Staff Books

Light-hearted, friendly guides and talks on various subjects.

- Rix's Browser Extensions
- Notification Methods
- HalpyBOT Dev Update #1
- HalpyBOT Dev Update #2
- HalpyBOT Dev Update #3
- HalpyBOT Dev Update #4
- HalpyBOT Dev Update #5

Rix's Browser Extensions

The Issue

So you probably have heard the words "Manifest v3" being thrown around in regards to an upcoming version of Google Chrome. This has raised a number of concerns among internet watchdogs, with a number of valid complaints being raised about the implementation, which according to the Electronic Frontier Foundation (EFF) will hurt Privacy, Security, and Innovation on the internet. But one more widely talked-about impact is that Manifest v3's implementation by Google will render adblocking extensions inoperable.

Many people utlilze and rely on adblocking software for a number of purposes. Some websites are in fact impossible to navigate without these extensions and tools. While yes, there is an argument to be made about the morality of adblocking software, with various schools of thought of when you should "whitelist" sites or channels, that is a decision that should be left with the user of the software, not the world's largest ad conglomorate. So, what can you do? Public outcry has not been enough to sway Google from their path. Thankfully, there are other options for web browsing that do not have this issue.

Enter FireFox. Yes, good old trusty FireFox. Sure, the software has had its ups and downs, but the current version of FireFox is fast, private, and my personal choice for daily driving. Why should you care about privacy? There isn't one "true" answer, but there are plenty readily available good ones. (Here, Let Me Google That For You) Simply speaking, in my opinion your data is your data and it's not Google's business or anyone else's. The world spent 521.02 billion USD on advertisments in 2021, and incredibly smart people have dedicated their lives to collecting data about you to serve you more topical ads - tracking every click, keystroke, and preference you have on the internet. Sophisticated algorithms can even know more things about you than your best friend or significant other, and all of that data is sold to the highest bidder. Among various technologies that can be used to identify users is Browser Fingerprinting, using various programs and techniques. Want to test how much personal data your browser is leaking and how identifiable it is? You might be surprised. Check here:

https://coveryourtracks.eff.org/

Firefox bills itself as privacy, speed, and security focused. What's nice is that I can also add a number of extensions and heavily customize it for my purposes. I am a fan of online privacy, but that is not to say I am perfect. I still use GMail, Google Search, and sometimes filter pages through Google Translate. But, I have taken steps to limit the amount of data I leak, and honestly, that's a good place to start. Completely De-Googling is often something that online privacy communities fixate on, and shun anyone who isn't willing to make that jump and sacrifice. I'll be the first to admit that Google, the largest ad company on the planet, is so incredibly convenient - but that's not to say that I like all the things their company does. Even taking little steps can help you lead a more secure, faster, cleaner life.

My Extensions

Online Privacy

> Some of these products can break some webpages, meaning that they must be manually disabled on occasion for some sites. In my mind, it's a hassle worth paying. YMMV.

- uBlock Origin (Link): An efficient wide-spectrum content blocker. Easy on CPU and memory.
- WebGL Fingerprint Defender (Link): Defending against WebGL fingerprinting by reporting a fake value.

- Canvas Fingerprint Defender (Link): Defending against canvas fingerprinting by reporting a fake value.
- Decentraleyes (Link): Protects you against tracking through "free", centralized, content delivery.
- Facebook Container (Link): OK, Look... yeah I still have a Facebook account, because if I don't have one I'll never know what my family is up to. At least I can lock it to its own little sandbox and not let it see all my off-site traffic anywhere that has a "login with Facebook" option.
- Neat URL (Link): Removes garbage parameters from most URLs. This is stuff like that utm_campaign garbage... it'll shock you just how much junk is in some links (Looking at you, Amazon!)
- Privacy Badger (Link): Another product from EFF, automatically learns to block invisible trackers slipped in by some sneaky actors.

Website Enhancements

- Enhancer for YouTube (Link): Ever want to control the playback speed, volume level, select playback quality automatically, or control YouTube more than can be done with the stock experience? Check here. You'll never want to go back after using this.
- No Coin (Link): Did you know that come websites use your browser to mine cryptocurrency? That slows down the webpage and causes your computer to use more resources. This extension is designed to block that behavior... mine your own business!
- Outline Sidebar (Link): This navigational aid generates a table of contents by extracting all the headings in the page.
- Reddit Enhancement Suite (Link): A Suite of Enhancements for Reddit... sorta self explanatory. Works best on the old reddit site, but I'm not completely against some of the changes in New Reddit.
- Skip Redirect (Link): Attempts to skip over URL redirects and automatically navigate you to the final URL. 60% of the time it works... every time.
- SponsorBlock (Link): Oh, you thought we were done messing with YouTube? Uses a crowdsourced database of video segments to skip over annoying reminders, intros, sponsor segments, and other stuff you really don't care about. And now a word from our sponsor... Ra-***SKIP***

Utility

> A number of these probably leak some amount of personal data, but I still use them because they're convenient

- TWP - Translate Web Pages (Link): Translate your page in real time using Google or Yandex.
- BitWarden (Link): My Password Manager's Firefox extension
- Grammarly (Link): Me fail english? Unpossible! (Makes my writing suck a little less)
- Mailvelope (Link): Ever want to send encrypted emails? You can with this. Secure, end-to-end webmail with OpenPGP standard encryption. Does require some setup, and both parties to have keys. If you do this you are truly a giant nerd. (I like you.)
- Mendeley Web Importer (Link): The only reason I got my senior thesis done. Probably not useful for you, unless you need to save and cite research papers and the like frequently.

A lot of the extension I use are either niche, overkill, require some tweaking of settings, or all three. Howver, some are simple drop-in tools that I use every day. Combined with neat features like Firefox's DNS over HTTPS, strict inbuilt tracking protection, and HTTPS-Only mode (seriously, if your website doesn't support HTTPS in 2022 what's wrong with you), and other settings you can find in Firefox's simple, easy-to-navigate menus, Firefox has become my favorite browser and has resulted in a much cleaner, ad-free experience on the net. In fact, I donate money to Mozilla every month for the work they do, to help keep Firefox relevant and secure and to avoid giving Google a total monopoly on how you browse the internet.

I hope this has been informative, and if you have any questions, please feel free to reach out! Here's to a faster, more secure future together.

Notification Methods

Hello, Seals!

A few weeks ago, I asked you to fill out a survey about how we currently provide notifications for cases. While we're still investigating possibilities, I wanted to just provide a few points of comment on what the early results say. I wanted to thank the 18 of you who filled out the survey, as it is only through your feedback that we can chart the best path forward.

The majority of you rely on Discord's own notification system in order to know when cases come in. This was a truth that we were already aware of, but it is good to have solid data on this. We'll continue to support Discord pings in the future.

A number of you had expressed interest in a dedicated Hull Seals app for various platforms, or integration with an existing notification service, or even sending SMS notifications. Unfortunately, we do not have the budget to offer a SMS-message based notification signup service. However, we are taking your feedback into account and seeing if we can integrate a push-based notification framework into the next iterations of HalpyBOT and services. We are also beginning preliminary investigations into potentially developing a Hull Seals API which could allow further expansion and integration with other services in Elite.

A few of you expressed frustration with case pings being moved to Trained Seals instead of Case Notify, or the lack of specific breakdown by platform for cases. I wanted to spend a moment and expand upon why these decisions were made.

The lack of segregation between platforms for pings is because the ping serves not only for Seals for the specific case but also for Dispatchers. Any seal can (and should!) Welcome a client. Dispatching doesn't *require* specialized training, and while we offer Dispatch training it is not mandatory to take the course in order to take the hat.

The migration of Case Notify to Trained Seal pings came about after a number of cases in short order across all platforms failed because no seals responded - or sometimes, not even welcoming or dispatching the client. In a few of these instances, multiple seals were active in chat and seemingly ignored the case! This wasn't an acceptable situation, and unfortunately the only recourse we had was to widen the net of who was pinged. Without responding to cases, the Seals don't exist. It has long been our policy that if we get back to a state where cases are responded to reliably and our numbers grow to a level that the staff are comfortable with, we will revert back to using the Case Notify system. Unfortunately, we are not there and cannot afford to cast this smaller net.

Thanks for reading, and as always, please let me know if there are any questions!

Rixxan

HalpyBOT Dev Update #1

Heya folks, time to don my Cyber hat for a bit.

It's been a little bit since the last update to HalpyBOT, and we wanted to give you an update on what we've been up to over in the Cyber world.

Over the last little bit, we've been focusing on cleaning up and improving a number of aspects of the core bot, some of which haven't received any polish since version 1.0 of the bot! We've just finished what we think is the last of the core improvements to existing code, and are hard at work starting development of the new Case Manager system (the long awaited case management tool suite!)

Here's a short list of most of the major improvements we've made to the bot - with thousands of changed lines so far:

- Established a new Config mechanism for increased stability and cleaner usage.
- Added a number of automated reviews to our code process to help prevent bugs.
- Updated some of the CLIs to use the updated database mechanisms.
- Refactored the CLIs to be more easily understood and maintained.
- Updated some of the CLIs to use the HalpyBOT Coordinate System.
- Simplified some CLI methods.
- Migrated HalpyBOT to use the Poetry package manager for easier maintainability.
- Removed the Delayed Case mechanism to be replaced by the upcoming Case Board mechanism.
- Added a Dice Roller command (And another command, Sorcerers and Wizards be careful!).
- Migrated HalpyBOT's global entry point to inside the HalpyBOT directory.
- Added a number of missing Docstrings for code documentation.
- Integrates the BotClient and TCP Server into a single thread for memory stability.
- Migrates a number of module-level independent globals to the HalpyBOT main class for easier maintainability.
- Improves the Help command logic to be more responsive.
- Simplifies the Drill module logic to remove duplicated code fragments.
- Completely refactors the EDSM commands system from the ground up.
- Refactors the Staff Notification module to fix a number of potential bugs.
- Updated the performance of the Manual Case system.
- General logic and performance enhancements.
- Updated the offline mode system to fix a few possible issues.
- Completely refactors the case announcement system.
- Removes the Twitter module.
- Improves the logic of the CommandHandler system.
- Improves error output when problems occur for better tracking by Staff and Users.
- Reworks the database check system to better maintain database connections.
- Updates the Coordinate system logic.
- Adds new Language Code, Database Connection, and Command file handlers to the HalpyBOT Class.
- Completely reworks the Client Configuration system for the botclient.
- Improved the message-sending system to avoid sending blank new lines.
- Updated additional database linkups to use improved logic.
- Adds a timed, repeating task system to help automate some tasks.
- Fixes a possible race condition on cross-thread reads and writes.

We're super proud of our work, and hope that you will like it too once version 2.0 is ready. If you want, you can even track our progress at https://github.com/HullSeals.

HalpyBOT Dev Update #2

Hey there folks!

With the ongoing process of HalpyBOT 2.0, I wanted to start developing a series of small dev blogs which I am hoping to be able to publish on an approximate monthly basis as to the developments that the Cyber team are cooking up in relation to SACM and HalpyBOT. Last month, I provided a small list of changes and features that we'd added, but wasn't very descriptive as to what some of these changes entailed. So, before we can take steps forward, let's step back for a moment and look at some of the more important chagnes in greater detail.

The Help Command

One of the things we've improved in HalpyBOT 2.0 is the contextual help system. Now, not only will the help

This even

allows you to query Help commands based on their aliases!

Error Handling

We've also put great investment into helping the error diagnostics process. Previously, if an error occurred, HalpyBOT would simply emit that an error happened, but wouldn't tell you, the user, what went wrong. In 2.0, we've revamped the error handling system to give you an easy-to-remember, descriptive code to give to our Cybers to not only give us an idea of what has gone wrong, but also something that is easier to search in our logs.

Error types you might see include Harbor, Walrus, Bearded, Leopard, or Ringed. No prizes for guessing what these are types of ;)

Most of the other 2.0 changes we've made so far aren't going to change what you see, and are all behind-the-hood types of changes, which you can read about briefly in The Previous Update. But for now, let's jump into what's been cooking recently!

Case Shouts

One of the most common aspect of HalpyBOT that Seals will see is the "shout" mechanic - where HalpyBOT informs us all of a new ~~settlement which needs our help~~ CMDR in need of rescue. This is one of the aspects of 2.0 which the Cybers were keen to improve. Currently, the Shout function is very basic, and includes a lot of hard-to-parse information. We've tried to improve the mechanic in several ways, including the reintroduction of Platform Short Codes, and clearer definitions as to what each thing means.

What are Short Codes?

Short codes are two to three-letter abbreviations we've used to try and improve quick, at-a-glance understanding of what a case's platform is. Some of you may remember this from the early days of HalpyBOT before the PC Split, or others might recognize a similar mechanic in other organizations. When short codes are combined with the keyword CASE, they become easy-to-understand word highlights which can be set up in almost all IRC clients.

So, instead of `PC - OdysseyCASE`, HalpyBOT will shout out something like `PCOCASE` - So much easier to parse! Common abbreviations will include:

- PCOCASE
- XBCASE
- PSCASE
- PCHCASE (For PC-Horizons (LEGACY))
- PCLCASE (For PC-Horizons (LIVE))

In addition, we're improving the Platform printouts in the main case tracking section to be more succinct and clear as to what they are referring to. Here's an example of some of the new case ping blurbs.

```
HalpyBOT[Dev|Rixxan] xxxx PCOCASE -- NEWCASE xxxx
HalpyBOT[Dev|Rixxan] CMDR: InHoomansLeftEar -- Platform: ODYSSEY
HalpyBOT[Dev|Rixxan] System: SOL -- Hull: 25
HalpyBOT[Dev|Rixxan] xxxxxxxx
HalpyBOT[Dev|Rixxan] System exists in EDSM, 0.0 LY South of Sol.
HalpyBOT[Dev|Rixxan] xxxx XBCASE -- NEWCASE xxxx
HalpyBOT[Dev|Rixxan] CMDR: InHoomansLeftEar -- Platform: XBOX
HalpyBOT[Dev|Rixxan] System: SOL -- Hull: 25
HalpyBOT[Dev|Rixxan] xxxxxxxx
HalpyBOT[Dev|Rixxan] System exists in EDSM, 0.0 LY South of Sol.
HalpyBOT[Dev|Rixxan] xxxx PSCASE -- NEWCASE xxxx
HalpyBOT[Dev|Rixxan] CMDR: InHoomansLeftEar -- Platform: PLAYSTATION
HalpyBOT[Dev|Rixxan] System: SOL -- Hull: 25
HalpyBOT[Dev|Rixxan] xxxxxxxx
HalpyBOT[Dev|Rixxan] System exists in EDSM, 0.0 LY South of Sol.
HalpyBOT[Dev|Rixxan] xxxx PCHCASE -- NEWCASE xxxx
HalpyBOT[Dev|Rixxan] CMDR: InHoomansLeftEar -- Platform: LEGACY_HORIZONS
HalpyBOT[Dev|Rixxan] System: SOL -- Hull: 25
HalpyBOT[Dev|Rixxan] xxxxxxxx
HalpyBOT[Dev|Rixxan] System exists in EDSM, 0.0 LY South of Sol.
HalpyBOT[Dev|Rixxan] xxxx PCLCASE -- NEWCASE xxxx
HalpyBOT[Dev|Rixxan] CMDR: InHoomansLeftEar -- Platform: LIVE_HORIZONS
HalpyBOT[Dev|Rixxan] System: SOL -- Hull: 25
HalpyBOT[Dev|Rixxan] xxxxxxxx
HalpyBOT[Dev|Rixxan] System exists in EDSM, 0.0 LY South of Sol.
```

The Case Board & Case Class

The core feature of SACM, upon which all other aspects will rely: the Case Board is the basic building block which will track all the cases we have active at any given time. This will create, close, and modify rescues while they're in our IRC systems, and will be the main interface for most folks with SACM systems. Ultimately, this is something that's hard to visualize, but it's been one of the most complex classes we've had to build for the bot.

The Case class represents in the bot a single rescue and all of the different trackable aspects of it. From responding seals to hull percentage to coordinates of a planet, this class can track it all. This versatile class will be key in making sure that you can get your limpets on target reliably and quickly.

Board Listing

Of course, simply having the board in software is no good without some way to actually interact with it. While yes, we are planning on having an external webpage from which you can view the status of several cases, we are always trying to make sure that as much functionality as possible can be done inside of HalpyBOT without relying on external tools. Given this, we're working on developing a manner to interact with the case board entirely from within IRC.

```
           Rixxan ^loadboard
HalpyBOT[Dev|Rixxan] Debug Data Loaded!
           Rixxan ^listboard
HalpyBOT[Dev|Rixxan] Here's the current case board:
HalpyBOT[Dev|Rixxan] Case 1: Client: Bob, Platform: ODYSSEY, Type: Unknown, Status: ACTIVE
HalpyBOT[Dev|Rixxan] Case 2: Client: Larry, Platform: ODYSSEY, Type: Unknown, Status: ACTIVE
HalpyBOT[Dev|Rixxan] Case 3: Client: John, Platform: ODYSSEY, Type: Unknown, Status: ACTIVE
HalpyBOT[Dev|Rixxan] 3 Cases on the Board.
```

Misc Utils

As we've built out the core functionality of the Case Board and SACM, we're working to add a number of more useful utilities to HalpyBOT. Most of these have been user-requested features, such as how long ago the last case was (Careful, don't say the Q word!) and more. We're always looking for feedback as to what sorts of features and utilities you'd like to see in HalpyBOT.

Here we've got an example of the new Last command, which does what it says on the tin.

```
           Rixxan ^last
HalpyBOT[Dev|Rixxan] Last case was 1 week 2 days ago.
```

We've also developed a small function to help us keep track of case clients when they join, leave, or quit the IRC system. Here's an example of how that system will work:

```
           Rixxan ^newcase Torngate 1 69
HalpyBOT[Dev|Rixxan] New case started: Board ID 2
                   * Torngate (a28000000@Rixxan.seal.hullseals.space) has joined
HalpyBOT[Dev|Rixxan] Client Torngate connected successfully. Welcome! Please wait for a dispatcher to respond to your case. (Case 2)
                   * ChanServ sets mode +q on #bot-test Torngate
                   * ChanServ gives channel operator status to Torngate
           Rixxan ^debugwelcome 2
HalpyBOT[Dev|Rixxan] [DEBUG] case 2 set to Welcomed.
                   * Torngate (a28000000@Rixxan.seal.hullseals.space) has left
HalpyBOT[Dev|Rixxan] Client Torngate left the channel. (Case 2)
                   * Torngate (a28000000@Rixxan.seal.hullseals.space) has joined
HalpyBOT[Dev|Rixxan] Client Torngate reconnected. Welcome back! (Case 2)
                   * ChanServ sets mode +q on #bot-test Torngate
                   * ChanServ gives channel operator status to Torngate
```

This will help Clients know that they have connected successfully, and will also help us track when clients lose connection or quit the server in a recordable, reportable format.

Other Changes

Some changes we make don't really have a big, flashy, showy aspect to them, and simply help us ensure that our bot will continue to function as stable as possible into the future. I won't bore you with code snippets, but here's a quick rundown on some of the other changes that have happened recently:

- Created more internal fuzes to help prevent catastrophic failures and easier debugging.
- Updated the announcer template to allow for more fields in Announcement loading data.
- Hands over internal timekeeping from DateTime to Pendulum.
- Reworks the WHOIS module to use the new Seal class.
- Refactor a number of dataclasses to use the new attrs.define decorators.
- Moves the MainAnnouncer from a module global to the main botclass.

- Establishes a number of supporting classes, enums, and datamethods to support the Case class.
- Fixes a crash that could occur if certain datafiles were missing.
- Simplifies some REGEX in the bot to reduce potential false positives.
- Rework the WHOIS module database link to use more reliable mechanisms.
- Fixed a bug that could occur when using the database link from the webserver.
- Added a number of new tests to assist in code validation.

Of course, we're constantly tweaking and improving things, so keep in mind that the images you see here might not be reflective of the final product. We're looking forward to continuing to show off more of 2.0 as it draws near! You can watch our progress and review the code yourself here, if you want:

https://github.com/HullSeals/HalpyBOT

Until next time!

HalpyBOT Dev Update #3

Wow, another month already? Guess that means it's time for an update on the progress of HalpyBOT!

It's been a very productive month, with many of the core features of HalpyBOT 2.0 and SACM beginning to come into focus. I'd just like to ask for your attention as the flight attendants demonstrate the safety features of this HalpyBOT. In the event of a water landing... wait that's not right.

Over the last month, among other things, we've finalized our first passes on 17 new case management commands, and also gone about re-ordering the location of some sections of the code. While that latter part may seem boring, it's actually incredibly important to the longterm maintainability of the bot. We've also spent a lot of time refactoring and improving logic paths all over the bot, which will result in a faster, less memory needy program overall. With over 70 commits across 50 files in HalpyBOT, and almost 1,500 unique line changes, it's been an incredibly busy month for our Cybers.

Circular Imports Can Bite My Tail ~ Rik079

Throughout the development of HalpyBOT, we've come across a number of places where errors might occur. This might be a problem with EDSM, or with our connection to the database, or much more. For each of these, we've had to develop a series of error handlers - often by creating new Exception types and handling each of them accordingly. Historically, we've written each exception type in the file where it's most likely to occur, and this has mostly worked. However, if we need to use the exception outside of that file, we've had to import it - which gives us issues of circular imports.

Circular imports can occur when two or more modules mutually depend on each other. If this happens, the bot can't load either file, and the whole situation comes tumbling down. This has been an issue we've had to increasingly deal with, coming up with more and more creative solutions in order to avoid it. To fix it, we've finally moved all of these exceptions to their own file, which seems like we should have done from the beginning... but hey, better late than never! This has vastly simplified our code in some places, and making it less likely for these sorts of errors to occur.
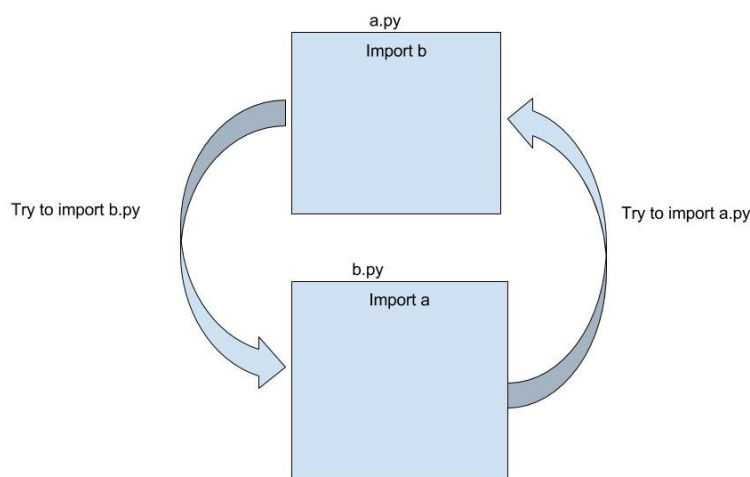
Case Creation

The Announcer now automatically loads in a case to the case board! This means that from the moment the case ping goes out, you can refer to a case by a given ID number, and can manage so many aspects of the case! In the future, we'll be setting you up to even add responders and issue facts - all from the power of your keyboard in IRC. We've also updated our case printouts to make case details look more friendly and readable! We even help

predict if a user's IRC name will be different from their CMDR Name.

A case ping with Case ID and Expected IRC User

Case Printing

Ultimately, the Cybers feel like it should be our goal to ensure that you never have to rely on any external services to respond to a case. While many of our new features are built with the long-awaited Dispatch Board in mind, we're also making sure you can manage every detail of a case from IRC - and that includes printing out the status of a case. To help with that, the bot has the ability to print out the details of every part of the case in DMs with you for quick reference - just ask!

The Board Listing - You can also filter by Platform or Case Type!

The Case Listing - Note how you can reference the case by ID number OR CMDR Name! How Convenient!

Case Updating and Management

Probably the single biggest feature introduced this month is the Case Management Mechanic. This allows us to track and update several key aspects of the case from the moment HalpyBOT shouts the case out onward. While several key aspects of this system were planted in the last update, much of that framework has now evolved into a fully armed and operational case board. Notably, HalpyBOT now actively takes notes on several key case actions, which will (in the future) be automatically fed into case paperwork - making your job easier!

Here's some of the new commands added this month:

Case Details

A number of details about a case can be changed, which allows cases to be significantly altered without needing a duplicate case entered. These include the Client's IRC Name, System, CMDR Name, Hull %, Platform, or even if the case is Black, Blue, KingFisher, or Seal. If it's a detail about the case's facts, it can be changed. Most of them follow the same format, so we won't list them all here.

```
           Rixxan  ^system 2 Delkar
HalpyBOT[Dev|Rixxan]  Client System for case 2 set to 'DELKAR' from 'SOL'
HalpyBOT[Dev|Rixxan]  The closest landmark system is Sol, 83.11 LY SW of DELKAR.
           Rixxan  ^ircn 2 Rik079
HalpyBOT[Dev|Rixxan]  IRC Name for case 2 set to 'Rik079' from 'InHoomansLeftEar2'
```

You can even rename the CMDR of a case, if they mistype their own name.

```
           Rixxan  ^rename
HalpyBOT[Dev|Rixxan]  Use: ^rename [Board ID] [New Client Name]
HalpyBOT[Dev|Rixxan]  Rename the client for the given case.
           Rixxan  ^rename 2 Rixxan
HalpyBOT[Dev|Rixxan]  Client for case 2 set to 'Rixxan' from 'InHoomansLeftEar2'
```

Of course, all of this will be recorded by HalpyBOT.

```
           Rixxan  ^listcase 2
HalpyBOT[Dev|Rixxan]  Here's the self listing for Case ID 2:
HalpyBOT[Dev|Rixxan]   General Details:
HalpyBOT[Dev|Rixxan]     Client: Rixxan
HalpyBOT[Dev|Rixxan]     System: DELKAR
HalpyBOT[Dev|Rixxan]     Platform: ODYSSEY
HalpyBOT[Dev|Rixxan]     Case Created: 11 minutes 19 seconds ago
HalpyBOT[Dev|Rixxan]     Case Updated: 44 seconds ago
HalpyBOT[Dev|Rixxan]     Case Status: ACTIVE
HalpyBOT[Dev|Rixxan]     Client Welcomed: No
HalpyBOT[Dev|Rixxan]     IRC Nickname: Rik079
HalpyBOT[Dev|Rixxan]  Case Details:
HalpyBOT[Dev|Rixxan]     Hull Remaining: 25
HalpyBOT[Dev|Rixxan]  Responder Details:
HalpyBOT[Dev|Rixxan]     Dispatchers: None Yet!
HalpyBOT[Dev|Rixxan]     Responders: None Yet!
HalpyBOT[Dev|Rixxan]     Notes:
HalpyBOT[Dev|Rixxan]        Client System set to DELKAR from SOL by Rixxan at 01:02:30
HalpyBOT[Dev|Rixxan]        IRC Name set to Rik079 from InHoomansLeftEar2 by Rixxan at 01:02:34
HalpyBOT[Dev|Rixxan]        Client Name set to Rixxan from InHoomansLeftEar2 by Rixxan at 01:05:58
```

Notes

Did you notice we had custom notes in an earlier screenshot? Why wait for your paperwork to get started on notes! HalpyBOT will automatically take notes for you for some details, such as changing case details, but you can also manually add, view, edit, or delete note entries straight in IRC before you even load your paperwork on the website.

```
HalpyBOT[Dev|Rixxan]  The closest landmark system is Sol, 83.11 LY SW of DELKAR.
           Rixxan  ^addnote 1 Did you know IRC Nicknames can't start with numbers? That's why we added CMDR_ to the front of it.
HalpyBOT[Dev|Rixxan]  Notes for case 1 updated.
```

Made a mistake? You can even update your notes.

```
           Rixxan  ^editnote 1 1 I'm an updated note!
HalpyBOT[Dev|Rixxan]  Editing line 'This is an example of a custom-added note! - Rixxan (2023-04-10T00:57:54.273195+00:00)' to the provided text.
HalpyBOT[Dev|Rixxan]  Notes for case 1 updated.
```

Updated Client Tracking

Last update, we introduced client tracking - watching for clients to join or leave the server. With the tracking of the Welcome command, we've made this more granular, with the message sent to a client on joining contingent on if the case has been Welcomed or not. It's these little things that will help clients relax and feel at home in our system.

HalpyBOT[Dev|Rixxan] Client Torngate left the channel. (Case 2)
                   * Torngate (42cebb5d@Rixxan.seal.hullseals.space) has joined
HalpyBOT[Dev|Rixxan] Client Torngate connected successfully. Welcome! Please wait for a dispatcher to respond to your case. (Case 2)

More Granular Web Requests

Historically, if the webserver for HalpyBOT received input it couldn't understand, it assumed that something was wrong, and issued an HTTP 500 error. We've taken steps to update this, and now have specific catches to return more specific error codes if Kingfisher coordinates are malformed, or the case already exists on the case board, which will also prevent duplicate case pings in IRC. This will help make case pings more informative and help prevent duplicate cases from being submitted.

The Minor Things

Some things just don't really have anything to talk about, other than saying that they happened. However, many of these are minor bug fixes, Quality of Life enhancements, or other general updates that are worth at least mentioning. We're always hunting down and fixing bugs as we find them, and working to make the best version of HalpyBOT we possibly can.

- Updated the configuration system to specify which channels are Rescue channels.
- Added Local Case ID's to the case announcement.
- Fixed a bug where the bot's user agent was missing a).
- Rearranged some file imports to be more in line with PEP8 Code Standards.
- Added a number of inline code annotations to help with future development.
- Simplified a few logic expressions.
- Moved a few functions around to be more logically accessable.
- Fixed a bug that could result in the edit fact command crashing.
- Fixed a bug that could break the Ping command.
- Fixed a bug that could cause the join channel command to fail silently.
- Updated the logic of the WHOIS Command.
- Added an "Unknown" Platform Fallback if something goes weird.
- Re-ordered the board isinstance checks to not fail if provided an integer.
- Converted a few EDSM classes to attrs dataclasses.
- Fixed a bug where the bot could be instructed to send a message with no content.

Next Steps

We're still hard at work on the next feature set. Even as I write this, our next set of issues is already in the oven, and we've got some ready to go right now that are just waiting on review! In fact, I submitted a bug fix while I was writing this very post that I found while making these examples for you. We'll hopefully be able to share more details about the next features here soon. Stay tuned! As always, you can watch our progress and review the code yourself here, if you want: https://github.com/HullSeals/HalpyBOT

See you next month!

HalpyBOT Dev Update #4

So much for "Monthly", eh?

So yeah, it's been two months since the last HalpyBOT Dev Update... sorry about that! I'll also apologize for not a whole lot of progress being made in these past two months - the prior 3 were hotbeds of activity and sped by, and are proving themselves to be the exception, not the rule. Generally speaking, you're not going to see an update here until it's passed its initial review and is packaged in the latest mainline Dev version of the bot. This may lead to some code being done for quite a while before we actually write about it here. Thankfully, we had a few things pass review this month, So let's jump right into it!

Responder Management

Last few blogs, we discussed Case Creation and how HalpyBOT will handle loading all the fancy details about a case. This included allowing trained seals to update certain case details, such as System, IRC Name, or even the CMDR name itself. That was an important step, but we were missing visualization on a couple more critical, blubbery, fish-smelling details - YOU! How do we know who has responded to a case, if the Client has been welcomed by Dispatch, or who is in charge?

Enter Responder Management.

Now, Dispatchers can actively add or remove responders from a case, or even add or remove other dispatchers! Welcoming a case adds you as a current Dispatcher, but we also have a number of key commands including !addresp, !adddisp, !remdisp, and !remresp. These commands (as well as !go and !welcome) can be used to track exactly who's doing what in a case. In the future, this means you won't have to manually enter all of this in paperwork.

```
HalpyBOT[Dev|Rixxan] | Here's the self listing for Case ID 1:
HalpyBOT[Dev|Rixxan] |   General Details:
HalpyBOT[Dev|Rixxan] |     Client: InHoomansLeftEar2
HalpyBOT[Dev|Rixxan] |     System: MAIA
HalpyBOT[Dev|Rixxan] |     Platform: ODYSSEY
HalpyBOT[Dev|Rixxan] |     Case Created: 2 minutes 59 seconds ago
HalpyBOT[Dev|Rixxan] |     Case Updated: 3 seconds ago
HalpyBOT[Dev|Rixxan] |     Case Status: ACTIVE
HalpyBOT[Dev|Rixxan] |     Client Welcomed: Yes
HalpyBOT[Dev|Rixxan] |     IRC Nickname: InHoomansLeftEar2
HalpyBOT[Dev|Rixxan] | Case Details:
HalpyBOT[Dev|Rixxan] |     Hull Remaining: 25
HalpyBOT[Dev|Rixxan] | Responder Details:
HalpyBOT[Dev|Rixxan] |     Dispatchers: Rixxan, JooGoo
HalpyBOT[Dev|Rixxan] |     Responders: Rik079
HalpyBOT[Dev|Rixxan] |     Notes:
HalpyBOT[Dev|Rixxan] |         None Yet!
```

Requirement Cleanup

HalpyBOT uses a very strictly-defined system to determine different permission and restriction levels on commands and users. From who can run a command to where it can be run, to what systems have to be in the green for it to operate, it's been our bread and butter. However, recently we found out that some things we were doing were a little over-complicated. This has allowed us to break apart the overarching "Require" system into its component parts, namely Permissions, Locations, and Systems. This allowed us to streamline our code process, and only import what we truly need. This results in less overhead, and a better coding experience for our devs. We also took the time to streamline and remove a **LOT** of redundant code in these sections. Yay!

Is This Thing On?

Sometimes, we all just need that little push. Now, if nobody welcomes a case within a 5-minute window, the Bot will send a reminder to all Seals, make a note of the event in the logs, and also send the Client a short little message asking them to hold on. This can be disabled by using the !welcome command or updating any aspect of the case in the system.

```
[18:25:18]    HalpyBOT[Dev|Rixxan] xxxx PCOCASE -- NEWCASE xxxx
[18:25:18]    HalpyBOT[Dev|Rixxan] CMDR: InHoomansLeftEar2 -- Platform: ODYSSEY
[18:25:18]    HalpyBOT[Dev|Rixxan] System: SOL -- Hull: 25
[18:25:18]    HalpyBOT[Dev|Rixxan] xxxx Case ID: 1 xxxx
[18:25:18]    HalpyBOT[Dev|Rixxan] System exists in EDSM, 0.0 LY South of Sol.
[18:30:11]    HalpyBOT[Dev|Rixxan] Hey there, InHoomansLeftEar2! It seems we're a little short on Seals right now.
[18:30:11]    HalpyBOT[Dev|Rixxan] Just hold tight and someone should be with you shortly!
[18:30:11]    HalpyBOT[Dev|Rixxan] NEWCASE has not been welcomed. Seals, Please respond! Case ID: 1
```

Just The Little Things

Of course, there are always little bugs and other fixes we're working on as we go, trying to build the best possible product. Here are a few small ones that made it into this window:

- Fixed a bug with the Dice Roller function
- Updated the argument checks at various places throughout the bot to be more accurate
- Removed some useless duplicate code in the whois section
- Updated some of the exception types to be more specific
- Fixed a bug that could occur if the bot tried to update a case with some special characters
- Fixed an issue with a case printout that could result in a weird-looking name for some statuses
- Added a missing await on one function

- Updated a command mapper to use dict.items()
- Updated a number of type annotations
- Fixed and wrote a bunch of new tests

Why so little these last months?

Admittedly, this isn't as cool or flashy of an update as we've had in updates past. It's only fair to ask why that is. Our team consists of a bunch of volunteers, and as volunteers, our ethos has always been that real life comes first. We all have our own careers, education, life, and futures to think about. I know for just myself alone that the last two months have seen 3 cross-time-zone trips, my final exams, graduation from uni, moving to a new city, and starting a new job. Others have had their own plans and situations change, and really, we've just been super busy trying to keep up with life. Unfortunately, this cut into gaming and coding time significantly. Not to mention, we've had to deal with a number of... let's call them incidents... regarding our site and abuse of services that have required our prompt attention. Unfortunately, these resources took away from our bot development, as well as a number of years off our collective lives from stress.

We're also limited by our review capability - writing a new feature requires (obviously) someone to write it, but it also requires a second level of validation from an uninvolved coder to review and make sure the changes are sound. This helps prevent stupid mistakes from making it into the production-bound pipeline, and also fosters discussion, and allows us all to learn. However, there's really only one or two folks who fit the bill for this "senior dev" in the Cybers, and their free time is always at a premium. And remember, the reviewer has to have been someone who didn't write a large chunk of the feature. This has resulted in a small backlog of things that are ready to go pending review, which we simply haven't had the time to kick forward. As I write this, one feature is awaiting review, and another is going to be ready to go in just a few days, but we don't have a solid timeline to get those approvals in.

So, what can we do about this? Simply put, we need more folks. If you know code or want to expand your knowledge, reach out! We're always willing to take on new folks with a drive to jump in and learn. It may look scary, but you'd be surprised how quickly someone can pick up the reigns. Just ask our very own Phoebe, who was where many of you are now just a few months ago, interested and wanting to get your feet wet.

The Future

We're starting work on our Hull Seals API, and a few more features for HalpyBOT. One of those features in progress is a feature requested by one of you! We're super excited to see how it turns out and are always interested in hearing your thoughts on tools you wish were in the bot. While 2.0 is taking a significant amount of time, we are sure it will be well worth it once it is ready. Until then, stay tuned and stay in touch. If you're interested in looking at the code yourself, feel free to here: https://github.com/HullSeals/HalpyBOT

Bye for now!

HalpyBOT Dev Update #5

Welcome one, welcome all, to your HalpyBOT Dev blog for the month of `{!REF_ERR}`!

It's been another relatively slow month, but that doesn't mean we haven't been working hard. We've got a new feature to show off, and another one just waiting for the final checks to be rolled off the assembly line and into staging. So anyways, off to the show! Apologies for the lack of shiny new toys or pictures this month. I promise next month will be cooler!

Case Flow

We've talked before about how Case Creation works, from adding cases manually to getting them from the site. This all works great - but what about the other side of things? We don't want dead cases cluttering up the board forever, so how do we deal with it? Enter case cleanup.

HalpyBOT has historically shouted out a message when paperwork is filed. Now, cases are now automatically removed from the board when this happens! The bot attempts to match announced paperwork to an active case and cleans it up accordingly. It's a simple thing, but it's an essential step in the process. If the bot messes up, the case can be removed by an admin from the board, however, this will be the exception, not the rule. We're continually working on testing and tweaking this system, so please bear with our growing pains.

Under the Hood

As always, each month we're trying to validate and ensure that everything is as optimized as we can get it. I can't tell you how many hours we've spent sometimes just staring at our code, trying to find ways to make it better, faster, and stronger. This update is no exception, with a few minor performance and memory improvements under the hood which are really too boring to elaborate on further.

We've also fixed a few bugs and tweaked a few logic processes to be simpler. Notably, we've tweaked how some decorators are formatted and simplified the logging setup for the internal API system. This all makes our code easier to read, maintain, and build on later down the road.

Reversions and Mistakes

We're not perfect, however. This month we discovered that one function we developed (the modification of case notes) wasn't quite working as intended, and in fact, had a completely redundant function in it that went nowhere, did nothing. That function has now been removed, and the behavior corrected. This reduces the memory intensiveness of the bot and reduces processing time for some functions.

So what's the holdup?

I know that this project is taking (generously speaking) a long freakin' time. As I mentioned Last Month, we're a tiny team and this labor of love has to be fit in during our very finite free time. We really need more help on all fronts, from writing code, reviewing it, or helping with testing when the time comes. I really do encourage you to at least think about if you can help us out. The Seals only survive with our members, and only with your help can we make this the best product we can.

Next Steps

As I alluded to earlier, we've already got another set of changes in review right now. It's one that a Seal actually requested, and we're super excited to be making it for you. I won't write about it here, but you can read the notes in the Git Pull Request. After that, we're into testing the bot and ensuring that it's up to the standards we expect, then working on the long-awaited Seal API! Keep your eyes peeled for more details on that coming down the pipe soon.

See you all next month!