

# 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 changes 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 command provide more information, but you can also query for help

```
Rixxan ^help
HalpyBOT[Dev|Rixxan] EDM
HalpyBOT[Dev|Rixxan] lookup, locate, distance, landmark, dssa, diversion, coords
HalpyBOT[Dev|Rixxan] Dispatch
HalpyBOT[Dev|Rixxan] mancase, manfish, tsping, last, listboard, listcase
HalpyBOT[Dev|Rixxan] Facts
HalpyBOT[Dev|Rixxan] allfacts, factinfo, deletefact, addfact, editfact, ufi
HalpyBOT[Dev|Rixxan] Utility
HalpyBOT[Dev|Rixxan] ping, dbping, edsm ping, serverstatus, utc, year, whois, whoami, forcejoin, rrjoin, opsignal, cybersignal, about, say, shorten
HalpyBOT[Dev|Rixxan] Drill
HalpyBOT[Dev|Rixxan] drillcase, drillkfcase, drillcbcase
HalpyBOT[Dev|Rixxan] Administrative
HalpyBOT[Dev|Rixxan] notifyinfo details, notifyinfo groups, addsub, settings nick, settings prefix, settings offline, settings warning_override, joinchannel, partchannel, shutdown
HalpyBOT[Dev|Rixxan] Fun
HalpyBOT[Dev|Rixxan] roll
Rixxan ^help lookup roll
HalpyBOT[Dev|Rixxan] Use: ^lookup [System Name]
HalpyBOT[Dev|Rixxan] Aliases: syslookup
HalpyBOT[Dev|Rixxan] Check if a system exists in EDM
HalpyBOT[Dev|Rixxan] Use: ^roll #d#
HalpyBOT[Dev|Rixxan] Roll the dice! (Up to 10 dice of any size)
Rixxan ^help roll
HalpyBOT[Dev|Rixxan] Use: ^roll #d#
HalpyBOT[Dev|Rixxan] Roll the dice! (Up to 10 dice of any size)
```

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.

```
[21:36:27] Rixxan ^fireball
[21:36:27] HalpyBOT[Dev|Rixxan] Kawoosh! Rixxan cast a fireball on chat! Rolling for damage...
[21:36:27] HalpyBOT[Dev|Rixxan] Failed to invoke the command! Error code: Harbor.
```

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

---

Revision #3

Created 7 March 2023 00:09:52 by Rixxan

Updated 7 March 2023 01:26:32 by Rixxan