CPILint @ sitMUC 2024

Automate your SAP Cloud Integration governance with open source

open source program™

CPILINT

Use with Cloud Integration

Who am I?

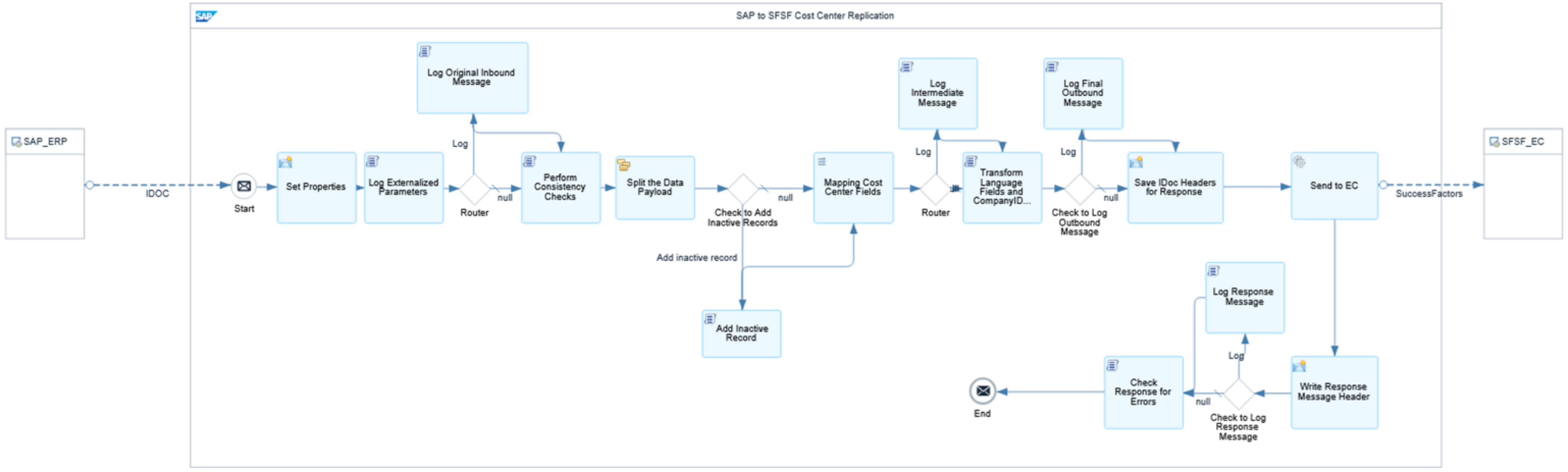
- Morten Wittrock
- Two decades of SAP integration experience
- Works at cbs Corporate Business Solutions
- Based in Heidelberg, Germany
- German last name but in fact Danish
- Part of the SAP Mentors program
- SAP PRESS author
- Frequent speaker at industry and community events
- Loves contributing to the SAP community
- Find me here:
  - LinkedIn
  - SAP Community

Corporate Business Solutions

SAP Cloud Integration in a nutshell

- SAP Cloud Integration is SAP’s process integration platform for the cloud
- One of the capabilities of the SAP Integration Suite
- SAP Integration Suite is SAP’s iPaaS offering
- Related to SAP XI/PI/PO, but built for the cloud from the ground up
- SAP Cloud Integration is based on the Apache Camel open-source integration framework
- Integrations are modelled visually using integration flows

Integration flow example


SAP to SFSF Cost Center Replication
SAP_ERP
IDOC
Start
Set Properties
Log Externalized Parameters
Log Original Inbound Message
Log
Router
null
Perform Consistency Checks
Split the Data Payload
Check to Add Inactive Records
null
Add inactive record
Add Inactive Record
Mapping Cost Center Fields
Log
Log Intermediate Message
Router
Transform Language Fields and CompanyID...
Log
Log Final Outbound Message
Check to Log Outbound Message
null
Save IDoc Headers for Response
Send to EC
SuccessFactors
SFSF_EC
Write Response Message Header
Check to Log Response Message
Log
Log Response Message
null
Check Response for Errors
End

Integration governance and guidelines

- SAP Cloud Integration gives you a lot of freedom in how to build your integrations
- This is a good thing, but it also makes it harder to achieve consistency across your integration landscape
- Consistency makes it easier to:
  - Build new integrations
  - Maintain existing ones
  - Debug failures
- Part of your integration governance should be guidelines for how to build integration flows
- Examples of such guidelines:
  - How to perform mappings
  - Which adapters you are allowed to use
  - Which scripting languages to use
  - Which naming conventions to follow

Integration governance and guidelines

- There is an inherent problem with such guidelines
- They often sit in a Word document in SharePoint or Confluence
- You can point developers and consultants to this document
- “This is how we do things around here”
- But are your integrations actually compliant?
- To find out you would have to manually inspect each integration flow
- This approach does not scale!
- The size of the problem increases with the number of developers and the amount of integration content
- This is the problem CPILint exists to solve

What is CPILint?

- CPILint is a tool that automates your compliance checks
- It ships with a bunch of built-in rules
- You pick the rules that your integration flows should comply with
- CPILint does the heavy lifting of checking each integration flow for compliance
- CPILint is a command line tool
- You can run it interactively from your computer or, for instance, in a CI/CD pipeline
- Released in 2019 with five new versions since then
- The latest version, 1.0.5, was released in August, 2024

CPILint’s rules

- CPILint currently ships with 24 rule variations
- Examples of rules:
  - NamingConventions
  - DuplicateResourcesNotAllowed
  - UserRoles
  - JavaArchives
  - SenderAdapters/ReceiverAdapters
- The rules are documented in the CPILint wiki
- Key takeaway: The rules are not best practices in and of themselves!
- Choose the ones that make sense in your context

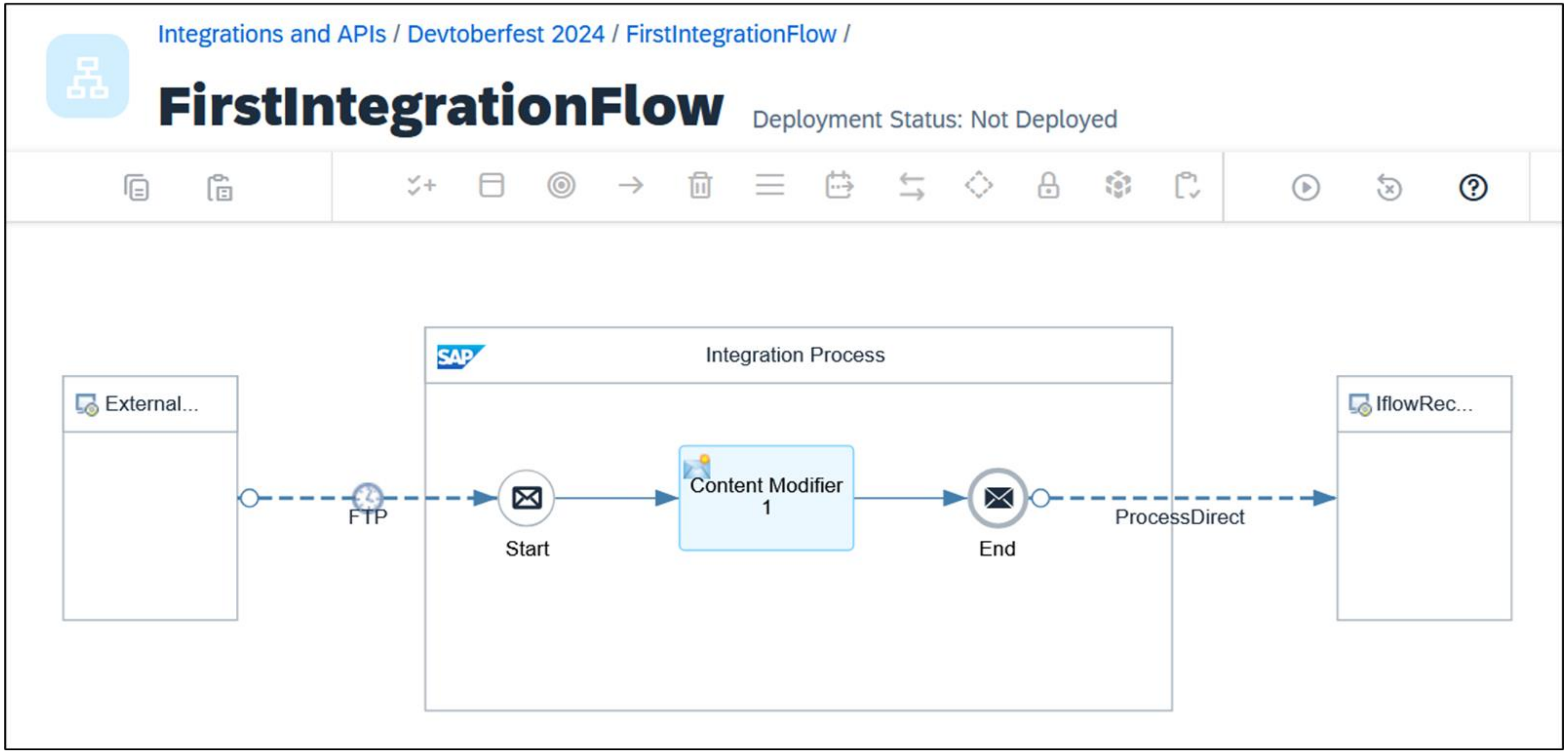
What's in a name?

- So why is the tool called CPILint, anyway?
- Let's break down the name:
  - CPI: Short for SAP Cloud Integration
  - Lint: A tool that flags "programming errors, bugs, stylistic errors, and suspicious constructs" (Wikipedia)
- Linters analyse your code and suggest improvements
- There are many linters out there like:
  - abaplint (ABAP)
  - UI5 linter (UI5)
  - ESLint (JavaScript)
- So: CPILint is a linter for SAP Cloud Integration

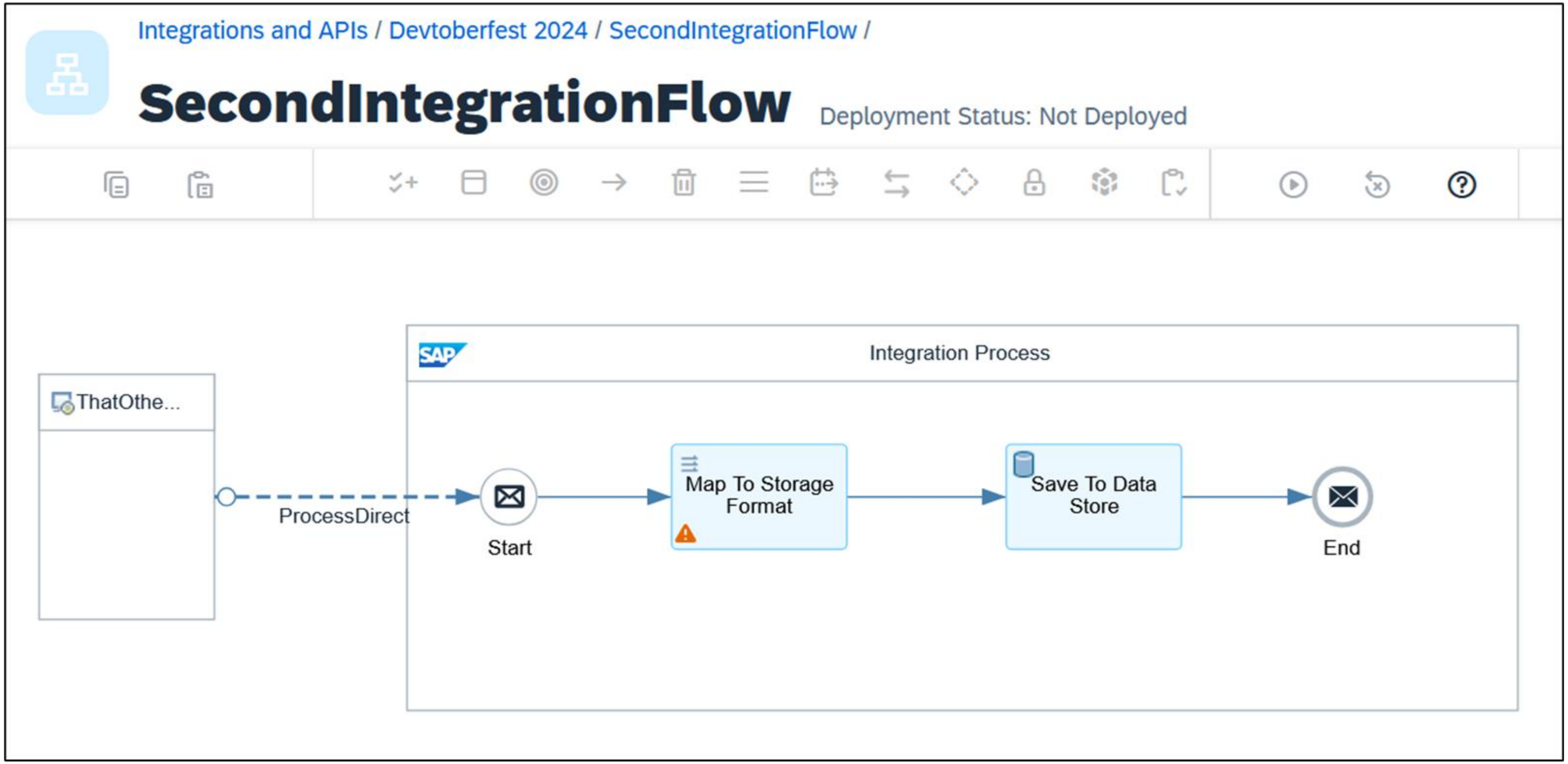
Demo

Let’s see CPILint in action!

FirstIntegrationFlow



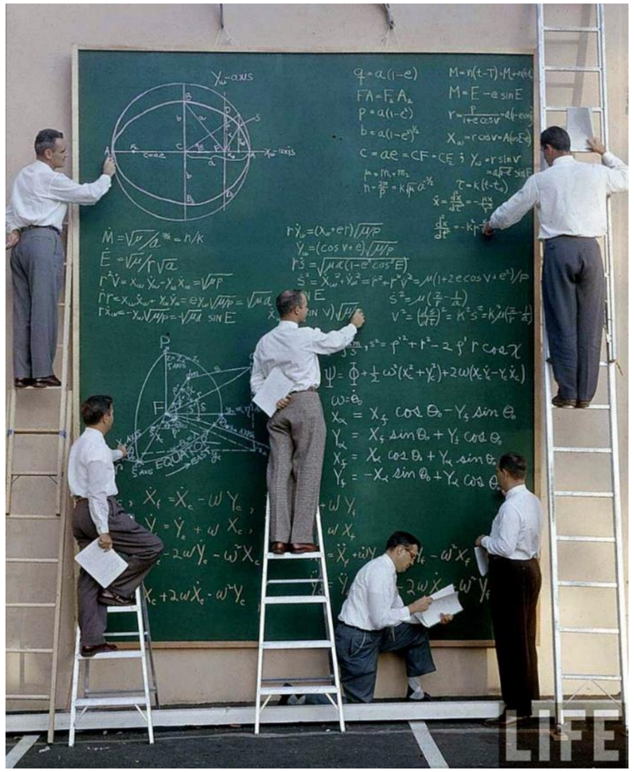
SecondIntegrationFlow


Integrations and APIs / Devtoberfest 2024 / SecondIntegrationFlow /

SecondIntegrationFlow Deployment Status: Not Deployed

How does CPILint work?

- Like other linters, CPILint performs static analysis
- In other words: It inspects your integration flows, but it doesn’t run them
- CPILint communicates with your tenant using the official OData API
- Integration flows are stored internally as XML documents in the BPMN format
- Most rules analyse these XML documents
- Some of them only inspect resources such as scripts and XSLT stylesheets



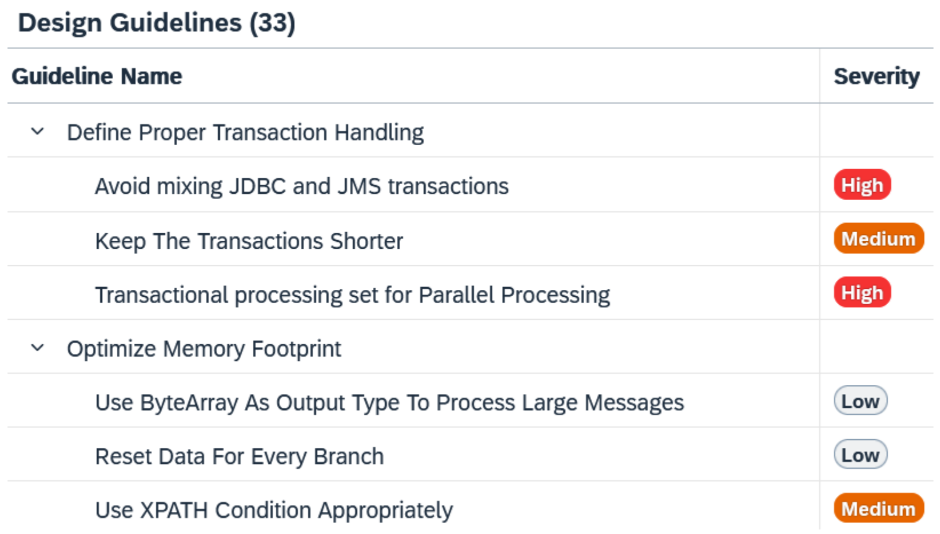
Why command line?

- Why does CPILint not have a nice Fiori UI?
- Part of it is personal preference
- But mostly it’s about flexibility
- You can:
  - Run CPILint interactively from your own computer
  - Run it in a scheduled script that sends you email in case of issues
  - Run it inside your CI/CD pipeline
- A GUI CPILint would not have this versatility at all

```
    **** COMMODORE 64 BASIC V2 ****
 64K RAM SYSTEM  38911 BASIC BYTES FREE
READY.
```

CPILint and the Design Guidelines

- The SAP Cloud Integration documentation has long had integration flow design guidelines
- As of April of 2024, you can run compliance checks directly in the UI
- The Design Guidelines feature and CPILint are very similar in concept
- But there is not a lot of overlap in the rules
- Should you use one or the other?
- Use both for even better coverage!



Design Guidelines (33)

Guideline Name	Severity
⌄ Define Proper Transaction Handling	
Avoid mixing JDBC and JMS transactions	High
Keep The Transactions Shorter	Medium
Transactional processing set for Parallel Processing	High
⌄ Optimize Memory Footprint	
Use ByteArray As Output Type To Process Large Messages	Low
Reset Data For Every Branch	Low
Use XPATH Condition Appropriately	Medium

CPILint is open source

- CPILint is free and open source
- What does that mean for you?
- First, that you have access to the complete source code:
  - On GitHub and included in every CPILint download
  - Want to learn how some part of the tool works? You can
- Second, that the license allows you to modify the code:
  - CPILint is licensed under the MIT License
  - Want to change or add something? Go right ahead; the license lets you do that

open source
initiative
Approved License®

Getting started with CPILint

- To run CPILint, you need a Java runtime (11+) on your system
- Go to GitHub and download the latest release
- Follow the installation instructions in the wiki
- That’s all there is to it!
- If you want to play around with the code, check out the build instructions

What’s on the road map?

- More rules
- Exemptions = certain integration flows are allowed to not be compliant with certain rules
- Even more names supported by the NamingConventions rule
- Extensibility = adding your own, custom rules
- Have a suggestion or an idea? Create a GitHub issue!

Thank you!

Questions are welcome