COPIA

Modern Backup Strategies for Industrial Automation

Benefits and Challenges of Backing Up Control Data

It's widely known that regularly backing up a company's data protects vital files and minimizes disruptions when issues arise. Backups safeguard against human errors, hardware failures, malicious attacks, power outages, and natural disasters. Ultimately they save time and money when failures occur.

The responsibility for most data backup falls to IT departments. IT teams often implement strict backup policies to protect business data, including customer, inventory, accounting databases, and employee files. Modern databases and data management tools assist their efforts. For example, in software development environments, Git tools are used to ensure that all data changes are tracked and stored in a centralized location for easy copy.

Unfortunately, there are areas of the company where IT's reach is often hindered, and crucial data is not backed up with the same rigor. One such area is the manufacturing network. Manufacturing networks are vital and complex, providing the programs and configuration files to the numerous control devices running the production machinery. Because of the common practice of isolating these networks, backing up their data usually falls to the operations team.

Backing up the automation programs on these networks can require significant work. The sheer number of devices, different brands, proprietary languages, locations, and people involved with these systems present challenges. As a result, control programs and related files are often managed using manual processes and stored in different locations without clarity regarding which files are the latest working versions. Many companies struggle to find the correct files to restore service when a running control device fails. In a worst-case scenario, production lines may be down for extended periods, losing companies hundreds of thousands of dollars or more.

Many companies struggle to find the correct files to restore service when a running control device fails.

A Manual, Cumbersome Process

The common practice for backing up a PLC is to directly connect a PC to the device and utilize a software tool from the manufacturer to facilitate the retrieval and saving of the program files. Once backed up, teams can copy the files to a centralized directory, preferably off site or in the cloud, for added security. This effort can become tedious and time-consuming if a factory has many devices and utilizes multiple PLC brands. For example, employees may use Rockwell Automation® Studio 5000® to retrieve device backups if it uses Allen-Bradley® PLCs. However, if they also use Beckhoff® controllers, they will need to use Beckhoff's TwinCAT® software too.

There are obvious disadvantages to this manual approach.

- Due to the amount of effort, companies generally will perform this manual process at an irregular and less-than-optimal frequency. They may not capture the latest updates to the PLC program.
- Additional work is required to keep a backup history and ensure older backups are not overwritten.
- Teams may waste valuable time backing up devices whose program files have stayed the same.

Some PLC vendors have attempted to improve the backup process with additional software or hardware features. Rockwell Automation FactoryTalk® AssetCentre®, for example, provides scheduling for automated device backup and will support a few different device types. Some controllers, such as a Siemens S7-1200 CPU, can use an optional SD memory card to help load and retrieve programs onto the PLC. Most large factories face lengthy manual processes to ensure a comprehensive data backup despite these improvements.

Change Detection with Visual Compare Adds Greater Value

It was mentioned above that keeping a history of backed-up files ensures older versions can be restored when needed. This routine storage of the previous version is standard practice in most traditional software development environments (outside of manufacturing).

Once a directory of older file versions is recorded, the ability to compare and visually render the difference between versions adds even more value. It enables developers to document a timeline of necessary changes for future development streams. Furthermore, it can help identify unauthorized changes. Seeing the difference in code accelerates troubleshooting and resolution if problems arise due to code edits.

Why Do Control Programs Change on the Fly?

One of the challenges faced by the operations team is that technicians can change control programs by connecting directly to the device. These “on the fly” modifications ensure the machines are running correctly. Unfortunately, these changes are often not documented and can go unnoticed by colleagues. Daily backups ensure that these types of changes are captured. Here are some real-world examples of situations where PLC code may be changed on the factory floor:

Hardware changes

- An axis load encoder was vibrated loose or knocked off the machine. Hence, a technician needed to reinstall or replace the sensor and reset the axis home position.
- An I/O card failed and needed to be replaced with a similar but not identical card kept on hand. The technician needed to update the hardware configuration on the PLC.

Reworking parts or custom runs of parts

- A custom order requires parts built to specifications slightly different from standard runs. A technician may need to modify the PLC files to apply slightly different parameters.
- Rework is needed on a pallet of parts, requiring a technician to edit the PLC program to skip some sequence steps.

A change in the environment causes inconsistent or flawed machine output

- An electrician may have installed new overhead lights, which impact the automated inspection system. New parameters need to be set.
- A garage door to the factory is opened, and humidity has risen. As a result, control logic needs to be adjusted to ensure the epoxied parts stay in the vacuum desiccator for longer.

Previous programming errors

- There’s a bug that wasn’t caught during initial testing, and the machine is ejecting unfinished parts. The program needs to be edited quickly.
- A program adjustment inadvertently impacts a downstream process, requiring further changes.

Regardless of the reason, it’s important to recognize that machine code is occasionally updated after a machine’s initial installation and commissioning. Recording and tracking those updates securely, with a clear view of what exactly changes, can greatly benefit any industrial controls team, OEM, or system integrator.

An Ideal Solution

Considering all the challenges manufacturers face in backing up control devices and understanding the proven processes that IT utilizes when backing up data from other business departments, it becomes easy to envision the ideal backup solution for the factory floor.

An ideal solution has the following characteristics:

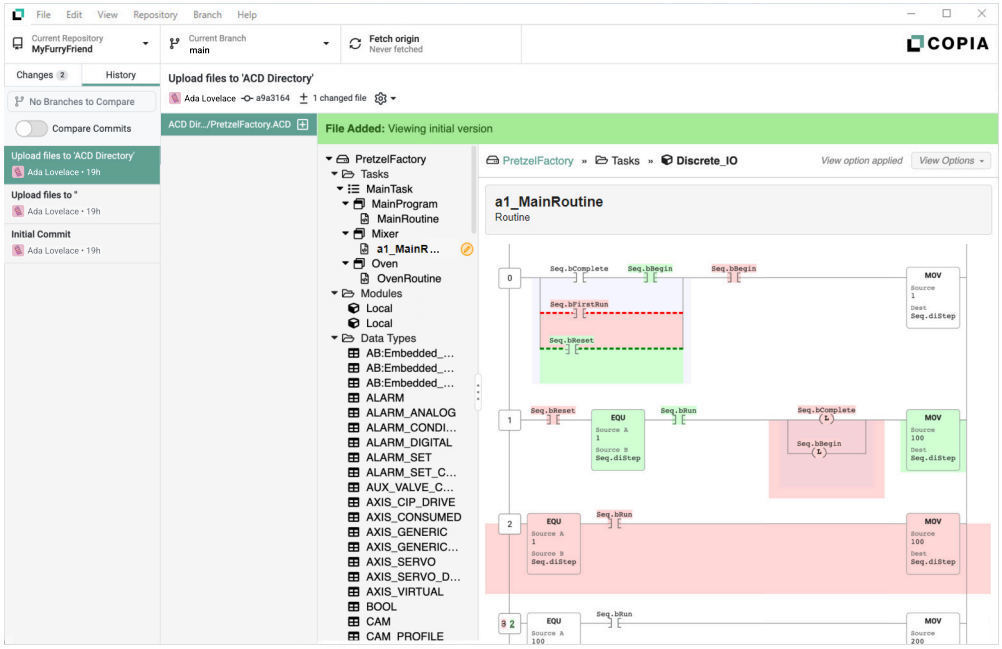
- It will work with different control devices.
- Backups can be executed according to a schedule or performed on demand.
- It will retain historical backups, only storing new files when it detects changes.
- It can compare a recently backed-up file to a previous version and alert users to changes.
- It can visually display the differences between file versions.
- It will store all data in a centralized, secure location, preferably remote.

Copia Delivers Modern Tools for Device Backup

Copia Automation was founded to bring modern development tooling to industrial automation. Its initial offering provides Git-based source control to PLC programming.

Git tracks all code versions during development, provides greater context about differences, and stores the code in a centralized repository. It empowers teams to collaborate while writing and reviewing programs.

Copia understands the visual languages of PLC programming, enabling users to see *Ladder Logic* or *Function Block Diagrams* outside the original development environments. Copia renders these formats and highlights differences between versions, empowering control teams to check code faster and more thoroughly. Users of Copia report greater productivity and code quality.



Copia Automation can display ladder logic with changes highlighted

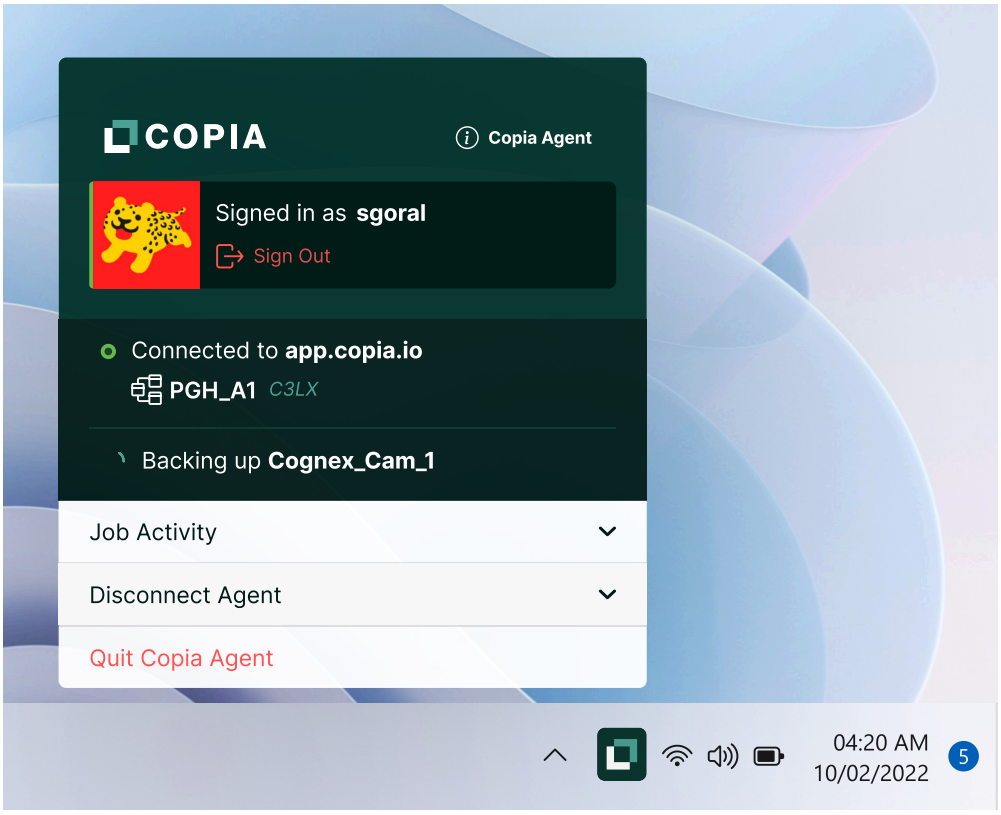
In 2022, Copia announced its DeviceLink™ product, developed to address the challenges of device backup. When coupled with Copia's Git-based version control, DeviceLink streamlines the entire backup process for the factory floor.

A comparison of DeviceLink's advantages over manual backup processes can be seen in the following table.

	DeviceLink	Manual Processes
Automatically back up different control devices	✓	✗
Schedule backups or perform backups on-demand	✓	✗
Maintain a history of backups for easy access	✓	✗
Automatically compare the latest backup with previous versions	✓	✗
Proactively notify users when code changes are detected	✓	✗
Highlight the differences between versions	✓	✗

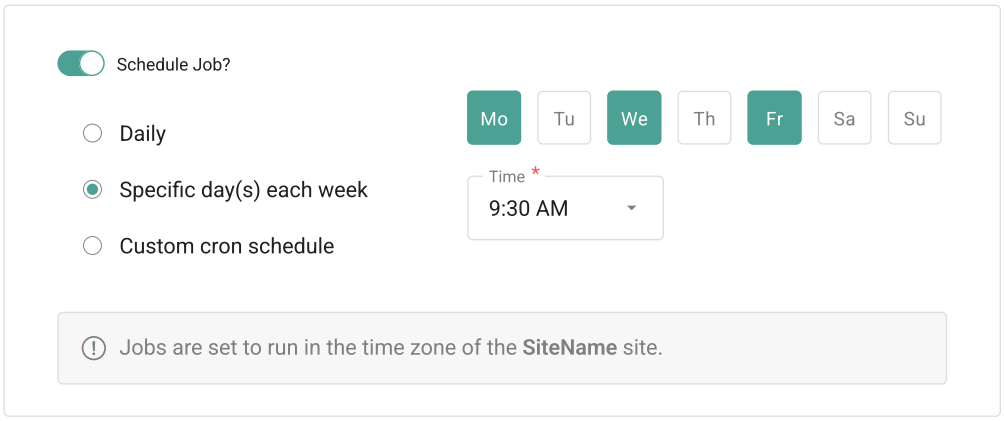
How DeviceLink Works

Copia Automation's DeviceLink requires a lightweight client-side application (an agent) to be installed on a PC with access to the manufacturing network. This agent will back up the code on the customer's devices and send it to Copia's Git-based source control system.



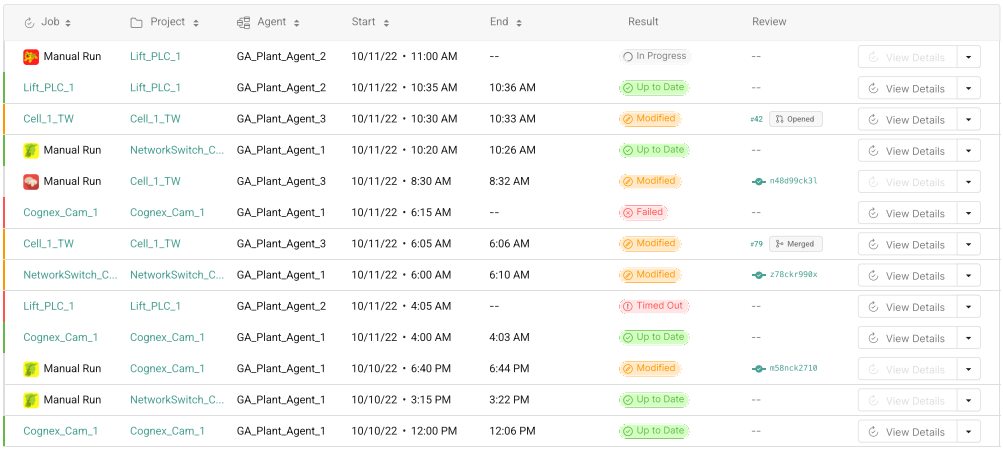
Copia's Agent is a lightweight app that can be installed on any PC or IPC on the manufacturing network.

Within the Copia application, jobs can be scheduled to back up devices and compare the files to their previous backed-up version. The jobs will control the frequency at which the Copia Agent will back up a device. They can be scheduled or run on demand.



It is easy to schedule automatic backups in Copia

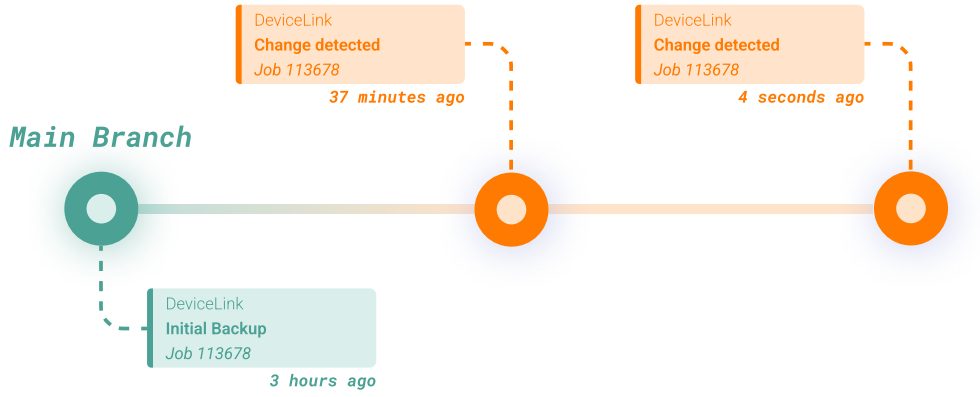
To perform backups, DeviceLink can utilize PLC development environments (such as Siemens® TIA Portal, Rockwell Automation® Studio 5000 Logix Designer® Studio 5000, Beckhoff® TwinCAT®, CODESYS® v3, etc.) or connect directly to devices via FTP or custom scripting. FTP and custom scripting usually require file lists to tell Copia which files to retrieve and ignore. Copia displays a dashboard of all jobs and their status, so users can easily see which jobs are running, have completed, failed, and when file changes are detected. Alerts can be customized so that users get notifications when file modifications are discovered.



Job	Project	Agent	Start	End	Result	Review	
Manual Run	Lift_PLC_1	GA_Plant_Agent_2	10/11/22 • 11:00 AM	--	In Progress	--	View Details
Lift_PLC_1	Lift_PLC_1	GA_Plant_Agent_2	10/11/22 • 10:35 AM	10:36 AM	Up to Date	--	View Details
Cell_1_TW	Cell_1_TW	GA_Plant_Agent_3	10/11/22 • 10:30 AM	10:33 AM	Modified	#42 Opened	View Details
Manual Run	NetworkSwitch_C...	GA_Plant_Agent_1	10/11/22 • 10:20 AM	10:26 AM	Up to Date	--	View Details
Manual Run	Cell_1_TW	GA_Plant_Agent_3	10/11/22 • 8:30 AM	8:32 AM	Modified	n48d99ck3l	View Details
Cognex_Cam_1	Cognex_Cam_1	GA_Plant_Agent_1	10/11/22 • 6:15 AM	--	Failed	--	View Details
Cell_1_TW	Cell_1_TW	GA_Plant_Agent_3	10/11/22 • 6:05 AM	6:06 AM	Modified	#79 Merged	View Details
NetworkSwitch_C...	NetworkSwitch_C...	GA_Plant_Agent_1	10/11/22 • 6:00 AM	6:10 AM	Modified	z78ckr998x	View Details
Lift_PLC_1	Lift_PLC_1	GA_Plant_Agent_2	10/11/22 • 4:05 AM	--	Timed Out	--	View Details
Cognex_Cam_1	Cognex_Cam_1	GA_Plant_Agent_1	10/11/22 • 4:00 AM	4:03 AM	Up to Date	--	View Details
Manual Run	Cognex_Cam_1	GA_Plant_Agent_1	10/10/22 • 6:40 PM	6:44 PM	Modified	n58nck2710	View Details
Manual Run	NetworkSwitch_C...	GA_Plant_Agent_1	10/10/22 • 3:15 PM	3:22 PM	Up to Date	--	View Details
Cognex_Cam_1	Cognex_Cam_1	GA_Plant_Agent_1	10/10/22 • 12:00 PM	12:06 PM	Up to Date	--	View Details

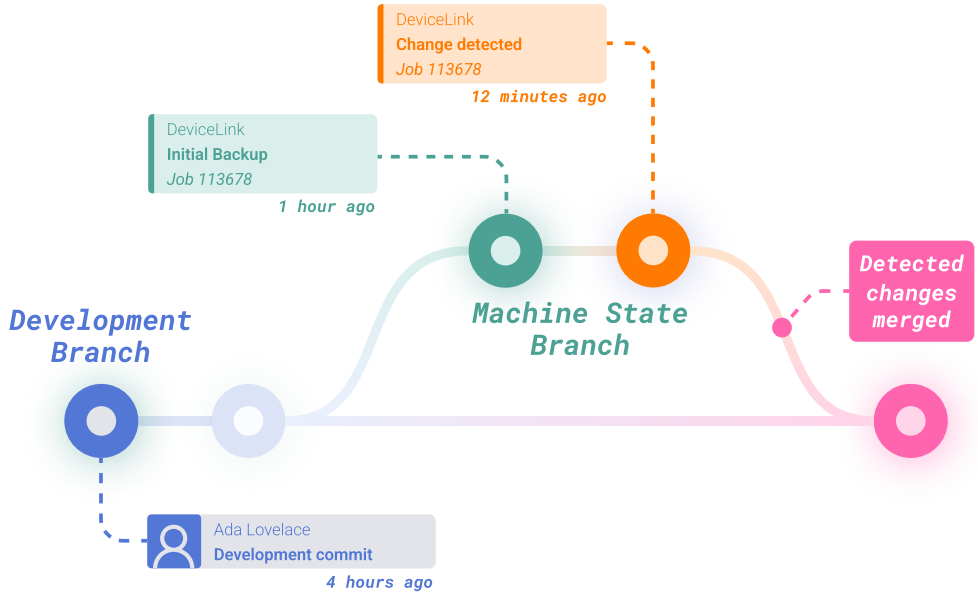
Copia shows all jobs and their status within one dashboard

Copia uses Git's branching and merging capabilities, which provide extreme flexibility for storing and managing backups. For example, if a machine technician simply wants to save and track all PLC code changes, they may use Copia to store the backups in a single main Git branch. Each new backup file version will be stored along that branch (as a committed change) and accessible.



In more complex environments, users may want to use multiple Git branches. For example, a System Integrator may actively develop control code for an entire manufacturing line. Using Copia, they set up a main development branch to track their ongoing work. When they start commissioning, they can utilize a second branch to capture field adjustments from their machine backups.

At any point, they can compare the machine state to the development branch and, if they desire, merge those changes to the development branch. Future development versions will then incorporate the latest field adjustments.



It's important to note that since Copia is cloud-based, all files and their histories are tracked in a secure, centralized repository. There is a complete audit trail of all activity, and any historical backup can be retrieved and utilized during a significant incident.

What is Git?

Git is a popular version control system that allows software developers to manage changes to their code over time. Linus Torvalds created it in 2005 to help manage the development of the Linux operating system. Today, over 100 million developers worldwide use Git to manage their code.

At its core, Git works by creating a repository, which is a place where Git stores all of the code and its entire history of changes. Developers can make changes to the code and save those changes as "commits" to the repository. Each commit represents a snapshot of the code at a particular moment, and Git stores each commit along with information about who made the change, when it was made, and what changes were made.

Git also provides features for branching and merging. Branching allows developers to create separate versions of their code, called branches, which can be worked on independently of the main code. Merging allows developers to bring changes on a branch back into the main code when ready.

Copia Automation is a Git provider that enables Git to be used with industrial automation file formats. It delivers a user interface that simplifies Git tasks when interacting with the Git repository. Copia Automation has built specific functionality for control engineers to better track, review, and collaborate with control programs during development and operation.

In summary, Git is a version control system allowing developers to manage code changes over time. It creates a repository to store the code and its entire history of changes. It also provides features for branching and merging to help developers work on their code independently and integrate changes into the main code when ready.

A Use Case of Copia's DeviceLink

A large plant continuously runs multiple manufacturing lines that fill and label bottles of salad dressing and condiments. The machines consist of 78 control devices that are accessible for maintenance and updates by both contractors and employees.

During the evening shift, a technician updates a PLC program to fix an issue with a bottle filling station. The changes ensure the bottles are filled correctly but have reduced output speed.

The company has Copia's DeviceLink running and is scheduled to back up the PLC daily during the early morning hours.

The following events will occur:

1. The DeviceLink agent will trigger the backup process, providing a status change as the job is started and completed.
2. The retrieved program files will be transferred to Copia's cloud servers via a secure outbound connection.
3. Copia's source control application will compare the files to the last backup. With a change detected, it will store the new version as a committed change and notify the plant manager via email or chat.
4. All differences between versions are highlighted for clarity. The plant manager can review the changes from a web browser and decide if further code adjustments are needed. Prior versions are immediately available if a rollback of the change is required.

With multiple scheduled jobs, DeviceLink can automatically back up all control devices. When a change is not found, it will simply indicate the job is up to date. Plant managers can quickly audit their control devices, backup status, and change logs from one dashboard.

Conclusion

A rigorous backup strategy makes excellent business sense but is difficult to perform efficiently in complex manufacturing environments. New tools streamline the process by automatically and frequently retrieving files directly from control devices. Copia's DeviceLink is one such tool. Combined with its Git-based source control, it not only retrieves and stores files but also detects, notifies, and visually displays any changes. Technicians and managers gain incredible visibility into plant operations while ensuring their files are secure and can be quickly restored if catastrophic incidents occur.

It's easy to get started. To automate your control devices' backup, contact us or request a custom demo of Copia's products at www.copia.io.

copia.io