

A crash course to the

# Renegade Projects Network Services

*as used by Total Destruction, ModEnc and all Strategy X Hostees.*

## Intro

Morning, people. Today I'm gonna introduce to you the Renegade Projects Network Services, namely

- **Operation Bughouse**

RPN's Mantis Bugtracker, located at <http://bugs.renegadeprojects.com>

- **The File Fortress**

RPN's download database, located at <http://files.renegadeprojects.com>

- **Visual Propaganda**

RPN's open gallery, located at <http://gallery.renegadeprojects.com>

and, to a smaller extend,

- **The Renegade Projects Network Forums**

to introduce you to some lesser-known and overlooked features.

First of all I'd like to mention that, apart from the Forums and Forum News, all RPN services are optional to use. You're not forced to use either of them. However, let me remind you of several advantages they provide:

- **They are ready-to go.** The systems themselves are running Just provide us with enough data to set up your specific accounts, and you've got a bugtracker, a download-db and a gallery for you screenshots in no time, with no hazzle for you.
- **They save bandwidth.** That's especially true for the download-db – you can still link to the File Fortress entry just like you would put a normal download link, but with the added advantage that the actually file comes from a different account, leaving your bandwidth to you.
- **You are on display.** Your screenshots, for example, will have a lot more hits if all of STX's userbase visits the gallery they're in, rather than just the users that are already visiting your site anyway.
- **They are easy to use.** Yes, especially Mantis looks mighty complex at first, but once I walked you through the standard procedures, I think you'll agree that looks can be deceiving.

And, last but not least, it's a question of usefulness. Why have five galleries with ten screenshots each, if we can have one gallery with 80 screenshots? Why rely on the forums for bugreporting if you have the possibility to use a full-featured bugtracker?

I am confident after our little tutorial you'll see that there's no reason not to use STX's services, and I'll be happy to answer all questions remaining.

# Operation Bughouse

Named after the Mobile Infantry's assault on Klendathu, this is *our* battlefield in the fight against bugs. Over the next view pages, I'll show you the basic procedures for

- **Bug browsing** (any user)
- **Bug reporting** (any user)
- **Bug updating** (the project's maintainers)
- **Project management** (the project manager)

## Bug browsing

So. For some reason, you want to look around the bugs of a certain project. Fine. Here's what you do:

Ideally, you should immediately see the main page, with an info post by me. If, for some reason, you see a login-form, either click „*Log in anonymously*“, or register, or, if you are registered, log in. Now that you see the main page, prepare to check the most important info in Mantis at any time: What project you're looking at. Glance at the upper right-hand corner. It'll probably say „*Project: All Projects*“, but if you've been there before, it might have saved something else for you. Now it's on you if you want to stay in „All Projects“ mode, or if you want to only look at a particular project. Since I'm going to use a Rock Patch bug for this tutorial, I select „*Rock Patch*“ from the dropdown-list (and click „*Switch*“ if it doesn't switch automatically).

Now, didn't change much, eh? Correct. Because all you changed was whether Mantis should display information about *All Projects* or just *Rock Patch*. *How* that information is displayed didn't change. Click on „*View Issues*“.

This is the key section for browsing bugs. Below the navigation, you see the *Filter* options, which control *which* bugs you see, and below them are the actual bugs. Now, let's analyze this once again. „*View Issues*“ shows you bug reports. „*Project*“ determines if you want to see all bugs, or just bugs from a certain project. And the *Filters* decide which of the project's bugs to show you, basing on your preferences. For example, in the default setting, closed bugs are not shown anymore. It's a *Simple Filter* with „*Hide Status*“ set to „*closed*“. I, personally, don't like this, and created the (public) *Advanced Filter* „*show all*“, where „*Hide Status*“ is set to „*none*“. As a result, I can see *all bugs* by default, while you can only see *non-closed bugs*. It is, however, fairly easy for you to change that ;). Just click on a certain category, e.g. „*Severity*“, select which bugs you want to see, and click „*Apply Filter*“, below the lower right-hand search field. For this example, I am still in Project Rock Patch, and selected to filter basing on Severity „*crash*“, resulting in only bug #[0000009](#) being displayed. If you want to reset the filter, just click on „*Use Filter*“ with „*[Reset Filter]*“ being selected. Alternatively, you can select a saved filter and use that one (e.g. my „*show all*“ filter). Note that filters and projects are independent from each other. While filtering on crash, we could easily switch projects to „*Whiteboy's Rules*“ to see if WBR has Severity: crash bugs reported. (Remember: Always keep an eye on your project selector.)

If you have the treshold of bugs you want to see, simply click on a bug's number to see the stored information about it. Still watching „*Rock Patch*“, using the „*show all*“ filter, I select bug #[0000008](#) - First Decade EXEs. (Coincidentally, the only closed Rock Patch bug so far.)

Now we're viewing the *Simple Details* of bug #0000008 – most of the fields are rather self-explanatory. You've got information about who submitted the bug report (Renegade, in this case), when (22.04.06 19:01), how severe it is, whether it is reproducible, which developer is assigned to look into it (pd, who else?), in which version it occurs, and, most importantly, the reporter's detailed description of the bug.

Below that, we see information about whether this bug is related to other bugs, attached files, and, even lower, we see user-added notes to the bug. In this particular bug, I re-posted pd's note that it'd be fixed in 1.08. That was before I set „*Fixed in Version*“ to 1.08.

Huh? „Fixed in Version“? Now you're just making stuff up!

Nope. Just like „Advanced Filters“, there are „*Advanced Details*“ as well. Click „*View Advanced*“ in the upper right-hand corner. Woah! Magic! Now there are a view more, mostly technical, fields. Information about the user's system, the projected type of fix, an ETA for said fix, and, ta-daa, „Fixed in Version“. The description part also gets extended, now showing steps to reproduce the bug.

As you can see, there are fields for everything you could imagine regarding a bug – developers get to know what they need to find the bug, and users can look up its status, who's assigned to fix it, and when to expect a fix.

But, how did the entry get there?

## **Bug reporting**

So you found a bug, and it wasn't reported already. What now? Well, if you don't tell the developers, they can't fix it! Therefore, you have to report it.

Click on „*Report Issue*“.

Since you've read the Bug browsing part, you should know all these fields already. Only this time, it's your turn to fill them out. There's not much to say here, since all fields are labeled descriptively, just don't forget that, the more you tell the developer, the more easy it is for him to hunt the bug down. Don't lie on the Reproducability field. If you haven't tried, tell him so. Don't overstate the Severity. If's only a missing name, it hardly qualifies for „major“. If you can describe exactly how to reproduce the bug, switch to advance mode and fill out „Steps to Reproduce“. If you know which version the bug occurs in, and you can select that version, select it. Fill in your computer's info (you can save profiles for that under „*My Account*“), since some problems are OS-related. If you have an except.txt, a log, a screenshot, anything helpful or proving the bug, attach it! The more information, the better.

When you're done, click „*Submit Report*“. You should be taken to the overview page, and see your bug as „new“ above the others.

Now that your bug is reported, soon a developer will stumble over it, and *update* it.

## Bug updating

So. Some evil user says your product is bugged. Better check that report out...!

As a developer (or its sub-class, „Updater“), you have the ability to update bugs, meaning you can provide the users and your fellow co-workers with up-to-date information about it. Simply click on the pencil graphic at the far left of a bug's column.

What you see now is basically an editable version of the View Bug page...but instead of being told the information, you can now tell it. You can assign the bug to a developer, tell the users which priority it has, give out information about its Resolution status (e.g. „fixed“, „unable to reproduce“ etc.), and change its general Status. The latter is especially important to let the users know something happens. Even if your first update is nothing but setting the Status to „Acknowledged“, it'll show the users that you are aware of the report – they know *something* is being done about the bug. Whether it comes out as „won't fix“ or „no change required“ is totally immaterial, as long as the users know they're not reporting for nothing. 'cause what reason would they have to spent their time writing you reports, if you don't even deem it necessary to let them know you read it?

When you're done, submit as usual, but never forget that you can always go advanced, to enter even more information...what kind of fix you guess it'll be, ETA, „Fixed in Version“...all the stuff you can see under „View Advanced“ can be set under „Update Advanced“.

In addition, there are some more things you can set when you simply view the bug. As a developer, there are additional forms allowing you to assign the bug on-the-fly and change its status (e.g. „Acknowledged“ ; ) ), but also new things, like buttons to clone the bug, move it, or to create a relationship-link to other bugs. The latter is useful if one bug results from another bug. For example, imagine, through a bug, a naval-only unit would be able to move to solid ground. Call that Bug A. Now a user reports that if that unit moves around on his land, it sometimes looks like it's diving straight into the ground or swimming. Call that Bug B. Now, Bug B would be the perfect candidate to be marked as a „*Child of*“ Bug A. Because, hadn't Bug A allowed the unit to be on land in the first place, its movement characteristics wouldn't be messed up. Fixing Bug A will probably resolve Bug B. (Notice, however, that Relationship is not limited to family issues. You can also mark duplicates and general connections through it.)

Righto. Now you fixed a whole lot of bugs, and what to update the „Fixed in Version“ tags. But wtf?! You can't select that version! Time to call up the *project manager*.

## Project management

Go to *Manage > Manage Projects > \$your\_projects\_name*.

Here's the place where you can set all necessary information to represent your project and ease reporting bugs for it in the bugtracker.

At the very top, there is general information, like whether it is a public or a private project, and which development status it has. Below that, the report-organizing part begins.

First of all, you can create *Subprojects* for your project; there is, for example, a public „Documentation“ subproject that many projects include. This allows your users to further specify which part of your project is broken.

Next up are the categories – they allow your users to tell you what type of bug they are describing. Rock Patch, once again our example, has categories for Crashes to Desktop, Internal Errors, but also Super Weapon related problems and Country and Side Issues. Simply create the categories you think are best for your project.

Btw, categories of subprojects are managed separately. They are independent from the main project's categories. So, while Rock Patch has the just mentioned categories, whenever a user selects one of the documentation subprojects, he'll be able to choose categories like Missing Information or Typos.

Below the categories are the versions. There are two types of versions – those checked as „released“ and those not. Users can only select released versions when reporting a bug, but unreleased versions are selectable by developers in the „Fixed in Version“ field. Rock Patch, for example, has an entry for 1.08. The latest version selectable for bug reports is 1.07b, but 1.08 can and has been selected for the „Fixed in Version“ entry. That's because 1.08 is missing the „released“ checkbox.

This technique also has a different use: In the navigation bar, there's a link „changelog“. It lists information for every version entered into the system, made up of a) the descriptions entered for the version, and b) the Fixed in ... issues. Rock Patch's version lists for 1.08:

– [0000008](#): [**Installation Problems**] First Decade EXEs (pd)

This is because bug #0000008 has been marked as „Fixed in Version: 1.08“.

The information given is, in order, bug no., as a link, the bug's category, the bug's title („summary“), and, in brackets, the developer assigned to the bug.

This means that, if you use the bugtracker extensively for your bug reports, even entering those you found yourself into the database, The Bughouse will be able to automatically generate you a changelog for fixed bugs for each of your releases.

But enough of that, there's one last thing on the project management page: Users. Using that form, you can assign users with different access levels to your project:

- **Viewers** are simple guests with no rights.
- **Reporters** are registered users, able to report bugs and post notes.

These are your normal, non-project-member user access levels. The line between them is somewhat blurred, since, through the activation of anonymous reporting, all „Viewers“ log into a certain anonymous account, essentially making them some kind of semi-reporters.

- **Updaters** can update the bug report, but only
- **Developers** can set a bug's status to „fixed“. In addition, there are
- **Managers** who can change the very project info we're talking about.

Those are the development-related levels, allowing you to fine tune each member's access level to the bugs. Note that you can add users as reporters or viewers to your project as well – they just won't have the powers to update bug reports.

Phew. I think that's about it about Operation Bughouse. Turned out to be more than I thought, but nevertheless, it's actually quite easy, isn't it? „Managa Projects“ takes you to project management, „Report Issue“ lets you report a bug, and if you change your project to „Rock Patch“, you only see Rock Patch related stuff. And all fields in the forms are labeled descriptively as well. Now if you're interested in using the bugtracker for your project, simply contact either DCoder or Renegade, and we'll have you set up in no time. If you register as a bugtracker user beforehand, it'll go even faster.

Next stop: Download server.

# The File Fortress

This one should go a lot faster than the Bughouse one.

## **User perspective**

There are two ways to come to the File Fortress: Through a download-link, or through the main entrance. If you came through a download link, you start directly at the file view. Otherwise, you start at the *category overview*. Simply select the correct category for the project whose files you want to download, and you'll proceed to the *file overview*. Here you can see all files released to the database by the project, as well as possible sub-categories. Click on a file link to see the *file view*. Now you'll get some information about the file you're about to download, how much is up to the one uploading it. The important part is the big, shiny „*Download*“ button at the bottom. And once you clicked that, you basically did everything you could do here. The only thing left is rating the download after you tried it out.

## **Admin perspective**

Log into the admin panel. You can now add, edit or delete your files. Since editing is nothing but changing the information you entered already, and deleting doesn't need explanation at all, we stick to adding here. Both the former actions can be done through „*Manage Files*“, but we're clicking on „*Add File*“ now. Once you see the form, you should already have understood it – as usual, all fields are clearly labeled.

You actually need to fill out *all* fields of the form when you upload something, but putting some gibberish into the screenshot and file URL fields is fine, they'll get overridden by the uploaded file.

And that's basically it. Don't forget to set the correct category for your download, and up your download is.

One more word about the File Fortress: Always remember that you can provide the link to the File Fortress just as you would a normal download link – the user doesn't *have* to go through it manually. If can even skip the file view part if you simply copy the File Fortress's download link and provide it instead of your normal link – so the user doesn't have to do a single extra click, but you save your bandwidth and have a reliable download counter.

Of course, given the vast amount of information you can provide with a download through the file view, and the ability to rate a download, linking to the file view is always preferable.

In addition, as you may have seen in the form, you can also put files into the database that are not hosted in it, by filling „*Download URL*“ instead of uploading a file. This may be useful if you want to have a certain file on your webspace (maybe in a password protected directory), but still want it listed in the DB. In addition, the current version of paFileDB also supports adding mirror addresses, but as off the time I'm writing this, I haven't upgraded yet. This will follow soon.

Woah...that mod your people can download now looks pretty cool! But where to show off?

# **Visual Propaganda**

Now, you or your users made a shitload of cool screenshots, and you look for a place to show them off? Look no further! The gallery is there for you.

## **Browsing images**

You start at the album list. Either click on one of the randomly selected pictures, or click on one of the toplist-links in the navbar, or click on an album name. Once you clicked that name, you see a thumbnailed list of images in that album. Click one of the thumbnails to view the picture in its original size. You can now rate the file, and, if you're registered and logged in, leave a comment. Also note that you can start a slide show, using the movie clap type of button above the image, and that there are „Next“ and „Previous“ buttons as well. And that's all you need for browsing.

## **Posting images**

To upload an image, simply hover over the „@“ in the navigation. A menu appears. (Here's the place where you can register and log in as well.) Click on „*Upload File*“. Put in or select the addresses of the files you want to upload. Hit „*Next*“ (or „*Continue*“, whatever the English or your localized version says). Wait for it... Then you should get a „successful“ message. Click „*Next*“. Fill in the form for each file you uploaded, clicking „*Next*“ until you get to the final „*Information*“ screen, telling you all went well. One more click on „*Next*“, and you're back at the album list. Time to browse to the selected gallery and check out the pics you just uploaded ;).

# **The Renegade Projects Network Forums**

Did you know...

- ...there's a calendar built into the forums, where you can add your release dates, competition ends, etc. for all to see?
- ...that you can also add private events to it?
- ...that there's a link „Show Forum Team“ both on the index and in the search, leading you to a list of moderators and admins?
- ...that the forums support RSS syndication of specific and all forums, through the „RSS Syndication“ link at the bottom?
- ...that, if you use Firefox, there's an RSS link in the lower right-hand corner, allowing you to add a dynamic favorite folder, which always links to the latest threads?
- ...that we recently installed an Arcade, reachable through the link near „Search“?
- ...that in the same line, the link „Help“ leads to a page where not only you are helped with certain standard tasks of the forum, but also our rules are saved?
- ...that you can subscribe to entire forums, and single threads, to be notified of replies?
- ...that there's a „Rate this thread“ option below all posts of a page?
- ...that you can switch to a printer-friendly version of a thread?
- ...that you can also switch to lite/archive mode, if you care for loading times and not for design?
- ...that there's a „Report“ button in every post, for when someone violates the rules, and no moderator is reading along anyway?
- ...that you should always click „Add into post“ when posting an attachment, or it won't be displayed?
- ...that, in your User CP, under „Edit Profile“, you can tell us which game you mod, for which mod you work, and which position you fill?
- ...that, on the same page, you can set yourself to „Away“, in case you go on vacation, to prison, get drafted by the army, or abducted by Aliens?
- ...that your chances of getting a „Happy Birthday!“ thread increase significantly if you actually save your birthday in your profile?
- ...that „Edit Options“ allows you to customize your default viewing options for threads and forums?
- ...that you have a buddy- and an ignore list?
- ...that you can add threads to a favorites list which is accessible through your User CP?
- ...that you can also save personal notes when you click on „Personal Pad“?

Okay, that should be it about RPN's and STX's services...guess you were pretty scared when you saw the amount of text on Operation Bughouse? Yet, if you have read it, I think you'll agree that's only because there are so many features, not because it's hard to use them.

Anyway, like I said, if there are any questions left, don't hesitate to ask. That's what we're here for. And now, go use the services ;)

**The End**