

Morgan Stanley



Bringing Devops enablement to your mainframe developers

Calder Sacks, Executive Director

February 2026

Bio Page

CALDER SACKS

Executive Director

Enterprise Z

Phone: +1 212 276-1952

Calder.Sacks@morganstanley.com

Years of Experience: 33 Years

Years at Morgan Stanley: 26 Years

Biography

Worked in many roles within the mainframe technology landscape. Started as a Natural mainframe programmer and moved into the Adabas DBA role. Lead the database engineering team at Morgan Stanley for 10+ years and now lead the team working on Mainframe integration strategy and technologies for our Cobol/DB2 mainframe environment.

Been working on technologies between mainframe and distributed since 2003.

- One of the first companies to call Java RPC services from our mainframe Batch and CICS processes back in 2005
- Setup Sybase access from the mainframe.
- Implemented SQL & ACI access to mainframe Adabas and CICS
- Running Kafka Connectors running on the mainframe to distributed Kafka

Our goal is integrating mainframe to be just another platform and to leverage what has already been built where possible

Table of Contents

Why do it ?

Our approach for leveraging Git

Our journey

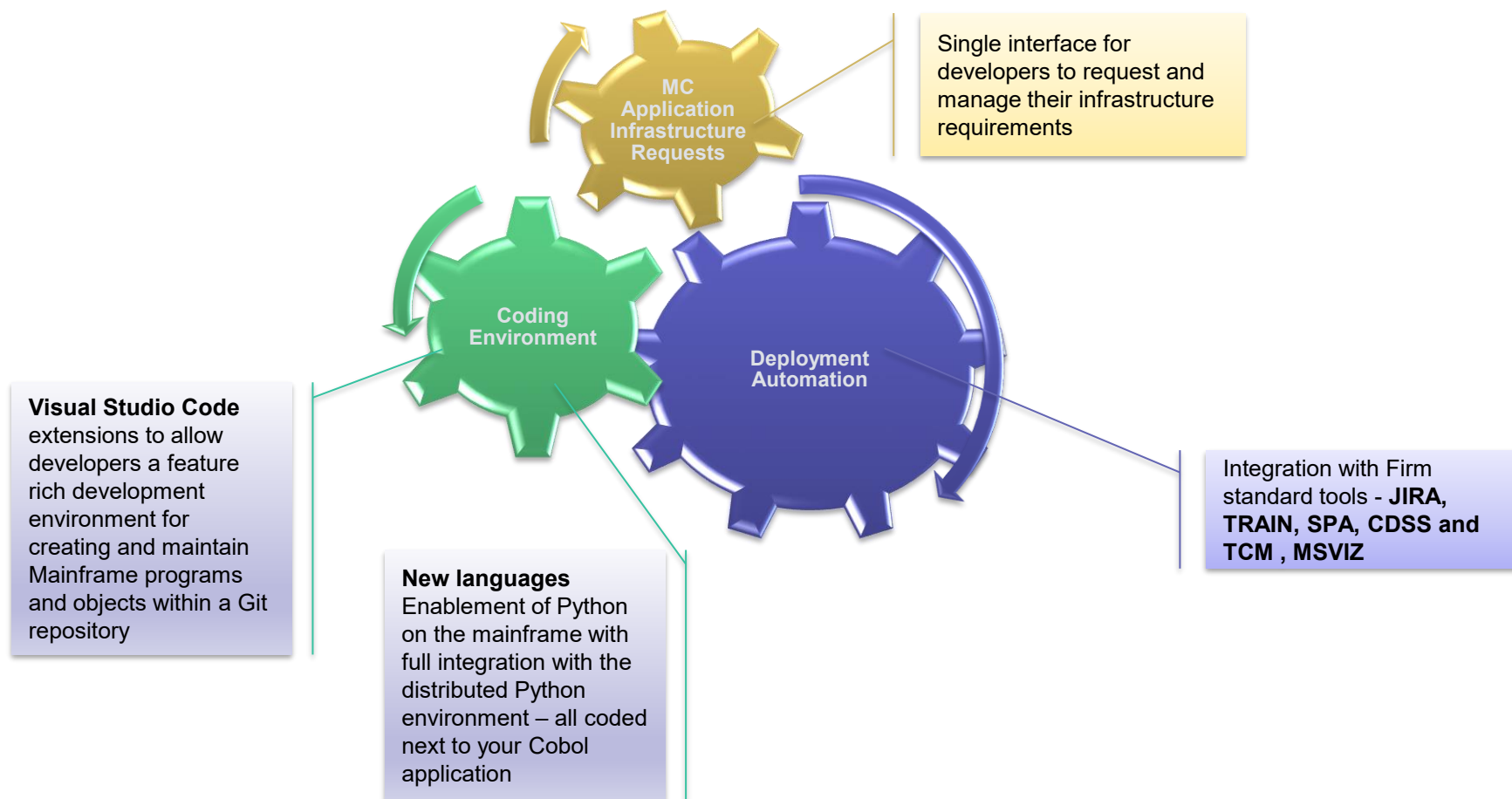
Lessons learnt

Why do Git for mainframe applications?

- We were looking at ways to integrate our mainframe code base into the broader change management eco system
 - Mainframe change management systems have been around for many years
 - A number of bridges have been built over the years between mainframe change control systems and the newer distribute change control systems
 - Some of the mainframe change control systems have a limited interface to Git – some are more integrated but none of them can interface with our change control workflow
- Within our space the distributed change control process and workflows have now become the standard we have to adhere to
 - There are tighter controls and reporting required, both internally and externally
 - Newer reporting and measuring systems are built around distributed code and not for the mainframe
 - Many of our application owners now have mainframe and distributed code and they have to manage them separately
 - We cannot build bridges fast enough to talk to all the new systems in the change control workflow
- Change systems and workflows have greatly evolved in the Git and CI/CD environment

Project Goal

Simplify and unify the development environment, by providing a modern feature and function rich set of tools in an easy-to-use environment, to improve the mainframe developer experience when making requests for mainframe resources and deploying changes to the mainframe environment with full integration into the Firm standard CI/CD tool set



Our approach to using Git

We had a few challenges to overcome

- How do you get only the changed programs ?
 - For Java we rebuild the whole app
 - For Mainframe we only want to build the 1 program that changed
- How do you copy from a distributed environment concept of ownership into a shared environment ?
 - Distributed apps tend to live in a certain set of directories that they own
 - Mainframe apps are all in a limited number of shared datasets
- How do you get your distributed CI/CD tooling to talk to the mainframe as just another node ?
- Getting distributed teams to understand mainframe call structures
- Getting mainframe people to understand distributed paradigms
- Do you put your code in Git on distributed side or keep it on the mainframe
 - Who becomes the Golden Source ?
 - And a few other lively debates....

Our approach – Key decisions first !

GIT WILL BE THE GOLDEN SOURCE

This decision was the driver behind much of the designs. We chose this approach because in our environment the distributed change control process is the new standard. To best integrate with it we need to be a part of it from the beginning of the application lifecycle.

- Support all pieces of an application
 - JCL, Procs, Parms, Control Cards, Stored Procedures, ISPF panels, REXX and many other component types

INTEGRATE WITH THE EXISTING SYSTEMS

Don't treat the mainframe as different. In the end it is just computer code, treat it the same as any other code.

BUILD ONLY WHAT IS REQUIRED

Mainframe is not distributed and Distributed Is not mainframe, identifying the differences and catering only for these helped us build targeted approaches such as our gateway.

It starts with a Modern IDE

The image shows a side-by-side comparison of a code editor (VS Code) with a mainframe-style code editor. The left editor shows a file named `PMCACDS1.jcl` with the following content:

```
1 //PMCACDSX JOB (DWR,754010,MCA--PB2,PAADMXF,PRD),
2 // 'AAA DAILY',USER=PRDAAA1,CLASS=7,MSGCLASS=N,REGION=0M
3 /* {{ environment }}
4 /*
5 //JOBLIB DD DISP=SHR,DSN=DWS.PROD.PRDEXEC
6 // DD DISP=SHR,DSN=DB32.DSNEXIT
7 // DD DISP=SHR,DSN=DB32.DSNLOAD
8 // DD DISP=SHR,DSN=MVSP.MIGRATON.SFSYLOAD
9 /*
10 /*
11 /*
12 /*
13 //PMCASAMP EXEC PMCAIVP1
```

The right editor shows the same code with a green callout box pointing to it: "Auto Complete and Real time Syntax checking are all part of the VS Code Plugins".

Below the code editors is a terminal window showing the execution of a PowerShell task:

```
Executing task: powershell -ExecutionPolicy Bypass -File M:\dist\mcappdev\PROJ\mfgit-core-launcher\prod\mfgit-lint-invoke
.ps1 -filePath jcl\PMCACDS1.jcl

Using baseUri: https://mfgit-gateway.webfarm.ms.com
Failed Rules Found in jcl\PMCACDS1.jcl:
{
  "ruleName": "jcl-4",
  "message": "Job Name does not match Member Name",
  "detail": [
    "# 00001: //PMCACDSX JOB"
  ],
  "warning": false
}
```

A yellow callout box points to the terminal output: "Side by side comparisons were never possible before on the mainframe".

At the bottom right, a blue callout box states: "We can now do Audit rule checking as you code instead of only at compile time".

Using Copilot to explain Cobol code

The screenshot shows the VS Code interface with a COBOL program open in the editor. The program is named 'MCACDS08.cbl' and contains the following code:

```
1 IDENTIFICATION DIVISION.  
2 PROGRAM-ID. 'MCACDSS2'.  
3  
4 DATA DIVISION.  
5 WORKING-STORAGE SECTION.  
6 COPY 'MCACDSS1'.  
7  
8 PROCEDURE DIVISION.  
9 DISPLAY " _MFGIT_NEWCHANGE_ ".  
10 MOVE 12346 TO EMP-ID.  
11 MOVE 'John Doe' TO EMP-NAME.  
12 MOVE 50000.75 TO EMP-SALARY.  
13 MOVE 50000.75 TO EMP-BONUS.  
14  
15 DISPLAY 'Employee ID: ' EMP-ID.  
16 DISPLAY 'Employee Name: ' EMP-NAME.  
17 DISPLAY 'Employee Salary: ' EMP-SALARY.  
18 DISPLAY 'Employee Bonus: ' EMP-BONUS.  
19 STOP RUN.
```

On the left, the GitHub Copilot chat window provides an explanation of the code:

Let me know if you need further assistance!

hirattak_msid
Please explain this cobol

GitHub Copilot
Used 1 reference

This COBOL program, named `MCACDSS2`, is a simple example that demonstrates basic COBOL syntax, including the use of a copybook, variable assignments, and displaying output. Here's a breakdown of the code:

1. IDENTIFICATION DIVISION

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. 'MCACDSS2'.
```

- Purpose:** Identifies the program and provides metadata.
- PROGRAM-ID:** Specifies the name of the program, which is `'MCACDSS2'`.

2. DATA DIVISION

```
DATA DIVISION.  
WORKING-STORAGE SECTION.  
COPY 'MCACDSS1'.
```

- Purpose:** Defines the program's data.
- WORKING-STORAGE SECTION:** Used to declare variables and constants that persist throughout the program's execution.
- COPY 'MCACDSS1':**
 - Includes a copybook named `MCACDSS1`.
 - A copybook is a reusable file that typically contains variable declarations or common code.

How can I install the Perl-based `rename` command on Ubuntu?

Ask Copilot

At the bottom, the terminal shows a message from the Zowe Explorer plugin:

```
Using baseUri: https://mfgit-gateway.webfarm.ms.com  
Failed Rules Found in jcl\PMACADS1.jcl:  
{  
  "ruleName": "jcl-4",  
  "message": "Job Name does not match Member Name",  
  "detail": [  
    "# 00001: //PMACADS3 JOB"  
  ],  
  "warning": false  
}
```

Using VS Code and leveraging your code in GITHUB gives you access to Copilot and more of the extended tools available

There are an array of other plugins that you can now use to dramatically enhance the overall developer experience

Changing our JCL scripts to Python for Git processing

Existing JCL loaded with ZOWE

z/Python scripting

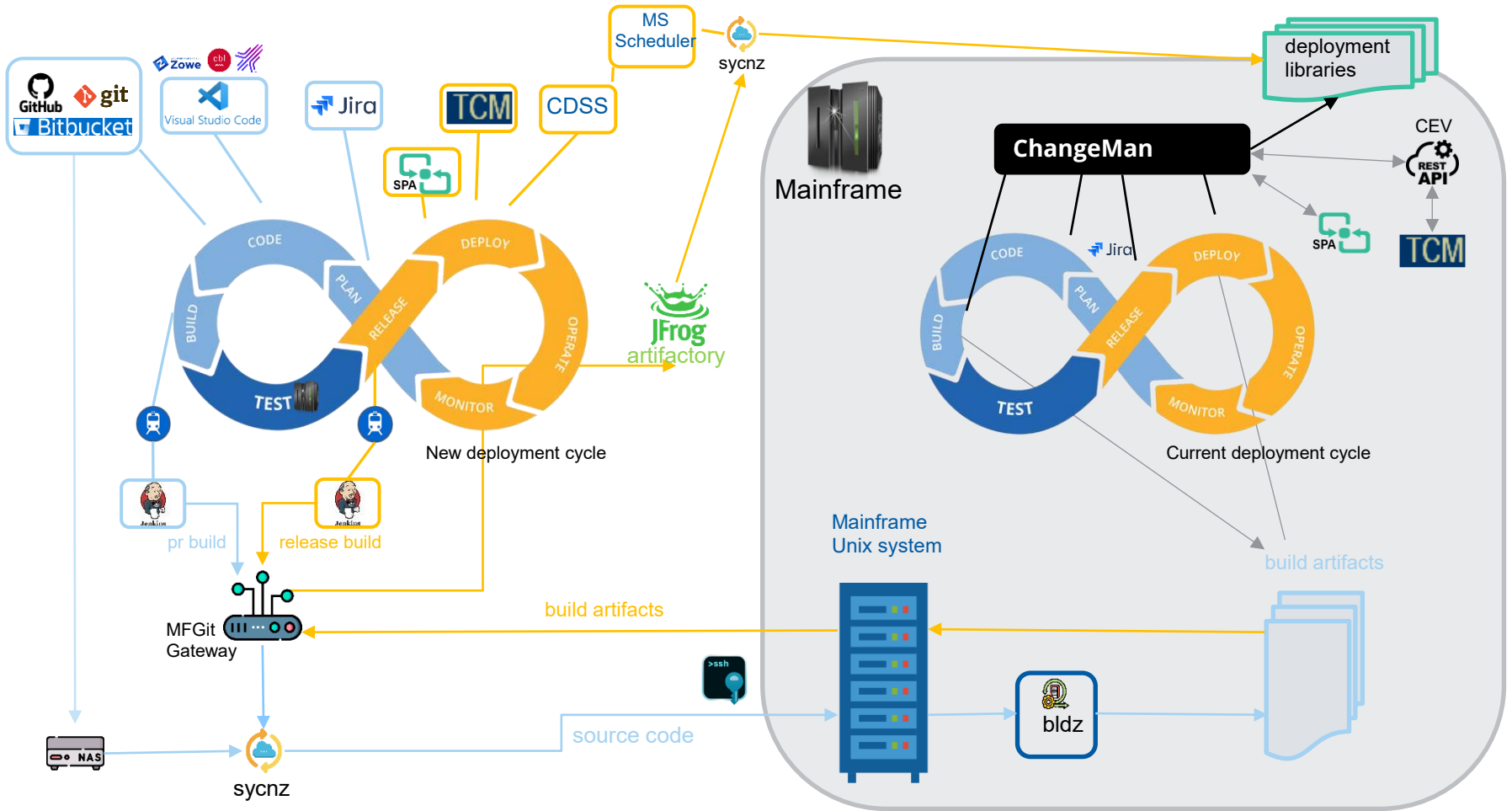
Getting Co-pilot help

The screenshot displays a development environment with three main components:

- Left Panel (JCL Script):** Shows a JCL script for job P814FH51JOB. It includes steps for allocating a temporary dataset (ALLOCATE TEMP DATASET), executing a program (EXEC PGM=IEFBRI4), and displaying the dataset (DISPLAY EXEC PGM=IEBGENER).
- Middle Panel (Python Script):** Shows a Python script using Click to create a CLI for updating a dataset. It defines a command `update_dataset_with_dd` with options for data definition and a function that calls the JCL script's `update_dataset_with_dd_test` function.
- Right Panel (Chat):** Shows an AI chat window with the following content:
 - Initial question: "Could you share the full command or script you're executing when you see this error?"
 - AI response: "can you add the step to create/allocate temporary dataset &&TEMP in the first step"
 - User question: "I can see you're working with a JCL file. Let me read the current content to add a step for creating/allocating a temporary dataset:"
 - AI response: "Read", "Now I can see your JCL! Let me add a first step to allocate a temporary dataset &&TEMP:"
 - User question: "Using 'Replace String in File'"
 - AI response: "Added temporary dataset allocation step!"
 - What I Added:**

```
*****  
/* ALLOCATE TEMP DATASET  
*****  
//ALLOC EXEC PGM=IEFBRI4  
//TEMP DD DSN=&&TEMP,  
// DISP=(NEW,PASS),  
// SPACE=(TRK,(1,1)),  
// UNIT=SYSDA,  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
```
 - What This Does:**
 - Step Name: ALLOC - Allocates the temporary dataset
 - Program: IEFBR14 - IBM utility that does nothing (just processes DD statements)
 - Dataset: &&TEMP - Temporary dataset (double ampersand makes it temporary for the job)
 - DISP=(NEW,PASS): Creates new dataset and passes it to subsequent steps
 - SPACE=(TRK,(1,1)): Allocates 1 track primary, 1 track secondary
 - UNIT=SYSDA: Uses system disk
 - DCB Parameters: Fixed block, 80-byte records, 800-byte blocks
 - User question: "Now &&TEMP is available for use in subsequent job steps! You can reference it in later steps with DSN=&&TEMP."
 - AI response: "Let me read the current JCL file and add a step at the end to display the &&TEMP dataset:"
 - User question: "Read"
 - AI response: "Now let me add a step at the end to display the contents of &&TEMP:"

The next step : Change the Mainframe Deployment Process



The next step is to move your code to Git

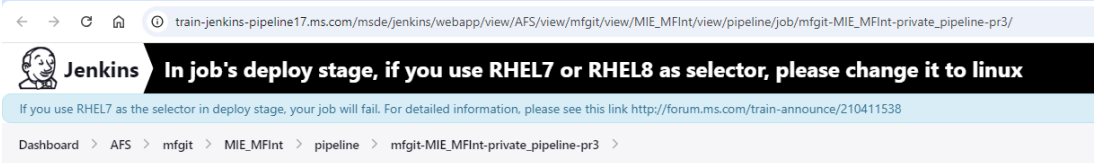
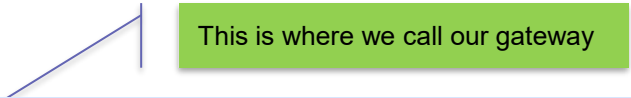
The screenshot shows a Bitbucket repository page for the path `wmmutualfunds/wm_mf_mnfrm_archiving / wm_mf_mnfrm_archiving`. The repository is currently on the `master` branch. A green callout box with a blue arrow pointing to the repository content contains the following text:

Using Git allows the mainframe components to now become part of the eco system
In our environment you can use GITHUB or Bitbucket

Source	Description	Size	Last Modified
..			
PICIRO03.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	1.84 KB	30 Apr 2025
PICPRO04.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	1.84 KB	30 Apr 2025
PIFSRO04.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	1.85 KB	30 Apr 2025
PIFSRO05.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	1.84 KB	30 Apr 2025
PIFSRO06.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	1.84 KB	30 Apr 2025
PMFSRVP1.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	5.1 KB	30 Apr 2025
PMFSRVP2.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	5.1 KB	30 Apr 2025
PMFSRVP3.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	2.76 KB	30 Apr 2025
PMFTRAX1.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	2.97 KB	30 Apr 2025
PMFTRVP1.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	5.1 KB	30 Apr 2025
PMFTRVP2.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	5.89 KB	30 Apr 2025
PMFTRVP3.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	4.72 KB	30 Apr 2025
PSCSRVP1.jcl	TIPS-34325 Data Management - Mutual Funds - Table Data Purge - Phase 2 and VSAM Cleanup	1.98 KB	30 Apr 2025

At the bottom of the page, it states: "Git repository management for enterprise teams powered by Atlassian Bitbucket".

Setup the Jenkins Pipelines to run



Status

✓ mfgit-MIE_MFInt-private_pipeline-pr3

Pipeline Job triggered by user.
 Quick Links for mfgit/MIE_MFInt [project page](#) .
 User is not allowed to edit the job configuration or replay a build. Should modify jenkinsfile instead in order to change the configuration.

Stage View

	Pre-Build	Build and Test	Declarative: Post Actions
Average stage times: (Average full run time: ~4min 57s)	1min 18s	2min 30s	4s
#37 Jan 15 15:00 No Changes	810ms		
origin/pull-requests/31/from Dec 05 16:21 No Changes	1min 55s	3min 30s	3s

✓ mfgit-MIE_MFInt-promotion_pipeline-deploy3

Pipeline Job triggered by user.
 Quick Links for mfgit/MIE_MFInt [project page](#) .
 User is not allowed to edit the job configuration or replay a build. Should modify jenkinsfile instead in order to change the configuration.

Stage View

	Promotion workflow	Deploy	Declarative: Post Actions
Average stage times: (Average full run time: ~1min 23s)	31s	44s	1s
2024.10.28-6 => qat Oct 28 14:37 No Changes	23s	31s	1s

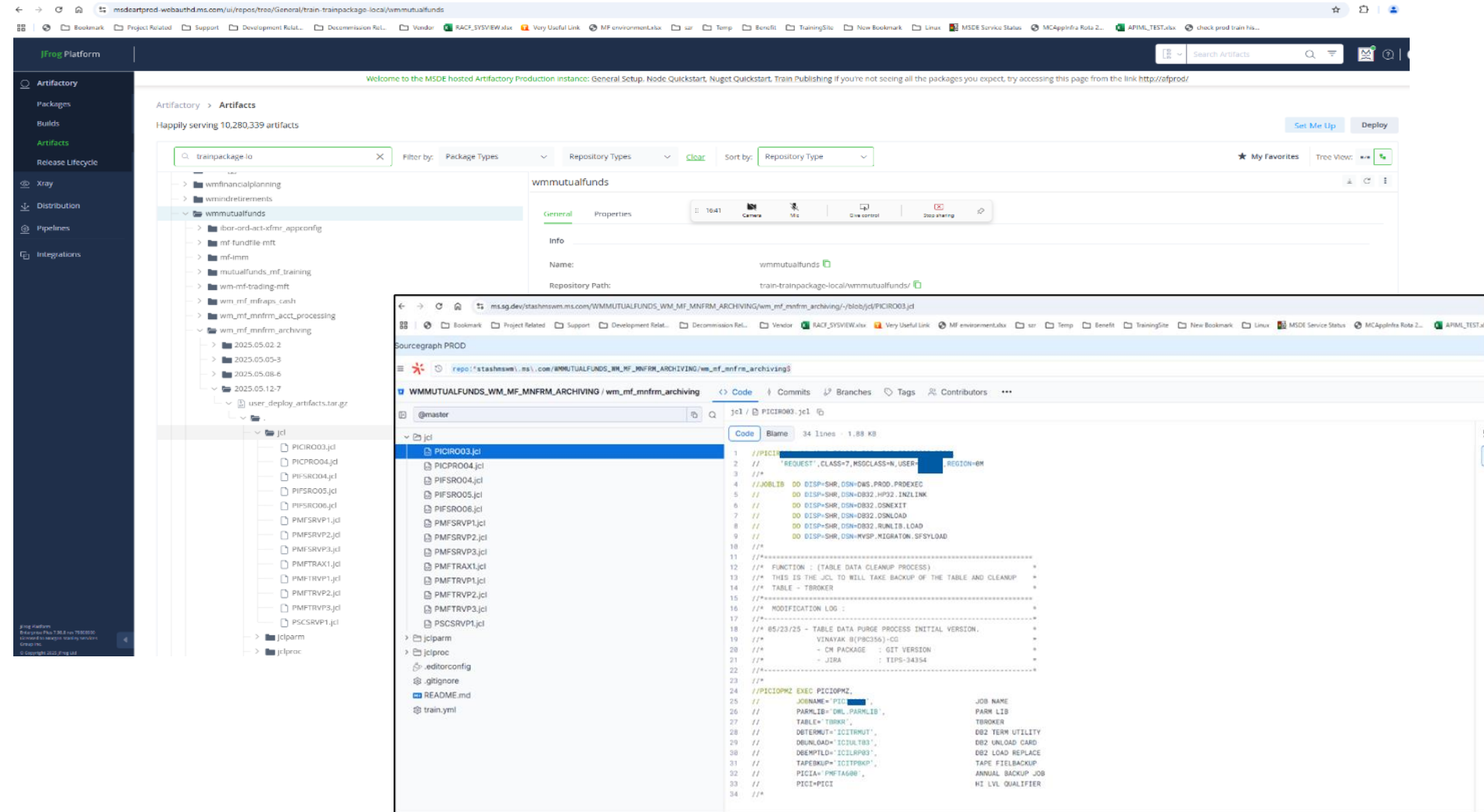
✓ mfgit-MIE_MFInt-release_pipeline-main3

Pipeline Job triggered by user.
 Quick Links for mfgit/MIE_MFInt [project page](#) .
 User is not allowed to edit the job configuration or replay a build. Should modify jenkinsfile instead in order to change the configuration.

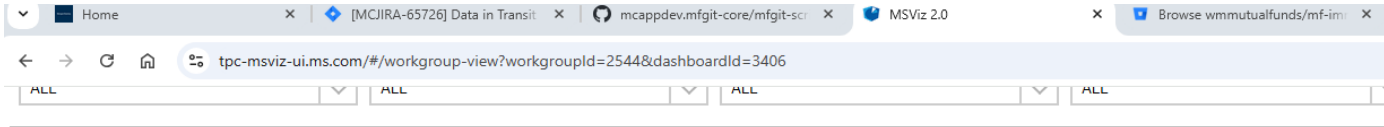
Stage View

	Release Workflow	Pre-Build	Release Build and Test	Dist	Declarative: Post Actions
Average stage times: (Average full run time: ~8min 19s)	77ms	50s	5min 0s	2min 10s	1s
2024.12.05-7 Dec 05 16:37 10 commits	111ms	57s	5min 49s	2min 38s	

The end result is that the Mainframe code can now be seen in tools such as Artifactory or Sourcegraph



Mainframe code can now be measured in standard reporting tools



DevOps Adoption

Group Name: (All) Team Name: (All) Super Division: Technology Division: WMT Super Dept: (All) Department: (All) System Name: (All) Year Month: 2025-May

IFPAT Reporting & Analytics

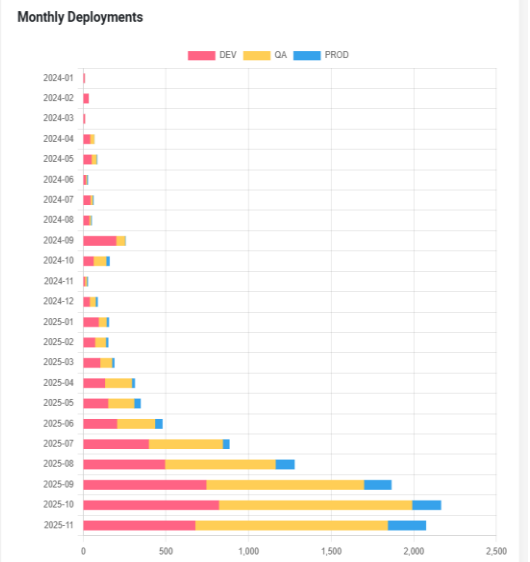
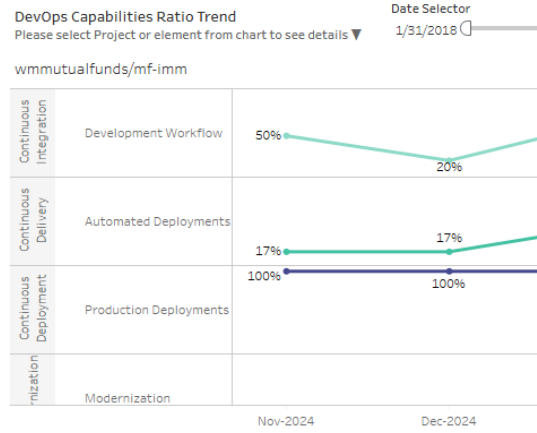
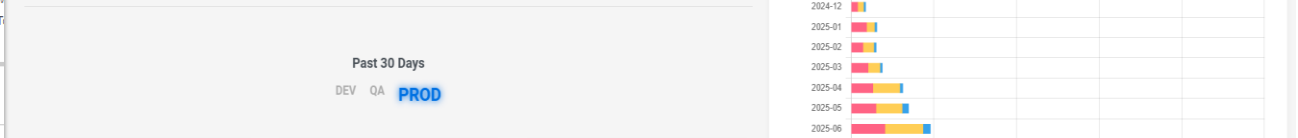
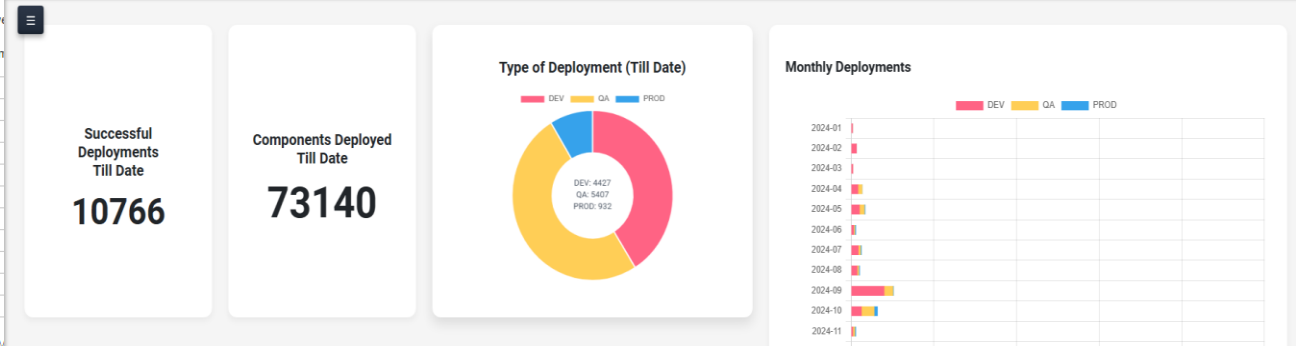
Last Data Refresh: 5/12/2025 5:02:14 PM

DevOps Capabilities by Project


Project	In Train	Last Activity Time	Investment Buy
wmmutualfunds/mf-fundfile-cons...	75%		50%
wmmutualfunds/mf-fundfile-db	80%		50%
wmmutualfunds/mf-fundfile-mft	65%		50%
wmmutualfunds/mf-fundfile-recon	75%		50%
wmmutualfunds/mf-fundfile-ssis	75%		33%
wmmutualfunds/mf-fundfile-svc	90%		50%
wmmutualfunds/mf-fundfile-vend...	95%		17%
wmmutualfunds/mf-imm	65%		50%
wmmutualfunds/mf-pdt-pub-svc...	85%		50%
wmmutualfunds/mf_jptl_eod_rec...	85%		17%
wmmutualfunds/mf_jptl_eod_re...	80%		17%
wmmutualfunds/mf_ops_svc	85%		50%

Legend: Application Security, Development Workflow, Automated Deployments, Production Deployments, Modernization

MFGIT Deployment Dashboard



Create dashboards and metrics for people to track progress and get help



EZ GIT Community

Following in ↶ 1 stream

[Leave group](#)

☆☆☆

TECH EXCELLENCE '25 RUNNER UP


BANKING TECH AWARDS '25 SPL. MENTION

581+

STRONG EZGIT COMMUNITY!

ANNOUNCEMENT: [MFGIT OUTAGE Window: January 23 9:15PM NYT - January 25, 2026 at 5:00AM NYT](#) [Show Details](#)

ADOPTION DASHBOARDS



1. [Training & Adoption Tracker](#)
2. [Consolidated Deployment Dashboard](#)
3. [Adoption Trendlines: Migration Tracker](#)
4. ***NEW:** [Change2Git Dashboard](#)

5. [WM Deployment Dashboard: WMTDevCentral](#)
6. [WM GIT Repos Stats: WMTDevCentral](#)

7. [Training instructor roster: MFGIT Adoption Training Roster.xlsx](#)
8. [Adoption Directory: Deprecated. Please refer Training Tracker instead.](#)

EVENT CALENDAR

< January 2026 >







None

RECENTLY JOINED

*New: [Changeman to MFGit Component Bulk Copy Utility](#)

MFGIT ADOPTION: START HERE

Modern DevOps practices can now be adopted by Mainframe users too. GIT for Change management is being implemented across the firm. GIT on Mainframe will replace Changeman as the source control tool, and teams can now get onboard the central Train for deployment of code. We have tons of resources to help you adopt these modern practices, and this Jive page will provide helpful links and guidance for anyone ready for the change. Below are a few important links to get you started:

	<ul style="list-style-type: none"> • Register and take this Video based classroom session if you are ready to implement MFGIT (Select based on whether you are a BitBucket user or Github user) <ul style="list-style-type: none"> ◦ MFGIT Adoption Training for Github users [Github is the firm standard] ◦ MFGIT Adoption Training for BitBucket users [Please check with your ITSO if you really need to use BitBucket. Otherwise, go in for the firm-standard Github]
	<ul style="list-style-type: none"> • MFGit - A one-stop page with step by step instructions for adoption of MFGIT process
	<ul style="list-style-type: none"> • Join the MFGIT Community Teams Channel • ONLY FOR THE Enterprise Z (MC) teams >> Please join this Teams Channel for posting your questions during training
	<ul style="list-style-type: none"> • Please find the recording of the KYI(Know Your Infrastructure) session on MFGIT (Mainframe Modernisation using DevOps), presented by Tejas Jadhav.
	<ul style="list-style-type: none"> • Frequently faced issues (FAQ) while implementing MFGIT • NEW! Component ownership issues while migrating to MFGIT? Raise a ServiceNow Ticket by going to the Support page on Change2Git. Also update the ownership of those components in MRC/
	<ul style="list-style-type: none"> • Distribution list to reach out to all 580+ Mainframe developers: mfgit_forum@morganstanley.com • Have feedback to share about our training? Please take the survey: MFGit Training Survey

Join the Mainframe GIT User Community

Changeman Block: Sep 15

Starting Sep 15, Changeman will be blocked. [Read more ...](#)


MFGIT Champions (Wealth Mgmt.)

MFGIT Internals

LATEST IN BLOGS!

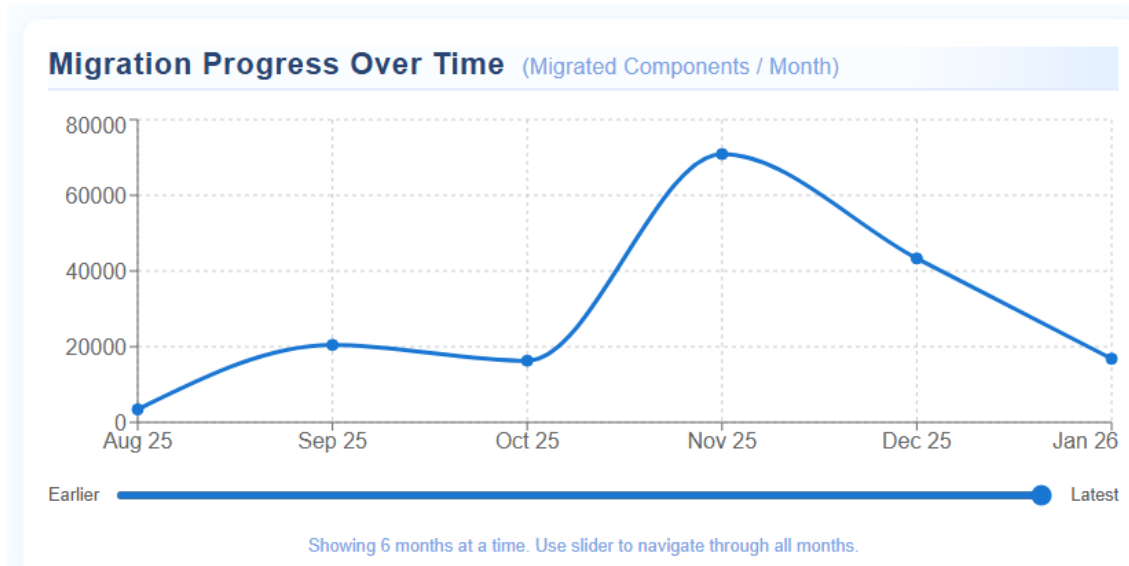
- [MFGIT AUDIT/LINT: Notes - Taka Hirata](#)
- [Leveraging Git for mainframe code - Calder Sacks](#)

USEFUL DOCS



- [SDLC Jargons](#)

Create dashboards and metrics for people to track progress and get help



Total Components Migrated

172035

Avg. Components Migrated Per Month (Over Past 3 Months)

43662

Lessons learnt

- Start with an IDE to get people more comfortable
- It is a journey
 - We started 2 years ago with VS Code and the plugins
- Replace functions one step at a time
 - Change control systems on the mainframe are purpose built
 - Pick the core components to start
 - Build new function support as you mature on the journey
- People need time to adapt and learn
 - Some teams will be quicker than others to adopt
 - It is a complete paradigm shift for developers
 - Mainframe engineering teams need to learn a lot of new concepts as well
 - Setup and structure is very different and requires in depth understanding

Lessons learnt

- Technologies we used
 - Cobol and JCL
 - Zowe and the mediation layer
 - Python and Java Script with extensive YAML files
 - USS on the mainframe
 - Team Build
- Most importantly - Setup a cross functional team to build it
 - Our team consists of 5 core people
 - Mainframe change control people that know the structure of the systems
 - Distributed people that know much of the distributed environment
 - The key is to identify people that are willing to learn both mainframe and distributed
 - Interactions with many of the teams outside of the core team requires “translation” and training

Questions ?