

Repo, Man
by
Daniel Messer

Daniel Messer(Tanglewood Hill Studio)
cyberpunklibrarian@protonmail.com
@cyberpunklibrarian@hackers.town

FADE IN:

INT. DAN'S OFFICE LOOKING THROUGH EXIT

After a few seconds, Dan wanders by

DAN
Oh hey! You're just in time! Grab a cup of
coffee and hop in the truck. I gotta go to
work!

CUT TO

DAN GRABBING COFFEE

CUT TO

DAN GRABBING KEYS AND BAG

CUT TO

DAN WALKING OUT THE DOOR AND GETTING IN THE TRUCK

During this shot make sure the house is visible.

CUT TO

MULTIPLE DRIVING SCENES

Music plays in the background as Dan is listening to the
radio.

CUT TO

DAN SIGNALLING AND PULLING INTO A LARGE PARKING LOT

CUT TO

DAN PARKING THE TRUCK

Dan shuts off the truck and looks up.

CUT TO

THE TRUCK PARKED OUTSIDE THE HOUSE

CUT TO

DAN SITTING IN THE TRUCK

Dan looks up the house and, after a beat...

DAN
Oh... yeah. Right. I work from home.

Beat

DAN
Huh. Okay well...

CUT TO

DAN IN HIS OFFICE

DAN
Hi!

I'm Dan, and I'm an ILS Administrator for Library Systems & Services as well as your friendly neighbourhood Cyberpunk Librarian. I've worked with the Polaris ILS for over two decades now, from the front side to the web site to the administrative explorer. Thanks for checking out my virtual presentation!

Before you start writing down a bunch of notes, I want you to know that there will be links at the end of this video that'll give you access to everything I talk about. Heck, it's a video, so there's a screenplay behind it. You can have that too, so you'll get a transcription as well. So kick back, have some snacks, and let's get started!

I've worked from home for years now and, when I first started, that was weird. Before that, all of the library IT people had to come in, gather around each other in a large area, and do their thing together in one space. The only reason I got sent to

work at home was because the library I was working for at the time needed to downsize their space and they decided that, ya know, maybe we *don't* need everyone here all the time.

I mean, almost all of my work happens in remote desktop. I could remote into servers from home just as well as I could from an office building. And, confidentially, I had better Internet speeds at home.

Then the pandemic happened and, over the course of just a few weeks, working from home wasn't so weird anymore. I won't point fingers at anything in particular, but I'm amazed how suddenly organizations and businesses pivoted from "We don't do work from home. We need you in the office" to "For god's sake don't come to the office." Turns out, working from home was possible all this time.

Now, many of the Polaris ILS admins I've talked to have either worked at home or they're still working at home.

So working from home meant a few changes for me, especially at the outset. I didn't have all the accoutrements of the office available so I needed to build up a stack of tools to help manage my occupation, information, documentation, communication, and lots of other words that end with the letters "tion." Sure I had access to a VPN and, thus, the shared drive at work. But let me tell you something...

For a storage space used by librarians, that shared drive was a nightmare world of documents, images, and media. Even worse, you never knew which documents were the current ones. Never mind which version is which, when you're looking at file names like "Circulation Policy Final Final Draft Final Revision Final.pdf" versus "Final Circulation Policy Revisions Final Changes.pdf"... I mean, what do you even do with that?

Moving away from that space gave me the opportunity to create my own space, one which would work for me, with me, and could

be modified if needed. I didn't have to ask someone to install software on my workstation or ask permission to set up something on a server. Turns out, when you're a nerd literally left to your own devices, you can revolutionize your workflow.

CUT TO

DAN AT DESK - DIFFERENT VIEW

DAN

I think it's a basic statement of library life that a pile is not a collection. A pile of files in a directory isn't as useful as a curated, ordered, and searchable collection. After working at home all this time, I built some structures that help me find the things I need, when I need them, as quickly as possible. And that's what I'm here to share today.

Of course, you don't have to do absolutely everything I'm talking about today. You might find something that you can adapt, maybe do it a little differently, use something that works for you. That's great! As a friend of mine says "Tune technology into your way of life and make the technology work for you."

Now, I have four important caveats, avertoj, warnings, advertisements... whatever.

First and primary: Leave confidential information in the most secure location. All of the tools and tips I'll share today are there to manage documents, data, code, and information that I can pick up online or in other places. Important and confidential stuff like patron information, item histories, circulation histories, and so on? I don't bring that into my system. Always always always leave your confidential data in the confidential databases on your secure Polaris servers.

Secondly, if you're going to build your own collection of tools then you're in charge of that collection. Librarians know all

about weeding, getting rid of outdated information, updating what needs updating, and so on. For me, that's no problem and I spend only a few minutes per month tidying my tools. But unless you want your own lovely system to turn into that nightmare world of a shared drive... keep your collection tight and clean!

Third: I'm a Mac, Linux, and open source doofus and that means I occasionally go above and beyond what any sane person might want to do when assembling a toolbox. Now, what do I mean by that? On the farthest wall from my desk are a bunch of computers that are all acting as servers. This little Mac mini here? It's running two virtual servers. One for my web development, BBS, and internet radio hobbies and the other is running a software stack that we'll talk about shortly. It's also running a digital periodicals library. And that's just *this* server.

Finally, I understand that many people *aren't* Mac and open source doofuses so I'll make sure to throw out things that'll work in Windows too.

All right then! Let's get started!

CUT TO

TITLE CARD - DOCUMENT MANAGEMENT

CUT TO

DAN AT HIS DESK

DAN

I can't speak too much about working with other ILS products, but one of the best things about Polaris is the documentation. You've got PDFs, you've got online docs, you've got Compiled HTML documentation of the database structure, and more. But beyond the ILS documentation I've got stuff about SQL Server, IIS, library policies and procedures, cataloguing procedures, software manuals and documentation, and so on.

Until recently, I helped manage libraries on different versions of Polaris. So I had documentation for 6.5, 6.7, and 7.0. That was its own special challenge when I needed to go look up some facet of whatever I needed to do.

And sure, you can pull down all those PDFs and DOCX files and drop them in a folder on your computer. You can even create a hierarchy of folders and drop them all in the proper locations. And soon, you too, will have a pile of files slowly growing into the nightmare shared drive. And while I can go spelunking about in a folder structure looking for what I need, you know what's better?

A search field.

And that's why I use document management software instead of simple folders. This is Keep It.

CUT TO

SCREENCAST OF KEEP IT

DAN (V.O.)

For my needs, Keep It is an excellent way to manage all the documents I may or may not need on a regular basis. All of the Polaris documentation gets dropped into Keep It along with any how to guides for me or for staff, library policies, procedural stuff, and whatever else. The great thing about Keep It is that you can drag almost anything into it and then index it, tag it, sort it, and then search for it. It's got built in text recognition so, even if that PDF isn't rendered in searchable text, it'll be searchable once Keep It runs through it.

Beyond library related things, I've got PDFs, images, documents, and so on for my personal things like my podcasts and writing projects.

While Keep It isn't a a free application, it's not expensive either. And it's been

well worth the money.

CUT TO

DAN AT HIS DESK

DAN

For the more hardcore solution, there's an app called DevonThink which is your Cadillac option.

CUT TO

SCREENCAST OF DEVONTHINK WEBSITE

DAN (V.O.)

DevonThink is a fantastic bit of software that costs just a little more than I care to pay for it right now. I won't say it's not worth it, but I will say that Keep It already does everything I need. I dunno, maybe if they have a sale or something.

CUT TO

BACK AT DAN'S DESK

DAN

Now, if you're not working on Macs then try Microsoft OneNote if you haven't already. OneNote does almost everything Keep It does and since you're already working in a Windows environment, OneNote plays very well with Word, Excel, and PowerPoint. OneNote is an oddity in the Microsoft sphere because everyone sees it on their computer, but I don't know a lot of people who use it. It's not bad, and I used it for a couple years myself.

There's also an app called Zoot that's available for Windows. I gave it a good run and while it's powerful, I found the UI to be a bit... busy. But hey, check it out if you have the interest.

CUT TO

SCREENCAST OF ZOOM WEBSITE

It's not an accident that I started out with document management because document management is knowledge management. I don't know about you but I'd say that most of the time when I'm searching documentation, it's to confirm something I think I already know, but don't want to be wrong about.

CUT TO

SCREENCAST OF SEARCHING POLARIS DOCS IN KEEP IT

DAN (V.O.)

I've saved lots of time with the Polaris docs alone because I no longer have to remember which document covered the thing I need to find. I can just search all of them.

CUT TO

DAN IN HIS OFFICE

DAN

In the way of a bonus pick, I wanted to mention something that's not quite document management but is more focused on *documentation* management.

CUT TO

SCREENCAST OF BOOKSTACK

DAN (V.O.)

You may have seen it a time or two if you've checked out the Useful Polaris SQL website. It's called BookStack and it's a free and open source way to maintain documentation that you and your team have written on your own. The great thing about it is how it works like a stack of books, hence the name. You create a Shelf, put

Books on it, and then fill those books with Pages.

And the best part is, all of that becomes searchable once it's in the system. If someone can't find something within the set hierarchy, well, that's what the search box is for.

The system supports code too, which is why I use it for the Useful Polaris SQL website. It handles syntax highlighting for SQL, C#, PHP, Ruby, CSS, Python, HTML, and a lot more. So if you and/or your team are looking for a way to manage your process documentation and such, this is a free and open source system. It's a server based software, so you'll need to install it on a server or in a VM.

Okay, cool, so that's documents and documentation. But there's something else I want to talk about when it comes to building your own repositories and it has to do with something that lots of ILS admins have to deal with on the regular.

CUT TO

TITLE CARD - MANAGING YOUR SQL (AND MORE)

CUT TO

DAN AT HIS DESK

DAN

I don't know about you, but sometimes the shortest distance between a librarian's question and the answer is a dozen lines of SQL. And if I'm going to take the time to write that stuff, then I'm going to take the time to save it in case it might be useful later. And if I'm going to take the time to save it, I'm going to make sure I can find that code later on.

Now, when I talk about curating your code and making it findable, I can already sense that some people are nodding their heads, some people have a pen hovering over a pad of paper, and a couple of you are wondering

if it's okay to drink beer during a virtual presentation.

Let me be the one to assure you that it is. Given my topic, it might be helpful.

But when I throw out the idea that if you have a pile of files ending with the letters ".sql" sitting in a folder somewhere on your computer and you're relying on your glorious naming abilities to find that file six months later?

Well, there are better ways.

So to cut right to it: I'm amazed, and ever so slightly horrified, how many ILS admins who write a bunch of SQL code don't use git.

Now before you throw your pen down in disgust, hear me out. Git doesn't have to be hard, especially if you're the only one using it. But, speaking as a nerd who's been writing semi-competent code since the Commodore 64, give me a few minutes to explain why I use it and why it might be useful for you.

And for those who already use it, kick back and nod sagely.

To offer a quick and dirty explanation of git, I'll say it's a way of managing your code not just from the aspect of storing your files but tracking the changes. That comes in handy when you make a few changes here and now your query isn't working. If you're using git, you can revert back to a previous version of your code and start fresh. A collection of code in a particular git system is called a repo, which is short for repository.

But more than that, git is to code what a document manager is to your PDFs. Don't remember exactly what the title of a file was, but you do remember some of the code that was inside? That works, just search for it. I've saved time and stress more than once when a Director has a stat request for data, and I know I've got a query that does the job, but I don't

remember what it's called. I barely remember what I had for lunch yesterday. Often, I'll recall within the code itself. A quick search later, and I'm working the ticket.

Quick side tangent here. For your own sanity, comment your code. Especially if you're going to drop it into some kind of management system. A few lines at the beginning of a query that describe what it does? That's gonna help you find that query weeks and months later. Heck, use hashtags to categorize things. Anything like that will be helpful when you search for it later on. Give future you the best chance of finding something that present you did today.

Now, the first thing you need to use git is a git server. Thankfully there are two major ones that most people have heard of: GitHub and GitLab. Me, I prefer GitLab for reasons we'll get into here in a few minutes. The good news is that both GitHub and GitLab offer free accounts and private repositories. For the most part, unless I'm sharing the code on the Useful Polaris SQL sites... I keep my SQL repo private. Not because I don't want people seeing the absolute magnificent brilliance of my code, but because I may have sensitive information in there. Honestly, people would probably laugh at the state of my workday queries.

But the thing is, git is a standard. So you don't have to use GitHub or GitLab if you don't want to. For years I used some software called Gogs which I hosted on my own server. And that's something you can explore too. You can host your own git system, on your own devices and servers, and it doesn't have to cost anything. I said I'd tell you why I prefer GitLab over GitHub and it comes down to hosting your own system. You can host your very own GitLab on your servers and it will look and act exactly like the public website at GitLab.com. I've got all of my private projects on my self-hosted GitLab and my public projects go up to the public GitLab.

If you want to share your queries with the world, you can do that. Or you can keep everything private. Or you can do both, that's the basis of Useful Polaris SQL project.

Now, I don't want to bore you with a whole walkthrough of the git process because there are multiple ways to approach it. I use the command line, but there are plenty of software solutions out there if you want to use something friendly and graphical. Tower and GitKraken are some good examples. Heck, if you're using Azure Data Studio or Visual Studio Code to write SQL, they've got git integration built right in. I'll have links to multiple solutions along with some resources on how to get started with git. For this presentation, I want to show you what you get outta git and how easy it can be.

So, by way of a demonstration:

CUT TO

SCREENCAST OF DAN'S GITLAB

DAN (V.O.)

Here we are in my self-hosted GitLab. As you can see I've got a bunch of SQL queries here. This is all the stuff I figured was worth saving, indeed, that's the description.

CUT TO

SCREENCAST OF DAN'S TERMINAL EMULATOR

DAN (V.O.)

Back on my computer, these queries live in a directory that I'm tracking with git. I've made a few changes to some of these files and I want to send that up the repository. Here's all I have to do.

First, I tell git to add everything in this directory so it can be tracked.

Then I tell git to commit the changes and I

provide a quick message about what these changes are.

Finally I tell git to push these changes from this computer, which is called the origin, up to main branch of the repository, which is unsurprisingly called main.

And that's it. I've committed those changes and they're in the repo now.

CUT TO

SCREENCAST OF DAN'S GITLAB

Scrolling through and pointing out updates

DAN (V.O.)

See? Here they are.

And, since we're here, I can search this repository in various ways. Say I need to find a query that produces data about circulation statistics. Now, know yourself and know your own jargon. I rarely refer to "circulation" when the word "circ" is three syllables shorter. So I'm going to search for "circ stats."

Searching for circ stats

DAN (V.O.)

Oh look, here are three different queries that not only have "circ stats" in the name, but also in my comments.

CUT TO

CLOSE UP OF DAN

DAN

Comment your code, people.

CUT TO

SCREENCAST OF DAN'S GITLAB

DAN (V.O.)

Or hey, I don't remember what the query was called. All I remember is that I joined the NotificationLog table in the Transactions database. Okay...

Runs a search for
PolarisTransactions.Polaris.NotificationLog

DAN (V.O.)

And here we go.

CUT TO

DAN IN HIS OFFICE

DAN

I'm only scratching the surface of what git can do. If you invest just an hour of time into learning what it can do for you, you'll be amazed, even if you decide not to use it.

But hey, I'm not gonna try and pull a fast one on you. You know what else is pretty good for handling a code repository, offering you search functionality and the ability to curate your collection?

A document manager. Feel free to roll this video back a few minutes if needed. Because you can totally stuff your code into a document management system. The thing you'll miss the most would be git's ability to track your changes. Since I don't just work in SQL but also in PHP, Python, HTML, CSS, and so on... I really want to track those changes because that's incredibly helpful to me. If it's not such a big deal to you, then hey, make the technology work for you.

But one last nerdy tidbit before we get outta here: Like Keep It and other document managers, git can track pretty much anything you want. You can keep tabs on all

kinds of projects by pushing them up to a git repository and that includes writing.

Because even this presentation, from outlining to scripting to editing to lists and links... I tracked all of it via git.

So... if you have the interest, check out this link. I'll have all kinds of things for you including further information and walkthroughs on how to use git. You'll find links to various document management software. And hey, since this is a scripted video, it's not too hard to provide a transcript. So you'll find the screenplay there among all the goodies.

If you have questions, drop me an email or find me on the fediverse. And hey, I'm usually hanging around the IUG Discord too. I'm happy to answer whatever I can!

Thanks for watching my presentation, enjoy the rest of the conference, and I'll see you soon at the Q&A session!

THE END

SCENE INDEX

INT. DAN'S OFFICE LOOKING THROUGH EXIT	1
DAN GRABBING KEYS AND BAG	1
DAN WALKING OUT THE DOOR AND GETTING IN THE TRUCK	1
MULTIPLE DRIVING SCENES	1
DAN SIGNALLING AND PULLING INTO A LARGE PARKING LOT	1
DAN PARKING THE TRUCK	1
THE TRUCK PARKED OUTSIDE THE HOUSE	2
DAN SITTING IN THE TRUCK	2
DAN IN HIS OFFICE	2
DAN AT DESK - DIFFERENT VIEW	4
TITLE CARD - DOCUMENT MANAGEMENT	5
DAN AT HIS DESK	5
SCREENCAST OF KEEP IT	6
DAN AT HIS DESK	7
SCREENCAST OF DEVONTHINK WEBSITE	7
BACK AT DAN'S DESK	7
SCREENCAST OF ZOOM WEBSITE	8
SCREENCAST OF SEARCHING POLARIS DOCS IN KEEP IT	8
DAN IN HIS OFFICE	8
SCREENCAST OF BOOKSTACK	8
TITLE CARD - MANAGING YOUR SQL (AND MORE)	9
DAN AT HIS DESK	9
SCREENCAST OF DAN'S GITLAB	12
SCREENCAST OF DAN'S TERMINAL EMULATOR	12
SCREENCAST OF DAN'S GITLAB	13
CLOSE UP OF DAN	13
SCREENCAST OF DAN'S GITLAB	14
DAN IN HIS OFFICE	14