DISPATCH SCIENCE

From File to Flow: Building Integrations with DataBridge

Craig McTavish Solutions Architect

Mathieu Savaria Lead Software Developer

October 2026 · Dispatch Science User Conference

What is DataBridge?

DataBridge is a cloud-native iPaaS (integration Platform as a Service) designed for Dispatch Science users to streamline operations and enhance efficiency with minimal IT involvement.

- Seamless connectivity through API, and file import/export, supporting ANSI X12 EDI, CSV, flatfiles and JSON
- Workflow Designer to automate processes, support for webhooks and C# scripts, and FTP file download/upload with error logs

Two types of audience

New to DataBridge

- "Integration" sounds like a six-month project
- No developer on staff to assign to it
- Partners are already asking for feeds

Already building with it

- Writing and deploying your own scripts today
- Want to know where the ceiling actually is
- Wondering what else it can reach

DISPATCH SCIENCE

An integration is a chain of blocks

Most blocks are configured, not coded. The script block is the only place code appears.

1 Fetch

Pick up the file, or catch the event

2 Map

Read the rows into structured records

3 Script

Apply your logic

4 Act

Create orders, write files, call an API

Build your integration block by block:

FtpFetchBlock → MapObjectsBlock → ScriptBlock → CreateOrderBlock

What you configure, no code needed

Connection

Which SFTP site or API endpoint to use. Set up once, reused by every integration.

Source & schedule

Which folder to watch, and how often to look in it.

File shape

Header row or not, delimiter, fixed-length layout.

LIVE IN SPEEDYSHIP

CSV Import (with Headers)

Connection	**DS Test SFTP**
Watched folder	**/CSV_in_Customer**
Polling interval	**Every 14 minutes**
File shape	**Comma-delimited, header row on**
Result	**Orders created in DS**

Not one line of code in any of that. Every value above is a field on a form.

Settings: definable values, no code change

Each integration carries its own configurable list of Settings. The script asks for them at run time.

- **API keys and tokens** — Credentials for the third party you are calling
- **Defaults** — Account, service level or vehicle type to fall back on
- **Thresholds and caps** — Maximum fee, minimum weight, tolerance windows
- **Feature flags** — Turn a behaviour on or off without a redeploy
- **Lists** — One setting holding a delimited list the script loops over

On the speedyship CSV import

Key1	`value1`
Key2	`list1,list2`

Key2 shows the pattern: one setting, a list the script walks.

One script, five accounts, five behaviours — changing what it does is a config change, not a code change.

DISPATCH SCIENCE

The linked mapping file

How it works

A spreadsheet is attached to the integration itself.

The script reads it cell by cell - every cell arrives with its sheet name, its row name and its column heading.

Lookups read like a sentence: find the row for this postal code, read the column for this service level.

What people put in it

- Partner SKU → DS parcel type
- Customer location code → DS account
- Postal code or zone → rate
- Partner status code → DS milestone

Your ops team edits a spreadsheet. The integration behaves differently tomorrow.

DISPATCH SCIENCE

Output logs you can actually use

- **Logged per block**

  Timestamp, duration, success and “completed with errors” tracked separately from outright failure.

- **Every payload kept**

  The input and output of each block is kept for you to examine

- **Per record, not per run**

  One run that created six orders logs six separate order-creation entries.

- **Failures point at themselves**

  Fetch ✓ → Map ✓ → Script ✗. You know in five seconds where to look.

- **Resubmit**

  Re-run a past run; the history records what it was re-run from.



DISPATCH SCIENCE

It’s not just pushing orders to a partner

Financial systems — invoice completed → accounting or ERP

Customer status feeds — milestones → EDI 214 or a REST push

Enrichment from external APIs — tolls and mileage, flight tracking, tax lookup

Internal automation — DS reacting to DS, no file transport at all

Notifications and data out — Teams or Slack alerts, scheduled BI extracts

DISPATCH SCIENCE

Three ways in, three levels of scope

What sets it off

- **Scheduled** — Watch a folder every N minutes
- **Event** — DS fires it on order, route or invoice activity
- **On demand** — You start it, or resubmit a past run

Who it applies to

- **Tenant** — Everyone, every order
- **Company** — One division or brand
- **Account** — One demanding customer, nobody else affected

Live in speedyship today

31
live subscriptions

9
different accounts

19
event types in use

Owning one yourself

A standard configuration path

1. Choose your integration type
2. Set your connections
3. Set the FTP folder and the schedule (depending on your integration type)
4. Fill in the mapping sheet (if you use one)
5. Fill in the settings
6. Write the script
7. Run it once and read the log

WITH A DEVELOPER

Write the script

- Grouping and splitting logic, conditional pricing
- Calls out to Dispatch or third-party APIs

Pick one manual process that happens every week. We will wire it up with you in a working session.

DISPATCH SCIENCE

Thank you for your time today.

Any questions?