

ESUG 2026 · CONFERENCE SESSION

A modern GemStone IDE, real Python, and AI on a live image

Five short demos in thirty minutes: developing Smalltalk in VS Code, letting an AI explore your data, running genuine Python inside the object database, richer inspectors, and one-click package loading — all against a single live GemStone image.

James Foster · GemTalk Systems · james.foster@gemtalksystems.com

5
live demos

30
minutes

3.7.5
GemStone/S 64

1
image, one object space

01 VS Code IDE

02 AI explores data

03 Python in GemStone

04 Enhanced inspectors

05 Loading packages

DEMO 01

A GemStone IDE inside VS Code

The Jasper extension turns Visual Studio Code into a full GemStone/S 64 development environment. Browse, edit, compile, debug, and test Smalltalk against a live image — and manage your GemStone infrastructure — without leaving the editor.

- ◆ Browse the whole image
Dictionaries, class categories, classes, and methods, driven by fast filters.
- ◆ Compile & evaluate live
Send Smalltalk to the running session and get the real result back — no rebuild.
- ◆ Debug and test
Step through code and run SUnit classes and methods, inspecting failures inline.
- ◆ Manage infrastructure
Configure the OS, versions, databases, processes, logins, and sessions from one panel.

What to watch for: the Jasper marketplace listing installing, then a live GemStone login from inside VS Code.

Video 01 · VS Code for GemStone

James Foster



Python Talks to GemStone
Client/Server Architecture



Watch on  YouTube

Starts at 0:12

[Open on YouTube ↗](#)

ESUG 2026 · CONFERENCE SESSION

A modern GemStone IDE, real Python, and AI on a live image

Five short demos in thirty minutes: developing Smalltalk in VS Code, letting an AI explore your data, running genuine Python inside the object database, richer inspectors, and one-click package loading — all against a single live GemStone image.

James Foster · GemTalk Systems · james.foster@gemtalksystems.com

5
live demos

30
minutes

3.7.5
GemStone/S 64

1
image, one object space

01 VS Code IDE

02 AI explores data

03 Python in GemStone

04 Enhanced inspectors

05 Loading packages

DEMO 02

Let an AI explore your data

Point an AI assistant at the same tools the IDE uses and ask questions of your live object model in plain English. It writes and runs GemStone code, reads back real results, and reasons about them — read-only, so nothing is committed.

◆ Ask in natural language

"How many invoices have an amount due in a different currency from the contract?"

◆ Real queries, real objects

The agent executes Smalltalk against the image and reports the actual answer.

◆ Follow-up reasoning

Drill into specifics — exchange rates, individual invoices — across turns.

◆ Safe by default

Exploration runs read-only; the agent even flags its own earlier mistakes.

What to watch for: the assistant catching a wrong assumption about exchange rates and correcting itself against the data.

Video 02 · AI queries

James Foster



Python Talks to GemStone
Client/Server Architecture



Watch on  YouTube

Starts at 1:44

[Open on YouTube ↗](#)

ESUG 2026 · CONFERENCE SESSION

A modern GemStone IDE, real Python, and AI on a live image

Five short demos in thirty minutes: developing Smalltalk in VS Code, letting an AI explore your data, running genuine Python inside the object database, richer inspectors, and one-click package loading — all against a single live GemStone image.

James Foster · GemTalk Systems · james.foster@gemtalksystems.com

5

live demos

30

minutes

3.7.5

GemStone/S 64

1

image, one object space

01 VS Code IDE

02 AI explores data

03 Python in GemStone

04 Enhanced inspectors

05 Loading packages

DEMO 03

Real Python, inside the object database

Not a bridge to an external interpreter — Python runs *in* GemStone. Every Python value is a first-class GemStone object, so an `import gemstone` lets Python reach straight into the live object model and work with your data directly.

- ◆ Python the language, GemStone the runtime
Run ordinary Python — including notebooks — against the “Grail (GemStone Python)” kernel.
- ◆ Reach into the image
`gemstone["InstalledAcmeSystem"]` hands Python a live GemStone object.
- ◆ Values are GemStone objects
No serialization boundary — Python data lives in the same object space as your Smalltalk.
- ◆ Shared persistence
Anything Python creates can be committed to the repository like any object.

What to watch for: a Jupyter cell importing `gemstone`, fetching a system object, and counting its invoices — all in Python.

Video 03 · Python runs in GemStone



Starts at 7:09

[Open on YouTube ↗](#)

ESUG 2026 · CONFERENCE SESSION

A modern GemStone IDE, real Python, and AI on a live image

Five short demos in thirty minutes: developing Smalltalk in VS Code, letting an AI explore your data, running genuine Python inside the object database, richer inspectors, and one-click package loading — all against a single live GemStone image.

James Foster · GemTalk Systems · james.foster@gemtalksystems.com

5
live demos

30
minutes

3.7.5
GemStone/S 64

1
image, one object space

01 VS Code IDE

02 AI explores data

03 Python in GemStone

04 Enhanced inspectors

05 Loading packages

DEMO 04

Enhanced object inspectors

Inspecting an object should tell you more than its raw slots. Custom inspector views present domain objects the way a person thinks about them — summaries, overviews, and formatted tables — right alongside the raw and meta views.

Domain-aware tabs

Summary, Overview, Contracts, Customer Invoices, Raw, Print, and Meta on one object.

Readable tables

Status, contract, licensee, amounts, and dates laid out like a report, not a slot dump.

Browser + inspector together

Navigate dictionaries and classes, then inspect a live global like InstalledAcmeSystem.

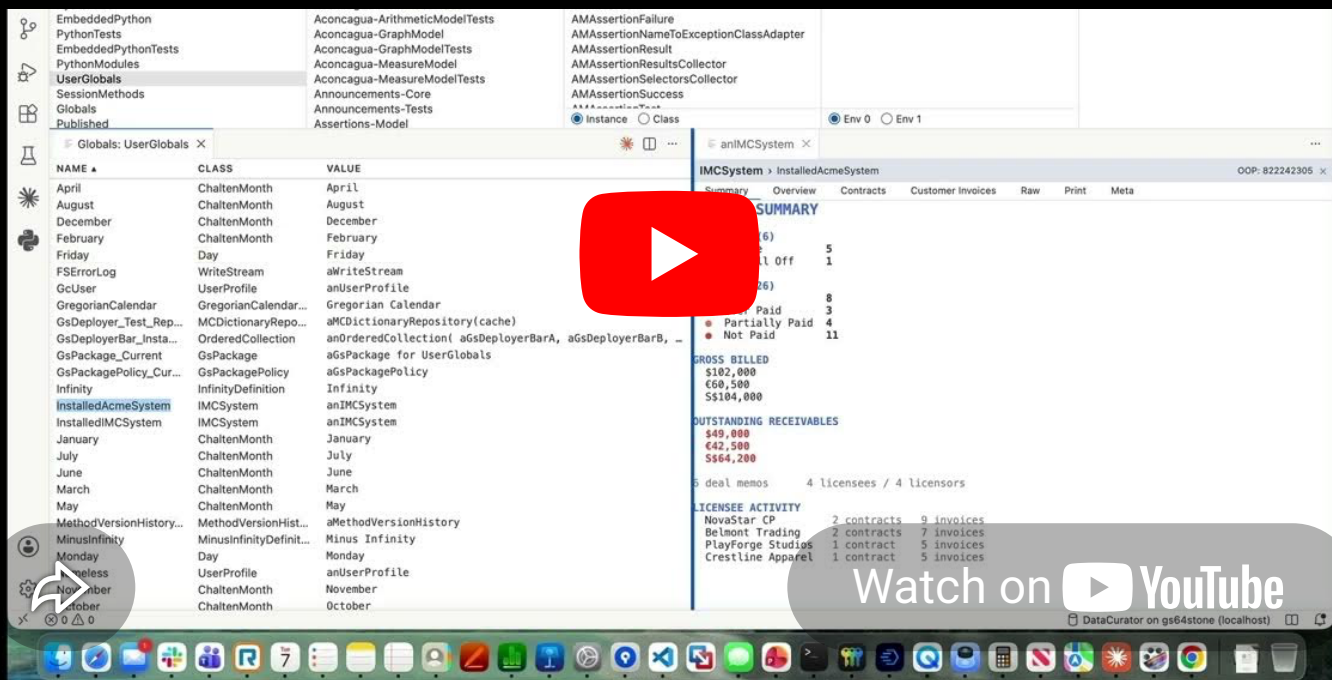
Still the real object

Every view is a lens on the same live GemStone object, OOP and all.

What to watch for: switching an IMCSystem inspector to its Contracts tab and reading the deal table directly.

Video 04 · Enhanced inspectors

James Foster



Full clip

[Open on YouTube ↗](#)

ESUG 2026 · CONFERENCE SESSION

A modern GemStone IDE, real Python, and AI on a live image

Five short demos in thirty minutes: developing Smalltalk in VS Code, letting an AI explore your data, running genuine Python inside the object database, richer inspectors, and one-click package loading — all against a single live GemStone image.

James Foster · GemTalk Systems · james.foster@gemtalksystems.com

5
live demos

30
minutes

3.7.5
GemStone/S 64

1
image, one object space

01 VS Code IDE

02 AI explores data

03 Python in GemStone

04 Enhanced inspectors

05 Loading packages

DEMO 05

Loading packages with Rowan

Bring code into your image without leaving the editor. The GemStone panel manages repositories and Rowan projects, so cloning a repository and loading a project — Seaside, say — is a guided, visible operation.

◆ Repositories & projects

See registered repositories and which Rowan projects are loaded at a glance.

◆ Load real frameworks

Pull in projects like Seaside and HelloSeaside and watch them appear as loaded.

◆ Sessions in context

Logins & sessions, versions, databases, and processes sit right above your projects.

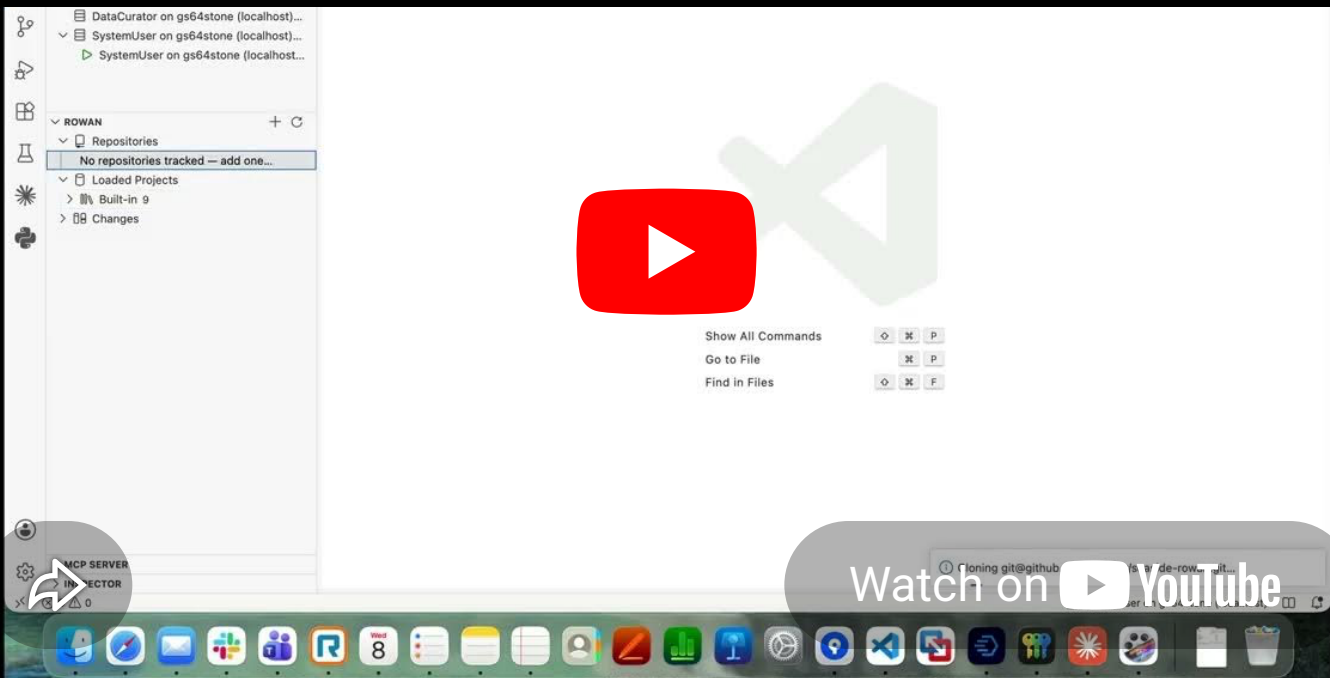
◆ Track changes

The Changes view shows what has moved since load, ready to commit.

What to watch for: a Rowan project going from a repository entry to “loaded” in the Loaded Projects list.

Video 05 · Loading packages

James Foster



Full clip

[Open on YouTube ↗](#)