Announcement
Timothy Leavitt · Nov 8, 2021

Open Exchange

Git for Shared Development Environments

If you're building solutions on IRIS and want to use Git, that's great! Just use VSCode with a local git repo and push your changes out to the server - it's that easy.

But what if:

- You're collaborating with other developers on a shared, remote development environment and want to avoid concurrent editing of the same file
- You're using editors based in the management portal for BPL, DTL, pivots, dashboards, etc. and want straightforward source control for your work
- You're still using Studio for some things and/or occasionally jump back there from VSCode - or, your team has not yet fully embraced VSCode, and some team members still want to use Studio
- You're working on a bunch of separate projects at the same time in the same namespace - say, several packages defined using the InterSystems Package Manager - and want to just work with all of them from one isfs editing view (rather than a bunch of distinct projects) with changes tracked in the proper git repo automatically

Then it wasn't so easy... until late last month, when we released **Git for Shared Development Environments (**Open Exchange **/** GitHub**).** You can get this extension using the InterSystems package manager:

```
zpm "install git-source-control"
```

Prior to this, the options for source control with Git were an old mostly-Windows-only, local development environment-only Git extension and a more recent Open Exchange project based on it but streamlining use a bit. There's also Port, which just deals with files and is version control system-agnostic.

What does git-source-control have that these packages don't?

- **Simple menu-based integration with git** that works on any operating system
- **A git user interface** to cover an expanding set of common git activities, without having to SSH over to the remote environment.
- **Concurrency control** for multiple users working in the same environment at the same time. Once you make changes to a class/routine/etc., it's yours until you discard or commit your changes. (We do have ways around this when needed, though!)
- **Package manager-awareness**: just zpm "load -dev /path/to/package" and, if /path/to/package/.git exists, changes to resources in your package will automatically be reflected in the right place on the server filesystem. The UI works with this too, based on the class/etc. from which it is launched.

All of this works from VSCode:

Spoiler

And Studio:

Spoiler

To give you control over and visual insight into your git repository:

Spoiler

We hope this enables your successful development of IRIS-based solutions, and welcome your feedback!

NOTE - To see the presentation launching this at the 2021 Global Summit, see this article: https://community.intersystems.com/post/video-git-gitlab-shared-developm...

#Best Practices #Embedded Git #Git #GitHub #Interoperability #InterSystems Package Manager (IPM) #Open Source #VSCode #InterSystems IRIS #InterSystems IRIS for Health
Check the related application on InterSystems Open Exchange