IDA Teams: Collaborative work with IDA

Over recent years, IDA's ecosystem has seen the addition of a handful of tools that all share a common characteristic: their goal is to make collaboration on projects easier.

In fact, we too at Hex-Rays have already made a move towards that goal some time ago with the introduction of the public lumina server.

However, recognizing this is not enough. We are now moving forward with a set of tools that we hope will help teams of reverse-engineers collaborate on the same (or similar) projects in the most efficient way possible.

1. What is IDA Teams

Roughly speaking, IDA Teams offers a client-server architecture:

- On the server-side, we will find a new component, the Hex-Rays Vault server.
- On the client side, we will find that an IDA Teams installation now install a few additional binaries: a pair of clients to connect to the server

In addition, IDA has gained powerful database "diffing" & "merging" capabilities (described in more detail below.)

1.1. The Hex-Rays Vault server

The server borrows heavily from existing source-code management (i.e. SCM) solutions, and maintains a ledger of operations performed on files that it hosts: changes, additions, deletions, …

The server should be made available to all members of a team destined to work on common projects. It comes with its own installer, and an "admin guide" explaining how to tune the installation & its basic principles.

1.2. The clients

On the client-side, IDA Teams will consist of an IDA installer, to be installed on each user's computer.

In addition to the regular IDA binaries, that installer will place two clients to connect to the Hex-Rays Vault server:

- `hv`: a command-line interface client
- `hvui`: a GUI client

2. Online vs Offline

One of the key differentiators between IDA Teams and the collaboration plugins that have been developed over the years, is that those plugins will typically use an "event broadcasting" mechanism, to let different running instances of IDA know what has changed in one of the user's database, so that other instances of IDA have a chance of applying it locally.

We considered that approach as well, but decided against it eventually, because this would require that all users are connected at the exact same time in order for changes to be properly replicated in all databases.

Instead, we took one more page from the source code management solutions' book and decided to improve IDA to be usable as a "diffing" and "merging" tool: IDA is now able to start in special modes, letting users view changes, and resolve conflicts.

This lets users perform their work locally, and share it with the rest of the team once it is ready.

We see the following benefits to this model:

- the "event broadcasting" mechanism is somewhat brittle: if a copy of an `.idb` doesn't receive one (or more) events, it could go out-of-sync
- it lifts the requirement that users must be connected at all times to the server - it's not only about being able to work on a plane; it's much more general than that: what if a user needs to restart their computer?
- the server holds a history of changes, attributable to users

3. Example

In this chapter we will present a small, concrete example of a small team consisting of two reverse-engineers (Fred and Harry), working on a brand new project: analyzing an executable called `mystery.exe`.

NOTE The example below, will assume that Alice first installed the Hex-Rays Vault server on the premises. Check out the "admin guide" (bundled with the Hex-Rays Vault installer) for more information!

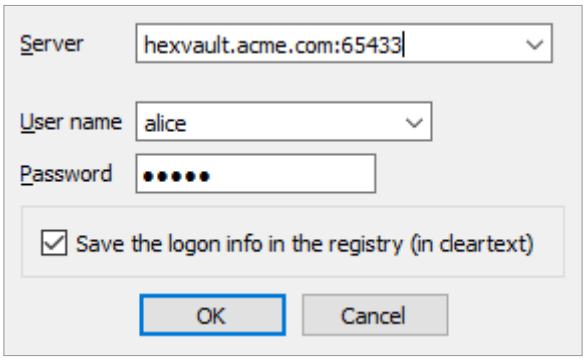
3.1. Alice lays the foundation for teamwork

In this scenario, Alice wants to get the team ready to hit the ground running, and so she will go ahead and perform an initial auto-analysis on the `mystery.exe` file, by opening it in IDA and letting the auto-analysis finish.

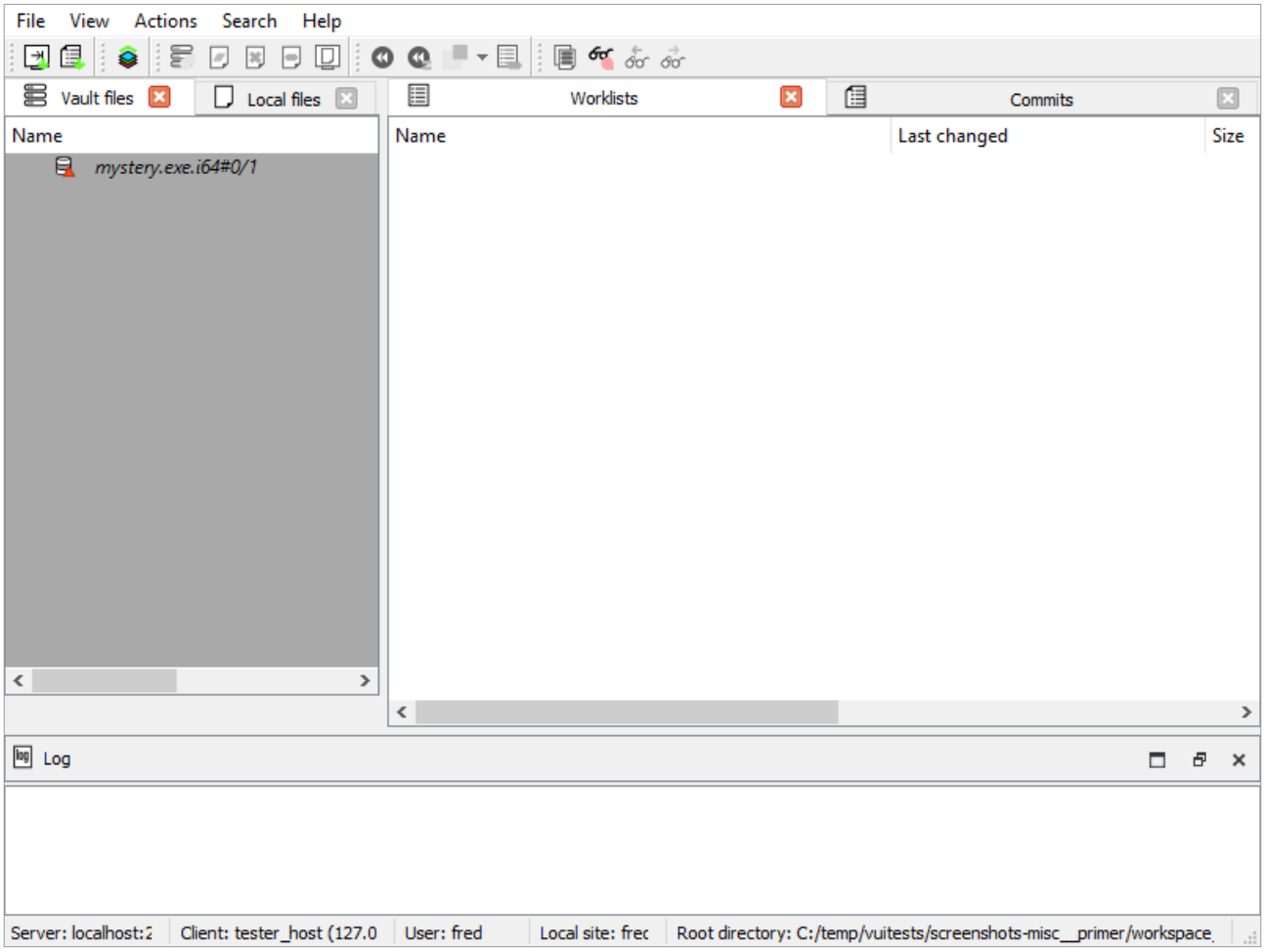
Alice will then use `hvui` to add the resulting `.i64` file to the Hex-Rays Vault server.

3.1.1. Making the file available

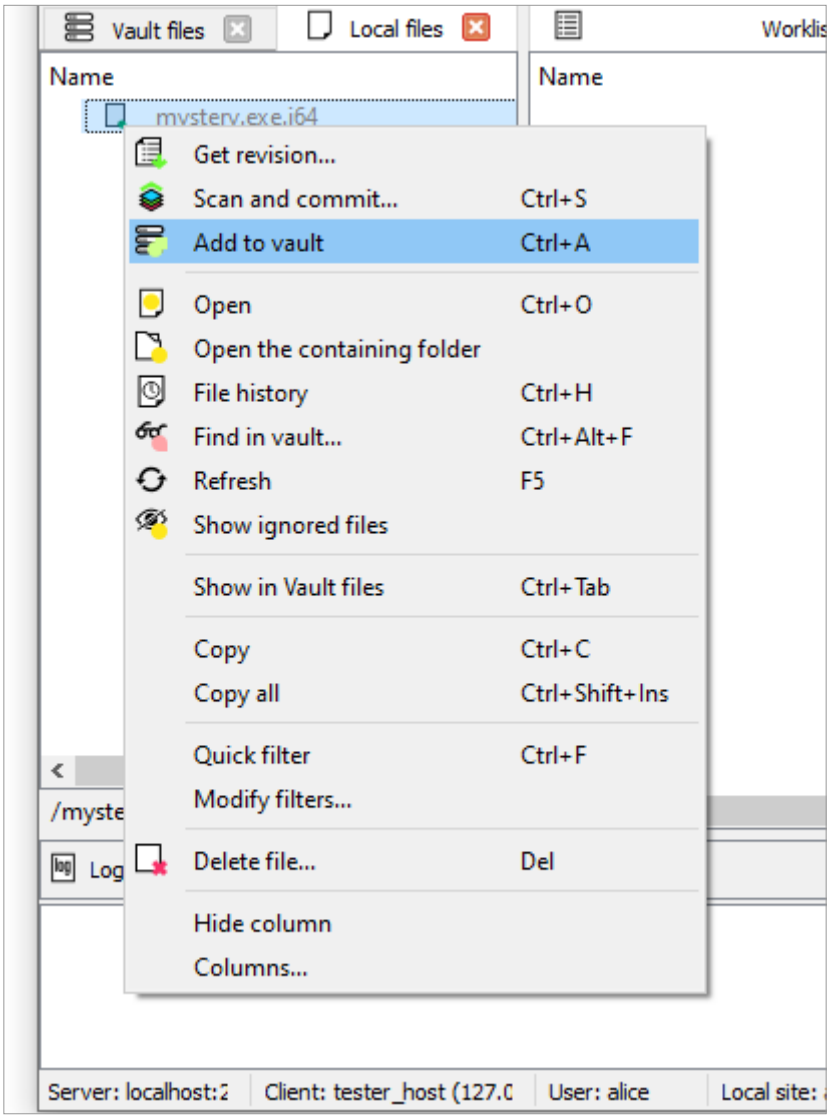
The first thing Alice will have to do when launching `hvui`, is to login:



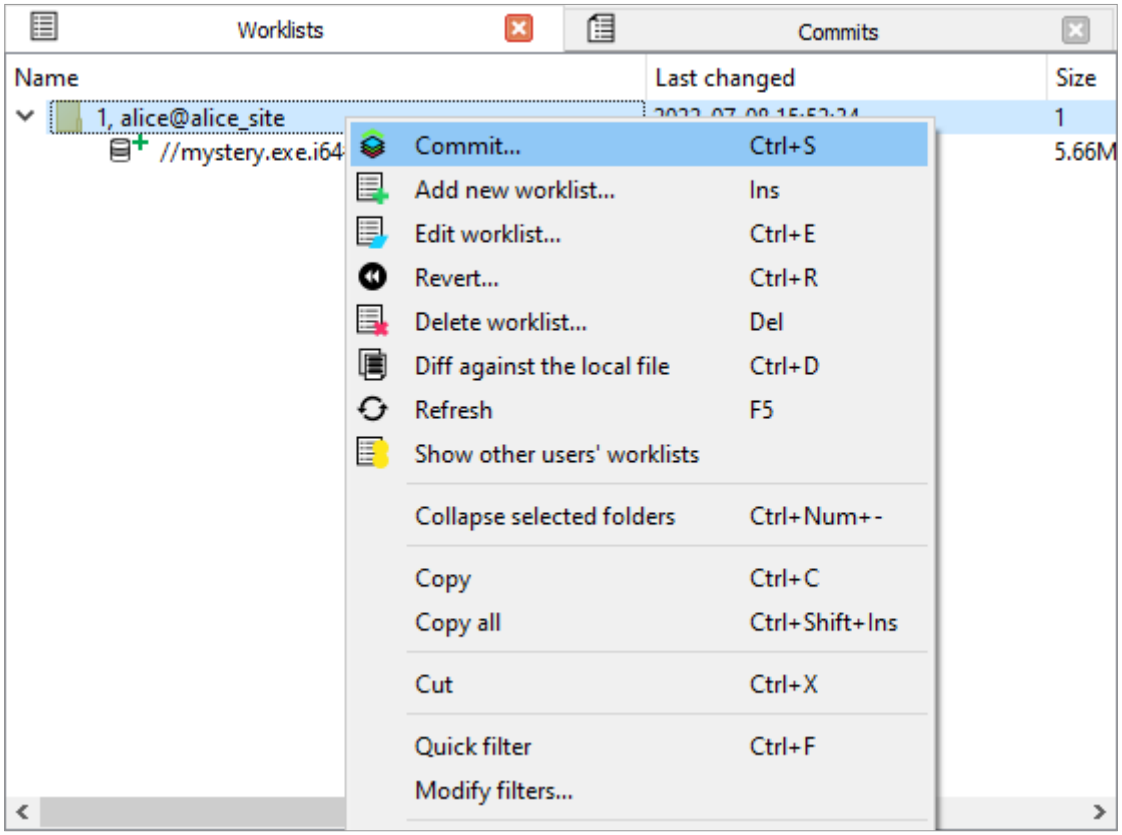
She will then be presented with the main `hvui` window:



From the "Local files" widget, she will instruct the Hex-Rays Vault server that the file should be added:



...and commit the freshly-created worklist:



3.1.2. Ready to go

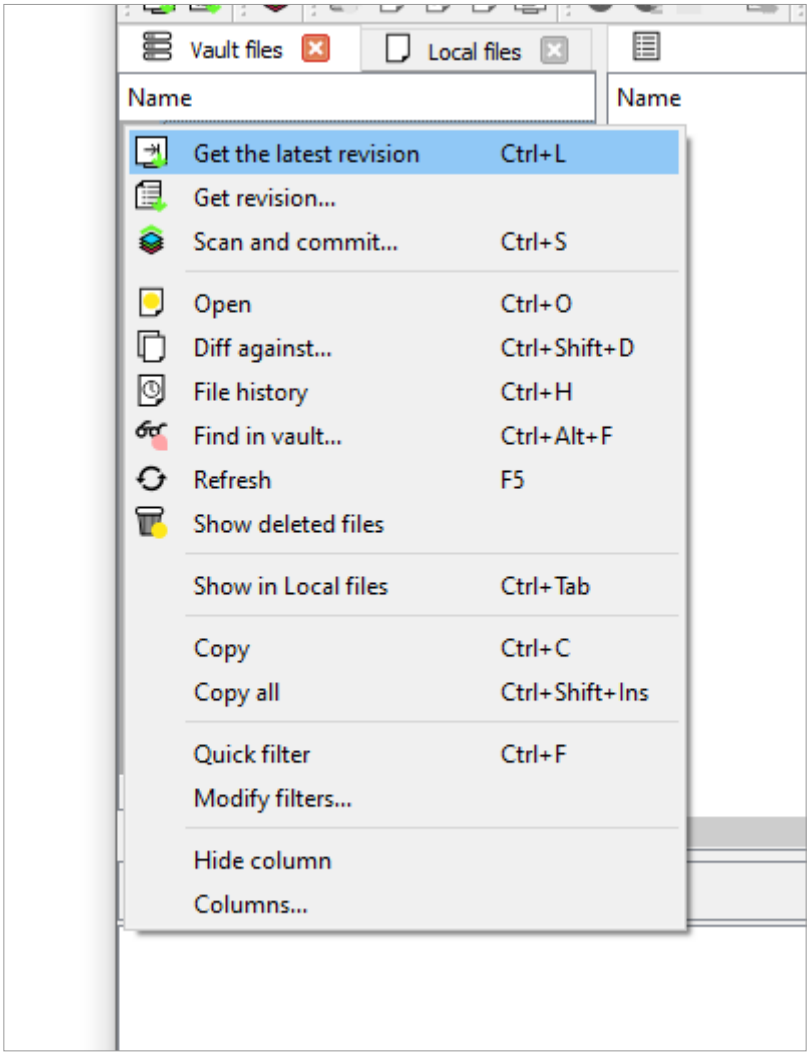
At this point, the initial IDA database is available for everyone to work on.

After a brief discussion with Alice, it was decided that Fred would do most of his initial investigation on filesystem-related functions, while Harry's focus would be more on the algorithms & data structures.

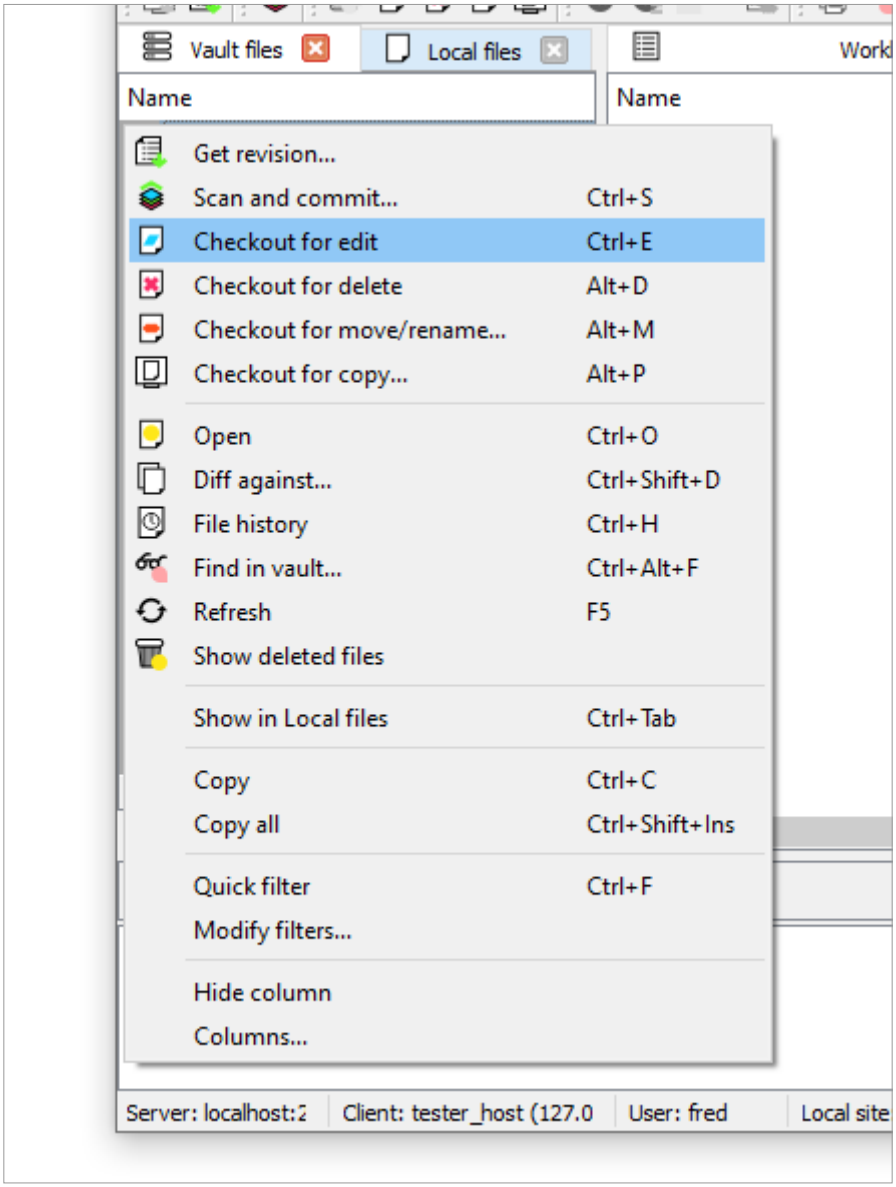
The very first thing both reversers will need to do, is to retrieve the file from the server (in this case, using the `hvui` GUI client).

3.2. Getting work done

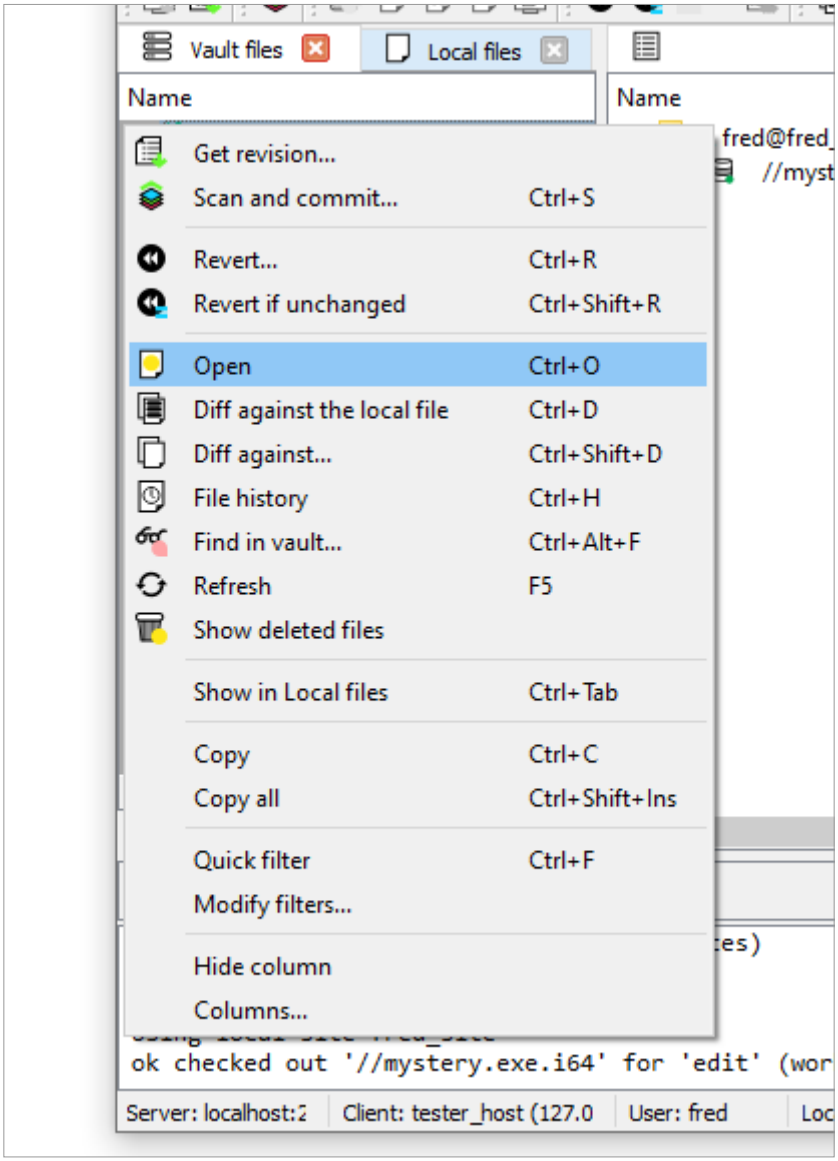
The first thing Fred and Harry will do after logging in, is to retrieve the file:

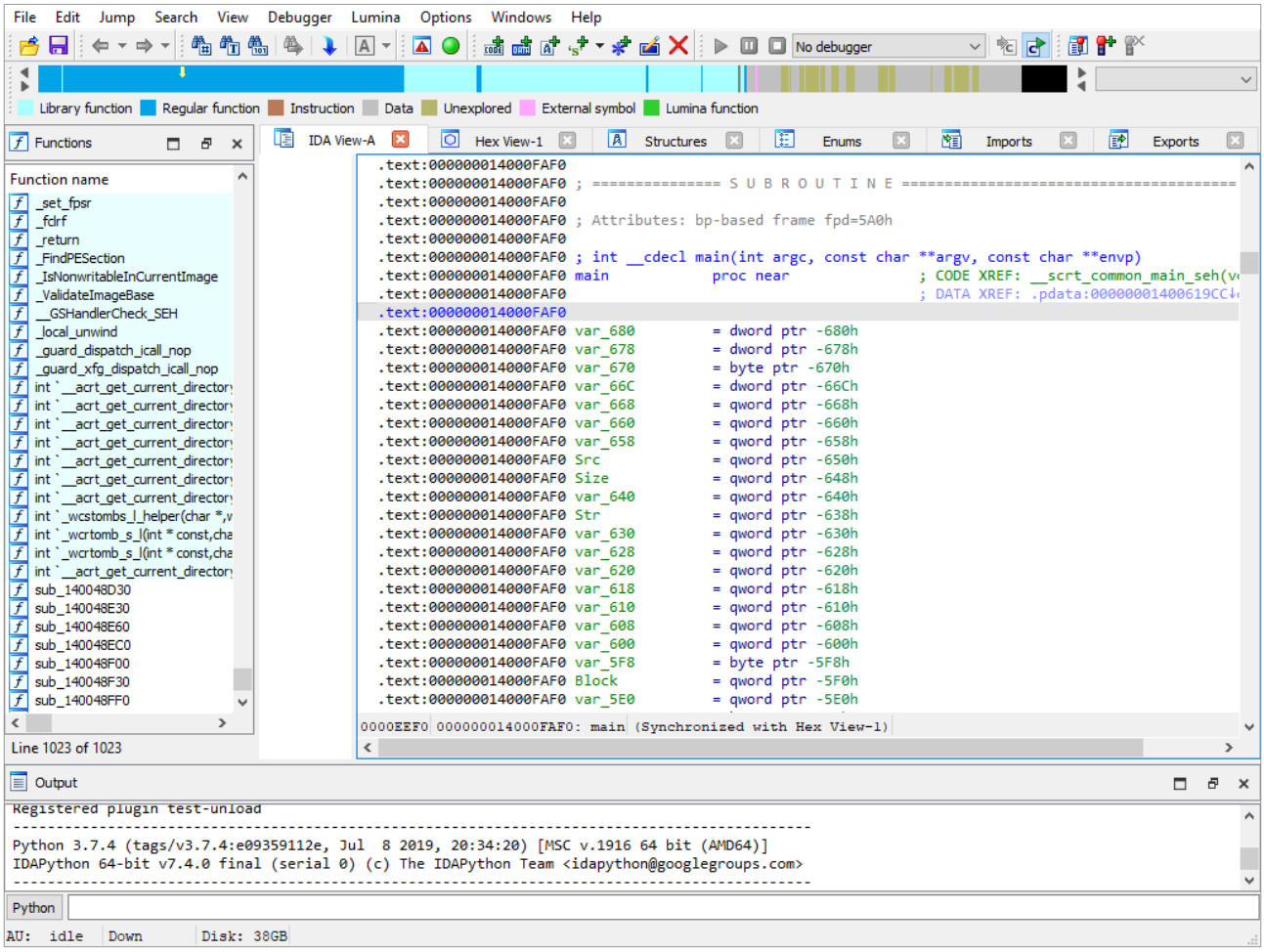


...then let the server know they will be working on it



...and open it in IDA:



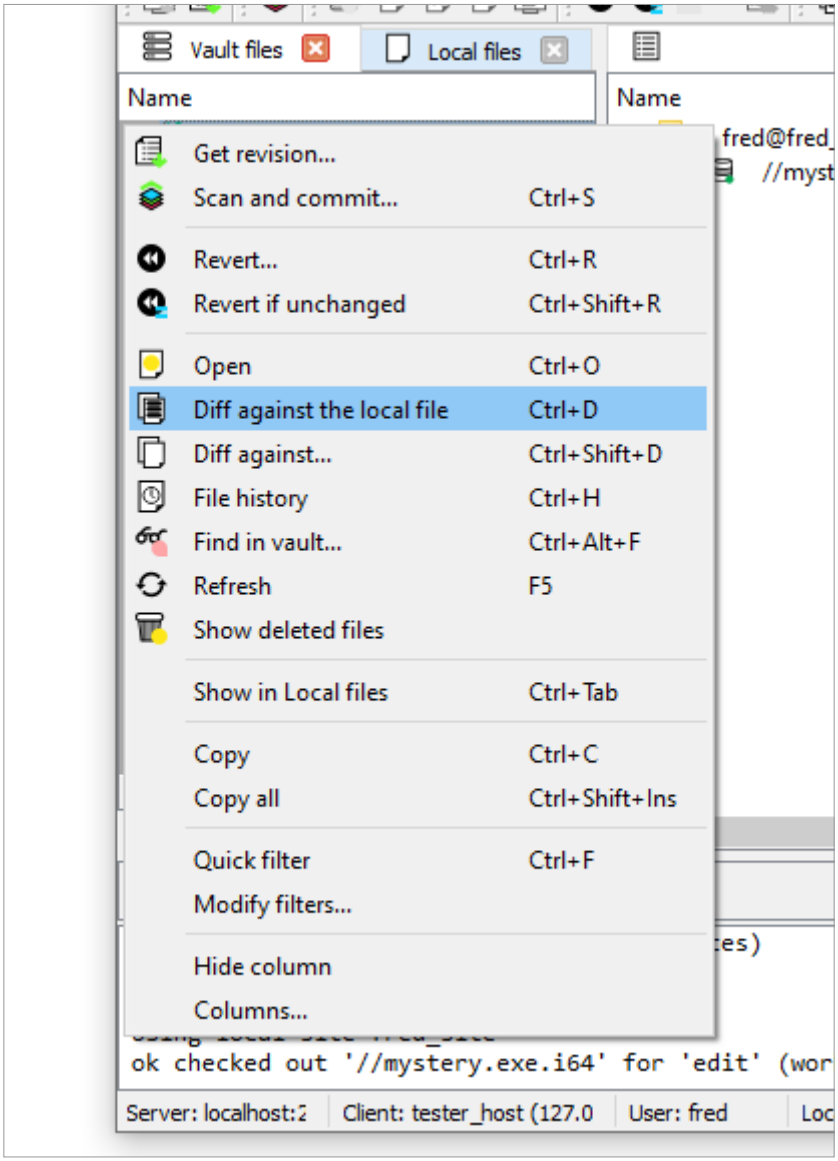


They are now all set, and ready to start analyzing their assigned aspects of the binary.

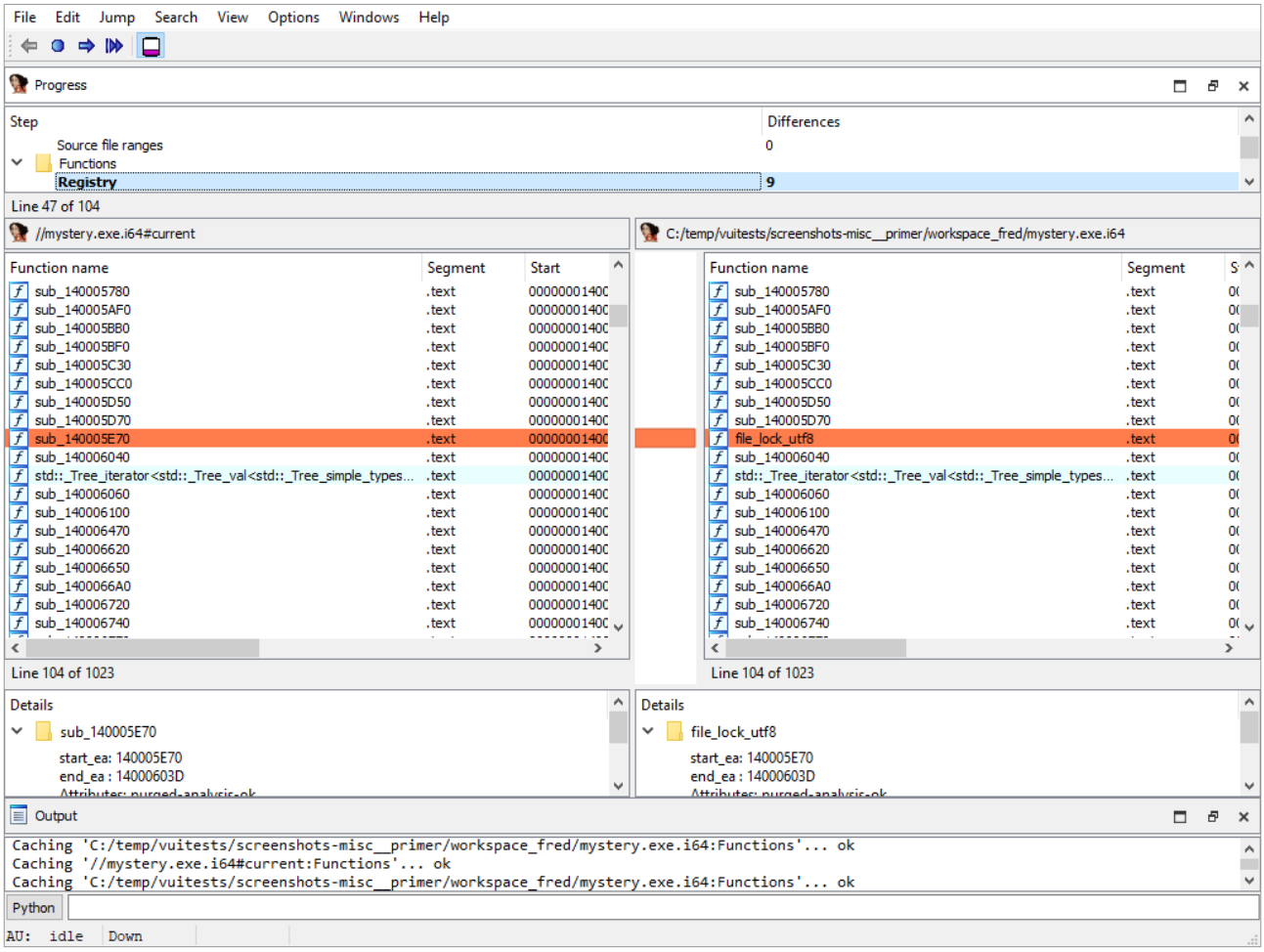
3.3. Fred reached a milestone

Fred has been doing good work, and is done with a first pass on the filesystem-related functions. He is now ready to submit his work so far, for others to benefit.

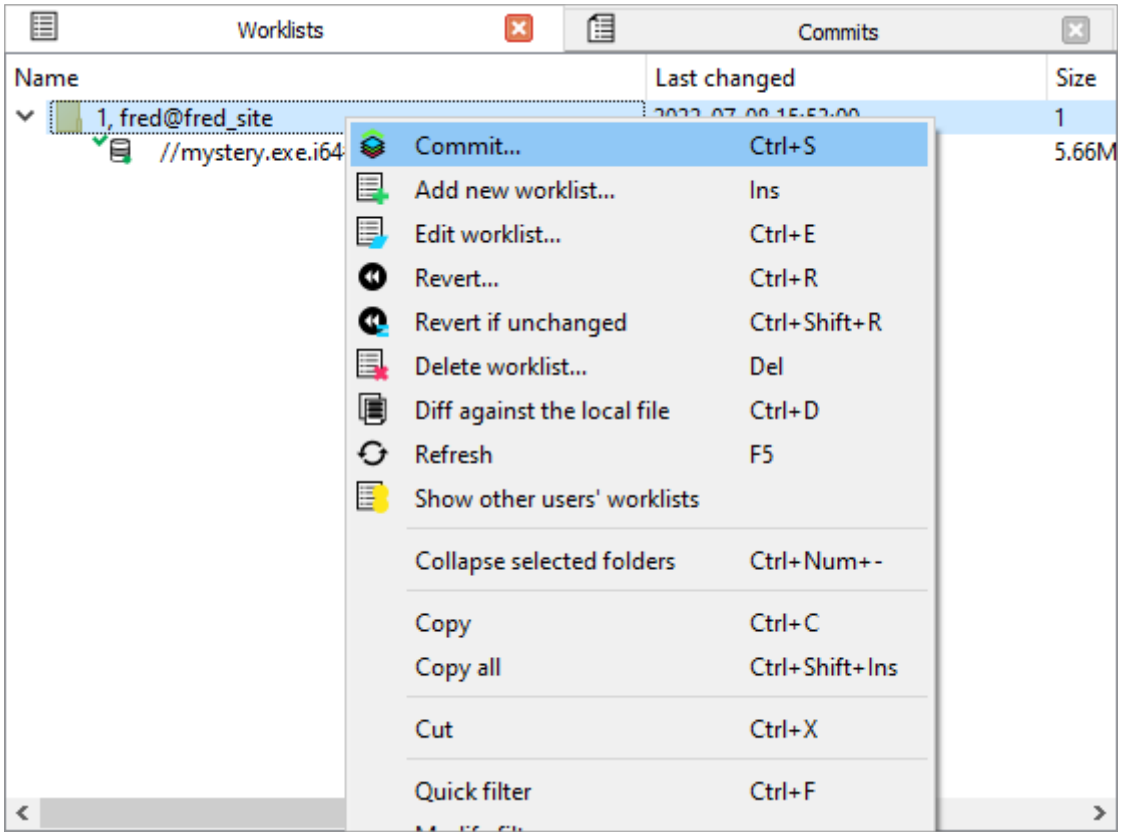
Before doing that, he wants to do a quick check to make sure that the changes he's done so far, all look good. He'll reach for `hvui`, and ask to see a diff of his local file against the server version:

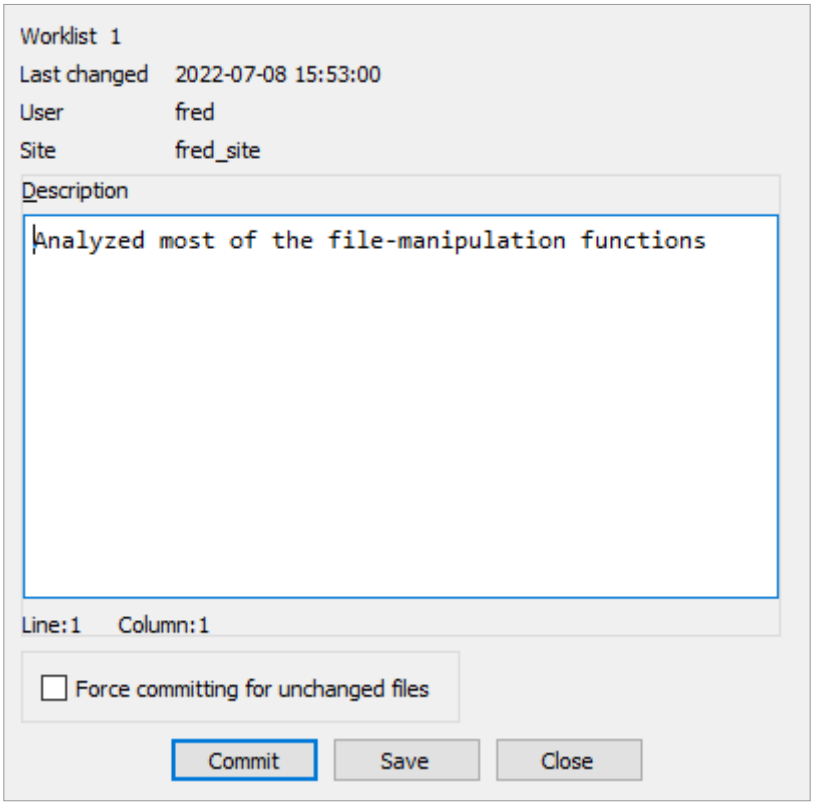


`hvui` will run IDA in a special "diff" mode, showing what changes have been performed:

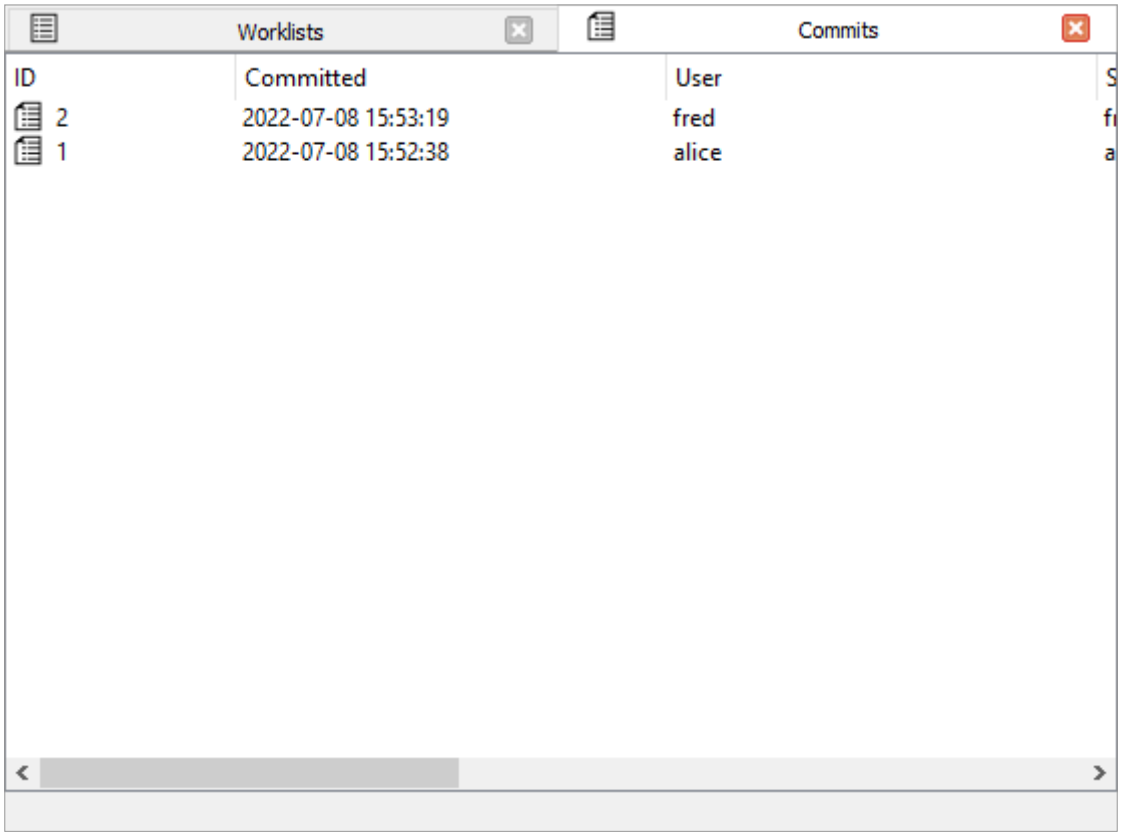


Satisfied with those, Fred will then write a small description…





and commit it to make those changes available to others:



Now that those changes are out of the way, Fred can continue analyzing other parts of the binary.

TIP Note that Harry doesn't need to import those changes *right away*: he's currently focused on figuring out some data structures.

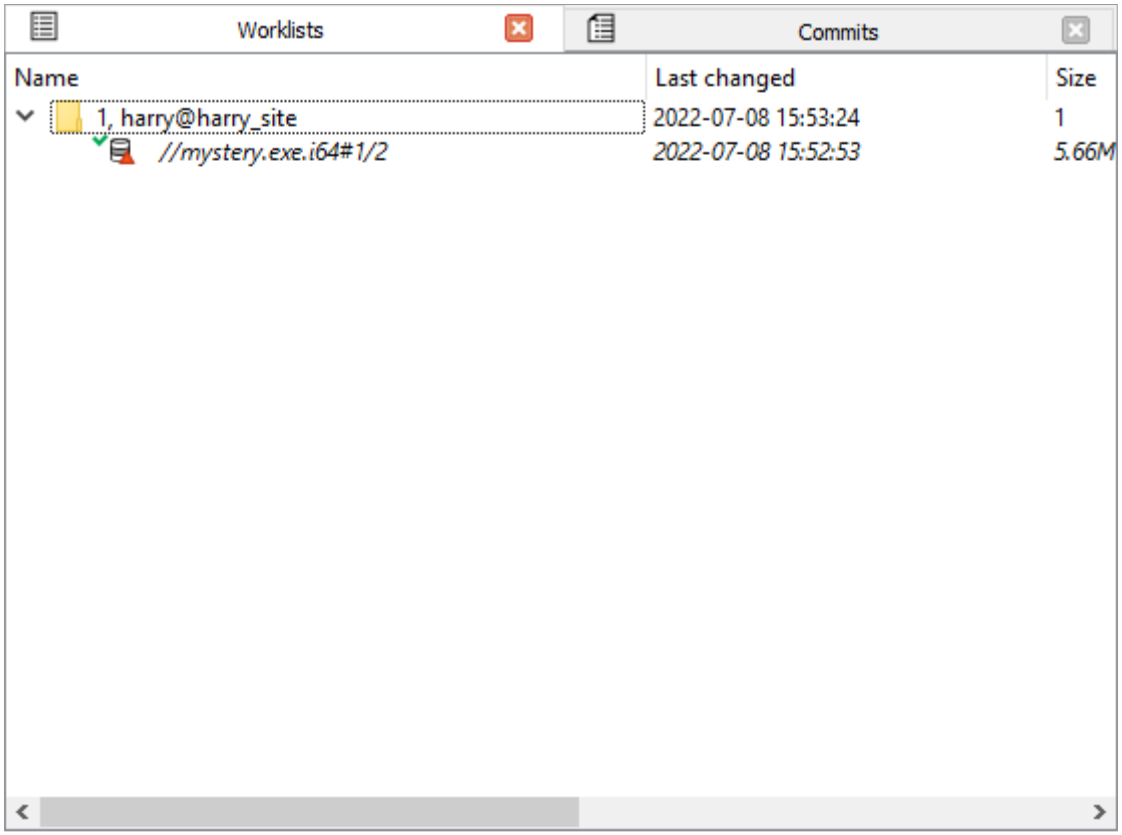
3.4. Harry has had a breakthrough

While Fred was busy analyzing filesystem-related functions, Harry managed to reverse-engineer some key data

structures, that appear to be used in many places in this binary.

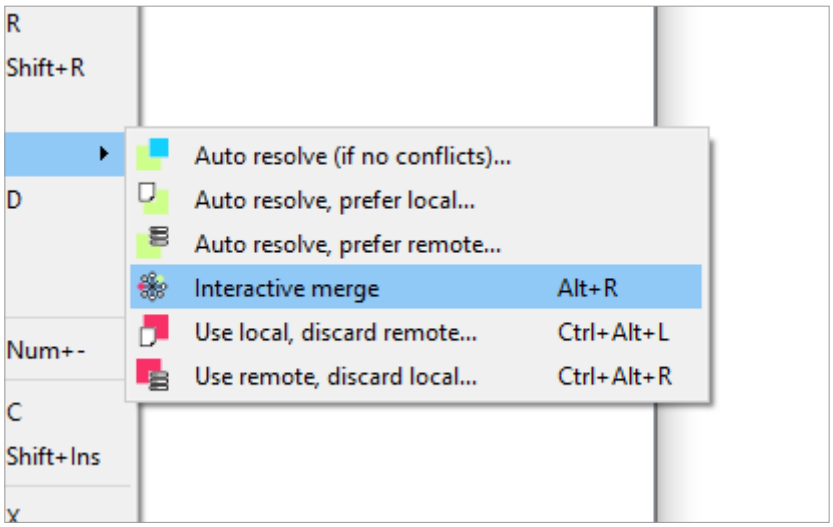
He has decided to make those available for others to use right away: he believes they will simplify their work.

However, When Harry goes to the "Worklists" widget to commit, he will notice that the client labels the file as outdated (notice the small, triangular badge as well as the `"#1/2"` hint):



3.4.1. Importing changes made by others, resolving conflicts

Before Harry can actually commit his changes, it will be necessary to first import the changes that were made in revision `#2` of the file.



TIP

Conflict-resolution algorithms

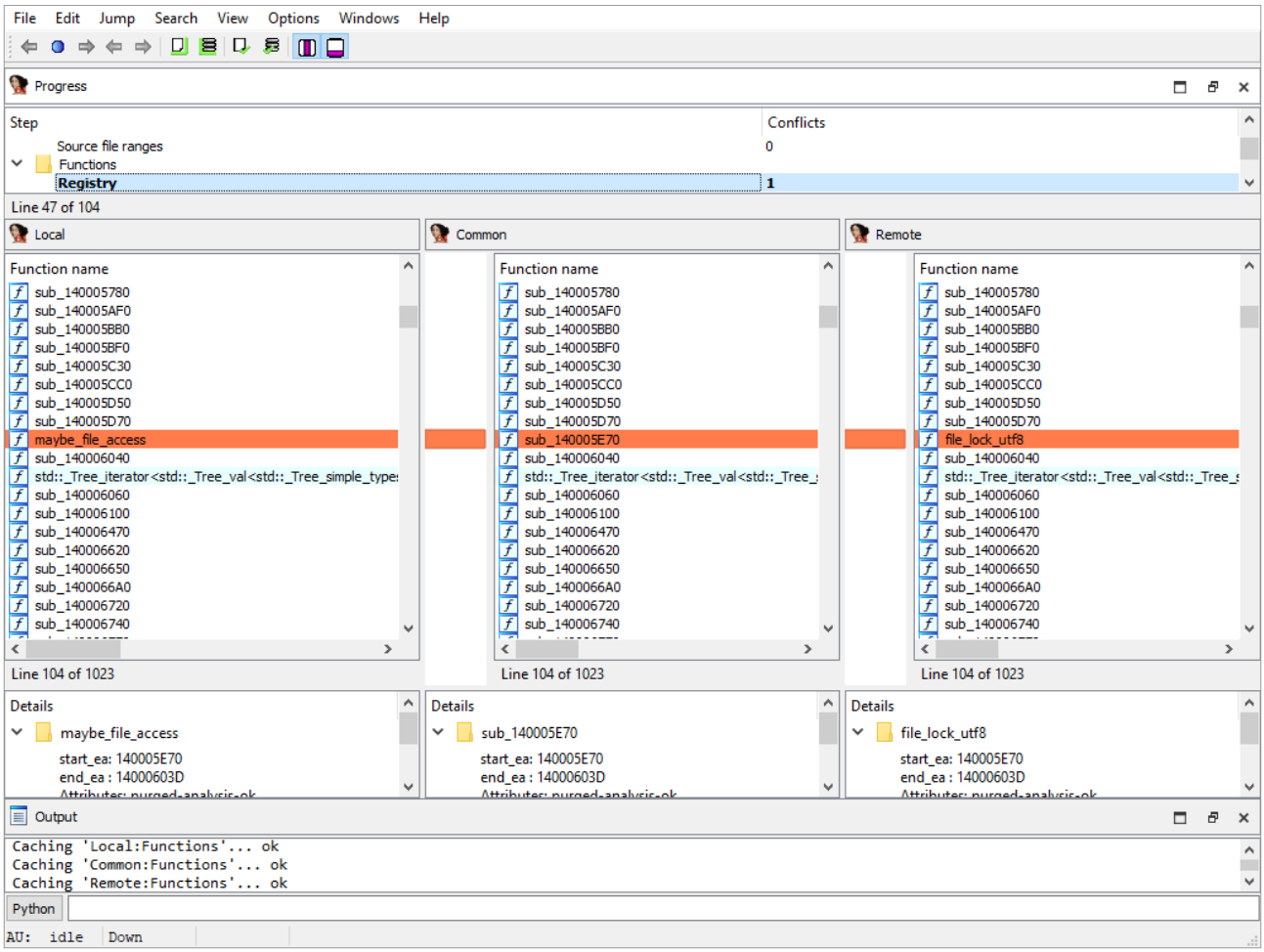
For the sake of the demonstration, we'll pretend Harry selected the "Interactive merge" option, to have fine-grained control over what gets picked in the event of a conflict.

There are many other merge modes, and in particular the "batch" modes that, in the event of a conflict, will automatically pick either the user's change or the other's change.

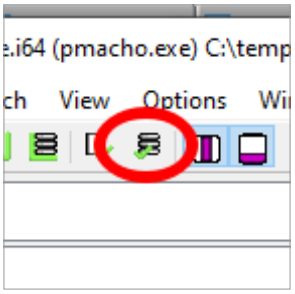
> In practice, "batch" modes will likely be used most of the time.

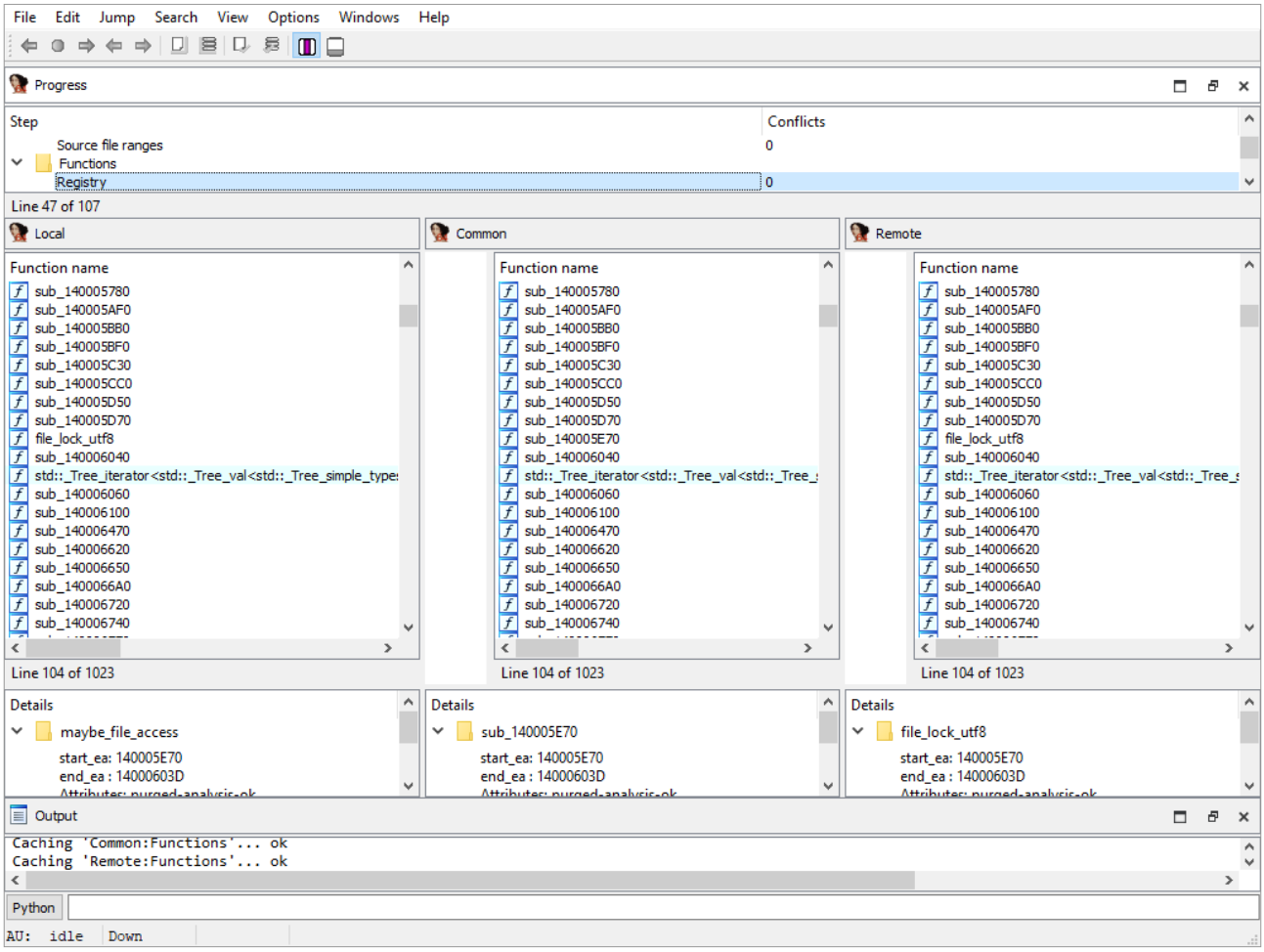
"Interactive merge" will cause IDA to open in merge mode, showing all conflicting changes. Harry will then be able to pick which change he prefers, at each conflict.

As it turns out, both Harry and Fred renamed the same function:

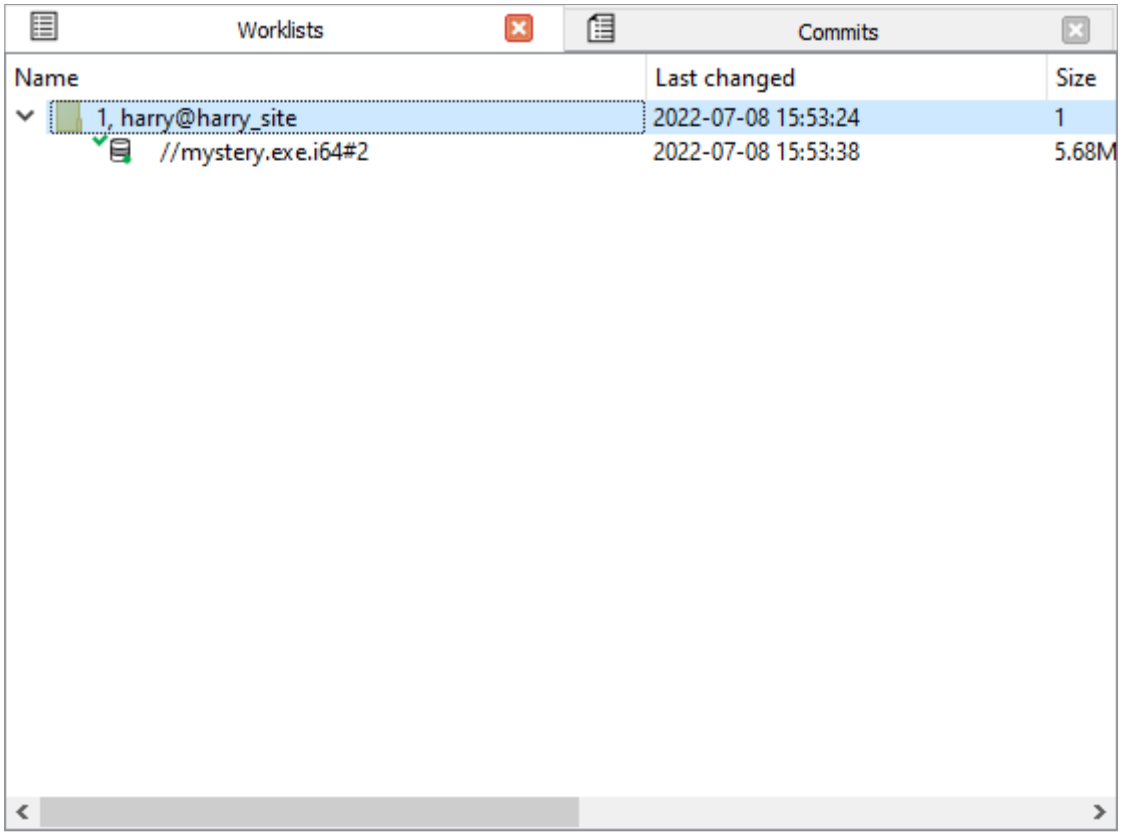


Since Harry finds Fred's version to be better, he'll pick that change:

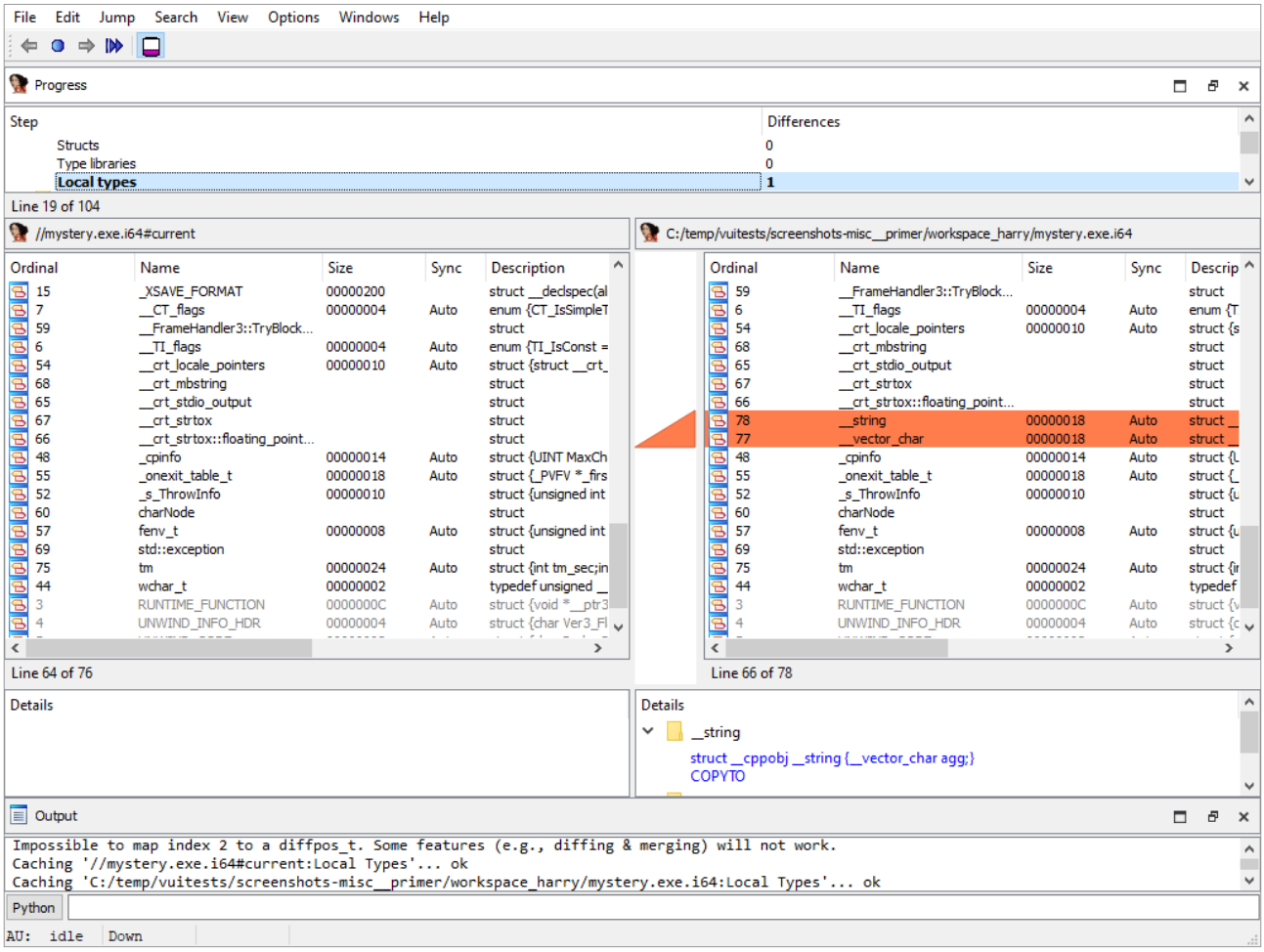




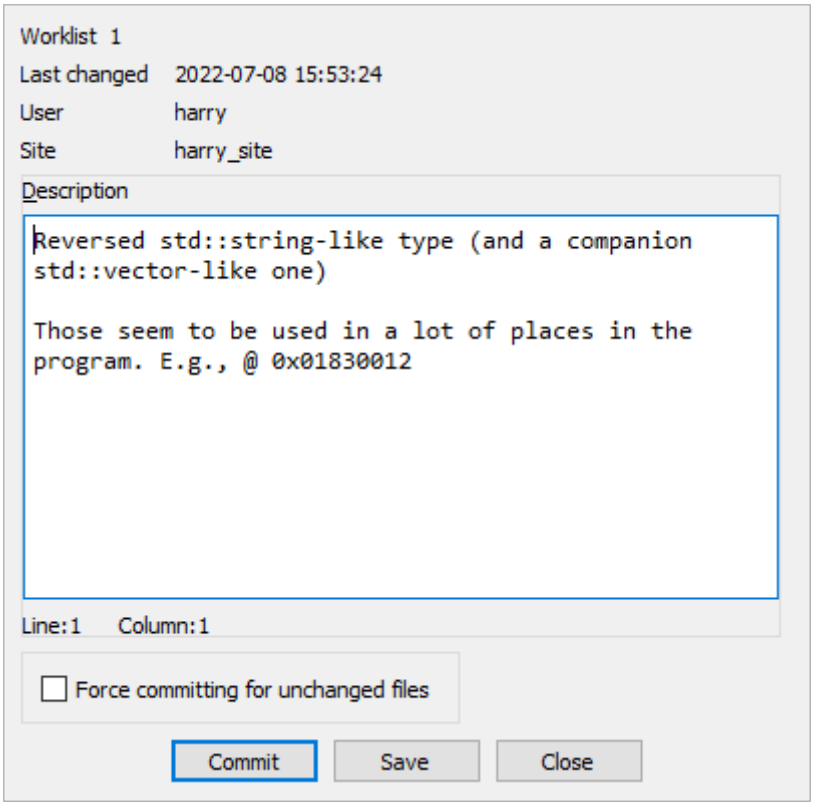
After all conflicts are resolved, IDA can be closed, and we can go back to `hvui`, which will now show the file as up-to-date:



Lastly, a quick check to ensure that the data structures are all in a good state for commit:



All good, let's commit!



3.5. In a nutshell

This is, in a nutshell, the purpose of IDA Teams: to propose robust, collaborative tools for teams of reverse-engineers to contribute to a project in the most efficient manner. And this is only the beginning!

4. Future plans

In its first release, IDA Teams will focus mostly on the "diffing" and "merging" aspects (the technical challenges to overcome to make this a possibility, were considerable.)

Later versions of IDA Teams will enjoy even more tools for collaboration (private "lumina" server(s), …)