

Mecha Interaction - For Dispatchers

This document is designed to lay out the commands and facts used by Dispatchers on a day-to-day basis, flowing through the rescue process section by section. It is not designed to be a comprehensive list of commands, as many are not relevant in day-to-day use. The comprehensive list of Mecha commands can be found in the Reference section of this page.

All commands and facts laid out in this document will have a short explanation. In many cases further SOP References are available in the Reference section. It is the responsibility of all Dispatchers to ensure they are familiar with the SOP and policy behind the commands they use, not just the commands themselves.

We always invite feedback and thoughts on this document which, by it's very nature, is fluid and evolving. Please get in touch if you have suggestions.

Cameron Welter (Clapton) and Jamie Torrance (Stillian)

Welcoming and Preparing the Client

Most cases are automatically generated by RatMama[BOT] announcing a new client when they join via the website, which will look like this:

```
RAT SIGNAL - CMDR Marenthy - System:  
LHS 3447 - Platform: PC -  
O2: OK - Language: English (en-US)  
(Case #1)
```

When a new client arrives, in all cases where O2 is OK, the first course of action is to prep the client. To do so, use:

```
!prep <Client Nick>
```

Sometimes a client will not know how to deactivate their ship's modules. You can instruct them how by using:

```
!(pc|x|ps)modules <Client Nick>
```

Occasionally, a client will join by other means and a case will not be generated for them automatically.

In this case, ask the client for their system name and platform **AFTER** the client has been prepped. You will then need to manually generate a case, using:

```
!inject <Client Nick> RAT SIGNAL  
<System Name> | <Platform>
```

Mecha can not auto-detect systems when opening a new case with !inject, so when opening a new case using the !inject command you must use:

```
!sys <Client Nick|Case Number>  
<System Name>
```

Mecha also cannot auto-detect o2 status. When a case is manually injected, it will assume the case is non-CR. If the client is on emergency oxygen you will need to update the case using:

```
!cr <Client Nick|Case Number>
```

Ensure the case is listed correct, either with the DWB (dispatch web board) or by using !quote <clientname/casenum> in a PM with mechasqueak.

Code Red

CR clients may be generated upon their arrival, and if so RatMama[BOT] will announce their arrival as:

```
RAT SIGNAL - CMDR Marenthy - System:  
LHS 3447 - Platform: PC -  
O2: NOT OK - Language: English (en-US) (Case #1)
```

Mecha will automatically prep this client by instructing them to note down their location, and save and exit to the main menu immediately.

Should a client go CR in the middle of a rescue, Dispatchers should use the CR prep command to generate this, using:

```
!prepcr <Client Nick>
```

The client can also be instructed to save and quit to the main menu, using the platform specific instructions below. However, in the first instance it is recommended to use !prepcr as it includes the instruction to note down location.

```
!(pc|x|ps)quit <Client Nick>
```

Once this is done, the case is marked as a Code Red case using the command:

```
!cr <Client Nick|Case Number>
```

Clients should be given clear instructions on the process of a Code Red rescue, to ensure there is no confusion and delays are kept to an absolute minimum. While many Dispatchers choose to type this out, it can also be automated and the command is:

```
!crinst <Client Nick>
```

Rat Assignment and Friend Requests

To assign a rat or rats to a case:

```
!(go|assign) <Case Number | Client Nick> <rat1> <rat2> <rat3>
```

Once the go|assign command is issued Mecha will automatically tell the client to add the rat(s) to their friends list.

Sometimes it's necessary to remove a rat from a rescue for whatever reason.

```
!unassign <Case Number | Client Nick> <rat2>
```

Clients may require instruction on how to send friend requests. This can be given with the command:

```
!(pc|x|ps)fr <Client Nick>
```

The Rescue

Finally, once Rats are assigned, the client must be rescued. The client must first send a wing request, and they can be instructed on this using the command:

```
!(pc|x|ps)wing <Client Nick>
```

Following which, they must drop their beacon. The instruction command for doing this is:

```
!(pc|x|ps)beacon <Client Nick>
```

Should this fail and the instancing SOP is used, the first step is to have the client conduct a supercruise hop. They can be given instructions on how to do this, with the fact:

```
!sc <Client Nick>
```

Once the client is rescued, the case is cleared using the command below. Note that the 'first-limpet Rat' value marks that person as 'first limpet.' It is not mandatory and, in cases of failure, will not be used.

```
!clear <Case Number | Client Nick>  
<First-limpet Rat>
```

As part of the Debrief process, the client will be instructed on how to filter the galaxy map and educated on fuel scoops. The link to the guide for this material is:

```
!kgbfoam <Client Nick>
```

Occasionally, the client may be unable to send a wing request and the Rats will need to conduct a "reverse wing." Should the client be unaware of how to accept a wing request, a short guide is available using the fact:

```
!invite <Client Nick>
```

Case Information

Detailed information is held about each case and should be kept as accurately as possible. The commands listed here are all 'administrative' commands and are used as part of this function.

Some clients may report their platform incorrectly when joining us via the website, or will join us via a manual Ratsignal and must have this set. This is done using the commands:

```
!pc <Case Number | Client Nick>  
!xb <case Number | Client Nick>  
!ps <Case Number | Client Nick>
```

Clients will regularly report the wrong system upon first joining us, particularly if they are new players confused between target system and their current system. Once this is found, the case can be corrected using the command:

Special Situations

Every case is different and some commands are used rarely enough, or do not fit neatly enough into any other category, that they have been classed 'Special.'

Many clients are unfamiliar with IRC as a method of communication and /or do not appreciate we handle many cases at once. Where this happens, Mechasqueak can be used to inform the client to only listen to instructions that include their username. This is done by using the command:

```
!multi <Client Nick>
```

Some cases are, once further information is obtained, deemed not to be a rescue after all. These cases are marked for deletion, for review and /or deletion by an Overseer or above. This must only be done once the Dispatcher has checked the case against the [How to File Cases / When to !md, !invalid and !delete cases](#) SOP, using the command:

```
!sys <Client Nick|Case Number>  
<System Name>
```

Occasionally the wrong Cmdr name will be given. This can be changed using the command:

```
!cmdr <Client Nick|Case Number> <New  
Cmdr Name>
```

Very rarely, the client will change their IRC nickname or this will otherwise be incorrect. The case can be updated with this change, using the command:

```
!nick <Client Nick|Case Number> <New  
IRC Name>
```

Notes are often added to the client's case, particularly during CR rescues. These are added using the command:

```
!inject <Client Nick|Case Number>  
<Text>  
e.g. !inject Stillian Client 300kls  
from main star, Rats bc+ and  
travelling
```

The client's last line in IRC can also be easily added to the case notes, using the command:

```
!grab <Client Nick>
```

All collected information on a case can be viewed using the !quote command. Note that this command should be used in PM to Mechasqueak, as it is multi-line and spammy.

```
!quote <Client Nick|Case Number>
```

For various reasons, a case may need to be marked inactive and re-visited at a later time/date. It may also need to be re-activated. This is toggled by:

```
!active <Client Nick|Case Number>
```

Finally, if the !inject or !grab commands have been used but must be edited, the command below can be used to achieve that. Note this should be used after having used the !quote function in PM to identify the line that requires editing:

```
!sub <Client Nick|Case Number> <Line  
to remove/edit> <New Text>
```

```
!md <Client Nick|Case Number>
```

Occasionally a case will come in where either the system or the circumstances mean that the level of danger for the Rats and/or the client is too high. Such rescues will be carried out in a Private Group. A good example of this is a system in which a Community Goal is active. The client can be instructed on how to join a private group, by using the command:

```
!pg <Client Nick>
```

You may find yourself short on rats in odd hours of duty, and other special occasions. Luckily, we have an alert twitter account for rats to follow! Mecha now supports the functions of TweetSqueak (thanks Edmonson for all your work). For you to be able to send a tweet from Mecha, you must be a drilled rat, and the command **must** be in #fuelrats. Never include any case-specific information in your tweet! Just leave it to the platform of the case and if it's a CR or not. To send the tweet use one of the following commands:

```
!tweet <message>  
!tweetcase <Client Name|Board Index>
```

Finally, some cases are just **epic** and can be marked as such. Epic cases confer a particular award on the Rat(s) involved and should be used wisely. Any Rat can nominate another Rat/case as 'epic,' although a second opinion is never a bad thing! Epic cases can also be given Operation names/titles if appropriate. The commands for these functions are:

```
!epic <Client Nick|Case Number>  
!title <Client Nick|Case Number>  
<Title>
```

Any Long-Range Rescue over 10,000lys is automatically **epic**, but still requires nomination. Any case that is deemed epic must be nominated by a Rat independent of the case. Cases can be nominated for any reason deemed epic by the nominator.



Epic Is broken

Fun fact, !epic is currently not implemented and doesn't work. Don't try using it, It does nothing!

SOP References

Dispatch SOP

The Dispatch SOP is "The Bible" and is essential reading. This document does not replace it, merely supplements it, and all Dispatchers should read the Dispatch SOP **fully** before referring to this page.

Rescue Standard Operating Procedures

The basis of everything we do, this SOP should be understood long before a Rat undertakes Dispatching duties.

How to File Cases / When to !md, !invalid and !delete cases

Possibly the least well known SOP at time of writing, we commend this SOP to you. It is a significant change to previous operating procedure, and all Dispatchers should thoroughly read and understand if they have not already.

Command-Reference

The comprehensive list of Mecha commands, where this document is only a day-to-day list.

Instancing and Avoiding Unwanted PvP

This SOP expands on the actions to be taken by Dispatchers when issues arise during rescues.

Xbox Troubleshooting

Xbox troubleshooting & issues can be significantly more complex than on PC. All Dispatchers, whether they themselves be PC or Xbox, should be intimately familiar with this SOP.

Special Rescue Scenarios

Special rescue scenarios are, thankfully, relatively rare. However, they can be stressful for an already-busy Dispatcher. This SOP guides Dispatchers through these unusual incidents and it is useful to have a working knowledge of it.

Long Range Rescue Standard Operating Procedures

The SOP for long-range rescues, it is appreciated that many Dispatchers may not come across these cases often. It is recommended to refer to this SOP when required to Dispatch such a case.