DATAFLEX BUILD SYSTEM

DF-CLI AS THE FOUNDATION FOR MODERN DATAFLEX WORKFLOWS

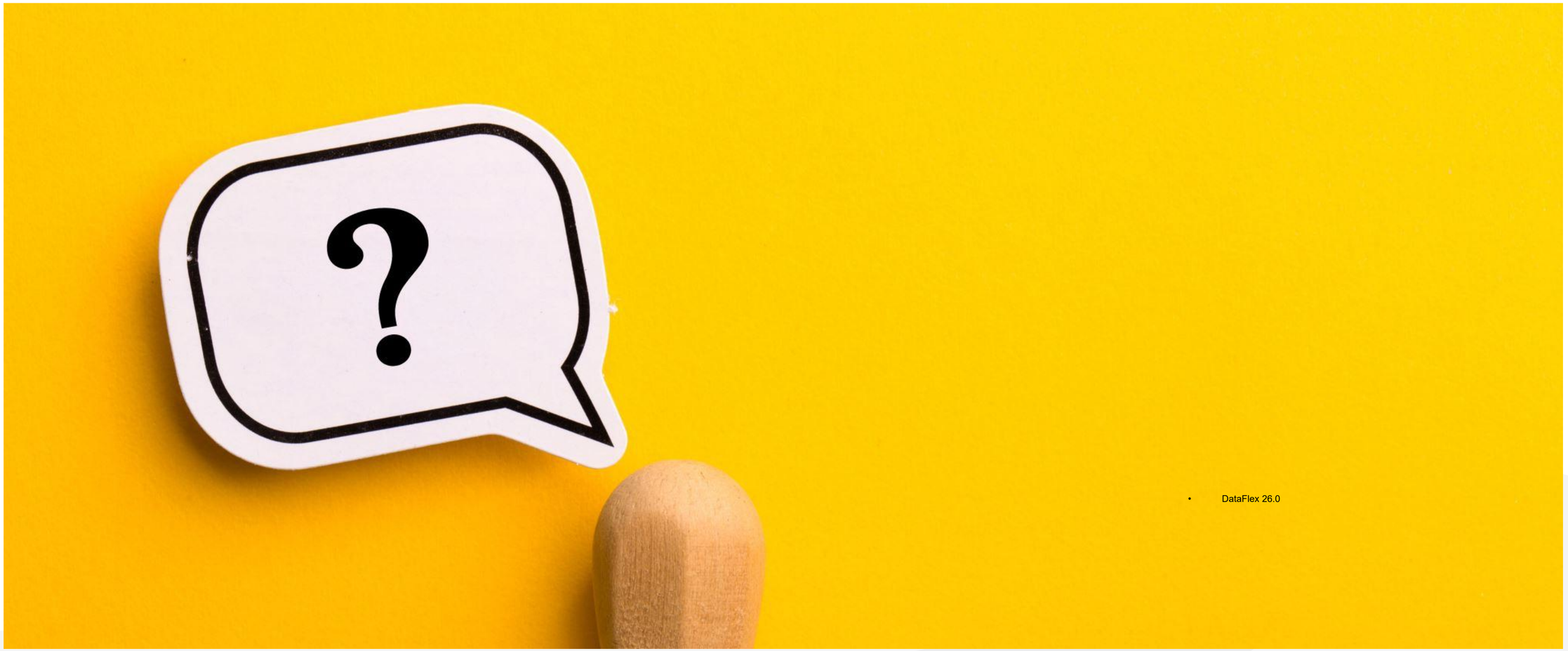
HENRI ALREADY SHOWED THE PACKAGE MANAGER PART, BUT IT IS SO MUCH MORE!

DataFlex 26.0

WE HAVE SPOKEN ABOUT ALL THESE PLATFORMS DATAFLEX IS GOING TO SUPPORT

DataFlex 26.0

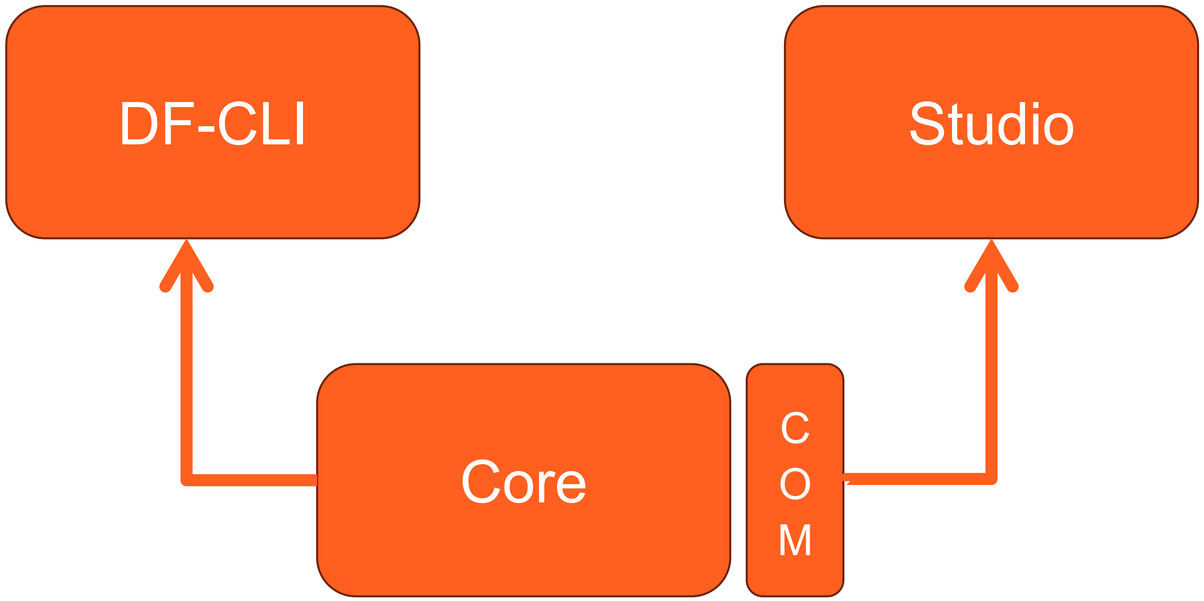
BUT NOW THE QUESTION BECOMES HOW ARE WE GOING TO DEVELOP ON THEM?!



- DataFlex 26.0

DATAFLEX 26.0 BUILD SYSTEM
THE STORY IN ONE SENTENCE

- Package manager was built to be platform independent from the very start
- To prepare for the future 27.0
- The DataFlex Studio through a COM interface allows for asynchronous operations
- The CLI is there for cross platform operations

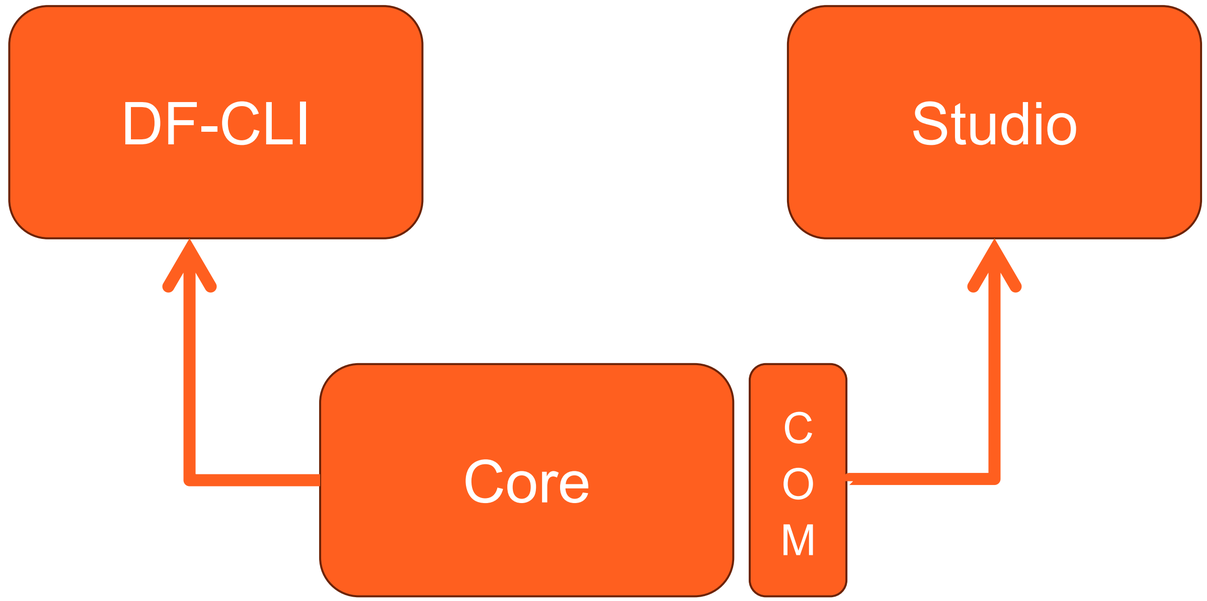


/* THE PACKAGE MANAGER STARTED THE JOURNEY.
DF-CLI BECAME THE FOUNDATION. */

DataFlex 26.0

DATAFLEX 26.0 BUILD SYSTEM
IS SO MUCH MORE

- Manages your workspace
- Build mechanics
- Security & Compliance



DATAFLEX 26.0 BUILD SYSTEM
THE STORY IN ONE SENTENCE

- The package manager was the original flagship project. The long-term product is broader: df-cli is now the core command-line build system that understands everything and is the center of the ecosystem:
  - Workspaces
  - Dependencies
  - Toolchains
  - IDE integration
  - Package publishing
  - Supply-chain metadata

LET’S START WITH THE WORKSPACE MANAGEMENT!

DataFlex 26.0

STRUCTURE
SWS, LOCK FILE, AND CONFIG.WS

- Before 26.0 workspace files were INI-based which of course stems from that fact that we were developing on Windows
  - There is no real INI-standard
- To plan ahead we decided on moving to JSON
  - To be more expressive
  - Provide more information in a more readable structure

DataFlex

WHAT MOVES INTO THE WORKSPACE

- Workspace paths (migrated from Config.ws)
- Projects
- Dependency intent
- Package metadata and settings

ON CONFIG.WS

- Previously the SWS file and Config.ws need to exist together to inform the Studio
- Now the Config.ws is the result of the SWS and is only needed and used at runtime
  - So don’t change the paths in Config.ws while working in the Studio it is only a file to change in a production environment

DF-CLI IS .SWS AGNOSTIC AND CAN STILL READ INI-BASED FILES!

LET’S TAKE A LOOK AT THE NEW SWS FILE

STRUCTURE
SWS, LOCK FILE, AND CONFIG.WS

- The lockfile is playing a crucial role in reproducibility
- Take the package expression that Henri used **“DataFlex-dev/Web UI<2.0.0”**
- Depending on the time that you open your workspace that expression might resolve to a different version
- Alice works on a project for a while, we issue an update, Bob needs to help out and opens that workspace
- To make sure you get the same version every time a lockfile locks expressions to a specific version
- Should be pushed and stored together with the sws in Git!

WITH THE WORKSPACE NOW IN JSON HOW DOES DFCOMP OR DFCOMPCONSOLE BUILD MY APPLICATION?

IT DOESN’T!

DATAFLEX 26.0 BUILD SYSTEM
A BUILD IS NOW A WORKFLOW, NOT A LOOSE COMMAND

LOAD WORKSPACE → RESOLVE DEPENDENCIES → PREPARE WORKSPACE → SELECT TOOLCHAIN → COMPILE

df-cli understands the new workspace format; dfcomp and dfcompconsole do not. Build automation therefore moves from loose compiler invocation to a workspace-aware build workflow.

WITH TECH STACK WE GET MULTIPLE TOOLCHAINS

DIFFERENT COMPILERS TO BUILD YOUR APPLICATION WITH!

A BETTER DECISION TO HAVE ONE CORE!

HOW DOES A BUILD WORK?
THE FLOW

1. Read a workspace
2. Load project details
3. Load dependencies and optionally install/repair them
4. For each project that it needs to build lookup the toolchain
5. When a project is to be build, we dynamically load dfcomp.dll/dfcomp.dylib/dfcomp.so
6. Then build the project using that loaded compiler
7. And optionally switch it out for the next project!

HOW DOES A BUILD WORK?
WHAT INDICATES THE TOOLCHAIN TO USE?

- A project.cfg file
- Already used since 20.0 for 64-bit
  - x86
  - x64

```
1    [Compiler]
2    Platform=0.1.5+native-64
```

- Only the studio was actually using these!
- Previously we had Bin64/DfCompConsole and Bin/DfCompConsole
  - This is no longer leading and will always use the specified one automatically switching
  - Override using –toolchain.
- The cool thing is that with 26.0 you can build older DataFlex projects as well (>=19.1)

DATAFLEX 26.0 BUILD SYSTEM
THE 26.0 COMMAND-LINE SHIFT

CONFIGURATION AND BUILDING

- config

Open, inspect or repair a workspace.

- build

Recommended workspace-first build command.

- build-file

Compile one source file with workspace context.

RUNTIME AND ENVIRONMENT COMMANDS

- run

Build when needed and launch.

- toolchain

Inspect installed DataFlex toolchains.

- system

Check the local environment.

DEMO!

DATAFLEX 26.0 BUILD SYSTEM
DEPENDENCY LIFECYCLE COMMANDS

DISCOVER AND INSTALL

- package search
- package details
- package install

These commands make dependency discovery and adoption part of the workspace workflow.

MAINTAIN AND CLEAN UP

- package update
- package upgrade
- package uninstall
- cache verify / prune

Lock resolved versions and branches in sws.lock.sws; held-back versions stay deliberate.

HOW DOES CACHING WORK?
LET'S SEE THAT IN ACTION!

SOMETHING MORE CONCERNING SECURITY

SECURITY
LICENSES

- Henri already spoke about this a bit; Licenses
- For both our security as well as yours!
- You can specify under what conditions a package can be used.
- Follows the SPDX format
  - (MIT OR BSD-3)
  - MIT
  - DocumentRef-DataFlex → ”licenseFile” → “LICENSE.txt”

DATAFLEX 26.0 BUILD SYSTEM
SBOM METADATA

- df-cli can generate a CycloneDX JSON Software Bill of Materials for a workspace. That turns package metadata and the resolved dependency graph into an artifact that can be archived, audited or passed to other security tooling.

Why it matters

- Auditing
- Compliance
- Vulnerability tracking
- Customer and partner trust
- Package accountability
- License metadata uses SPDX identifiers or expressions.

DATAFLEX 26.0 BUILD SYSTEM
PUBLISHING PACKAGES

- A workspace defines package metadata such as version, SPDX license, icon and installation behavior. A valid license is required when publishing packages to the cloud service.

- Example flow
- df-cli login
- df-cli package pack MyLib.sws
- df-cli package push MyLib.sws product

Build, package, publish and consume — with accountable metadata.

SO, LET’S TAKE A LOOK AT ALL THIS SBOM BUSINESS!

DATAFLEX 26.0 BUILD SYSTEM
REFERENCE PAGES

- df-cli Command Line Interface - high-level overview and core areas

https://docs.dataflex.dev/Tools/DfCli/Usage/

- df-cli Command Line Reference - commands and options

https://docs.dataflex.dev/Tools/DfCli/Command%20Line%20Reference/

- Workspace documentation - JSON .sws structure, dependencies and SBOM

https://docs.dataflex.dev/Tools/DfCli/Workspace/

QUESTIONS?