Montage User Manual 1.0.0

Montage Overview

This User Manual provides the details for how to use Montage as a user, or consumer of packages and products. For details on creating packages and products and listing them on the Montage marketplace exchange, visit the Montage Merchant Manual document found here.

Montage allows teams and communities to effortlessly build digital ecosystems that scale. We think of an ecosystem as a set of related software packages, applications and products that are used together. These ecosystems can be organized based on programming languages, tools, market verticals, or any other organizing concepts that makes relevant packages, applications, and products easier to find and use. Montage provides infrastructure to support generic ecosystem requirements, including packages, products, licenses, installers, and more. Montage can be extended to support any existing or new ecosystem.

You can read this Overview section to get a clearer understanding for what Montage is, or you can skip ahead to the "Getting Started" section.

Introduction to Montage

Montage, Packages, and Activities

Montage is a digital infrastructure platform designed to support **composable digital ecosystems** at scale. It provides a generic model of **packages**—modular units that can depend on other packages across diverse technologies like **NuGet**, **Maven**, **pip**, and more.

Montage doesn't replace existing package systems. Instead, it enhances them with a unified model and a focus on **power and ease of use**.

At the core of Montage is the concept of **abstract packages**. From this foundation, Montage introduces a specialized concept called an **activity**.

An **activity** represents something you want set up and configured on your system—such as a workflow, a software project, or any logical unit composed of multiple configurable parts.

For example, a software development activity might include:

- The source project to be built
- All its internal and external dependencies
- Build tools, IDEs, and scripts required for setup

This **activity** acts as a top-level representation of everything needed to **orchestrate** and run that system—without requiring manual intervention.

Activities can be created and used **locally**, or bundled into **activity packages** for sharing and deployment. When a user installs or "attaches" an activity package, they get a fully configured environment guaranteed to work as intended.

Ecosystems

In Montage, an **ecosystem** is a flexible abstraction for any digital community or platform.

Montage provides built-in support for the core needs of scalable ecosystems, including:

- Automated orchestration of packages and their relationships
- Organization and discoverability of reusable components
- Integrated documentation
- A two-sided digital marketplace connecting contributors and consumers

The ultimate goal of Montage is to offer **reusable infrastructure** that powers **any digital ecosystem**—enabling faster growth, better collaboration, and easier reuse across domains.

Introduction to the Montage Launcher

Montage Launcher

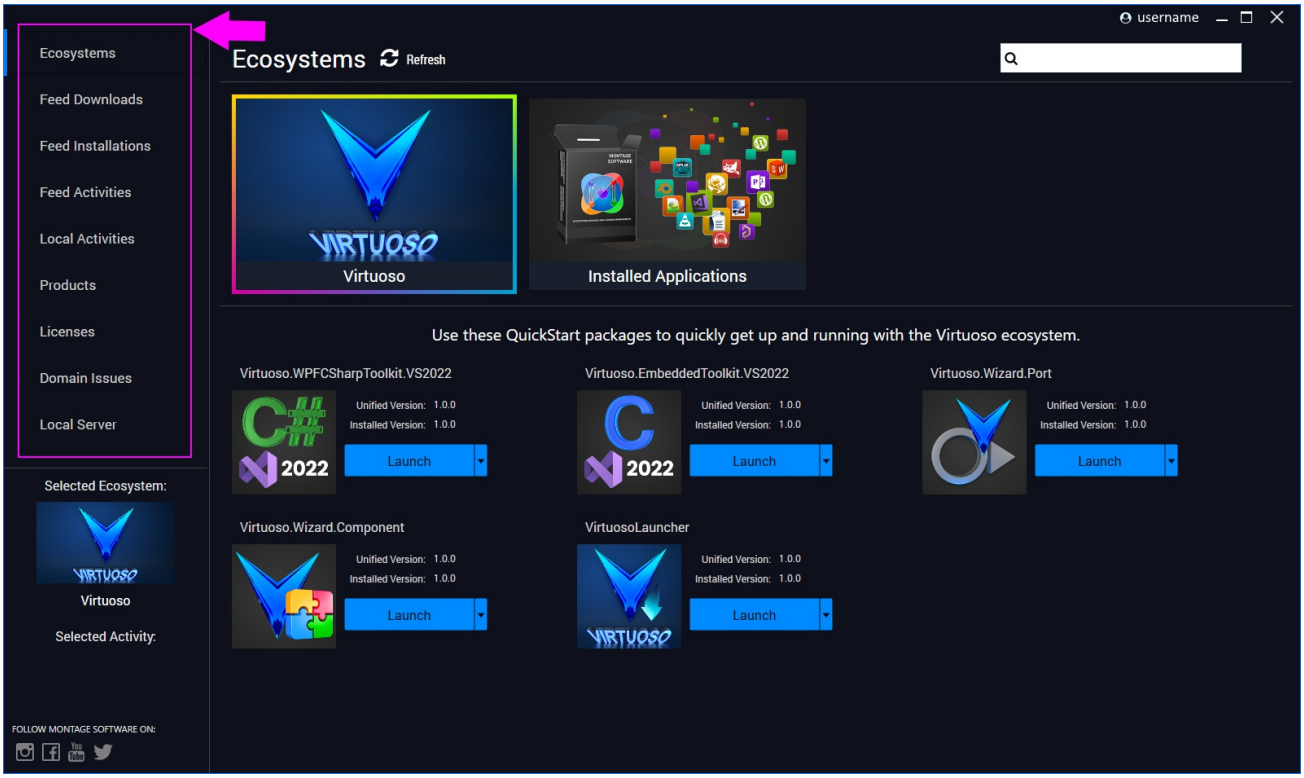
The Montage Launcher is a desktop application that provides a clean interface for managing activities.

It also supports digital ecosystems, like Virtuoso, by functioning as a local server for ecosystem applications to handle orchestration and other complexities.

When ecosystems issue requests to the Montage Launcher, the Montage Launcher displays a popup in the lower right hand corner of the screen, above the task tray, to alert the user that the Montage Launcher is being used.

The Montage Launcher desktop application's view is organized with 9 tabs along the left, as shown below. These tabs are detailed in the sections that follow.

- Ecosystems
- Feed Downloads
- Feed Installations
- Feed Activities
- Local Activities
- Products
- Licenses
- Domain Issues
- Local Server



Ecosystem Selection and Context in the Launcher

The currently selected **tab** is highlighted by a **thin blue line on the left**.

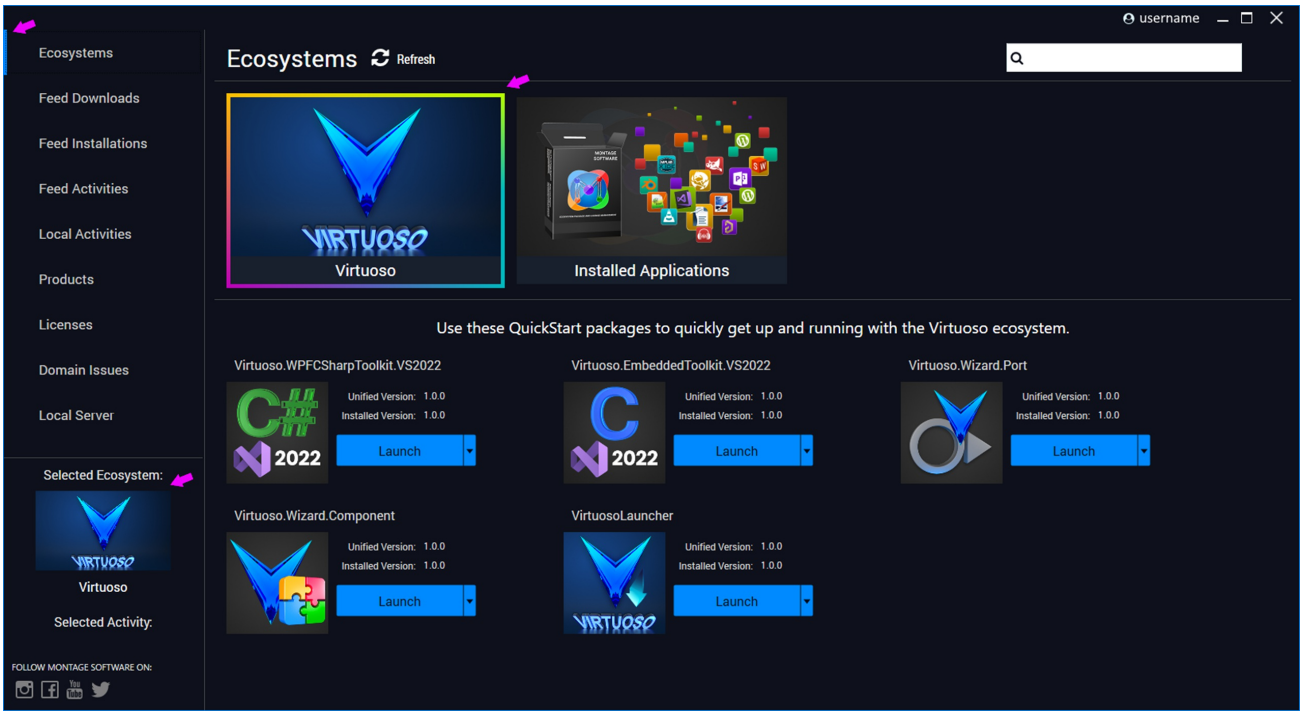
Within the **"Ecosystems" tab**, available ecosystems are displayed **horizontally across the top**.

The **selected ecosystem** is:

- Highlighted with a **multicolored border** in the list view.
- Shown in the **bottom-left corner** of the Launcher window.

The selected ecosystem determines the **context** for manual actions such as:

- Browsing
- Downloading
- Installing content



QuickStart Packages

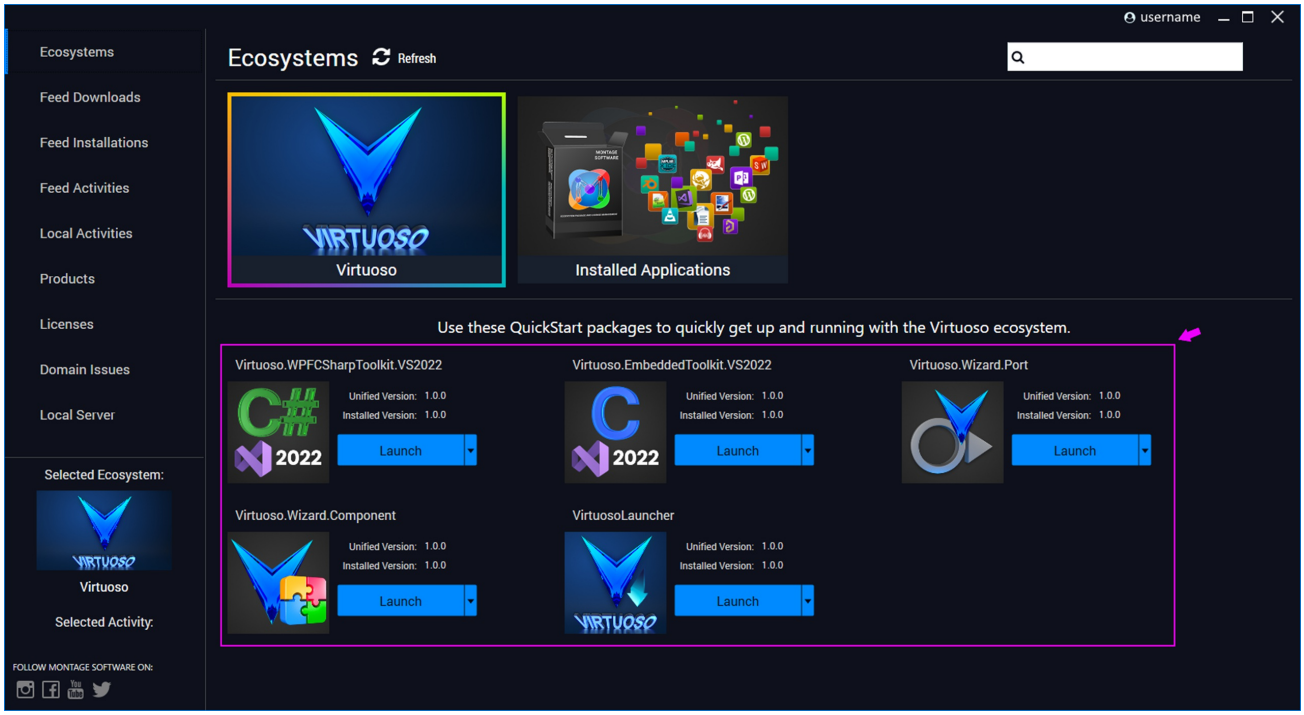
QuickStart packages are specially designated packages that help new users get started quickly with an ecosystem. Ecosystem creators can define these as entry points—typically representing common workflows or essential tools.

Like all Montage packages, QuickStart packages can serve as **placeholders** that reference one or more other packages to represent a complete workflow.

QuickStart Packages in the Virtuoso Ecosystem

The Virtuoso ecosystem defines several QuickStart packages:

VirtuosoLauncher
The core framework for building no-code applications in Virtuoso. Most other packages depend on this.
Virtuoso.WPFCSharpToolkit.VS2022
A workflow package that depends on VirtuosoLauncher.
It adds support for creating no-code **C# WPF desktop applications** in **Visual Studio 2022**.
Virtuoso.EmbeddedToolkit.VS2022
A workflow package for **virtualizing embedded systems**.
Depends on:
Virtuoso.WPFCSharpToolkit.VS2022
Which depends on VirtuosoLauncher
This layered dependency ensures that all necessary components are installed to simulate embedded systems in a no-code environment.
Virtuoso.Wizard.Port
A standalone desktop application for creating no-code **port definitions**.
Virtuoso.Wizard.Component
A standalone desktop application for creating no-code **component definitions**



Montage Launcher Tabs

This section details the tabs in the Montage Launcher desktop application.

Feed Downloads Tab

Feed Downloads Tab

The **Feed Downloads** tab allows you to **discover and download packages** into the currently selected **ecosystem**.

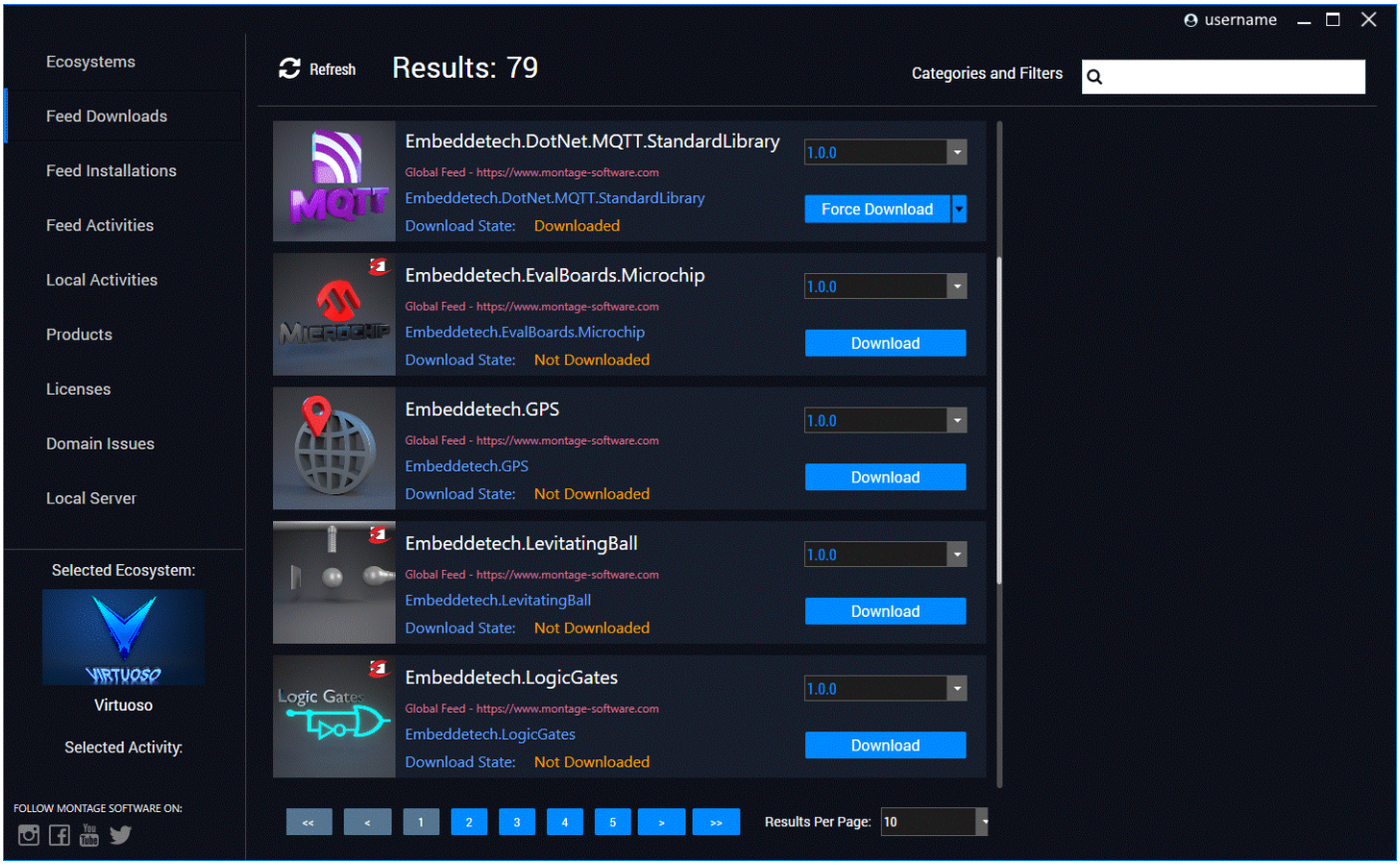
Each ecosystem extends the **Montage Launcher** with specific logic for:

Handling how downloaded packages are processed
Determining whether a package has already been downloaded

When an ecosystem is selected, the Launcher:

Displays the download status of packages
Enables filtering and searching across available packages
Facilitates downloading new packages

The packages shown in the **Feed Downloads** tab are those that have registered themselves with the **selected ecosystem**, and they are collected from **all package feeds** currently installed in the Launcher.



Feed Installations Tab

Feed Installations Tab

The **Feed Installations** tab lets you **discover and install software packages** that are directly associated with the selected **ecosystem**.

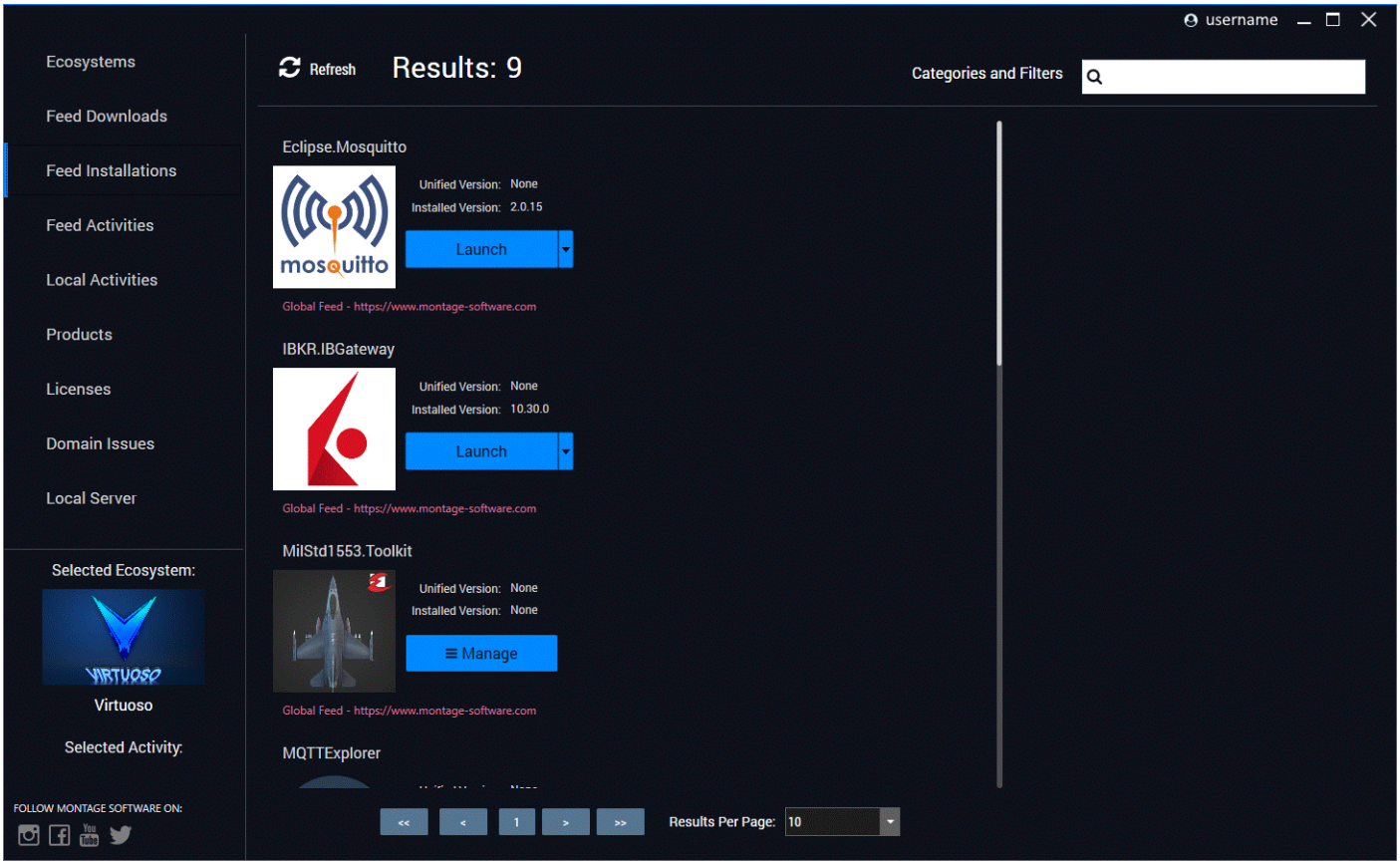
These are known as **"installable" packages**—they behave like standard Windows applications and can be installed directly onto your system.

Key Differences from Downloadable Packages

Unlike the packages shown in the **Feed Downloads** tab, installable packages:

- **Do not require additional context** (such as project details or environment targets)
- May or may not allow **side-by-side** installation, where multiple versions of the same package can be installed to the same **installation target** at the same time.
- Can be **installed and managed with a simpler interaction**
- Are generally **launchable** packages

This makes them ideal for users who want to quickly install and run ecosystem-related tools without additional configuration.



Feed Activities Tab

Feed Activities Tab

The **Feed Activities** tab allows you to **discover and install activity packages** from your installed package feeds, based on the currently selected **ecosystem**.

What is an Activity Package?

An **activity package** is a special type of Montage package that represents a complete **project or installation target**. It is designed to be fully installed and configured so it can **build or run on your machine** with no manual setup.

Activity packages define all dependencies required for the target to function, including:

- **Internal dependencies** like C#, Java, or Python libraries
- **External tools** like IDEs, build systems, SDKs, and other installable software

How It Works

To ensure consistency and reliability:

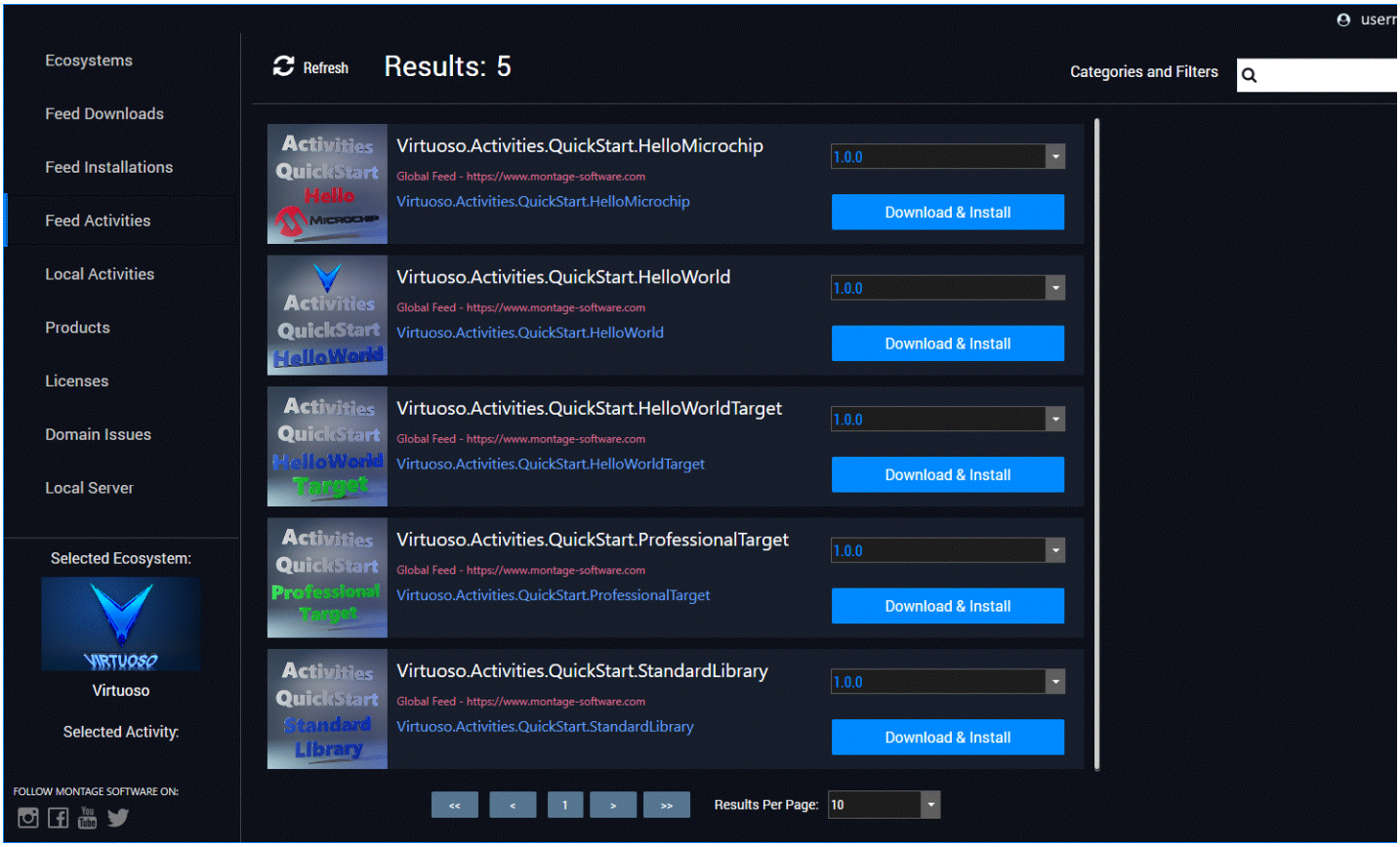
- **Dependencies** across all activities are **unified**, resolving overlaps and avoiding conflicts.
- **Governance policies** are applied to validate that packages meet the needs of both **consumers and creators**.
- After unification, **policy checks** ensure the entire activity is **licensed and valid** before installation.

One-Click Installation

Once validated, the entire activity can be installed with a **single click**. Montage handles:

- Dependency resolution
- Tool installation
- Environment setup

This enables fast, reliable setup of complex environments with minimal effort.



Local Activities Tab

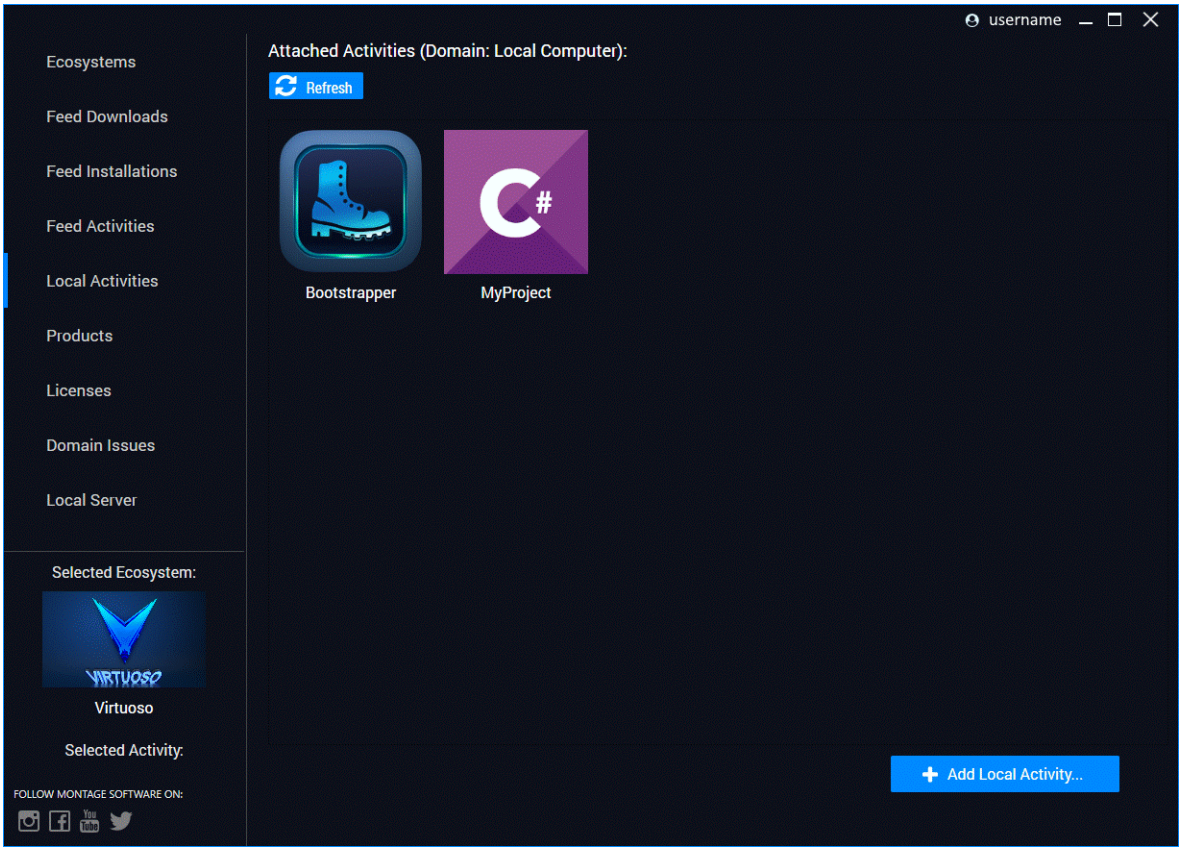
Local Activities Tab

The **Local Activities** tab displays all activities that are currently installed or attached to the Montage Launcher's **unification domain** (i.e., your entire PC).

In the example shown below, two activities are installed:

The **"Bootstrapper"** activity
The **"MyProject"** activity

Activities that have already been created can be manually installed using the "Add Local Activity" button.



Understanding Activities and Installation Targets

Activities are a core Montage concept that apply across different development languages and tools. Each activity contains:

- One or more **root package requests**
- Each root request is tied to a specific **installation target**, such as a file system location or project environment

These root requests must be registered to an activity and serve as entry points for dependency resolution.

Example: Bootstrapper Activity

The details of the Bootstrapper activity can be viewed by clicking on its icon.

When a user installs a regular desktop application via the Montage Launcher, the Launcher automatically:

- Uses the default **Bootstrapper** activity
- Resolves the **"Installed Applications"** installation target as the target to be used for the application

If you click the **Bootstrapper activity**, you'll see:

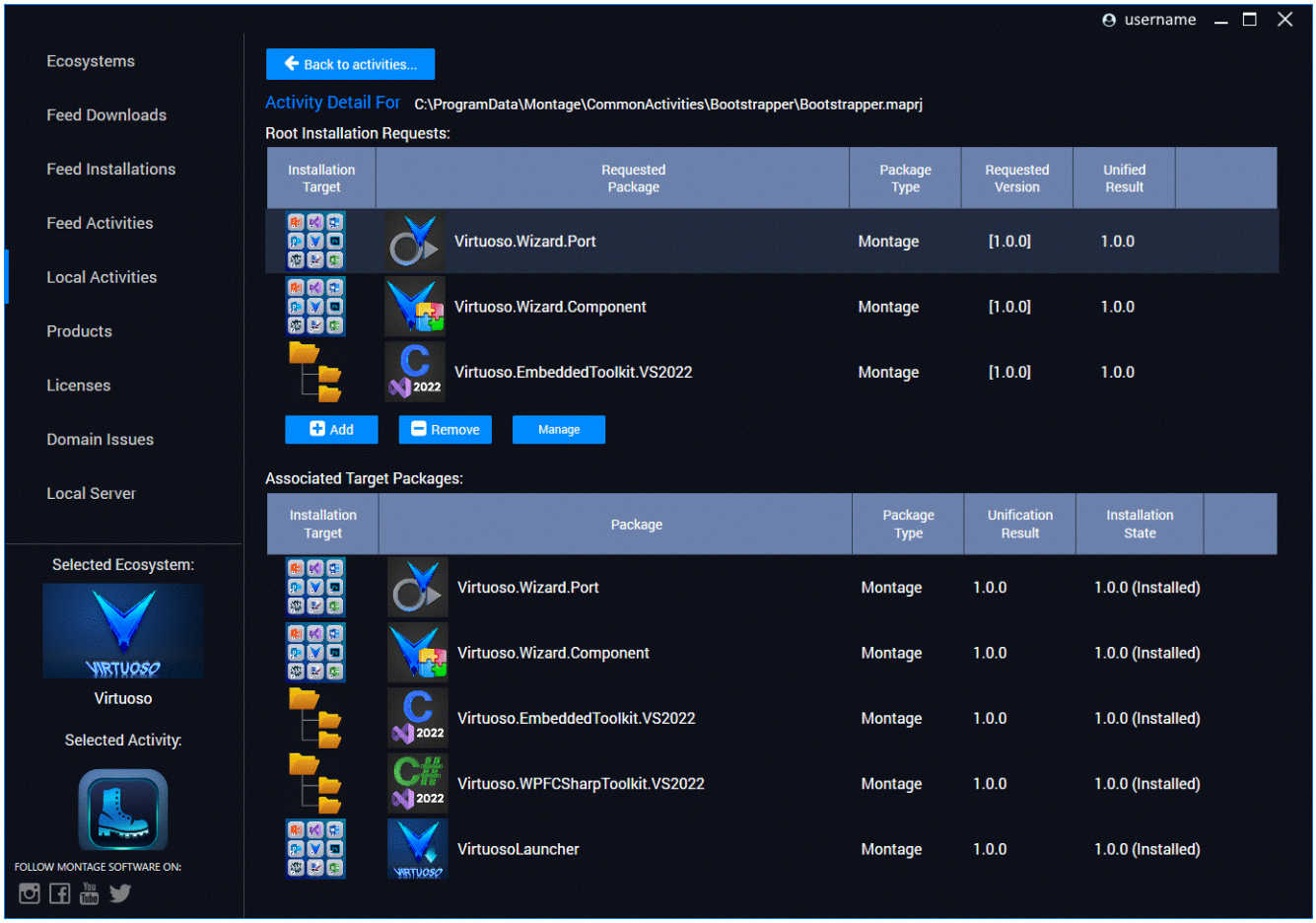
- Its **root package requests**
- All **associated package dependencies**

In the example:

- **Virtuoso Port Wizard**, **Component Wizard**, and **Embedded Toolkit** are requested
- The Embedded Toolkit depends on the **Virtuoso C# Toolkit**
- The C# Toolkit in turn depends on the **Virtuoso Launcher**

Each package is linked to its appropriate installation target:

- The **Virtuoso Launcher**, as a desktop app, installs to the **Installed Applications** installation target
- The **C# Toolkit** and **Embedded Toolkit** install to the **File System** installation target



Example: MyProject Activity

The **MyProject** activity includes several root requests and a more complex dependency graph.

Key highlights:

- The activity manages installations across **multiple installation targets**, including:
  - A **C# project**
  - The **File System**
  - The **Installed Applications** target
- Packages and targets in this activity include:
  - The **Virtuoso C# Toolkit** (File System)
  - The **Virtuoso Launcher** (Installed Applications)
  - Several packages installed directly to the **C# project**

Key Concepts Demonstrated

- **Multiple installation targets per activity**:

Activities aren't bound to a single context—they may span across overlapping desktop apps, file systems, and projects.

Mixed package ecosystems:

Montage supports packages from multiple sources, including **Montage, NuGet** and other package feeds.

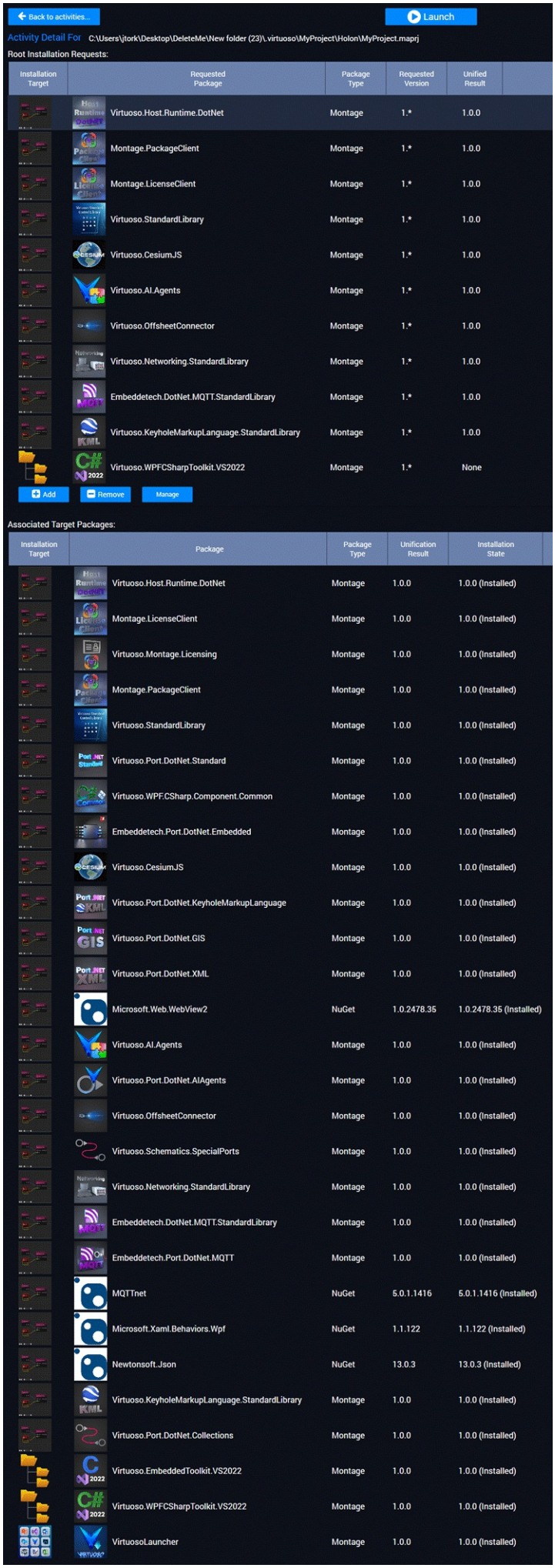
Version flexibility:

Montage extensibly handles **non-Semantic Version** notations as used by Microsoft.Web.WebView2 and MQTTNet, ensuring heterogeneous ecosystems remain compatible.

Unified orchestration:

If an activity package were created from this activity, Montage would orchestrate the entire graph—including tooling, libraries, and environment targets—ensuring everything works together seamlessly.

Montage's power lies in its ability to **orchestrate packages across diverse environments**, apply **governance**, and deliver a **scalable, unified experience**—all tied back to the activity abstraction.



Products Tab

Products Tab

The **Products** tab displays digital products that are available for purchase by Montage users.

What Is a Product?

A **product** in Montage is a **digital asset** that unlocks or activates specific features in a software package. These products are often tied to licensing, advanced features, or commercial use permissions.

How Products Are Used

While ecosystem products are shown in the **Launcher** for discovery, they are **most commonly encountered during workflow orchestration**.

For example:

A user installs a Montage **activity** made up of multiple packages from various creators.
Some of these packages may **require product purchases** to meet usage policies.

Policy Evaluation and Purchase Flow

When you install an activity:

Montage unifies the activity's dependency graph, tailored to the user's unfiication domain (PC).
It then **evaluates policy requirements** defined by each package creator.
If any policies aren't satisfied, Montage checks whether **digital products can provide encrypted "claims"** to meet those policies.
If needed, Montage will:
Identify the required products
Present them for purchase **before** installation proceeds

Why This Matters

This approach ensures:

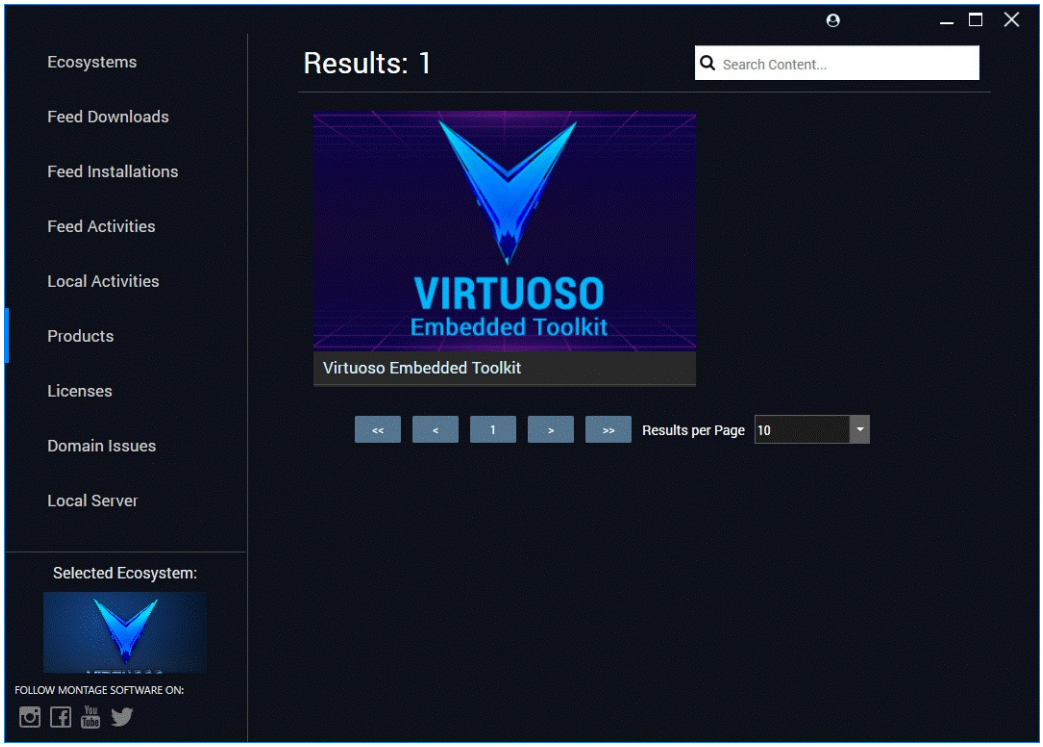
Users understand **total cost upfront**
No surprise purchases are required during installation of complex projects
License compliance is enforced automatically

By integrating product validation into orchestration, Montage enables confident, scalable use of composable packages across content creators.

Important Note:
Package policies in Montage include both **static** and **dynamic** policies.

Static policies are evaluated **after** unification and **before** installation.
Dynamic policies are evaluated at **runtime**, when specific features are used.

This means a package can be installed successfully, and only when a user attempts to access a **feature gated by a policy**, Montage will prompt the user to **purchase the required product** and guide them through the process.



Licenses Tab

Licenses Tab

The **Licenses** tab shows all licenses that your organization currently holds.

What Are Licenses?

When you purchase a Montage product, it grants you **one or more licenses**. These licenses control access to features or usage rights and may have expiration terms or floating access.

How Licenses Work

Activation:
After purchase, licenses must be **activated** before use. This starts the license period.
Access:
Once activated, licenses can be used by your organization's members.
If the license is a **floating license**, it can be **released (freed)** and reassigned as needed.

What You Can See

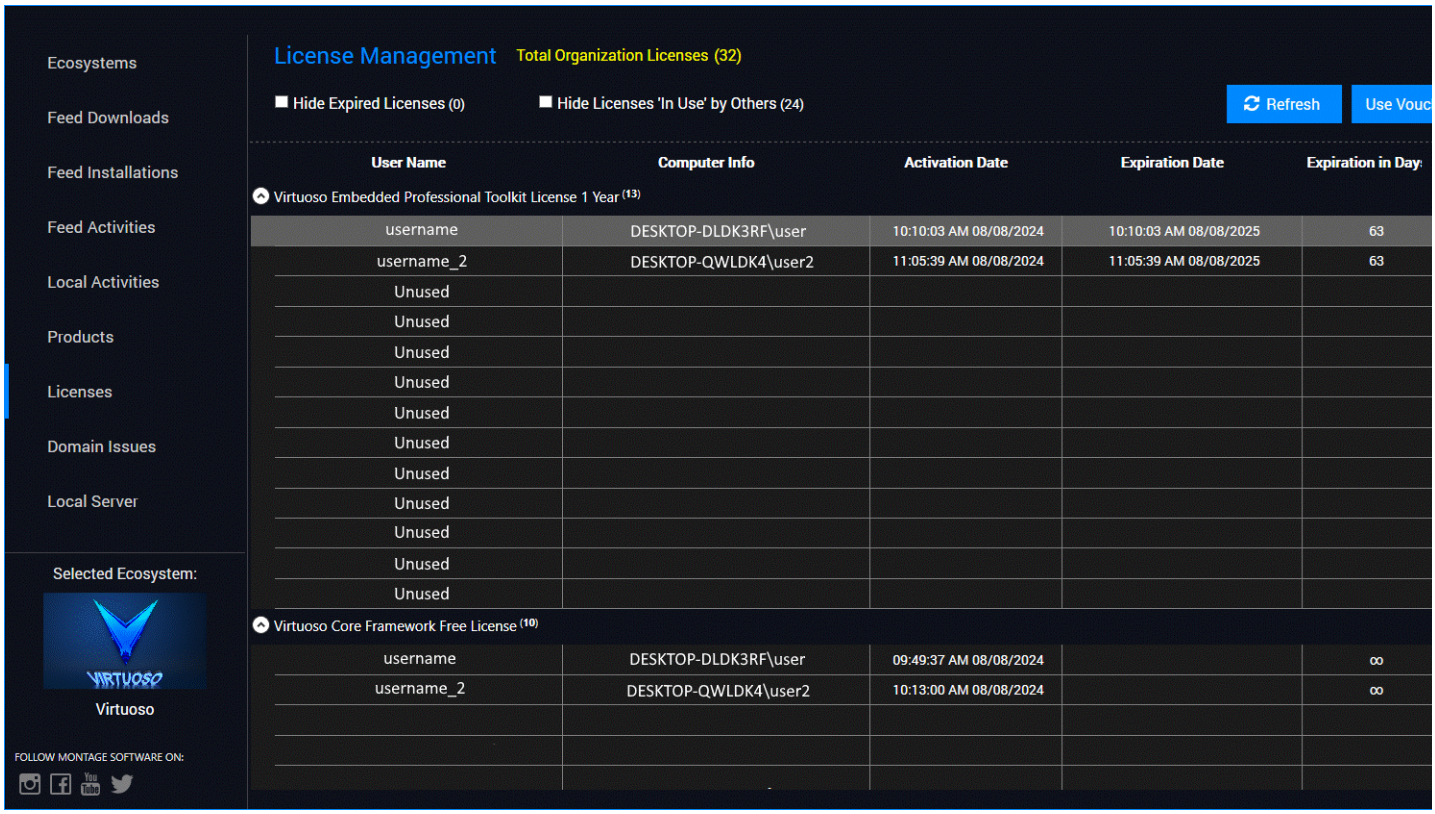
The Licenses tab provides detailed info for each license:

Activation status
Activation date
Current usage status
Days remaining
A **"More Info"** button to view the full license agreement

Special Cases

Some licenses may be:
Free
Granted automatically
Permanently valid (never expire)
You can also **redeem a voucher** in this tab, if a content creator has provided one. This allows you to obtain a license **without a purchase**.

The **Licenses** tab provides a **simple and centralized location** to view and manage **all licenses held by your organization**—across all content creators and ecosystems.



Domain Issues Tab

Domain Issues Tab

Domain Issues Tab

The **Domain Issues** tab provides tools and interactions to **identify and resolve issues** affecting your Montage domain unification.

What Is a Domain Issue?

The Montage Launcher manages a **unification domain**, which ensures that **all activities and their dependencies** are in a consistent, unified state.

A **domain issue** occurs when:

Dependencies are **no longer valid**
Activities or installation targets are **manually removed**
External changes **break the unification state**

When domain issues exist, the Montage Launcher **pauses unification-related actions** until they are resolved.

How Domain Issues Occur

The most common causes include:

A project or its **activity file is deleted**
A **new root installation request** is added or removed outside the Launcher
Changes are made **while the Launcher is closed**, such as deleting activities or modifying files

The Montage Launcher detects these inconsistencies and:

Automatically detaches activities that have been deleted
Prompts the user to resolve any added or orphaned root requests

How to Resolve Them

When an issue is detected:

The Launcher displays a **popup** asking how to proceed

- Options include **"Ignore For Now"** or resolving immediately

Ignoring the issue allows use of basic features (e.g., downloading packages)
However, **unification-related operations** can't be completed until all issues are addressed

All unresolved and ignored issues appear in the **Domain Issues** tab, where you can:

Review the specific discrepancies
Delete, accept, or ignore conflicting elements
Restore the domain to a unified state

Why It Matters

Think of the Montage Launcher as a **control system**:
It continuously attempts to maintain the domain in a valid, unified configuration. When that's no longer possible, user intervention is required to restore domain integrity.

Local Server Tab

Local Server Tab

The **Local Server** tab allows you to view server activity logs and configure how the Montage Launcher functions as a local server. It includes two sub-tabs:

Local Package Server Requests
Server Configuration

Local Package Server Requests Sub-Tab

This sub-tab displays all requests processed (or being processed) by the Montage Launcher.

Left Panel: Lists individual server requests
Right Panel: Shows detailed status messages for the selected request, including:
Actions taken
Notes and any problems encountered

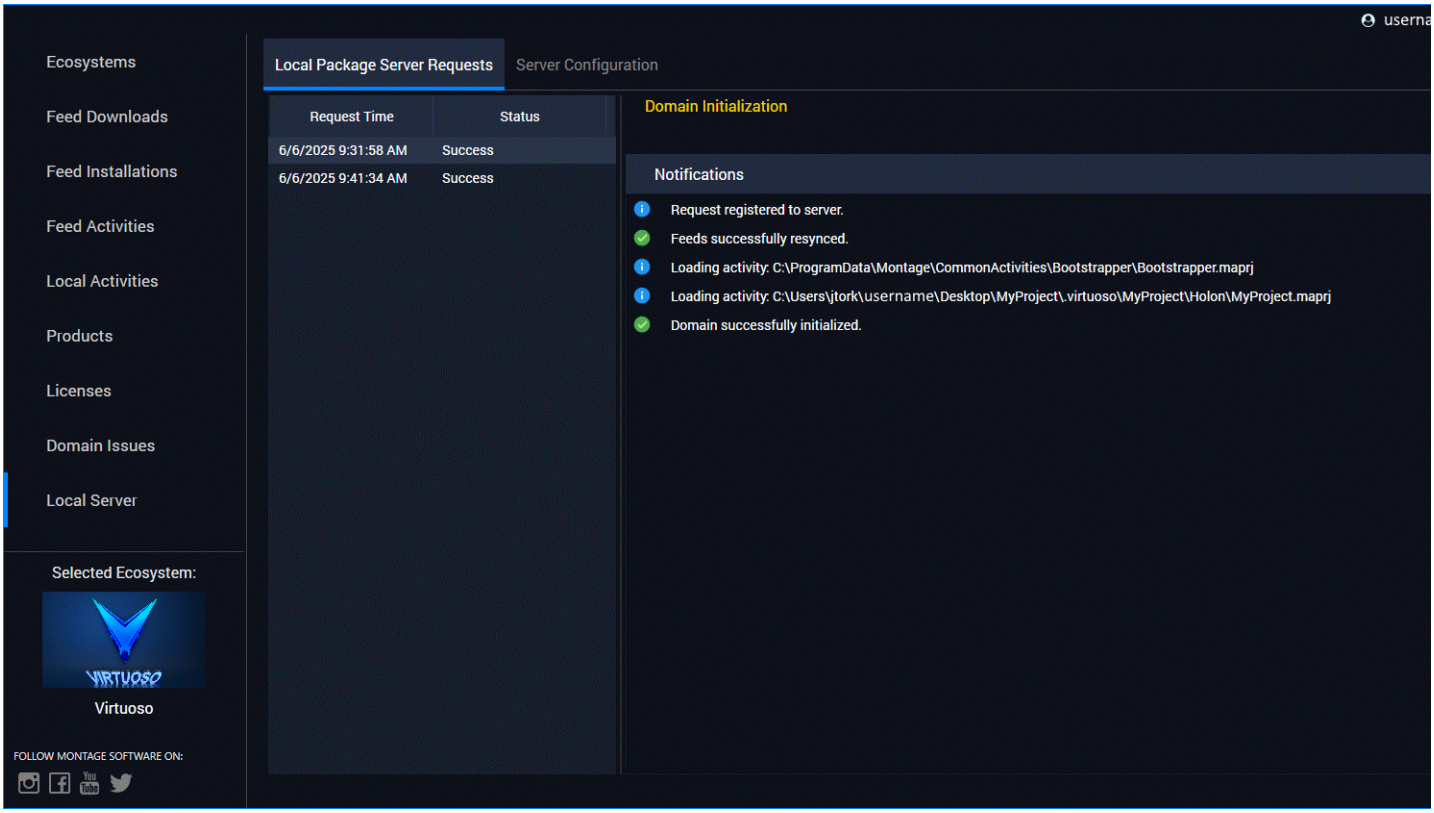
Requests may originate from:

Manual actions by users in the Montage Launcher
Ecosystem clients using the Montage Launcher as a server

The Montage Launcher serves as both a desktop application and a request-processing server, handling:

Package discovery
Dependency unification
Licensing
Orchestration

All requests are **queued and executed in order**.
To keep users informed, the Launcher displays a **notification in the bottom-right corner of the desktop** when it modifies the system.



Server Configuration Sub-Tab

This sub-tab allows you to manage the Montage Launcher's server settings.

Key Functions:

Administrator Privileges: Indicates whether the Launcher has elevated rights (typically required for orchestration and installation)
Start/Stop Server: Toggle server activity
Port Configuration: Change the server's listening port for specialized client scenarios

Feed Sources:

The Montage Launcher works with both **global** and **local** package feeds:

Global Feed: https://www.montage-software.com
Local Feeds:
Select folders on your computer or network shares
Ideal for **self-hosted**, **on-premises**, or **proprietary content**

You can **disable the global feed** and run fully local if needed.

Montage Extensions

Montage Extensions extend Launcher functionality:

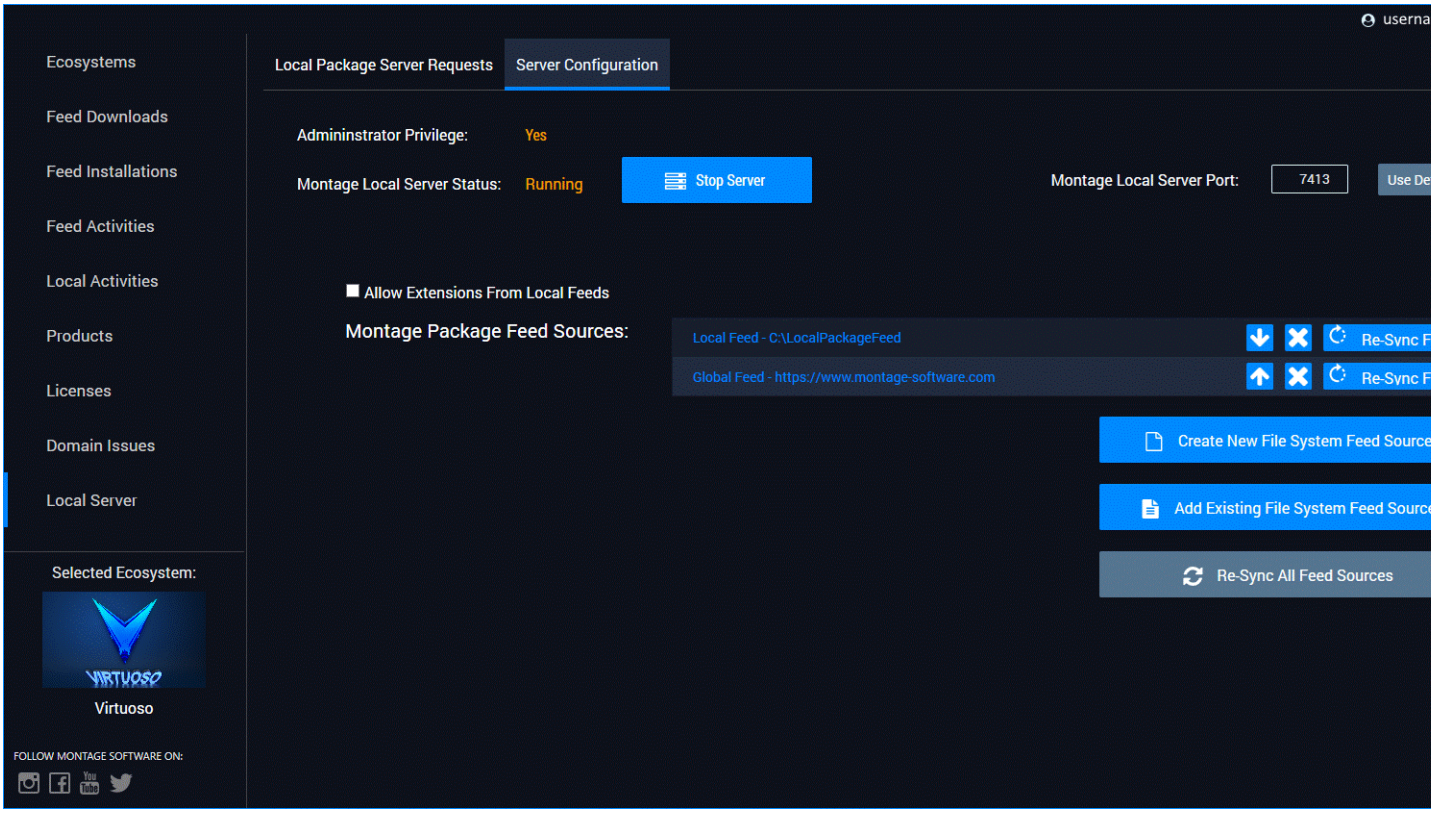
Deployed via packages through feeds (just like other packages)
Run with Launcher privileges
Must be **digitally signed** to be trusted when sourced from global feeds
To enable extensions from local feeds, you must check **"Allow Extensions From Local Feeds"**

□□ Local feed extensions are **not validated** and may pose a security risk.

Managing Feeds

You can manage the feed order and availability:

Add/Remove Global Feed
Add/Remove Local Feeds
Set feed priority using the ↑/↓ buttons
Create a new local feed via **"Create New File System Feed Source"**
The default local feed path is C:\LocalPackageFeed



Dependency Management

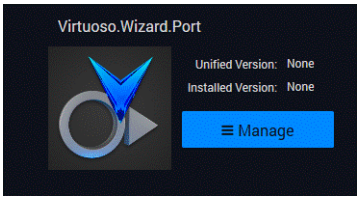
Dependency Management

The Montage Launcher performs most of its functions behind the scenes for ecosystem client tools and software. The services provided to ecosystem software includes:

- Creating new activities
- Creating new installation targets
- Downloading packages to the ecosystem
- Installing packages to installation targets
- Evaluating software package policies at design time and runtime
- Providing licensing resolution and product purchasing interactions for software packages at runtime

Although the ecosystem software generally hides the details of installing packages, it is helpful to understand how dependencies can be manually requested or removed in the Montage Launcher.

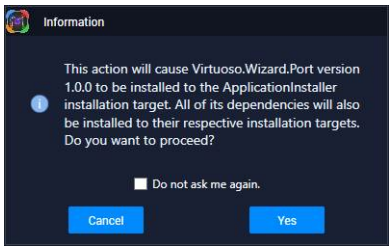
The QuickStart packages in the Ecosystems were the first place where manual installation requests can be made. Using the Virtuoso.Wizard.Port package (the Virtuoso Port Wizard desktop application) as an example, we see its initial state is not unified and not installed:



Click “Manage” to manage the package. Note that for this package, the Activity Context is already specified as the common “Bootstrapper” activity by mousing over the boot icon. The Target (installation target) is also already specified as the common “Installed Applications” installation target.

The “Simple Version Selection” tab is shown, because no other packages transitively depend on the Virtuoso Port Wizard, and so we are free to simply select which version we would like to install from the list of all versions of the package.

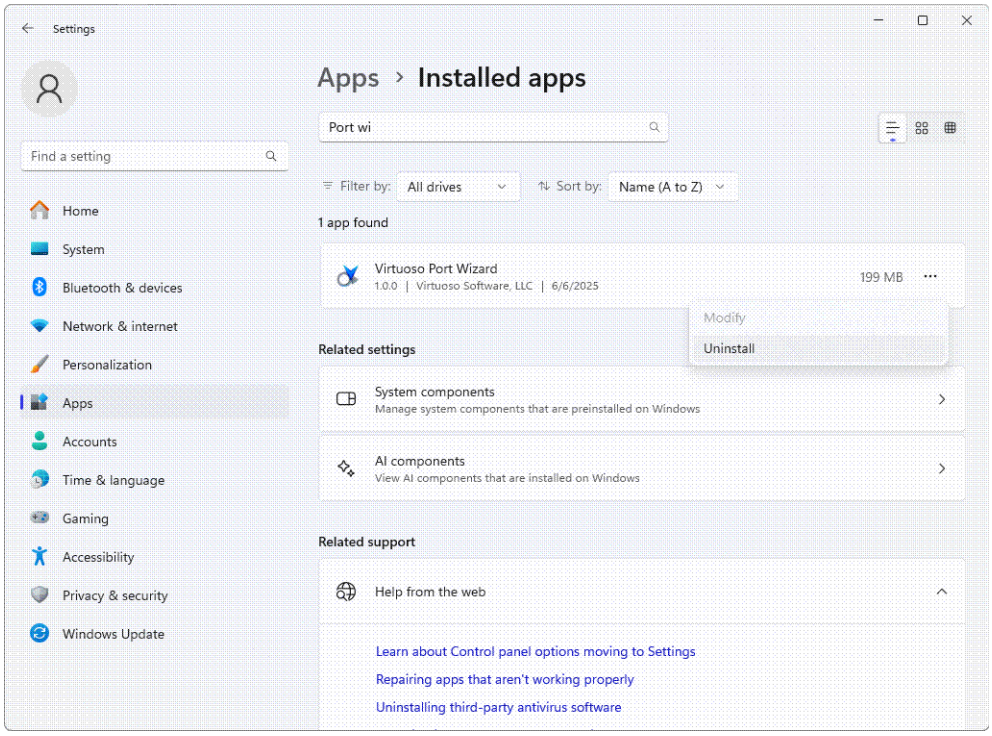
By clicking version “1.0.0”, we are prompted to confirm that we want to install version 1.0.0 of this package.



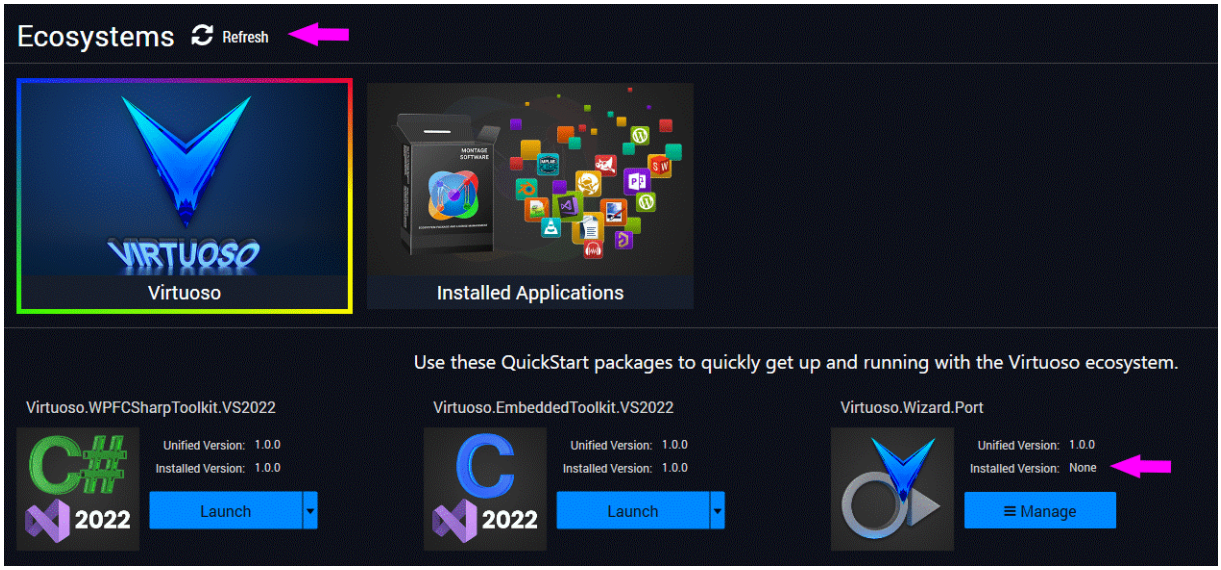
Click “Yes’, and the Montage Launcher will download and run the installer for the Virtuoso Port Wizard. The installer may run behind the Montage Launcher, so you may need close the package management dialog to see the installer. Follow the installation as usual. You will need to click “Exit” after the installer runs to close the installer.

After the installer runs, we see the package is unified and installed to version 1.0.0.

We could uninstall the package by clicking the “(No Version Requested)” option. However, instead, close this dialog and then manually uninstall the Virtuoso Port Wizard in Windows:



After uninstalling, we see that the Port Wizard is still unified to version 1.0.0, but has no installed version:



To resolve this, click “Manage” again, and we see 1.0.0 is the selected version, but there is now an “Install” button next to 1.0.0.

Click the Install button and Montage will rerun the installer. After the installer completes, the Launcher will refresh and show the package as installed to 1.0.0:

Now click on the “(No Version Requested)” item. After being prompted to confirm, the Launcher will remove the Port Wizard from the domain, however the Port Wizard will still be installed, as shown below. The Launcher does not automatically uninstall packages just because they are no longer a requirement for the domain.

To uninstall the Port Wizard, click the “Uninstall” button and the Port Wizard will be uninstalled.

Key Takeaways

Software continues to transform every aspect of business and daily life, and new software solutions to wide-ranging problems continue to proliferate. These solutions, however, are commonly just one smaller part of a larger problem. Companies do not need to become natural language processing experts and roll their own chat box, for example. Companies need to quickly find the best chat box solution and effortlessly incorporate it into their business CRM strategy. Alternatively, using Montage a company can leverage a third-party chat box and bundle it into a new software and product composition. Software composability will fundamentally drive the future of software architectures, as ease of integration becomes paramount, and as the need for specialization, expertise, and ease of integration increases.

Tomorrow’s killer apps and business system juggernauts will need to pull from a rich montage of composable software solutions with breakneck speed to market. Montage provides the infrastructure needed to support this software component digital economy of the future.