyNumber] field, but it's always empty.	SCENARIO: If I don't see at least one message come through every hour, I know something's wrong.	SCENARIO: I realize the value I expect in the [Insurance PolicyNumber] field is being sent, but is in a different field.
SOLUTION: Inspect audit logs and quickly track down failed messages and determine if and where the policy number is being sent.	SOLUTION: Set an alert to be notified immediately if no traffic comes through in an hour, allowing you to catch disruptions early.	SOLUTION: Apply a config modifier to remap [Insurance PolicyNumber] to the HL7 field where the correct data is being sent, so data lands where it belongs. Reuse this mapping across any connection or environment.
Translation Sets	Filters	Promotion
<i>"I want to standardize the values used in my data."</i>	<i>"I want to control the data that passes through."</i>	<i>"I want to test in staging, then move assets from staging to production."</i>
How Translation Sets Work	How Filters Work	How Promotion Works
SCENARIO: I received "WI" in the [Patient.Demographics.Address.State] field and need that to show up as "Wisconsin" in my downstream reports.	SCENARIO: We're only live with 3 clinics at the health system right now, but we're receiving additional data from locations that haven't gone live yet.	SCENARIO: I set up a config modifier to remap [Insurance PolicyNumber] to the correct HL7 field, tested it in staging, and now I'm ready to push it to production.
SOLUTION: Create custom translation sets and link to existing connections to standardize this value across connections, providing flexibility in managing diverse coding systems.	SOLUTION: Set a filter to block data from non-live sites to ensure clean and actionable data in your downstream systems.	SOLUTION: Promote assets from staging to production without a manual rebuild, saving time and reducing manual errors.