

nova
net



Eric's new in
the neighborhood.

Mac's new on
the planet.

Mac and Me

Out of this world and into your heart.

AN R. J. LOUIS PRODUCTION "MAC AND ME" CHRISTINE EBERSOLE
JONATHAN WARD KATRINA CASPARY LAUREN STANLEY
First Introduction JADE CATEGORIES Mac ALAN SILVESTRI Mac BROOKS ARTHUR
Edits TOM WALLS Director of Photography NICK McLEAN Executive Producer MARK DAMON · WILLIAM B. KERR
Produced by STEWART RAFFILL AND STEVE FEKE Produced by R. J. LOUIS Directed by STEWART RAFFILL

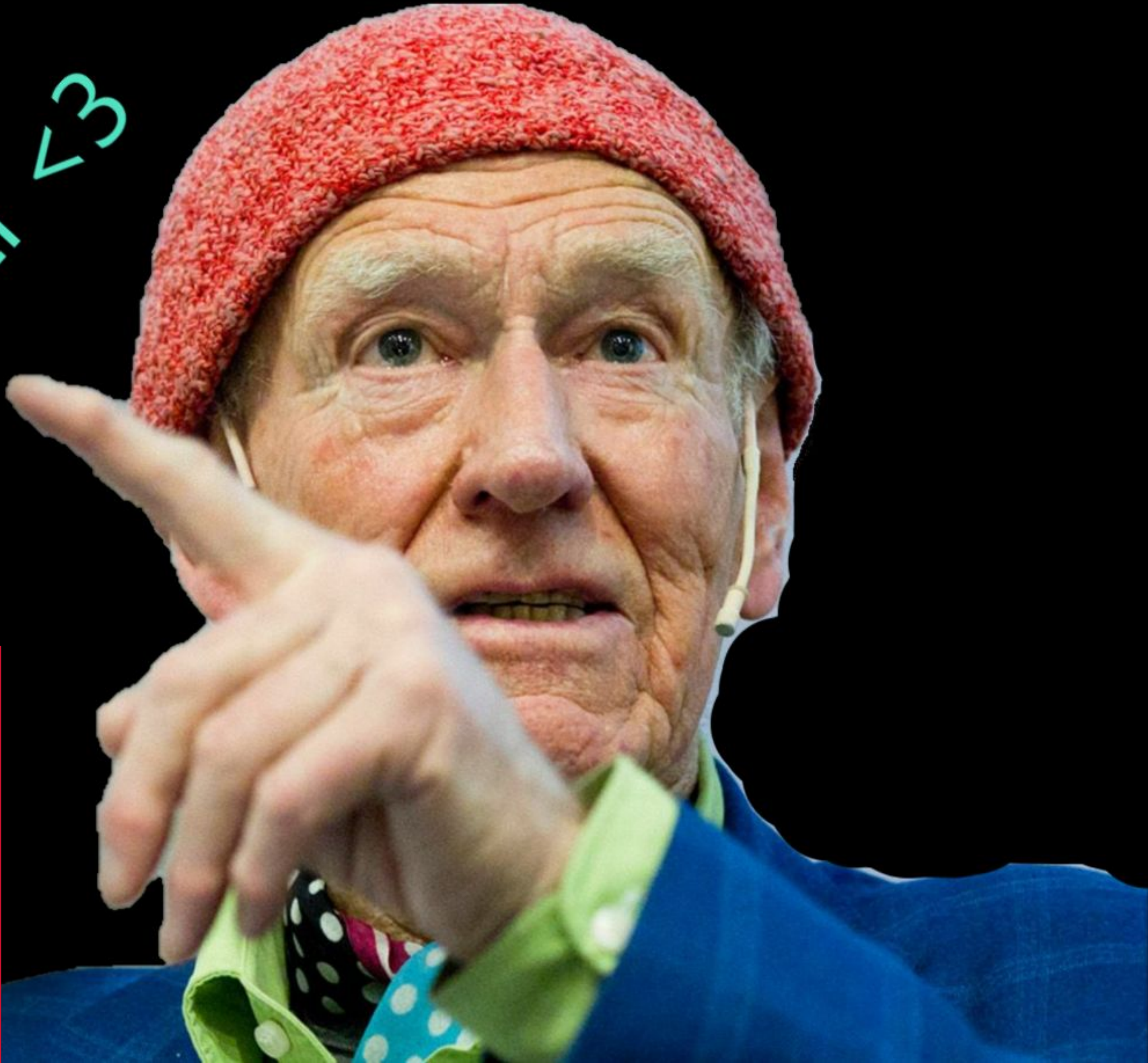
Printed by DeLuxe An ORION PICTURES Release PG PARENTAL GUIDANCE SUGGESTED
SOME MATERIAL MAY NOT BE SUITABLE FOR CHILDREN

Johan Grønstad

Azure CLI & me

Command line all the things (in Azure)

Azure CLI v3



nova
net

```
$ az resource show -g atlas-dev-rg -n atlas-dev-insights --resource-type microsoft.insights/components --query "properties.InstrumentationKey"  
"6492651c-5538-11e9-b5c4-677fba5051d1"
```



**HOLY CRAP
I'M BATMAN!**




```
1  #!/bin/bash
2  # secrets.sh
3  ENVNAME=$(echo $1 | tr '[:upper:]' '[:lower:]')
4  if [ "$ENVNAME" ≠ "local" ] && [ "$ENVNAME" ≠ "dev" ]
5  then
6      echo "usage: ./secrets.sh local|dev"
7      echo "example: ./secrets.sh dev"
8      exit 0
9  fi
10
11  azCmd="az.cmd" # Git-bash does not accept 'az'
12  if [ "$(uname)" = "Darwin" ] || [ "$(expr substr $(uname -s) 1 5)" = "Linux" ]
13  then
14      azCmd="az"
15  fi
16
17  echo "Environment:  $ENVNAME"
18
19  dotnet user-secrets set mysecret "Global Azure Bootcamp"
20
```



```
32
33 # secrets.sh modified
34 echo "Environment:  $ENVNAME"
35 echo "Logged into:  $($azCmd account show --query name)"
36 echo "Logged in as: $($azCmd account show --query user.name)"
37
38 dotnet user-secrets set mysecret "Global Azure Bootcamp"
39
```



```
57  # AZURE KEYVAULT ANNOYING STRINGS
58  # mysecret = fire
59  $ MYSECRET=$($azCmd keyvault secret show --vault-name mykeyvault-$ENVNAME --name mysecret --query value)
60  $ echo "This is $MYSECRET!"
61  This is "fire"!
62  $ MYSECRET=$($azCmd keyvault secret show --vault-name mykeyvault-$ENVNAME --name mysecret --query value | sed -e 's/^"//' -e 's/"$//')
63  $ echo "This is $MYSECRET!"
64  This is fire!
```


novα
net

```
Target.create "Azure-login" (fun _ →  
  let azureServicePrincipalId = getArg "--servicePrincipalId"  
  let azureServicePrincipalPassword = getArg "--servicePrincipalPassword"  
  let azureTenantId = "6798f877-a1d1-41fb-a343-0d3e3a99876a"  
  Az.login azureTenantId azureServicePrincipalId azureServicePrincipalPassword  
)
```

```
Target.create "Apply-arm-template" (fun _ →  
  let armTemplates = __SOURCE_DIRECTORY__ + "/arm-templates"  
  let resourceGroup = sprintf "atlas-%s-%s-rg" applicationName env  
  Az.applyArmTemplate env resourceGroup armTemplates  
)
```


novα
net

```
let az arguments workingDirectory =  
|   execProcess "az" arguments (fun o →  
|   { o with DisplayName = "Azure CLI"; WorkingDirectory = workingDirectory })
```


novα
net

```
module Az =  
    let exec arguments = az arguments "" ▷ ignore  
  
    let login tenantId servicePrincipalId password =  
        az (sprintf "login --service-principal --username %s --password %s --tenant %s" servicePrincipalId password tenantId) ""  
        ▷ ignore
```



```

let private execProcess name arguments createOptions =

    let arguments = // Split a command on whitespace, ignoring quoted sections
        let regex = new Regex(@"[ ](?=(?:[^\"]*" * "\"[^\"]*" * "\"")*[^"]*$)", RegexOptions.Multiline)
        regex.Split(arguments)
        ▷ Array.filter(String.IsNullOrWhiteSpace >> not)

    let joinArgs = String.concat " "

    let options = createOptions defaultOptions

    let cli = ProcessUtils.tryFindFileOnPath name
        ▷ function
        | Some cli → cli
        | None → failwithf "Can't find %s on path" (if String.IsNullOrEmpty options.DisplayName then name else options.DisplayName)

    try
        CreateProcess.fromRawCommand cli (arguments)
        ▷ CreateProcess.withWorkingDirectory options.WorkingDirectory
        ▷ CreateProcess.redirectOutput
        ▷ CreateProcess.disableTraceCommand
        ▷ CreateProcess.addOnSetup (fun () →
            Trace.tracefn "%s> \"%s\" %s \n" options.WorkingDirectory cli (joinArgs >> options.CensorTrace < arguments)
        )
        ▷ Proc.run
        ▷ fun res →
            if res.ExitCode <> 0 then failwithf "Step failed: %0" res.Result.Error
            res.Result.Output.Trim().Trim(' ')

    with ex →
        failwithf "Error calling %s %s dir: %s \n %0" name (joinArgs >> options.CensorTrace < arguments) options.WorkingDirectory ex

```


nova net

Johan Grønstad

