

Get and set default apps with **Utiluti**

2026
MACADMININS
CONFERENCE

WEDNESDAY, JULY 08, 2026
CIRCA MACOS 26.5.2; UTILUTI 1.5

Ross Matsuda Apple Systems Engineer at Ntiva

MacAdmins Slack: @Xirias

www.sudoade.com



Join the shared notes

<https://bit.ly/4wxh82R>



Why are we here?

- Get & Set Default Apps
- Change user behavior through system behavior
- Observe fleet state
- Enforce change through data-driven decisions



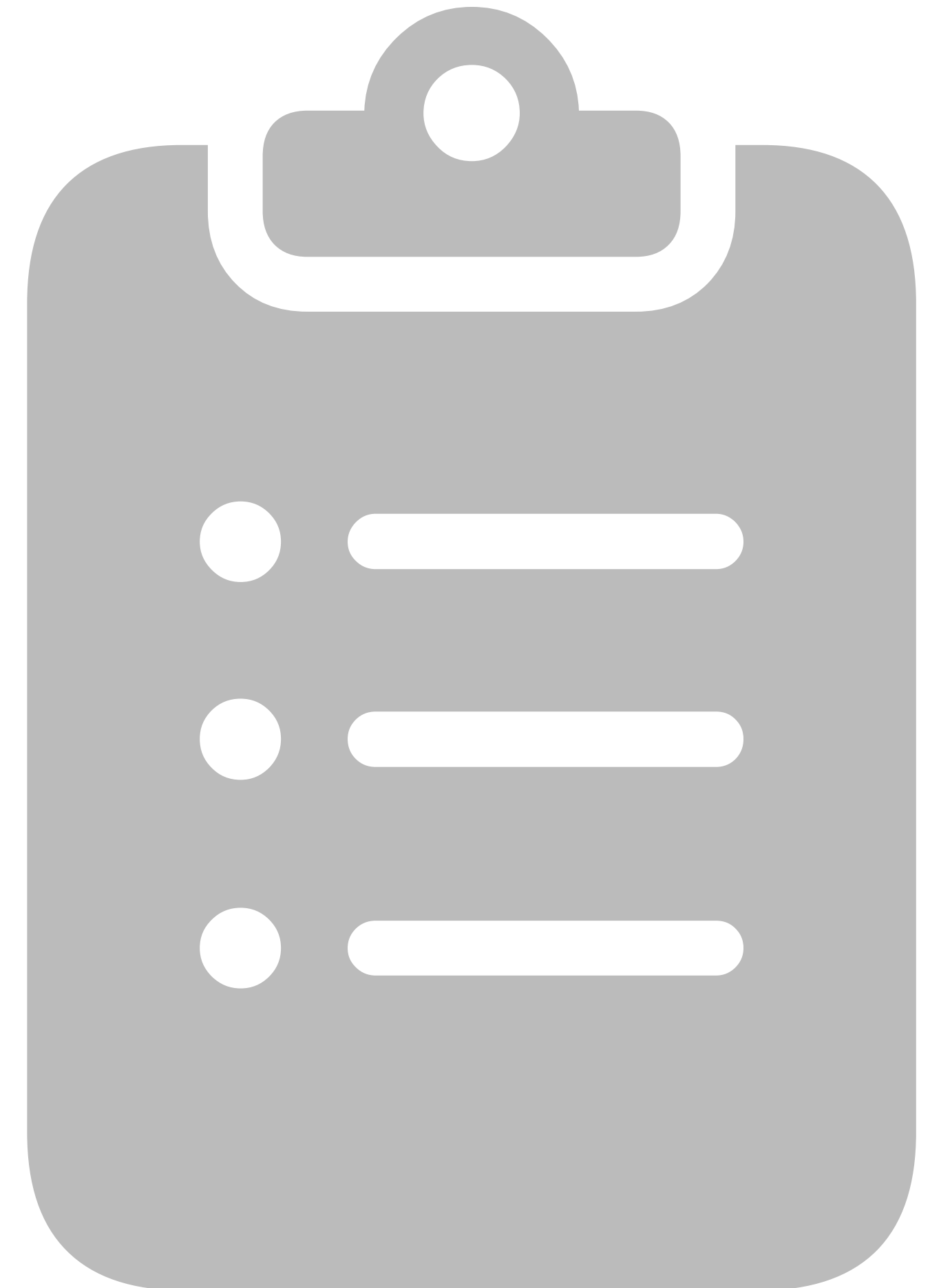
What do we get?

- Drive adoption of new software
- Covertly encourage recommended practices
- Security



UTI - Kind of like file types

- Uniform Type Identifiers
- Not *exactly* file types
- I'm going to conflate file *type* with file *extension*
- Includes URL schemes



Uniform Type Identifier

- Reverse DNS structure
 - com.adobe.pdf
- Hidden from users for simplicity
- One UTI can handle multiple file extensions:
 - "public.jpeg" references
 - **.jpg** and **.jpeg**



Uniform Type Identifier

Examples

- `com.adobe.pdf`
- `public.jpeg`
- `org.openxmlformats.wordprocessingml.document (docx)`
- `com.apple.iwork.numbers.sffnumbers`
- `public.tar-archive`
- `com.apple.mobileconfig`

URL Schemes

- Think services
 - mailto
 - http
 - ssh
 - tel



GET UTILUTI

scriptingosx | Armin Briegel



[HTTPS://GITHUB.COM/SCRIPTINGOSX/UTILUTI](https://github.com/scriptingosx/utiluti)

Getting Started

- Q2 2026 - Version 1.5
- Test against supported OSes
- Start in Terminal
- Move into scripts



Getting Started

- Study the Github repo
- ReadMe.md contains code examples



Getting Started

- Call Utiluti normally in Terminal
- Read the man page

`man utiluti`

- Read the help argument `-h`

`utiluti -h`

`utiluti type -h`

Poke it with a stick



Read/Get

UTI / File Extension

- What's your first project?
- Find your UTIs from extensions
- **utiluti get-uti \$extension**
`utiluti get-uti pdf`
`com.adobe.pdf`

UTI / File Type



Read/Get

UTI / File Extension

- Need the UTI for a *specific* file?
- **utiluti file get-uti /path/to/\$file**

UTI / File Type



Read/Get

UTI / File Extension

- Get an overview on a file type extension
- **utiluti type info -e \$extension**
`utiluti type info -e pdf`

UTI / File Type



```
rossmatsuda@Mac ~ % utiluti type info -e pdf
```

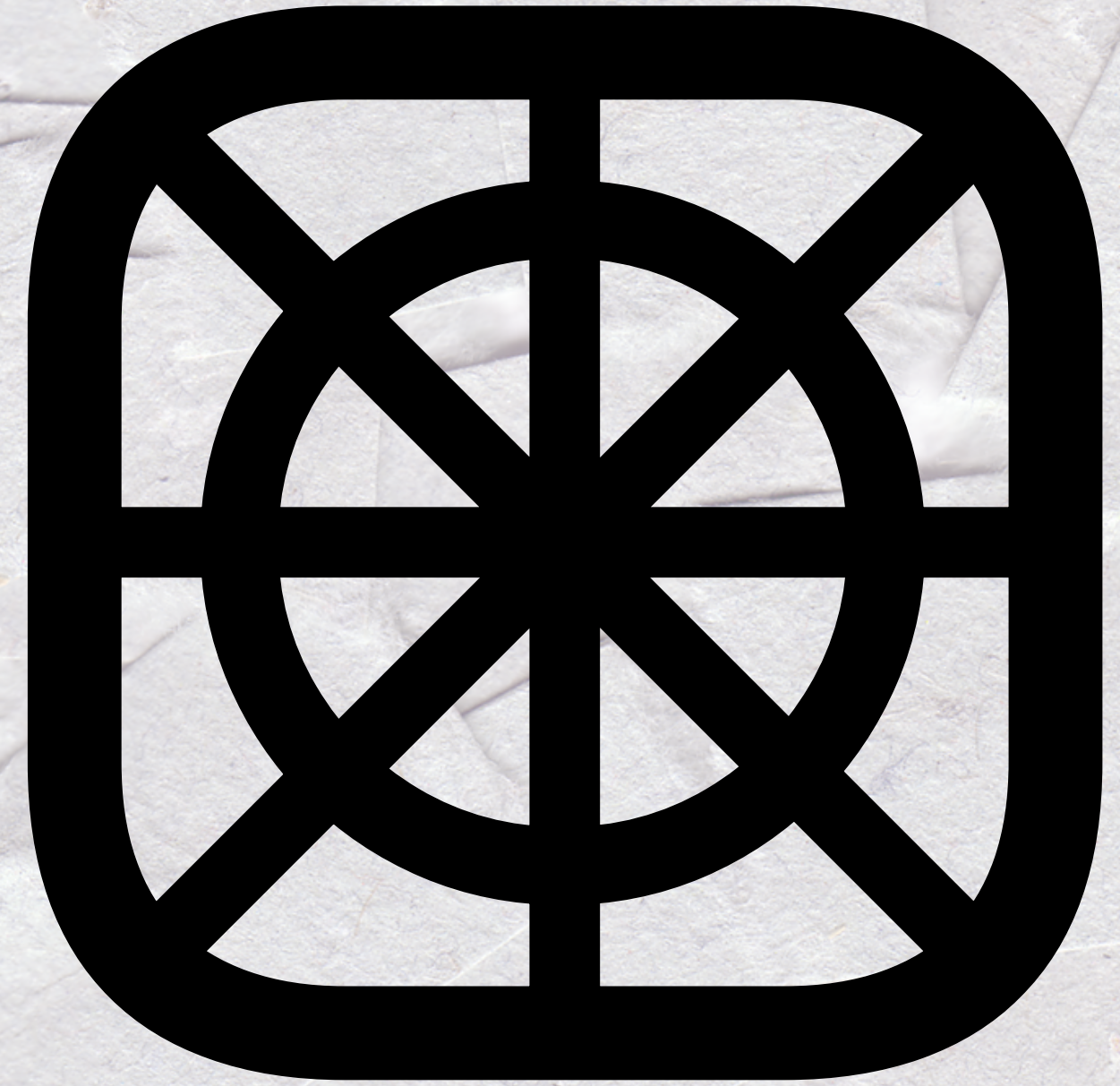
```
uniform type identifier: com.adobe.pdf  
public.mime-type: ["application/pdf"]  
com.apple.nspboard-type: ["Apple PDF pasteboard type"]  
com.apple.ostype: ["PDF "]  
public.filename-extension: ["pdf"]  
default app: com.apple.Preview (/System/Applications/Preview.app)
```

Read/Get

Application Path / Bundle ID

- What app is the default for a UTI?
- **utiluti type get \$UTI**

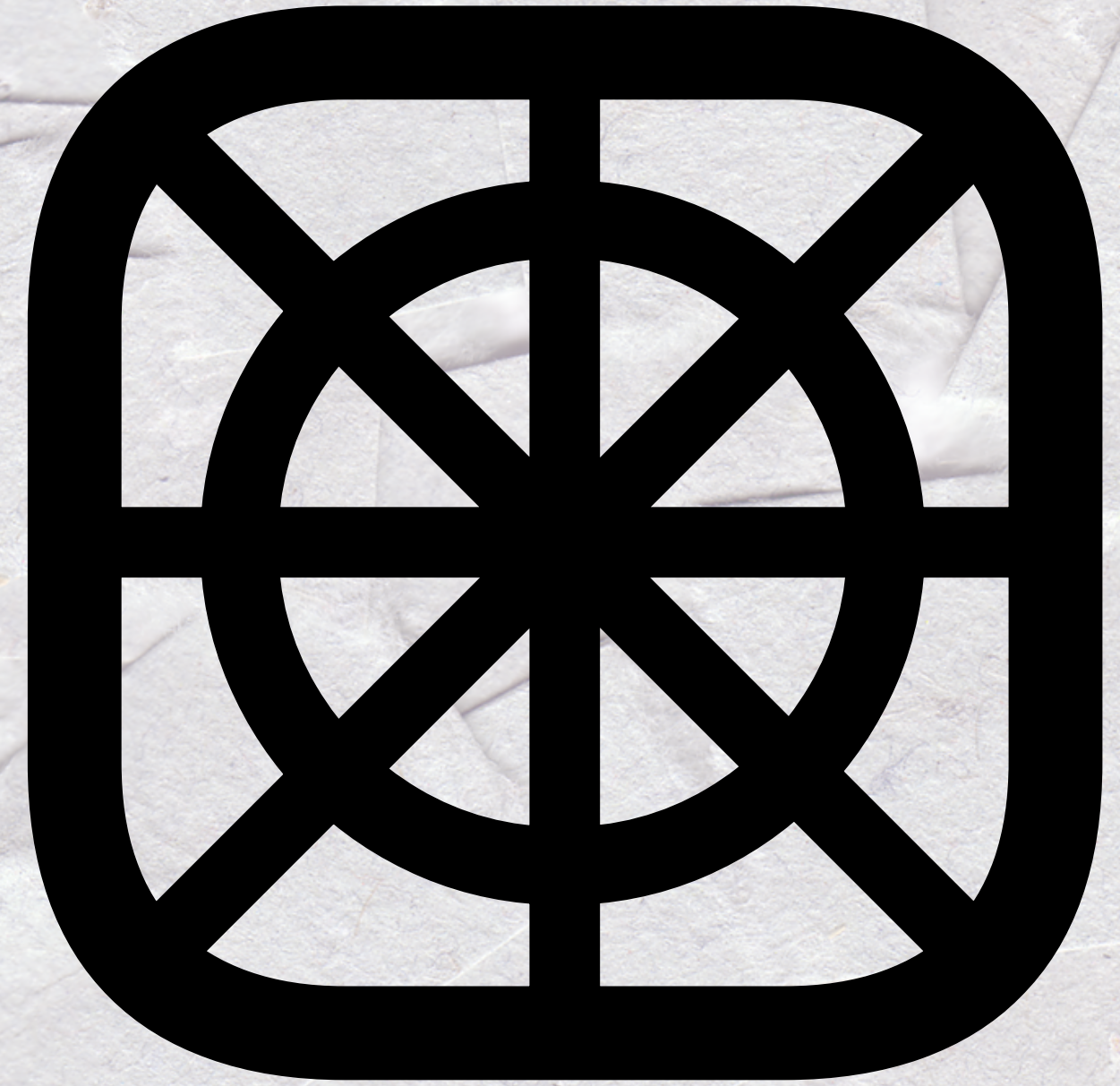
```
utiluti type get com.adobe.pdf  
/System/Applications/Preview.app
```



Read/Get

Application Path / Bundle ID

- What app is the default for an extension?
- **utiluti type get -e \$extension**
`utiluti type get -e pdf`
`/System/Applications/Preview.app`

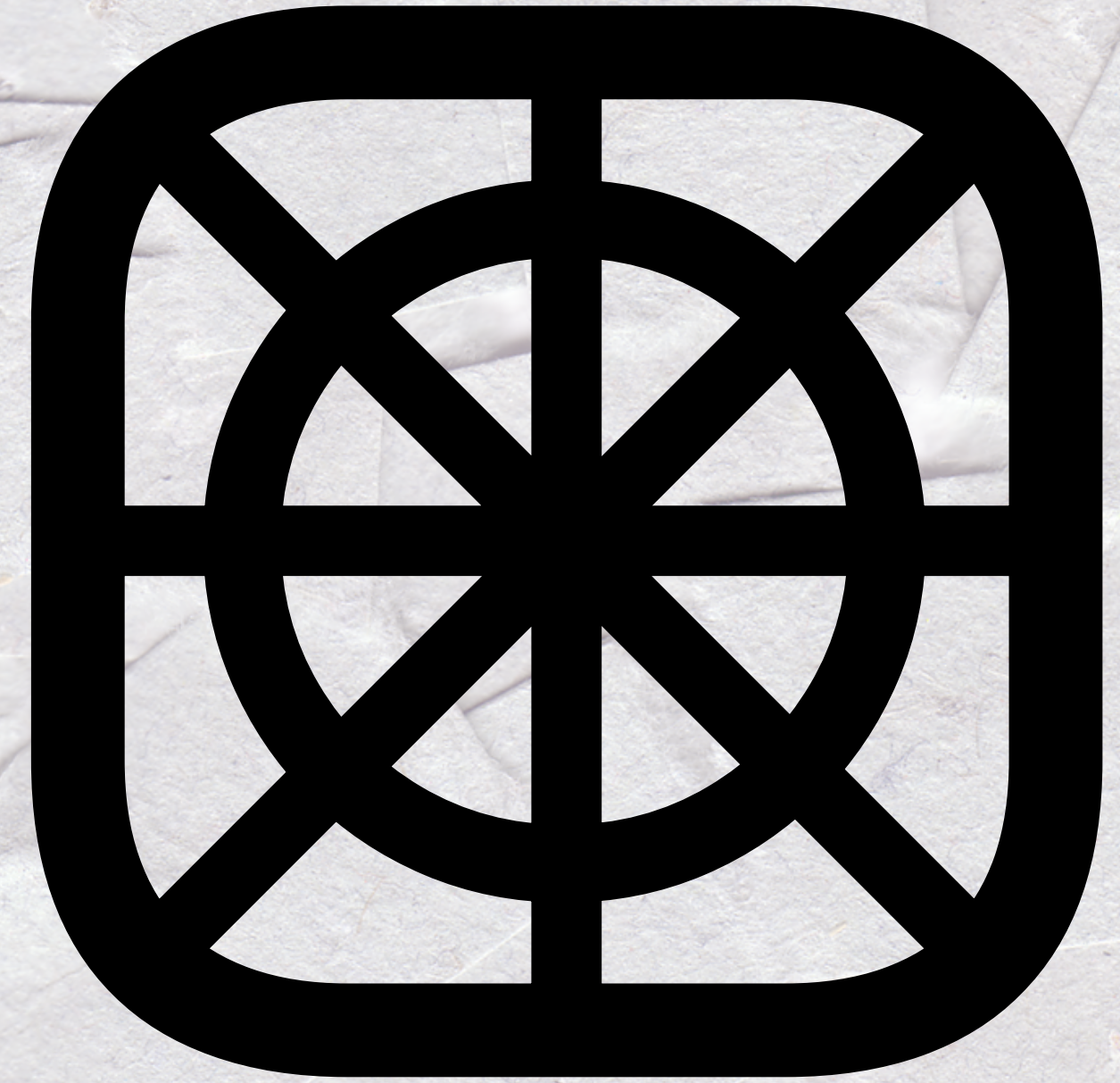


Read/Get

Application Path / Bundle ID

- What apps are registered with a UTI?
- **utiluti type list \$UTI**

```
utiluti type list com.adobe.pdf  
/System/Applications/Preview.app  
/Applications/Microsoft Word.app
```



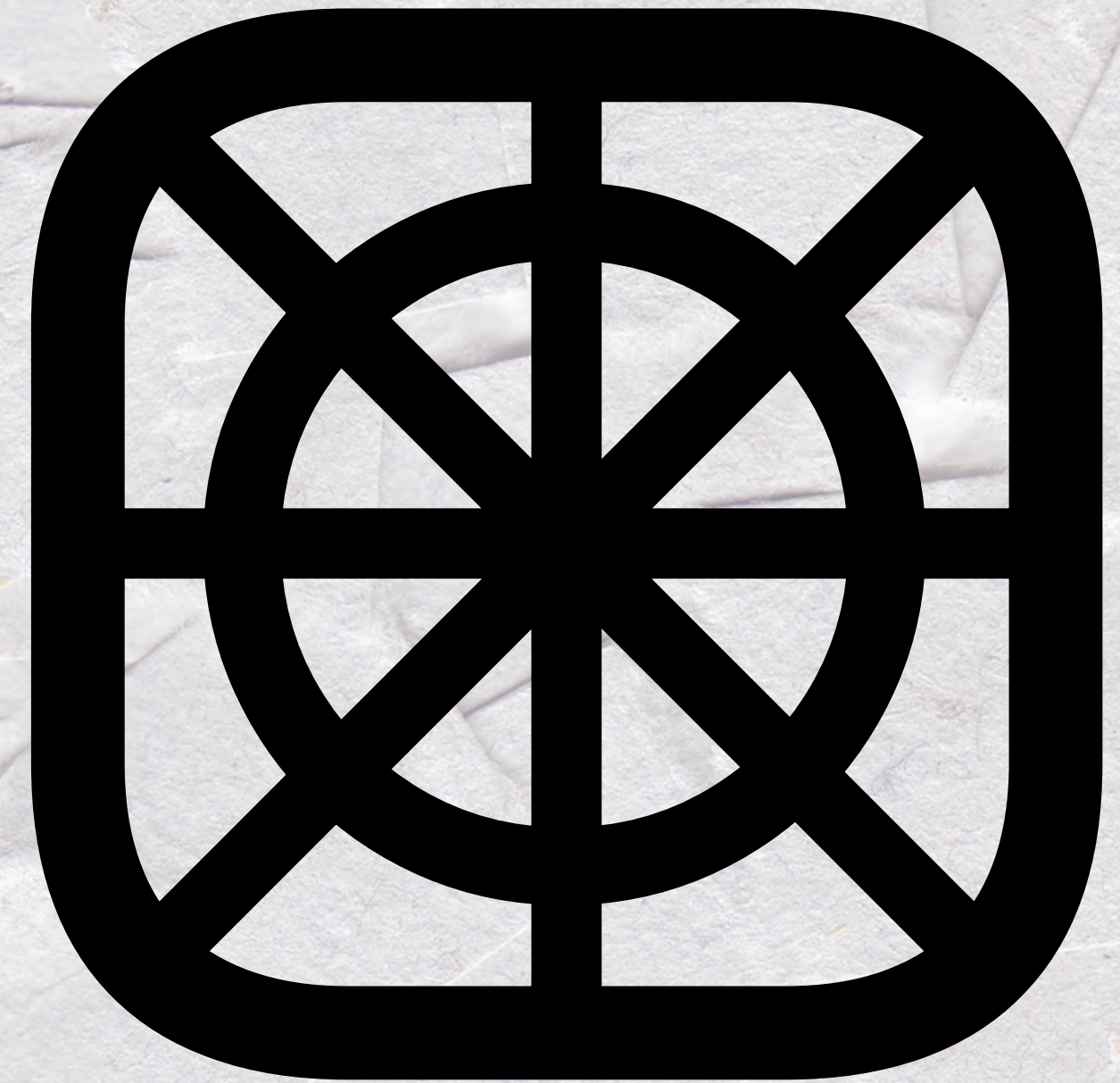
Read/Get

Application Path / Bundle ID

- What app *bundle IDs* are registered with a UTI?
- **utiluti type list \$UTI --bundle-id**

```
utiluti type org.openxmlformats.wordprocessingml.document  
--bundle-id
```

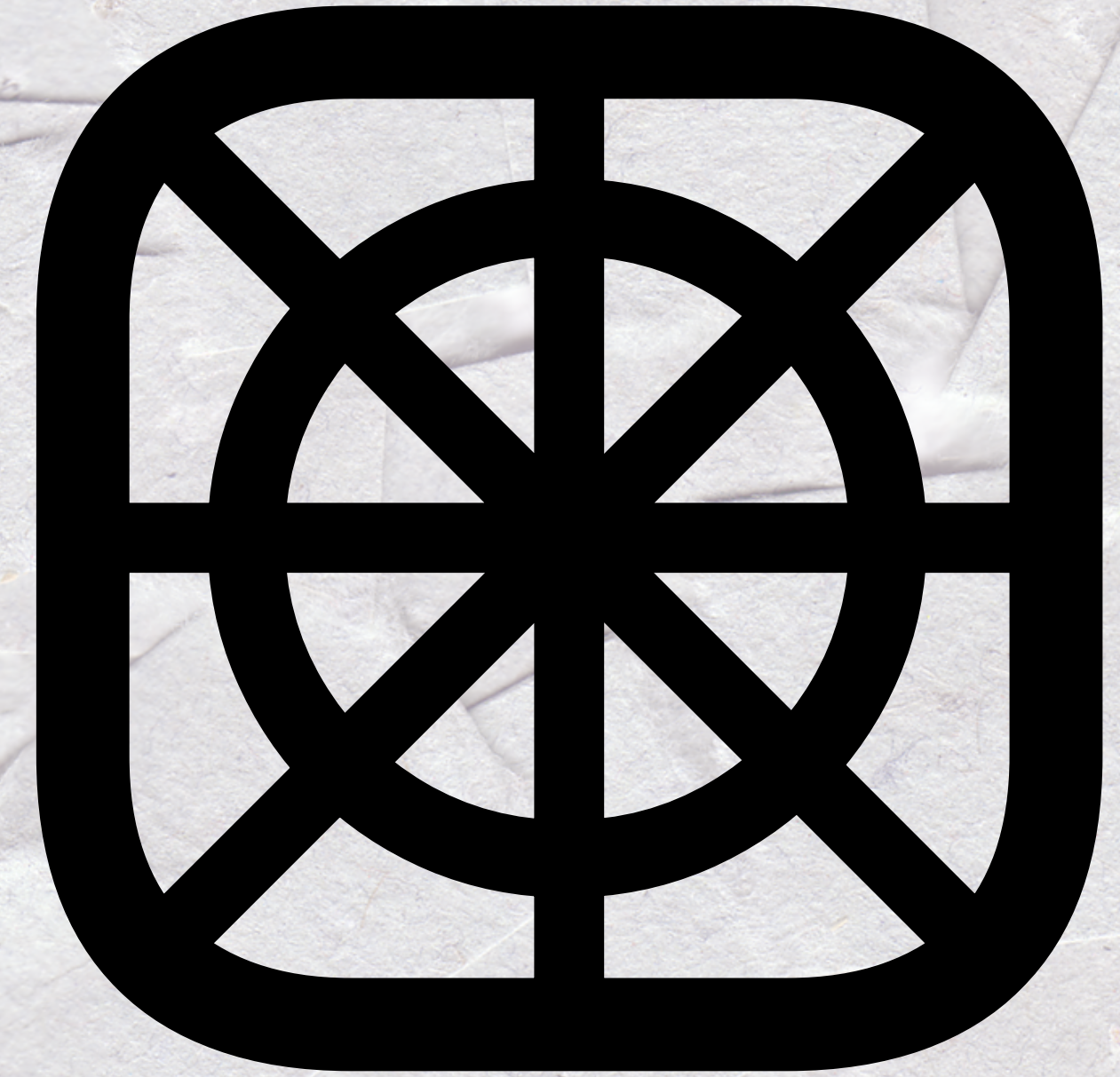
```
com.microsoft.Word
```



Read/Get

Application Path / Bundle ID

- Get an app's bundle ID
- **utiluti app id \$path/to/app.app**
`utiluti app id “/Applications/Microsoft Word.app”`
`com.microsoft.Word`

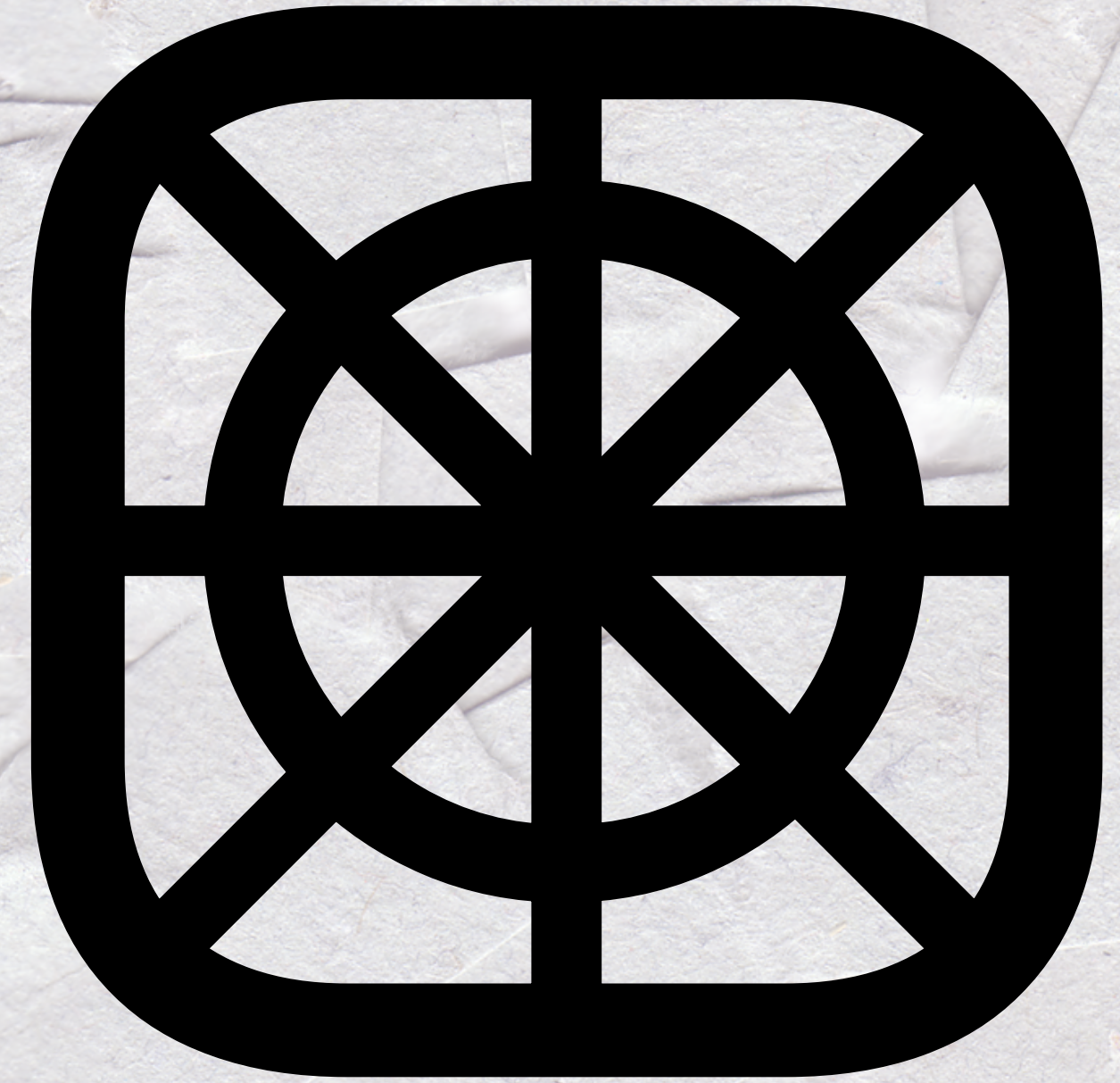


Read/Get

Application Path / Bundle ID

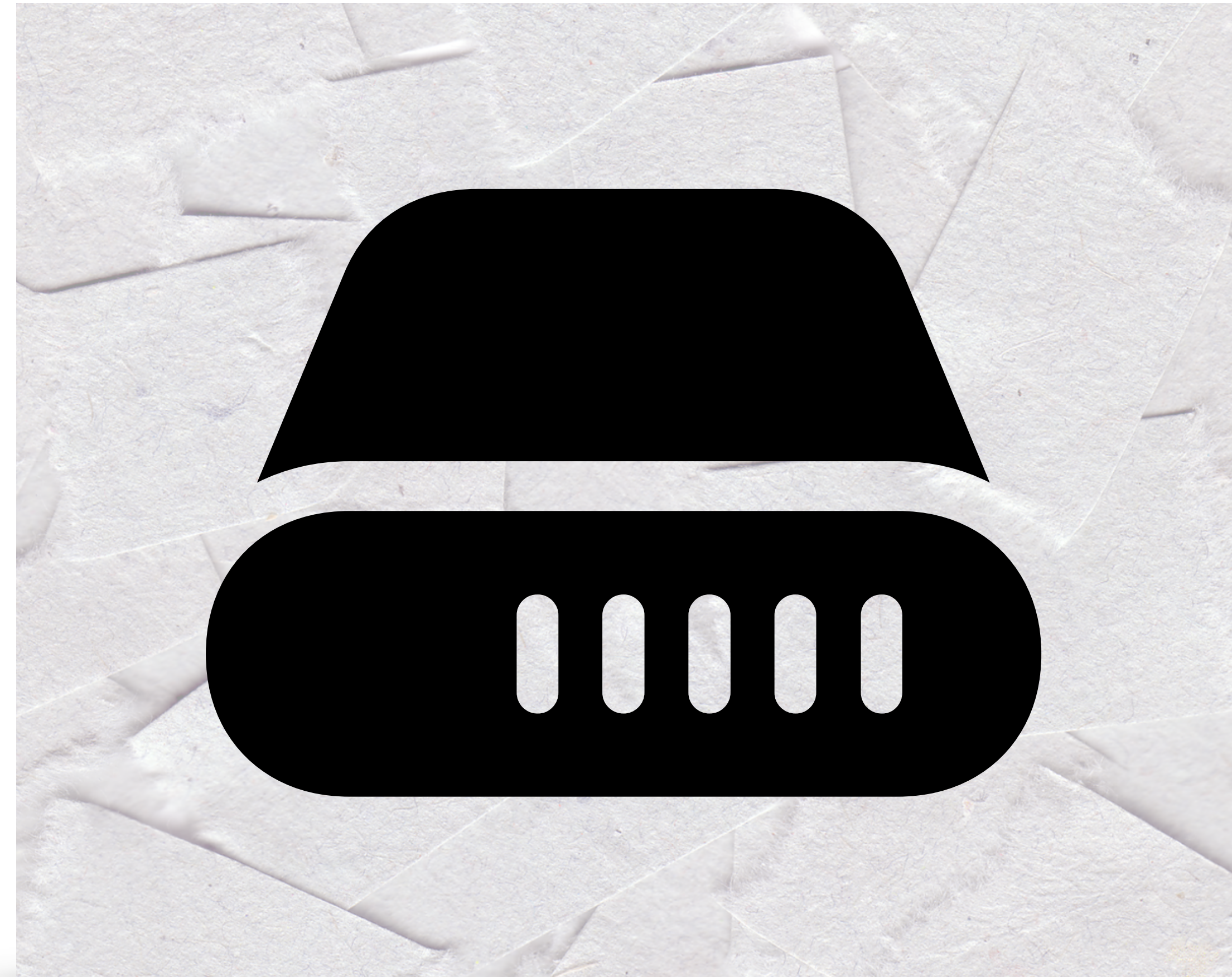
- If you have a Bundle ID, but need to know the app
- **utiluti app for-id \$bundleID**

`utiluti app for-id com.microsoft.Word`
`/Applications/Microsoft Word.app`



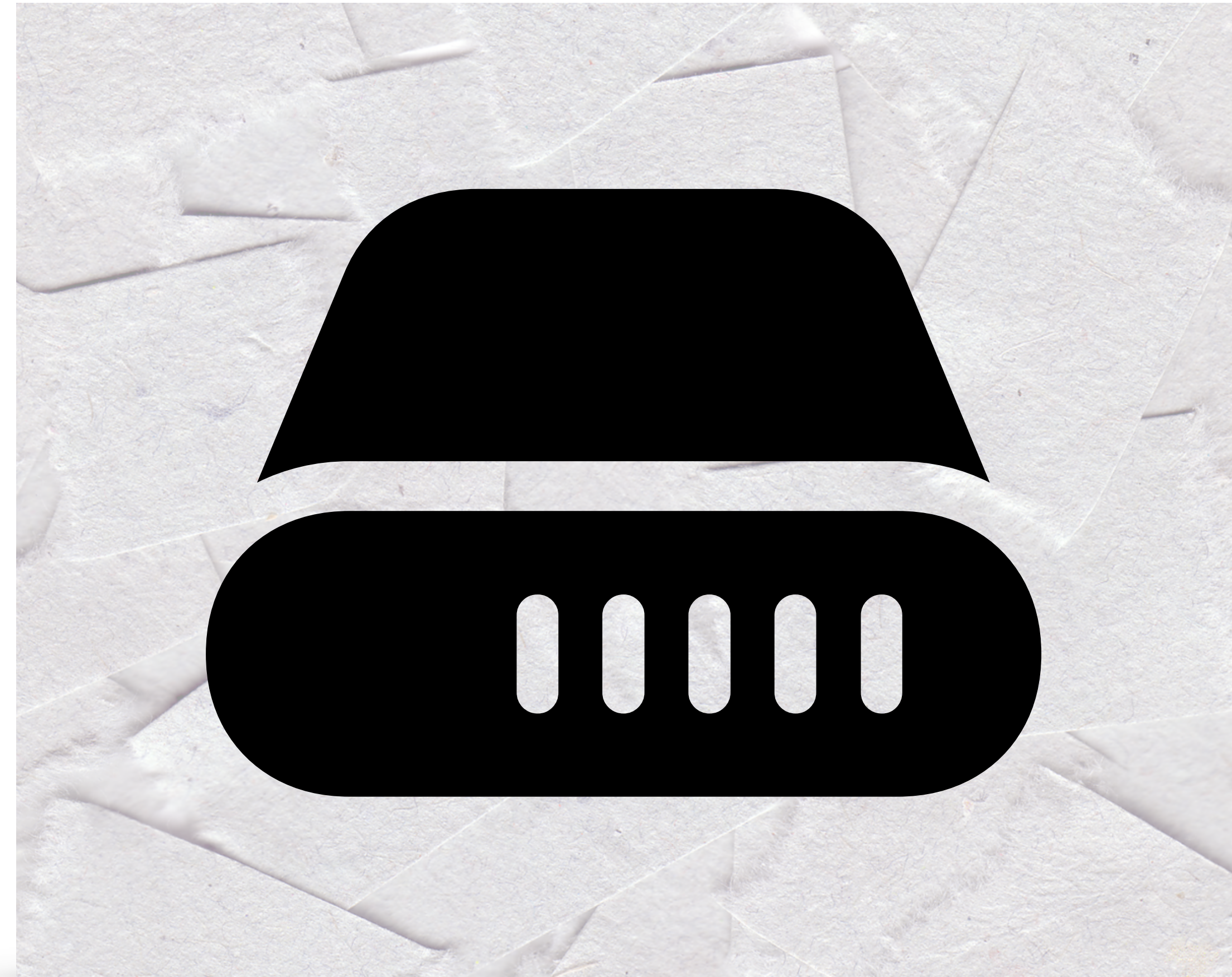
Making Changes

- Keep it simple
- Connect UTIs to app bundle IDs
- Use scripts, XML files, or configuration profiles to address multiple UTIs at once



Making Changes

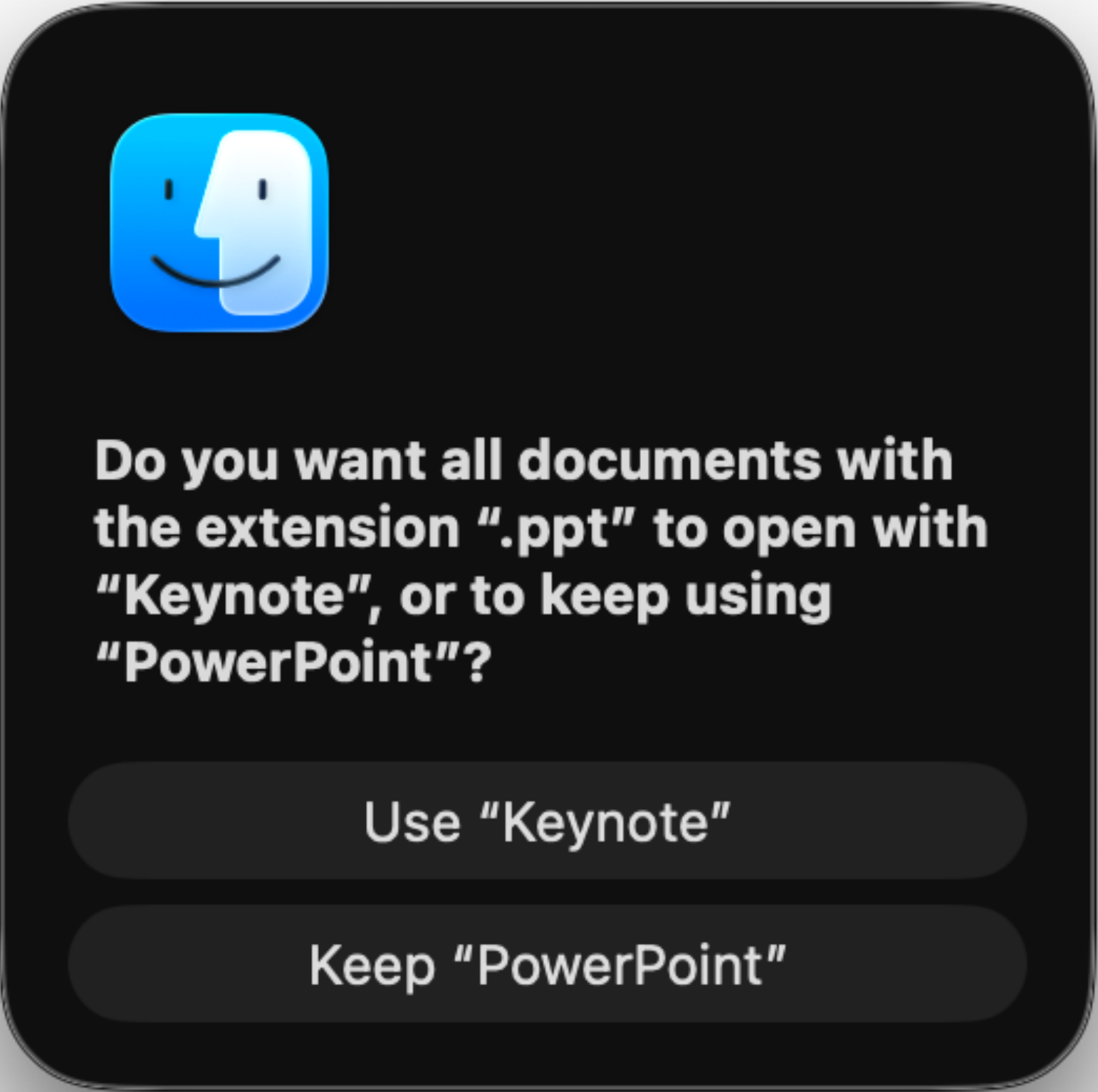
- More and more UTIs/URLs cannot be changed silently. Test!
- Focus on user-driven remediation with UI feedback.



Making Changes



Making Changes



Making Changes



Do you want all documents with the extension ".ppt" to open with "Keynote", or to keep using "PowerPoint"?

Use "Keynote"

Keep "PowerPoint"



Do you want all documents with the extension ".doc" to open with "Pages", or to keep using "Word"?

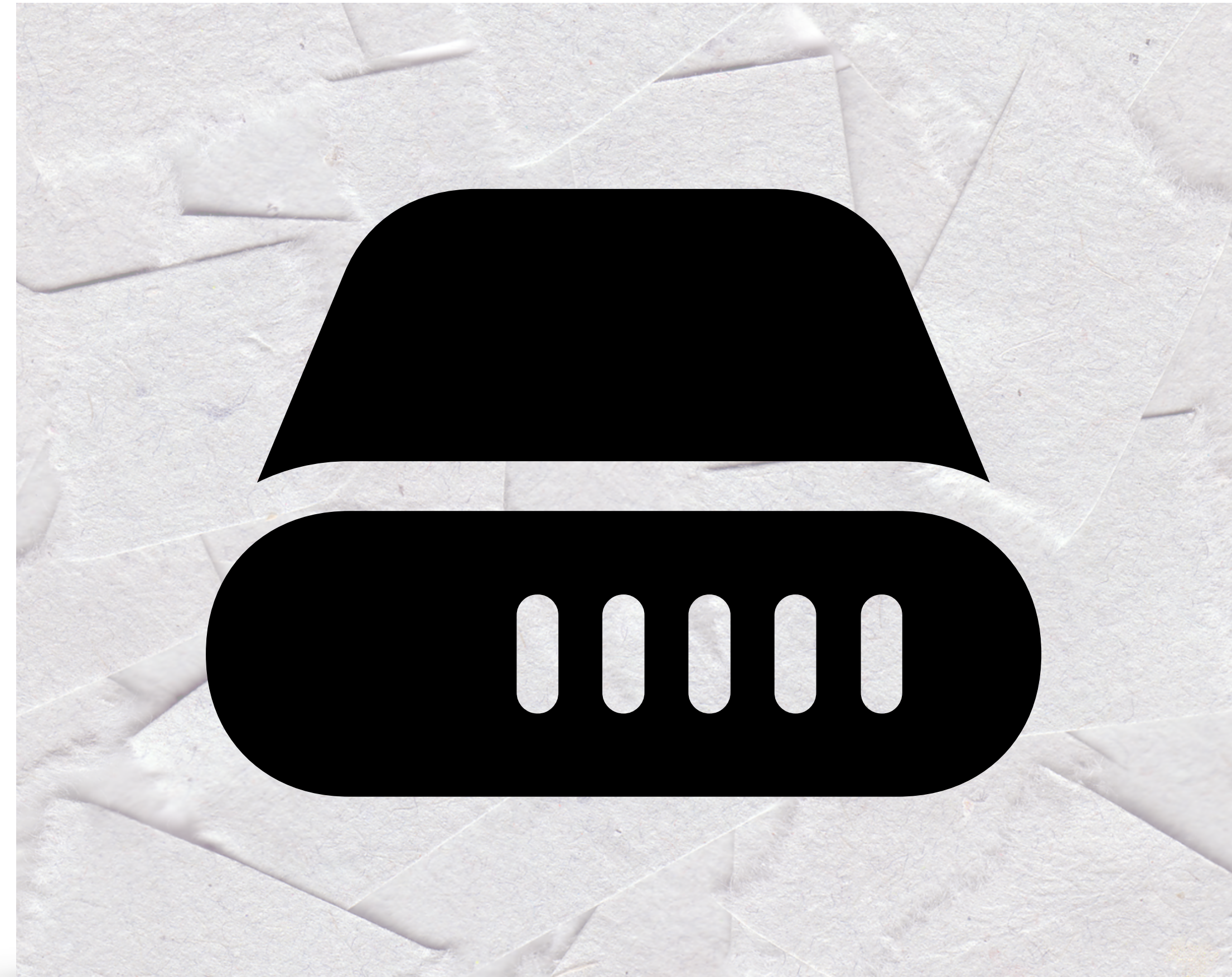
Keep "Word"

Use "Pages"

Making Changes

Setting defaults

- Set UTI to an app bundle ID
- **utiluti type set \$uti \$bundle-id**
`utiluti type set`



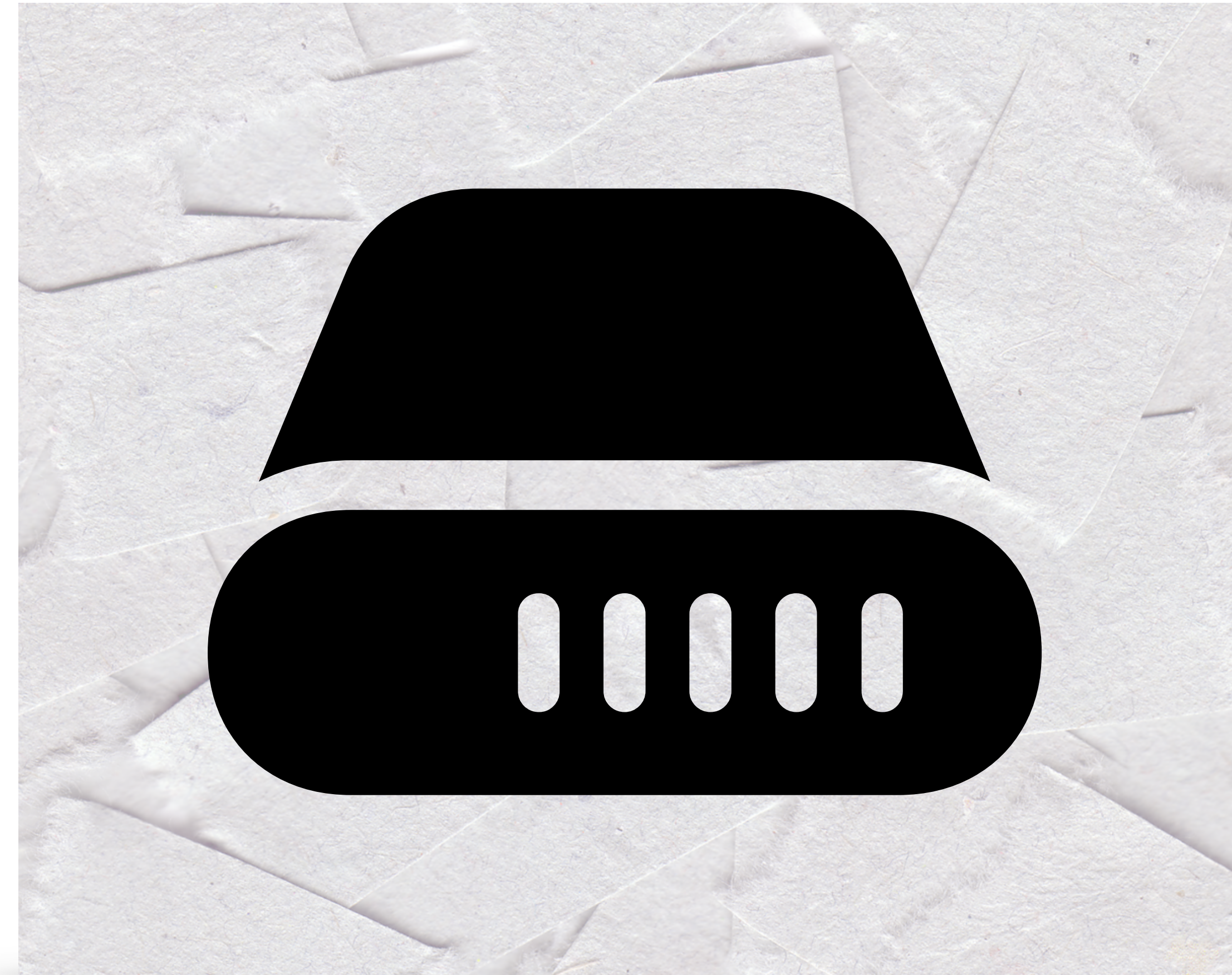
```
utiluti type set $uti $bundle-id
```

```
utiluti type set public.plain-text com.barebones.bbedit  
set com.barebones.bbedit for public.plain-text
```

Making Changes

Setting defaults

- Set extension to an app bundle ID
- **utiluti type set -e \$extension \$bundle-id**
utiluti type set -e
- Note - always need bundle ID, can't provide application path



```
utiluti type set -e $extension $bundle-id
```

```
utiluti type set -e txt com.barebones.bbedit  
set com.barebones.bbedit for public.plain-text
```

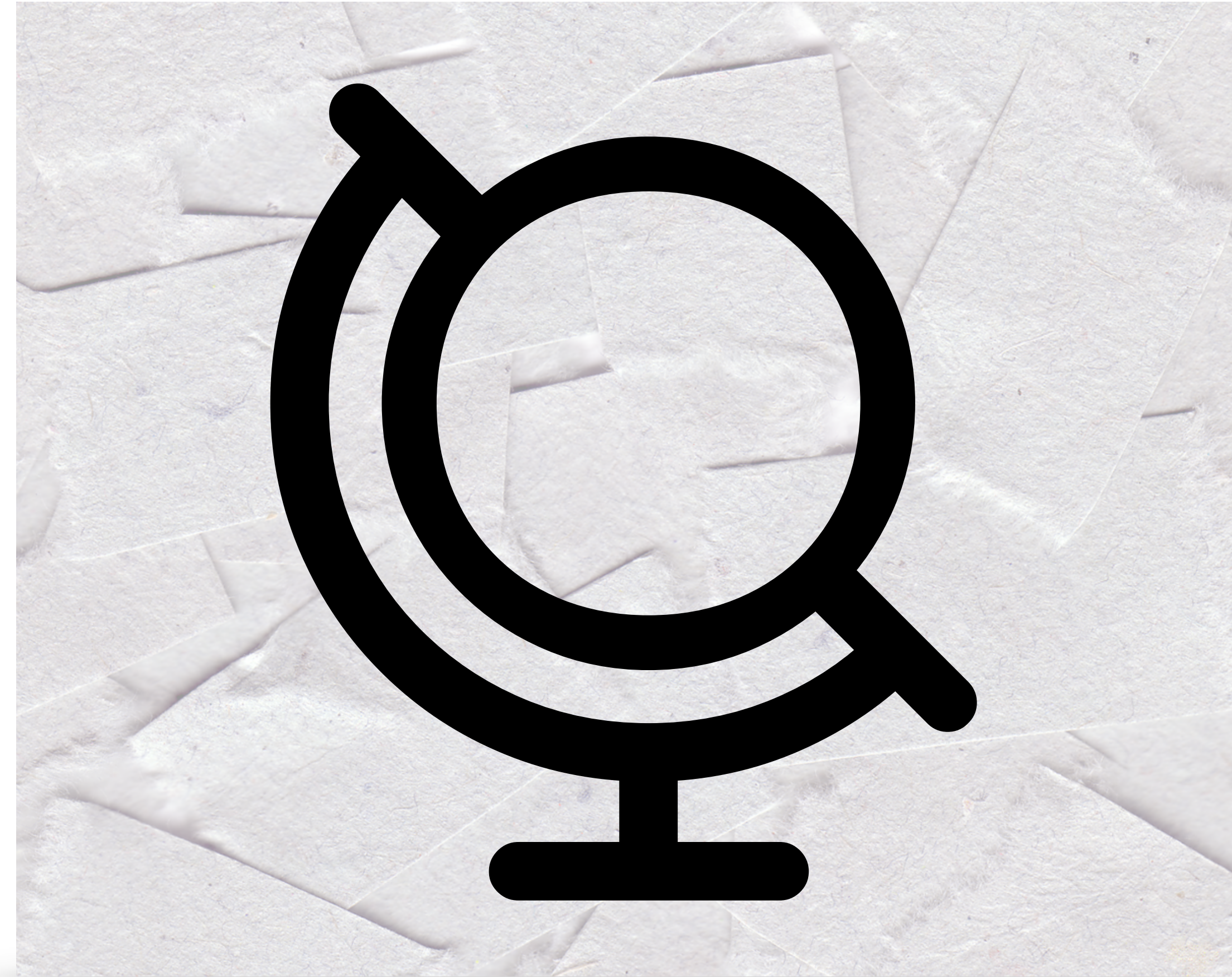
URL Schemes

Get apps for URL Schemes

- Get the app path for a URL scheme
- **utiluti url \$urlScheme**

`utiluti url mailto`

`/System/Applications/Mail.app`

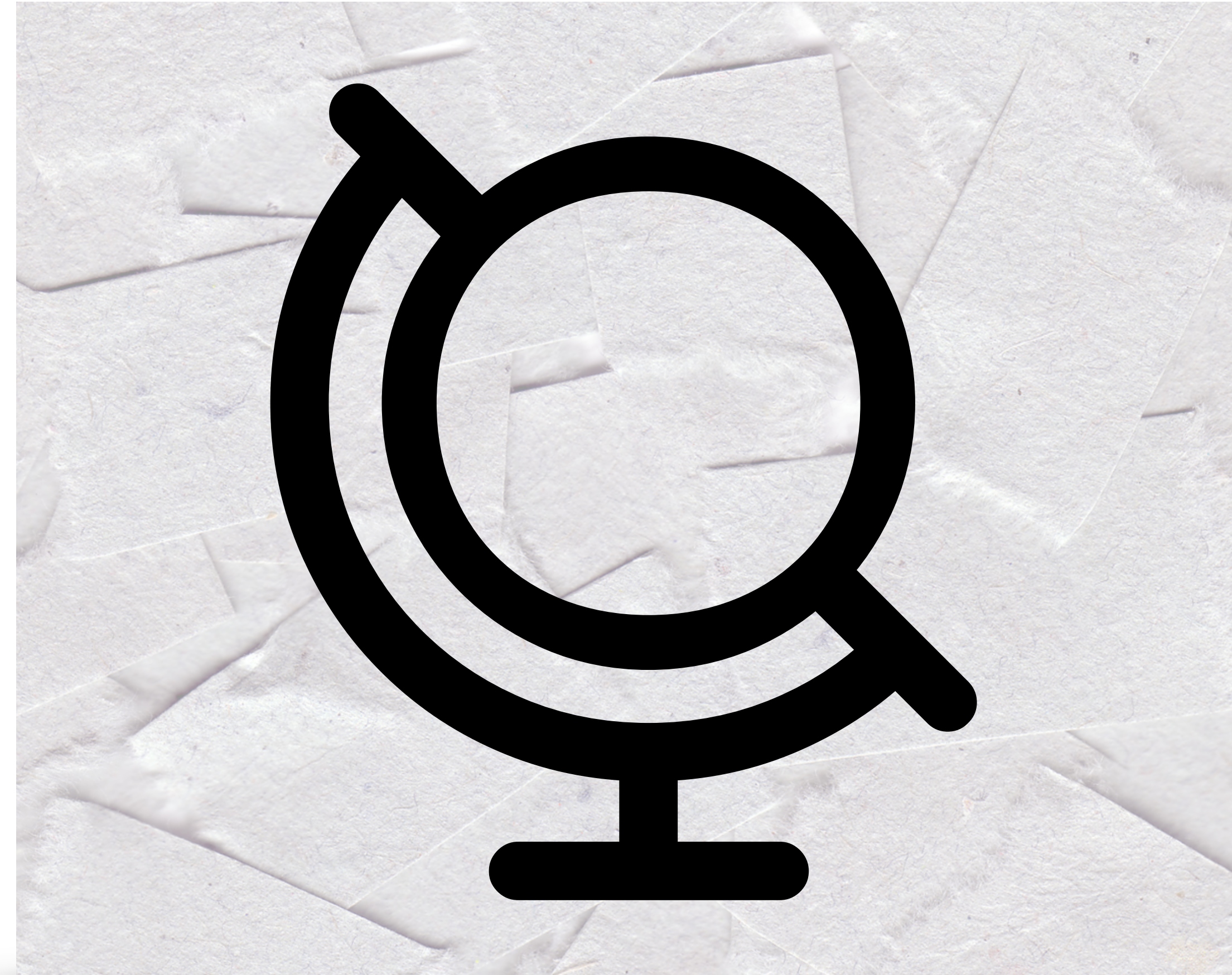


URL Schemes

Get apps for URL Schemes

- Get the app bundle ID for a URL scheme
- **utiluti url \$urlScheme --bundle-id**

```
utiluti url mailto --bundle-id  
com.apple.mail
```



URL Schemes

Get apps for URL Schemes

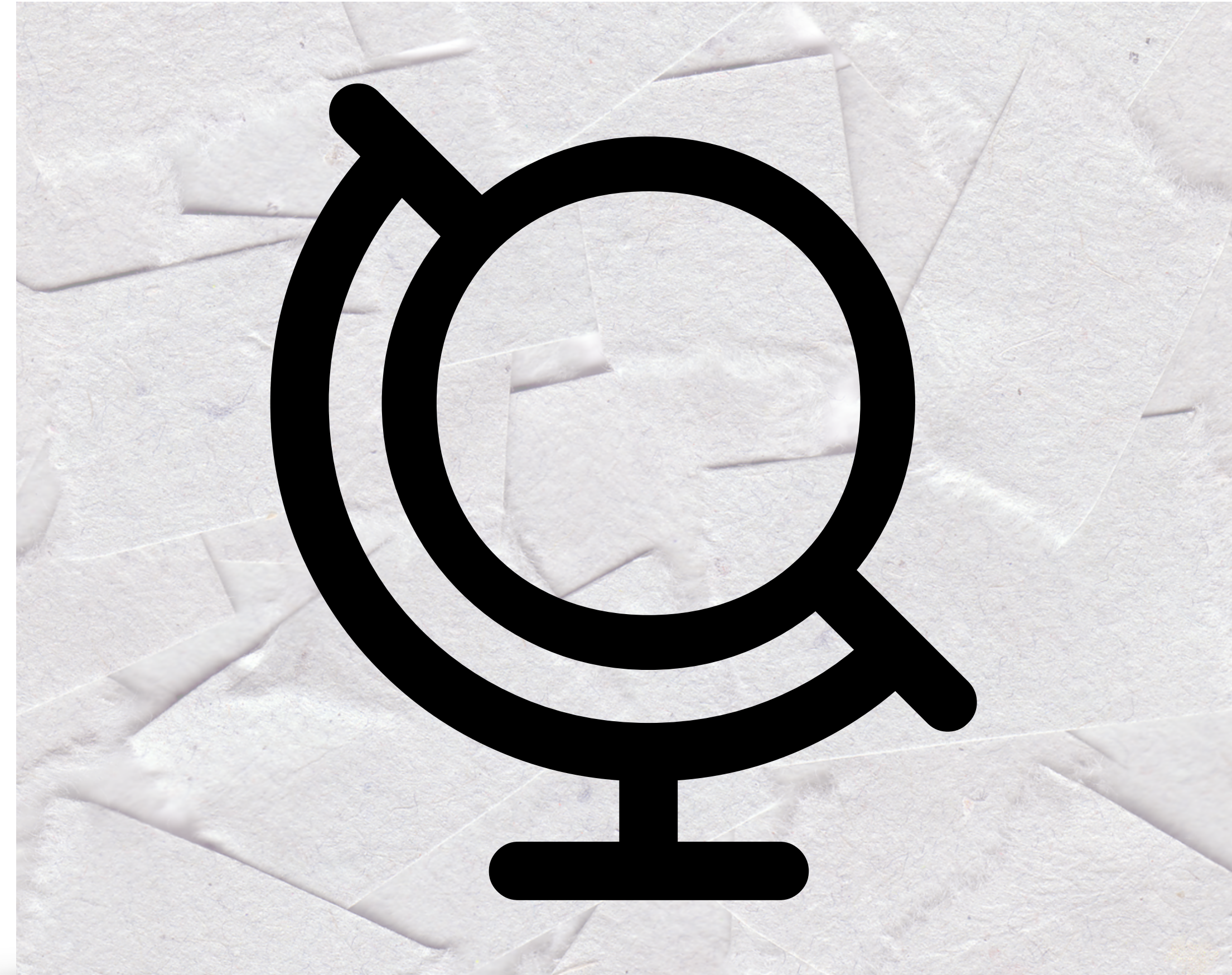
- List *all* registered apps for URL scheme
- **utiluti url list \$urlScheme**

```
utiluti url list mailto
```

```
/System/Applications/Mail.app
```

```
/Applications/zoom.us.app
```

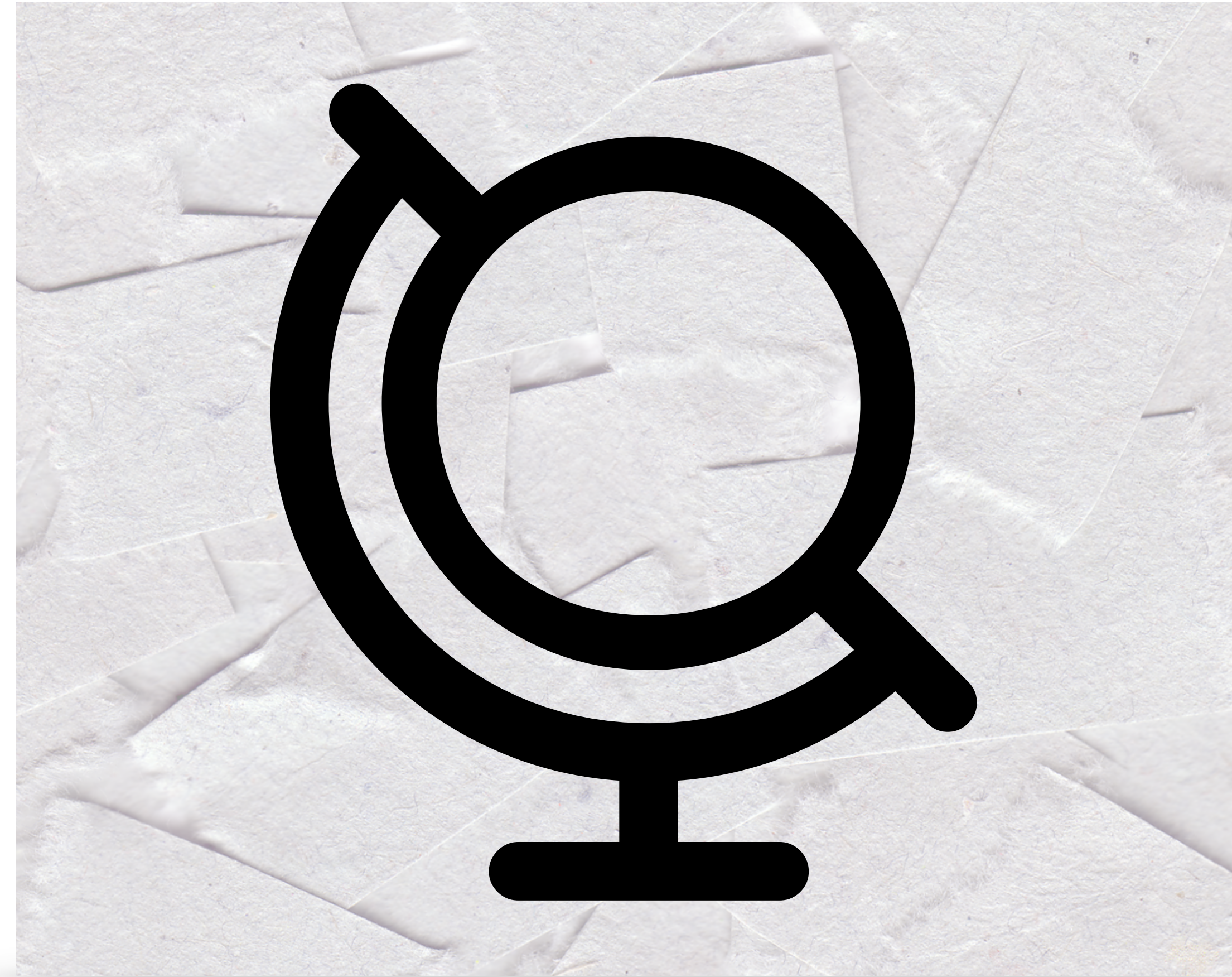
```
/Applications/Microsoft Outlook.app
```



URL Schemes

Get apps for URL Schemes

- List all apps registered with a URL
via bundle ID
- **utiluti url list \$urlScheme --bundle-id**
`utiluti url list mailto --bundle-id`
`com.apple.Mail, etc.`

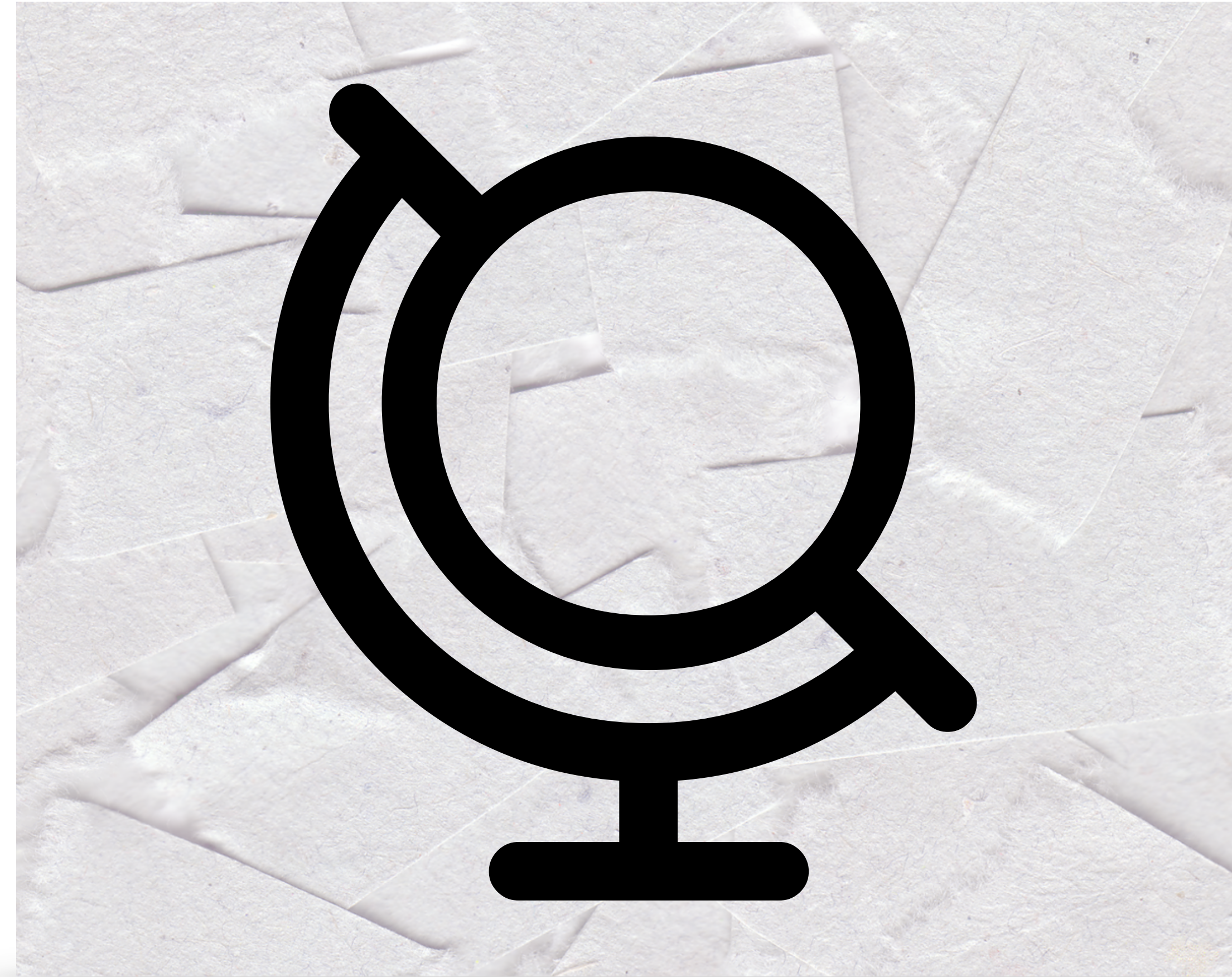


URL Schemes

Set

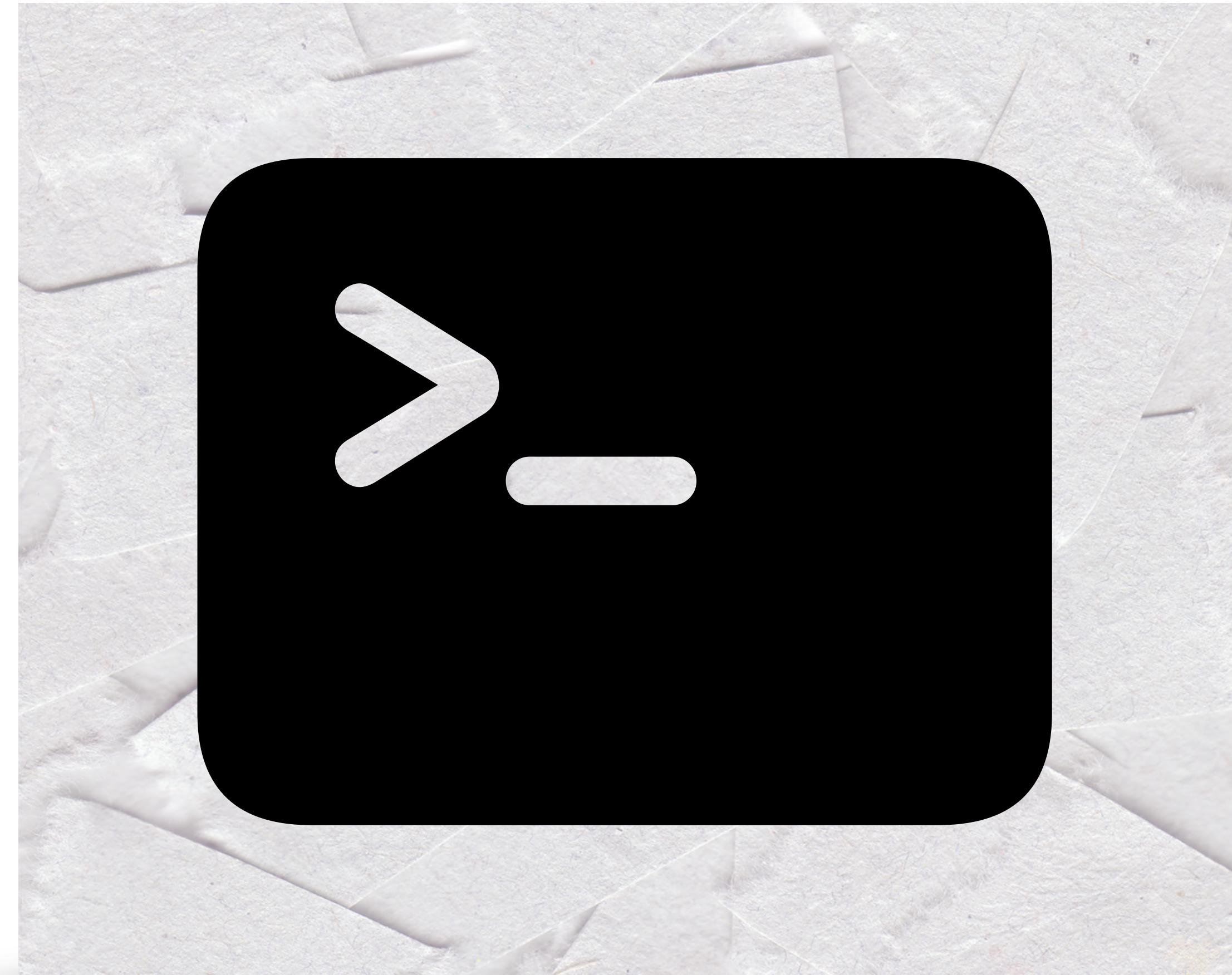
- Set default app bundle ID for a URL
- **utiluti url set \$urlScheme \$bundleID**

```
utiluti url set mailto com.apple.Mail  
set com.apple.Mail for mailto
```



Scripting

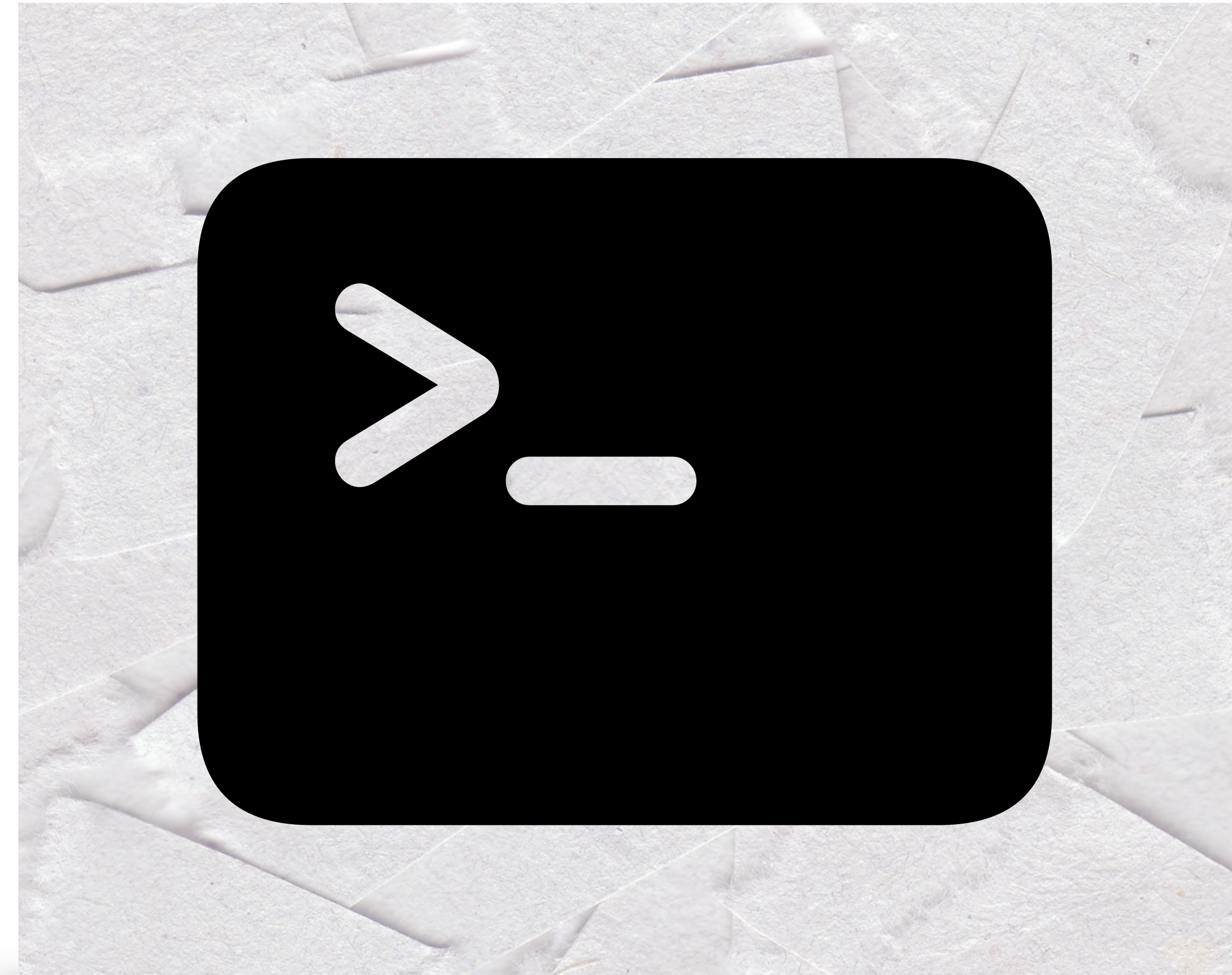
- `/usr/local/bin/utiluti`
- Make this a variable you can call for cleaner code
- `$utilutiPath`



Scripting

- All scripted commands must run in user space

Run as current user



```
currentUser=$(/usr/bin/stat -f%Su /dev/console)
```

```
rossmatsuda@Mac ~ % /usr/bin/stat -f%Su /dev/console
```

rossmatsuda

```
uid=$(id -u "$currentUser")
```

```
rossmatsuda@Mac ~ % id -u rossmatsuda
```

502

Run as current user:

```
sudo -u $currentUser $command
```

Run as current UID:

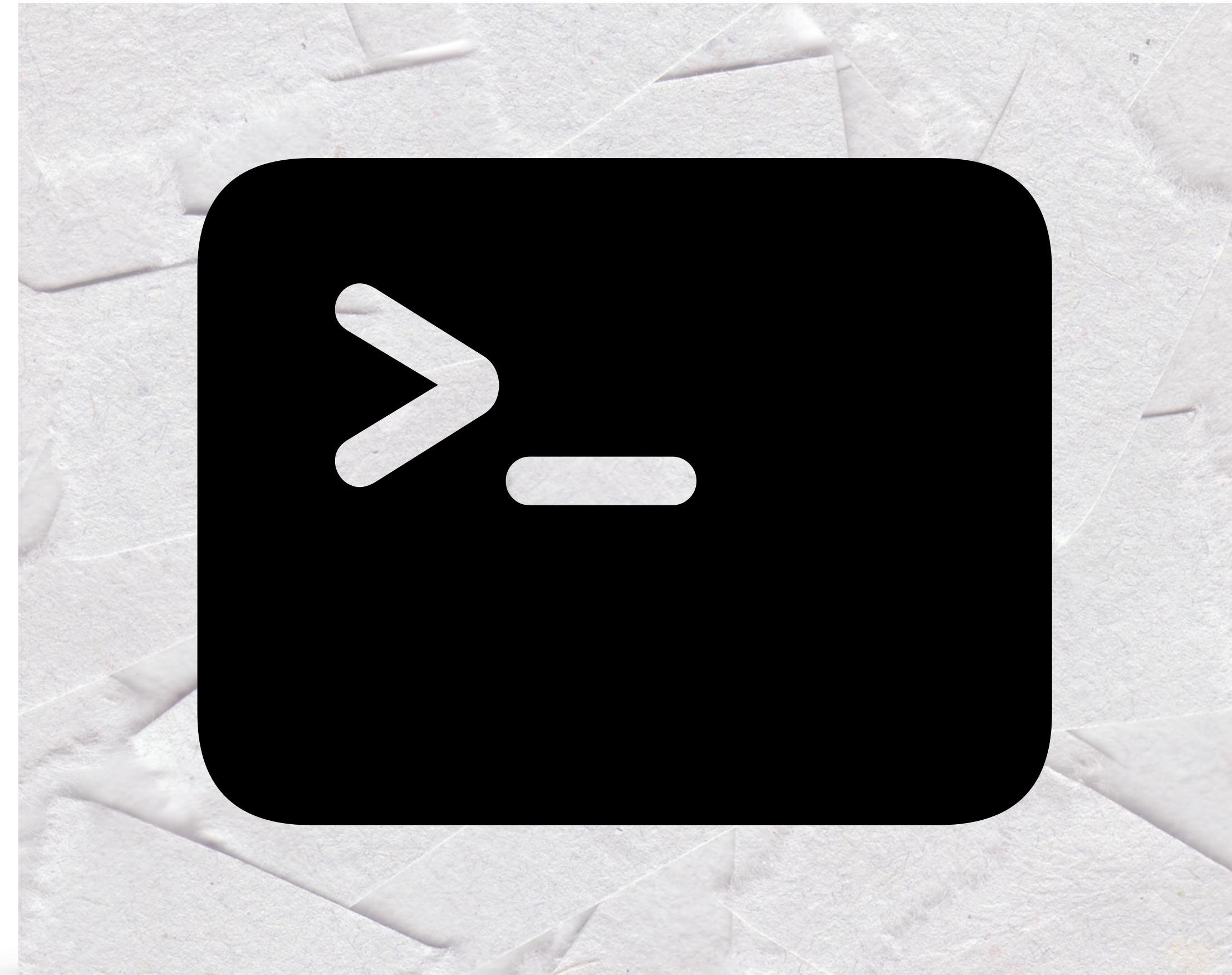
```
/bin/launchctl asuser "$uid" $command
```

Run as ***both***:

```
launchctl asuser "$uid" sudo -u "$currentUser" $command
```

Invocation block

- Make it easy to invoke Utiluti as the current user
- Make this reusable - "white label"



Code block for invocation:

```
utilutiPath="/usr/local/bin/utiluti"
```

```
currentUser=$(/usr/bin/stat -f%Su /dev/console)
```

```
uid=$(id -u "$currentUser")
```

```
utilutiCall="launchctl asuser "$uid" sudo -u "$currentUser" $utiluti"
```

```
$utilutiCall get-uti zip  
public.zip-archive
```

Condition Script

- Current device status:
 - Target app installed?
 - Does current URL/UTI have an associated Bundle ID?
 - Does that bundle ID match the one you want?



Condition Script

- Example: Set default browser to Safari
- We'll write this so it will work with both UTIs or URLs



```
#!/bin/zsh
# utiluti - Set Default App - Condition
# Ross Matsuda | Ntiva | April, 2025
# Tool documentation: https://github.com/scriptingosx/utiluti
# Set to execute if exit code is 1 (error detected), exit code 0
means no action is required

# Condition logic:
# 1 - if target app is not installed yet, halt
# 2 - If UtilUTI is not installed yet, proceed so we can assess
# 3 - if both installed, check current handler. If handler
already set, halt
    # This must be able to iterate through an array and only run
if an app doesn't match
```

```
##### VARIABLES #####  
# Bundle ID of target app - cannot use path.  
handlerBundleID="com.apple.Safari"  
  
# Array of UTIs or URLs - logic handles either  
utiArray=(  
    http  
)  
##### /VARIABLES #####  
  
##### DO NOT EDIT BELOW THIS LINE #####
```

```
##### VARIABLES #####  
# Bundle ID of target app - cannot use path.  
handlerBundleID="com.apple.Numbers"  
  
# Array of UTIs or URLs - logic handles either  
utiArray=(  
com.microsoft.excel.xls  
org.openxmlformats.spreadsheetml.sheet  
org.openxmlformats.spreadsheetml.sheet.macroenabled  
org.openxmlformats.spreadsheetml.template  
org.openxmlformats.spreadsheetml.template.macroenabled  
)  
##### /VARIABLES #####  
##### DO NOT EDIT BELOW THIS LINE #####
```

```
# Is handler installed?
```

```
handlerInstalled="$(mdfind "kMDItemCFBundleIdentifier ==  
'$handlerBundleID'")"
```

```
if [ -z "$handlerInstalled" ]; then  
    echo "Handler not installed, do not run"  
    echo "Exiting"  
    exit 0  
else  
    echo "Handler detected - proceeding"  
fi
```

```
# Install UtilUTI if not present

utilutiPath="/usr/local/bin/utiluti"

if [ -e "$utilUTIBin" ]; then
    echo "UtilUTI already installed - proceeding"
else
    echo "UtilUTI not installed, run installer."
    exit 1
fi
```

```
# Get current user - utiutil must be run in userspace

currentUser=$(/usr/bin/stat -f%Su /dev/console)

if [ "$currentUser" = "root" ] || [ "$currentUser" = "" ]; then
    echo "At loginwindow, defer until next check-in"
    exit 0
fi
```

```
uid=$(id -u "$currentUser")
```

```
utilutiCall="launchctl asuser "$uid" sudo -u "$currentUser"  
$utilutiPath"
```

```
#Trigger UtilUTI assessment by iterating through array

for uti in "${utiArray[@]}" ; do
    echo "Assessing $uti"

# Branch syntax - UTI vs URL
urlCheck="http|ssh|mailto|tel"

    if [[ $uti =~ ^($urlCheck)$ ]]; then
        # URL detected
        verb="url"
    else
        # UTI detected
        verb="type"
    fi
```

```
# Syntax: utiluti $verb --bundle-id $UTI

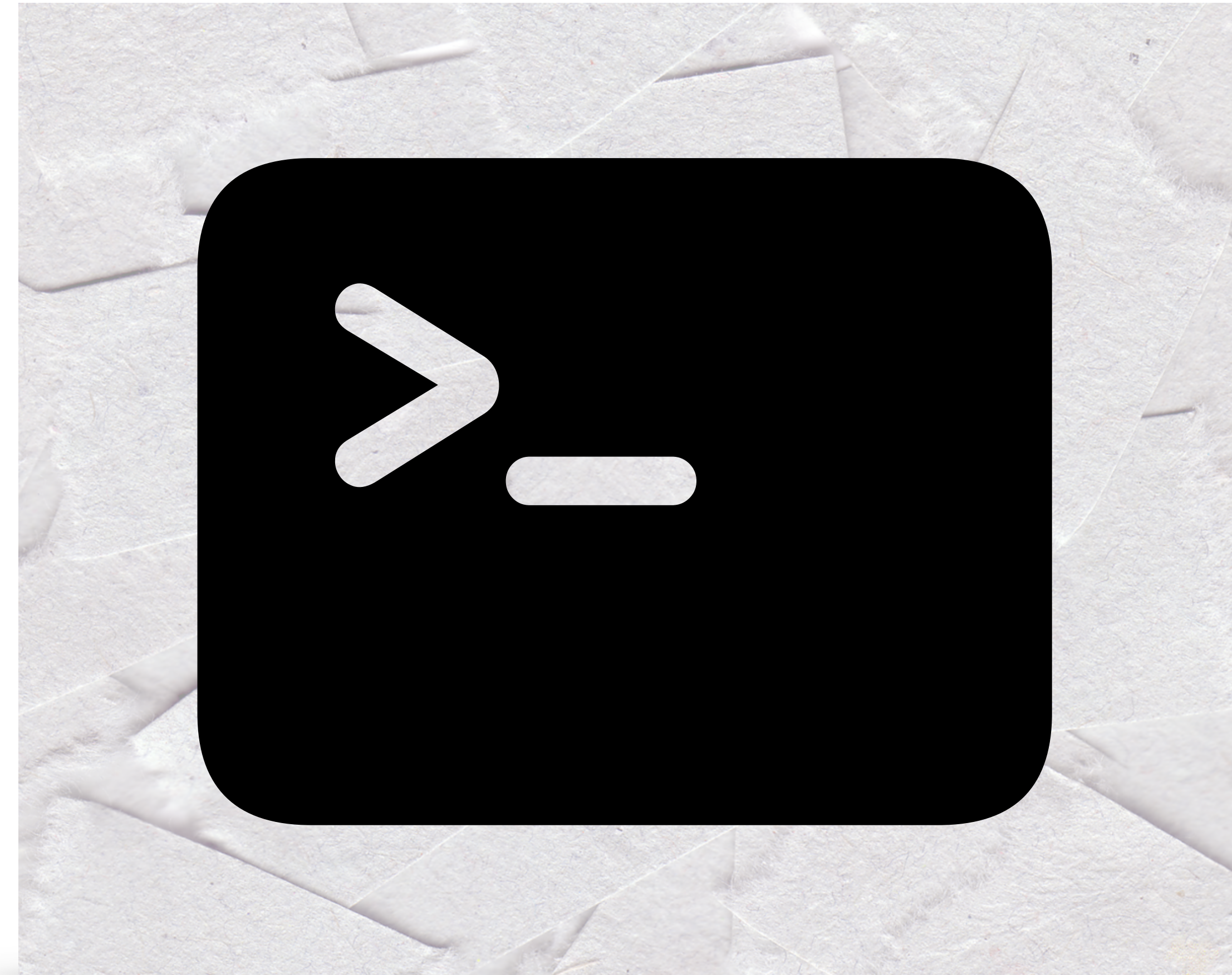
currentHandlers="$($utilutiCall $verb --bundle-id $uti)"
echo "Current handlers are $currentHandlers"

# 2 - check if desired handler is already the default app
if [[ "$currentHandlers" == *"$handlerBundleID"* ]]; then
    echo "$uti correctly assigned - continue testing"
else
    echo "$uti requires update, run script."
    exit 1
fi

done
echo "All tests passed, UTI already configured."
exit 0
```

Installation Script

- Install Utiluti if needed
- Sanity check that the target app is installed
- Set the UTI/URL for the current user



```
#!/bin/zsh
# utiluti - Set Default App (1 app)
# Ross Matsuda | Ntiva | April, 2025
# Tool documentation: https://github.com/scriptingosx/utiluti

# Note: Throws an error if an app with the target handler bundle
ID is not installed

# Note: For URLs, only http can be set, not https or public.html.
If you need to use either of those two, just set http
```

```
##### VARIABLES #####
softwarePath="/Library/Addigy/ansible/packages/Set Default app -
Web Browser to Safari (1.0.0)" # Copy from Addigy
utilUTIpkg="utiluti-1.5.pkg"

# Bundle ID of target app - cannot use path.
handlerBundleID="com.apple.Safari"

# Array of UTIs or URLs - logic handles either
utiArray=(
    http
)
##### /VARIABLES #####

##### DO NOT EDIT BELOW THIS LINE #####
```

```
# Install UtilUTI if not present

utilutiPath="/usr/local/bin/utiluti"

if [ -e "$utilutiPath" ]; then
    echo "[RESULT] UtilUTI already installed"
else
    echo "[TASK] Installing UtilUTI"
    /usr/sbin/installer -pkg "$softwarePath/$utilUTIpkg"
    -target /
    # chmod 755 "$utilutiPath"
    echo "[RESULT] UtilUTI installed"
fi
```

```
# Is handler installed?
handlerInstalled="$(mdfind "kMDItemCFBundleIdentifier ==
'$handlerBundleID'")"

if [ -z "$handlerInstalled" ]; then
    echo "Variable is null or empty - handler not installed"
    echo "Exiting"
    exit 1
else
    echo "Variable is not null - handler detected - proceeding"
fi
```

```
# Get current user
```

```
currentUser=$(/usr/bin/stat -f%Su /dev/console)
```

```
uid=$(id -u "$currentUser") # Note double-wrap required for  
modifying URLs
```

```
utilutiCall="launchctl asuser "$uid" sudo -u "$currentUser" $utiluti"
```

```
#Trigger UtilUTI update by iterating through array
for uti in "${utiArray[@]}" ; do
    echo "[TASK] setting $uti"

# Branch syntax - UTI vs URL
urlCheck="http|ssh|mailto|tel"

    if [[ $uti =~ ^($urlCheck)$ ]]; then
        # URL detected
        verb="url"
    else
        # UTI detected
        verb="type"
    fi
```

```
# Syntax: utiluti url set $UTI $APPBUNDLEID
```

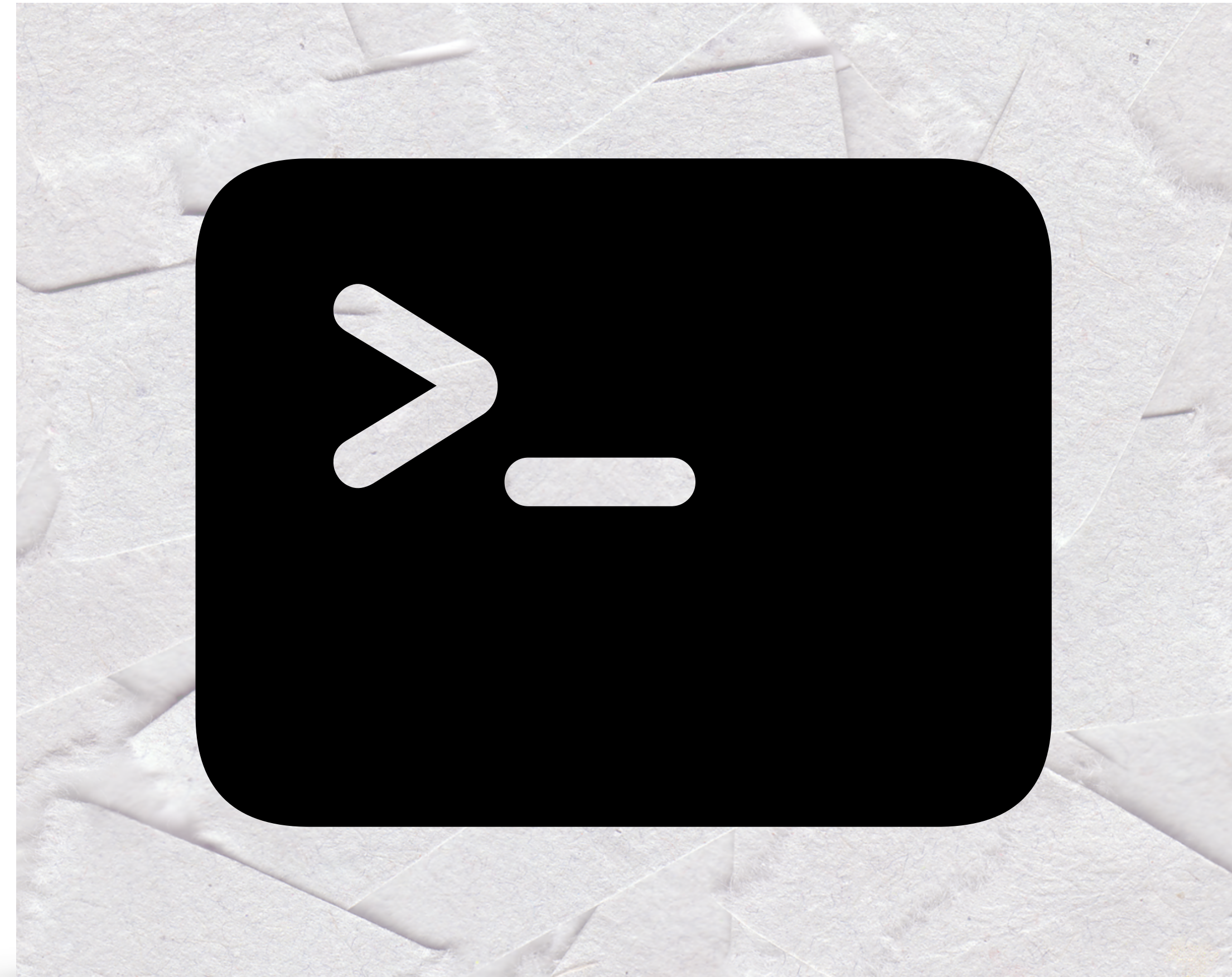
```
$utilutiCall $verb set $uti $handlerBundleID
```

```
# Note: For web browsers, the script will hold here until the  
user makes a selection from the popup.
```

```
echo "[RESULT] $uti handler updated"  
done
```

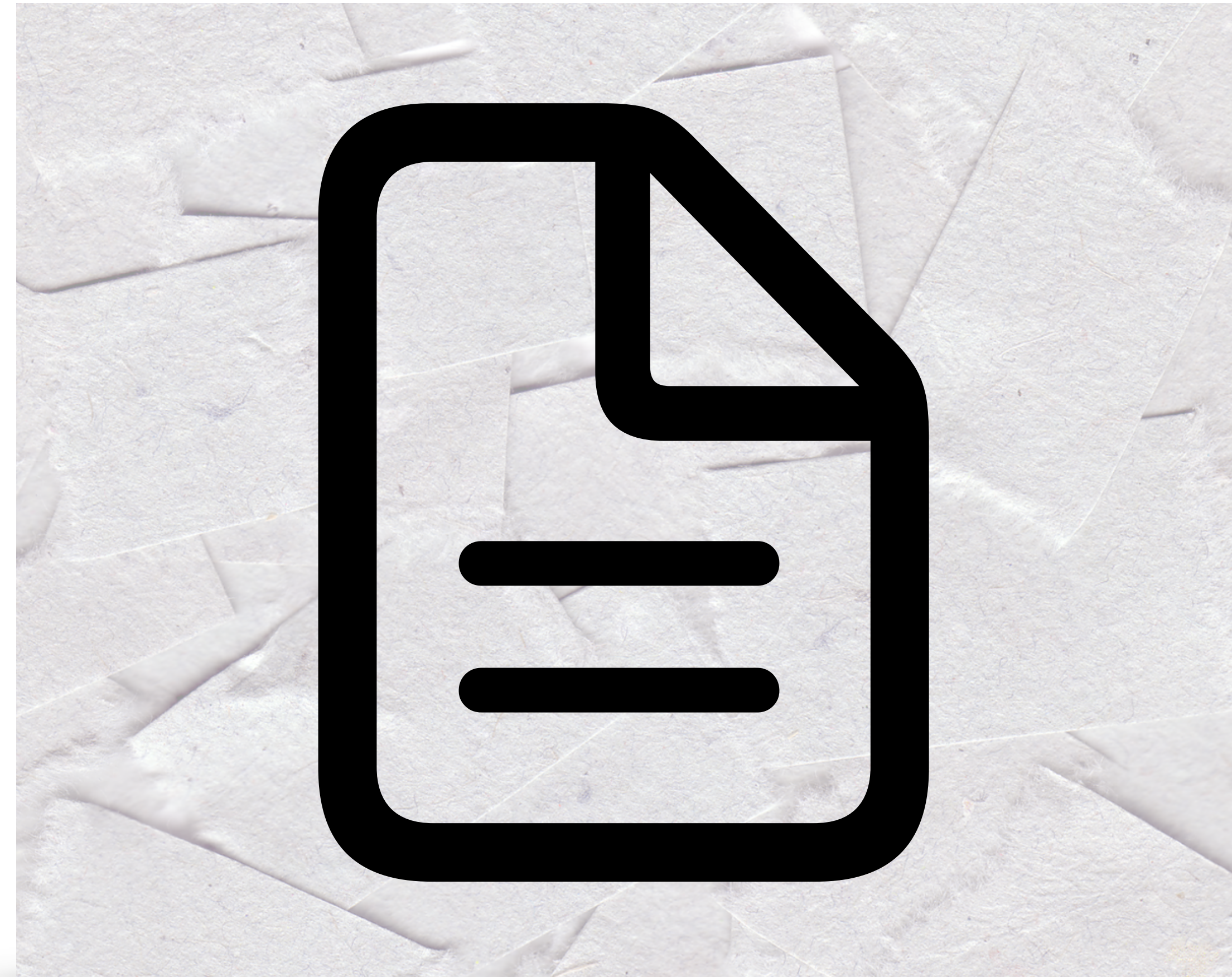
Making Changes

- Bulk changes with .xml file
- You could also scale your script to accept arrays of handlers, or just copy/paste your code.
- Review the guidelines in GitHub.



Homework

- Script changing multiple UTIs and handlers in one action
- Research UTI inheritance and change multiple UTIs via their parent
 - `public.jpeg` as a subtype of `public.image`
- Create a user-interactive notification for your self service UTI scripts



THANK YOU!

[HTTPS://GITHUB.COM/SCRIPTINGOSX/UTILUTI](https://github.com/scriptingosx/utiluti)



www.sudoade.com

FEEDBACK: [HTTPS://BIT.LY/PSUMAC26-1168689](https://bit.ly/psumac26-1168689)