



Introducing Brightside

An Introduction to our Command Line Interface into the z Platform

Elliot Jalley – 30th May 2018 – 5.06

Prague Technology Days

May 30 - June 1, 2018



For Informational Purposes Only

This presentation was based on current information and resource allocations as of **May 2018** and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Certain information in this presentation may outline CA's general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this presentation "as is" without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages. CA confidential and proprietary. No unauthorized copying or distribution permitted.

Agenda

1

MEET THE FOLKS BEHIND BRIGHTSIDE

2

THE NEED TO MODERNIZE THE MAINFRAME DEVELOPMENT EXPERIENCE

3

INTRODUCING THE BRIGHTSIDE COMMAND LINE INTERFACE

4

HELP US BUILD BRIGHTSIDE

Team Brightside - Prague!



The need to modernize the mainframe development experience

A Developer's World

Project Brightside

What is available for today's developers to use?

- Lightweight text-editors for quick editing with community plugins
- Powerful IDEs for specialized languages
- CLIs to interact with services
- Choice of powerful scripting languages for build and automation
- Continuous Integration and Delivery orchestration tools



A Mainframe Developer's World

Project Brightside

What is available for today's developers to use if they are working on the Mainframe part of their application?



Eclipse-based IDEs that are customized for mainframe development



TSO/ISPF interfaces



JCL/Rexx for build and automated system testing

Let's Listen

Project Brightside

What are developers asking for with the mainframe?

Associate Software Engineer @ CA:

“I wish I could just use modern testing frameworks for writing mainframe automated tests.”

r/Mainframe user:

“I've recently been thinking about working on something to give me better integration for my z/OS coding and a local editor (likely Atom... *maybe* VSCode).”

Stackoverflow user:

“Which plugin should I use to get started with Jenkins to manage mainframe components in PDS format?”

Our Goal

Project Brightside

Enable developers and DevOps engineers to control, script and develop on the mainframe like any other cloud platform and allow them to make use of familiar, industry-standard, open-source tooling. By accomplishing this, Brightside:

- Addresses the skills gap by making mainframe development more attractive and accessible to next-generation developers
- Improves the application delivery lifecycle by allowing mainframe applications to be integrated into enterprise DevOps pipelines

Introducing Brightside CLI

Brightside Introduction

- CA Brightside is a Command Line Interface that provides a simple and streamlined way to interact with the IBM z Systems platform

ZOS-CONSOLE

ZOS-FILES

ZOS-JOBS

PROVISIONING

ZOS-TSO

FILEMASTER

ENDEVOR

CICS

DB2

Brightside Introduction

- Script-friendly
- DevOps Enablement
- Modern Developer Experience

```
$ bright
DESCRIPTION
-----

Welcome to the Brightside CLI! A "brighter" way to "mainframe"!

Brightside CLI is a command line interface (built on Node.js) that
simplifies and streamlines your interaction with the IBM z Systems
platform.

If this is your first time using Brightside CLI or you would like more
general information, issue the command "bright help explain brightside".
This command will explain the syntax of Brightside CLI commands and how
Brightside works.

If you are not sure where to start, or how to perform a specific task,
use the "bright help search all <term>" command (for example, you can
issue "bright help search all "dataset"") to search the help for
commands, descriptions, options, and more.

To send feedback to the maintainers of Brightside CLI, issue the
following command to open the feedback page in your default browser:
bright contribute send feedback

USAGE
-----

bright [group] [action] [object] [options]
```

Brightside Introduction



```
cmd (Admin)
C:\Users\SOLSU01> cd C:\Users\SOLSU01
C:\Users\SOLSU01> bright --h

DESCRIPTION
-----
Welcome to the Brightside CLI!

Brightside CLI is a command line interface (built on Node.js) that simplifies and streamlines your interaction with the IBM z Systems platform.

If this is your first time using Brightside CLI or you would like more general information, issue the command "bright help explain brightside". This command will explain the syntax of Brightside CLI commands and how Brightside works.

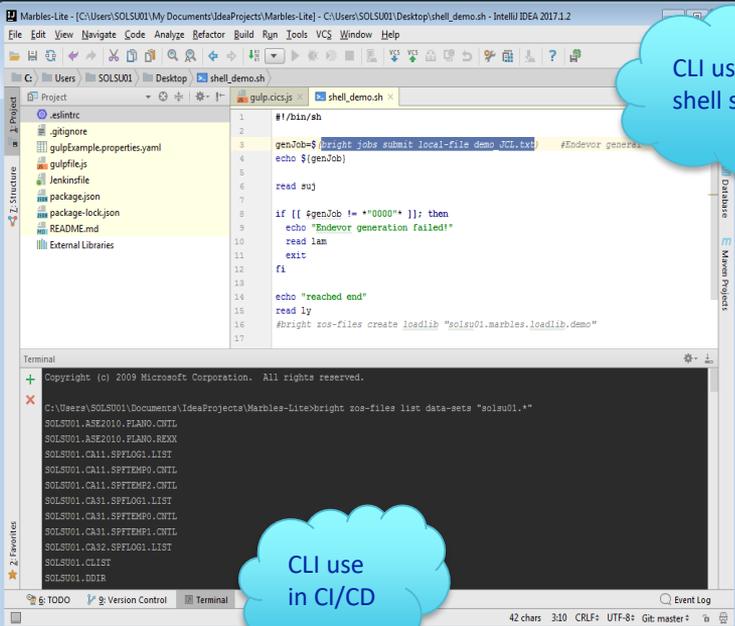
If you are not sure where to start, or how to perform a specific task, use the "bright help search all <term>" command (for example, you can issue "bright help search all "dataset"<") to search the help for commands, descriptions, options, and more.

To send feedback to the maintainers of Brightside CLI, issue the following command to open the feedback page in your default browser:
bright contribute send feedback

USAGE
-----
bright [group] [action] [object] [options]

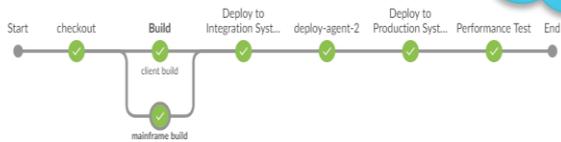
GROUPS
-----
ca-disk      (experimental) Interact with CA Disk Backup and Restore
cics         (experimental) Issue commands to interact with CICS regions
compiler     (experimental) Compile source code on the mainframe
config | cfg Configure Brightside
contribute   Generate Brightside Framework code and help

cmd.exe [166]8456
10/20/2016 1:11:11 PM  PRI: 8641  (3,85)251  10776 100%
```



```
Marbles-Lite - [C:\Users\SOLSU01\My Documents\IdeaProjects\Marbles-Lite] - C:\Users\SOLSU01\Desktop\shell_demo.sh - IntelliJ IDEA 2017.1.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
C:\Users\SOLSU01\Desktop\shell_demo.sh
Project: gulp.cicjs
1 #!/bin/sh
2
3 genJob=$(bright jobs submit local-file demo UCL.txt) #Endevor generation
4 echo ${genJob}
5
6 read suJ
7
8 if [[ $genJob != *"0000"* ]]; then
9   echo "Endevor generation failed!"
10  read lan
11  exit
12 fi
13
14 echo "reached end"
15 read ly
16 #bright zos-files create loadlib "solsu01.marbles.loadlib.demo"
17

Terminal
+ Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\SOLSU01\Documents\IdeaProjects\Marbles-Lite>bright zos-files list data-sers "solsu01.*"
SOLSU01.ASE2010.PIANO_CNFL
SOLSU01.ASE2010.PIANO_REXX
SOLSU01.CA11.SPFL01.LIST
SOLSU01.CA11.SPTEMP0_CNFL
SOLSU01.CA11.SPTEMP2_CNFL
SOLSU01.CA31.SPFL01.LIST
SOLSU01.CA31.SPTEMP0_CNFL
SOLSU01.CA31.SPTEMP1_CNFL
SOLSU01.CA32.SPFL01.LIST
SOLSU01.CLIST
SOLSU01.DOIR
```



```
graph LR
  Start((Start)) --> checkout((checkout))
  checkout --> Build((Build))
  Build --> DeployInt[Deploy to Integration System]
  DeployInt --> deployment-agent-2((deployment-agent-2))
  deployment-agent-2 --> DeployProd[Deploy to Production System]
  DeployProd --> PerfTest((Performance Test))
  PerfTest --> End((End))
  Build --- client-build[client build]
  Build --- mainframe-build[mainframe build]
```



```
Marbles-Lite - [C:\Users\SOLSU01\My Documents\IdeaProjects\Marbles-Lite] - C:\Users\SOLSU01\Desktop\shell_demo.sh - IntelliJ IDEA 2017.1.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
C:\Users\SOLSU01\Desktop\shell_demo.sh
Project: gulp.cicjs
1 #!/bin/sh
2
3 genJob=$(bright jobs submit local-file demo UCL.txt) #Endevor generation
4 echo ${genJob}
5
6 read suJ
7
8 if [[ $genJob != *"0000"* ]]; then
9   echo "Endevor generation failed!"
10  read lan
11  exit
12 fi
13
14 echo "reached end"
15 read ly
16 #bright zos-files create loadlib "solsu01.marbles.loadlib.demo"
17

Terminal
+ Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\SOLSU01\Documents\IdeaProjects\Marbles-Lite>bright zos-files list data-sers "solsu01.*"
SOLSU01.ASE2010.PIANO_CNFL
SOLSU01.ASE2010.PIANO_REXX
SOLSU01.CA11.SPFL01.LIST
SOLSU01.CA11.SPTEMP0_CNFL
SOLSU01.CA11.SPTEMP2_CNFL
SOLSU01.CA31.SPFL01.LIST
SOLSU01.CA31.SPTEMP0_CNFL
SOLSU01.CA31.SPTEMP1_CNFL
SOLSU01.CA32.SPFL01.LIST
SOLSU01.CLIST
SOLSU01.DOIR
```



Jenkins



CLI use in Source Build tools

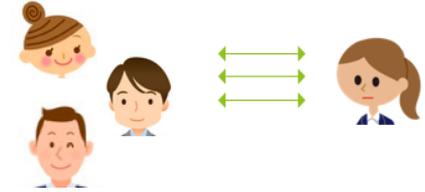
Developers use their **tools of choice**



She can use her **favorite code editor**

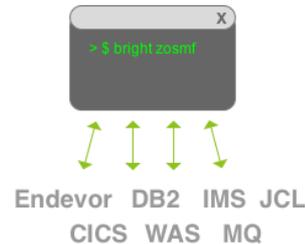


She works on a team, so **source control tools like Git** are essential



Using BrightSide, Michelle can code for the mainframe **as she would for any platform**

Using BrightSide commands, she can interact with the **mainframe's data and tools**



Michelle issues a few simple commands to **download the code to her PC**

Developers use their **tools of choice**

! Test Failed



Michelle edits the source and **issues a BrightSide command to generate a build**



Michelle can **debug and rebuild** her code without leaving her code editor



She can use **modern build tools like Gulp** and run a local unit test



Michelle commits her code, which triggers **Jenkins and her code pipeline** (QA, DEV, PROD)

Michelle gets a message that her code was **successfully tested and deployed**

Final Thoughts

- Brightside is a new software solution that bridges the MF / Distributed divide to make the mainframe easily accessible and programmable by the next-generation developers and DevOps Engineers.
- CA Service plugins are being developed with 100% automated DevOps enablement as the goal.

Final Thoughts

In an article, titled “The Return of the CLI: CLIs Being Used by API-Related Companies,” Bill Doerrfeld writes:

“A CLI is a powerful ally for improving developer experience, as it could be used in an **interactive mode** to debug, troubleshoot, or explore in real time, or used in a high-level, **abstracted way** with shell scripts to unite local and CLI actions.”

<https://developer.ibm.com/tv/wp-content/uploads/sites/62/2016/03/npm-1024x576.png>



[Project BrightSide - Mainframe Interaction Framework : Sneak Peek](#)

Help us Build Brightside

Benefits of Being a Validation Partner

- By joining Project Brightside on validate.ca.com, you will
 - Gain access to beta source code to try out Brightside CLI features first!
 - Provide direction on features so that Brightside will best fit your needs
 - Be invited to our Sprint Reviews to speak with our teams and see the latest developments and direction
- There is no commitment to purchase anything. We value your input and would like you to try out our product. We enjoy the benefits of the Brightside CLI at CA and hope you will too!

More Brightside at Prague Technology Days

Meet the team at our Booth
Wednesday afternoon



FileMaster Plus REST API and
its integration with other
Devops tools

17h30
Wednesday



Elliot Jalley

Principal Product Owner

Elliot.Jalley@ca.com



in

Call for Speakers Now Open



Share your experience. Be the teacher. Enhance your resume.

Register your session today!

Learn more: ca.com/caworld