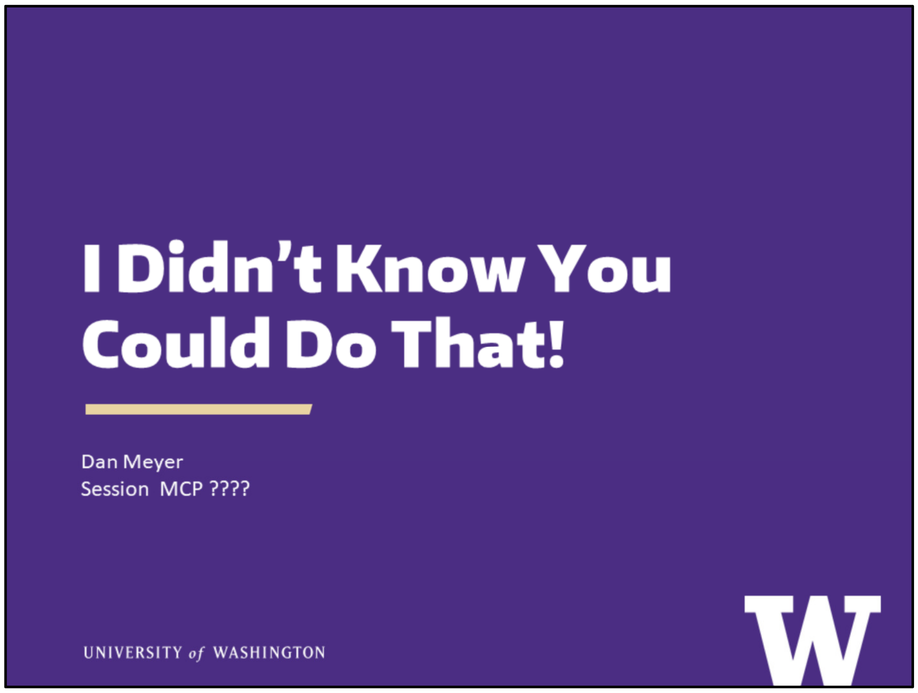
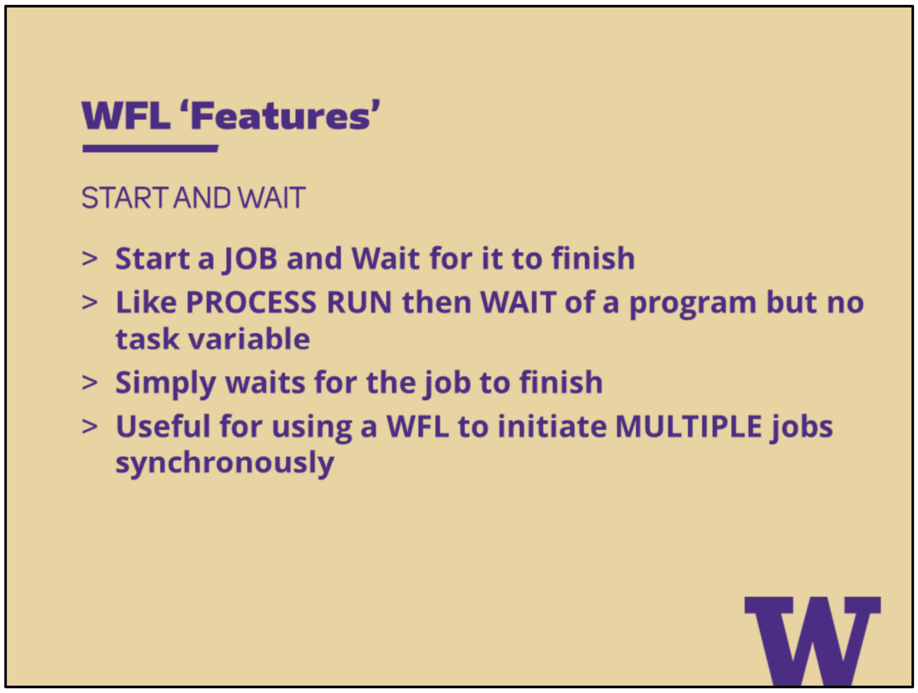


I Didn’t Know You Could Do That!

Dan Meyer
Session MCP ????

UNIVERSITY *of* WASHINGTON



WFL 'Features'

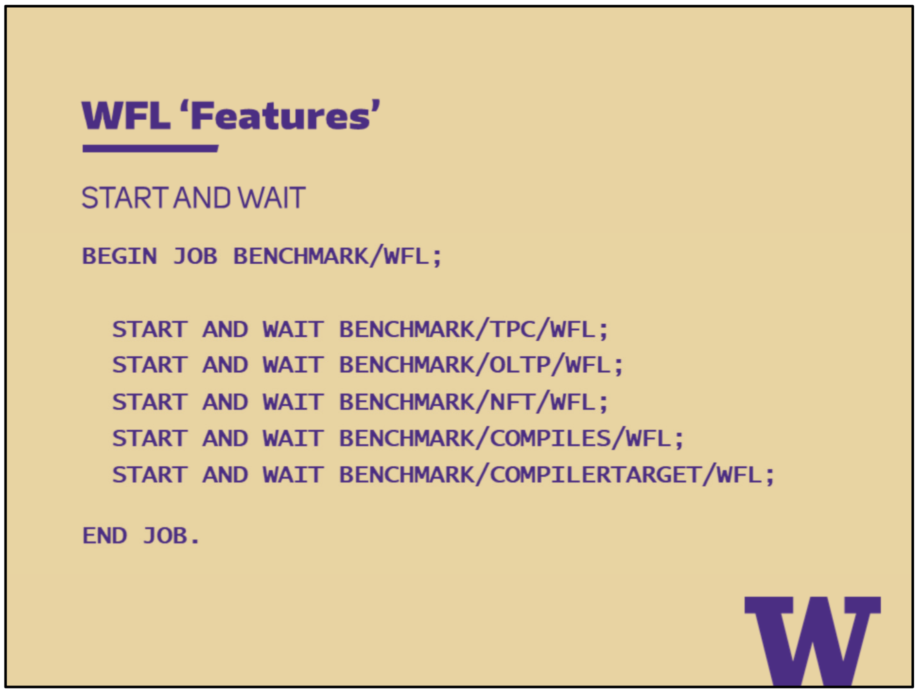
START AND WAIT

> Start a JOB and Wait for it to finish
> Like PROCESS RUN then WAIT of a program but no task variable
> Simply waits for the job to finish
> Useful for using a WFL to initiate MULTIPLE jobs synchronously

During an upcoming blocktime, Dan was scheduled to run six benchmarks within a 30 minute window. To initiate them one after another, he was going to have to sit there and really pay close attention to his emulator window. He was trying to figure out how to 'automate' it, one thought being to have each job create a "COMPLETED" file and then a WFL monitoring that file's residency, when he stumbled on the START AND WAIT statement. As the manual says, it starts a job and then waits for the job to finish. PERFECT.

Looking at this function a little more, it is like a PROCESS RUN statement where you initiate a task, and then wait for the task to complete using its task variable. Except you don't supply a task variable, the initiating JOB simply waits for the child job to finish.

So this is very useful when you need to initiate several jobs synchronously, like in his benchmark case.

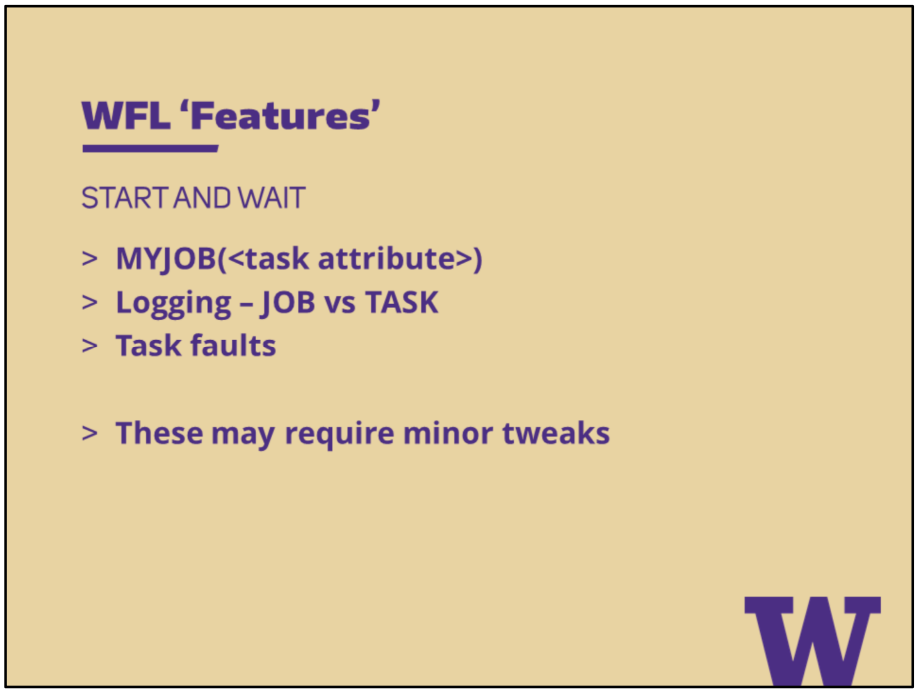


WFL 'Features'

START AND WAIT

```
BEGIN JOB BENCHMARK/WFL;

  START AND WAIT BENCHMARK/TPC/WFL;
  START AND WAIT BENCHMARK/OLTP/WFL;
  START AND WAIT BENCHMARK/NFT/WFL;
  START AND WAIT BENCHMARK/COMPILES/WFL;
  START AND WAIT BENCHMARK/COMPILERTARGET/WFL;

END JOB.
```

Here's the very simple WFL he wrote to control the series of benchmarks.



WFL 'Features'

START AND WAIT

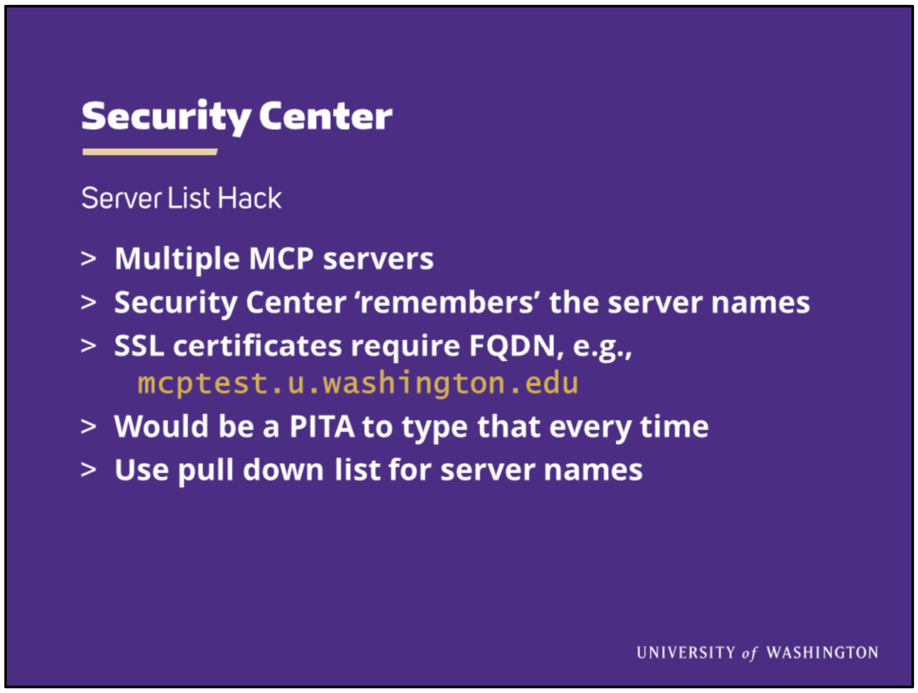
> MYJOB(<task attribute>)
> Logging – JOB vs TASK
> Task faults

> These may require minor tweaks

There are a few gotchas:

- The MYJOB attribute in the jobs being initiated doesn't necessarily produce the same result, so the subservient jobs should use MYSELF
- The subservient jobs become "TASKS", so a LOG <job number> of the subservient JOB finds nothing
- A fault in a subservient JOB is treated as a TASK fault

In general, you may need to review how errors and job/task attributes are used at various levels.


Security Center

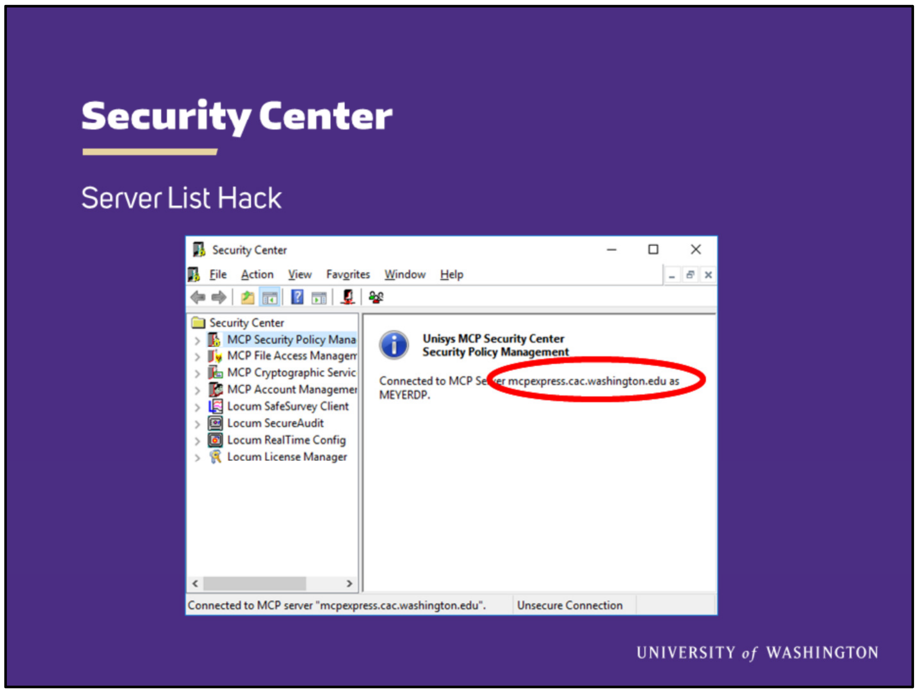
Server List Hack

> Multiple MCP servers
> Security Center 'remembers' the server names
> SSL certificates require FQDN, e.g., `mcptest.u.washington.edu`
> Would be a PITA to type that every time
> Use pull down list for server names

UNIVERSITY of WASHINGTON

At the University of Washington we have multiple CS4390s and Forward cabinets. What does one do with all this idle capacity? Well, we setup multiple DE1000 environments for testing purposes, for example, one is setup for the DBAs with a LOT of disk space so they can run various versions of Data Bridge and test a reorg process. As you've seen in other presentations this week, we are committed to using SFTP, IPsec, TCPIP Filter Rules – all sorts of security related MCP features. So with six or seven MCP environments, its nice that Security Center remembers the names of the servers you've connected to for managing certificates and policies.

Dan recently setup Security Center to use SSL for all connections. There is a new SSL certificate for each server and in the certificate it specifies the fully qualified domain names for each server. As a result, upon connection, the server name you enter to connect to the server to needs to match the FQDN that was registered with the certificate … such a PITA. But, since Security Center DOES remember those names – you can see the list and select a host if you click the down arrow on the right of the Connect Message Box - now the list contains the "common" name and the "FQDN" name for each system.



Security Center

Server List Hack

Connected to MCP Server mcpexpress.cac.washington.edu as MEYERDP.

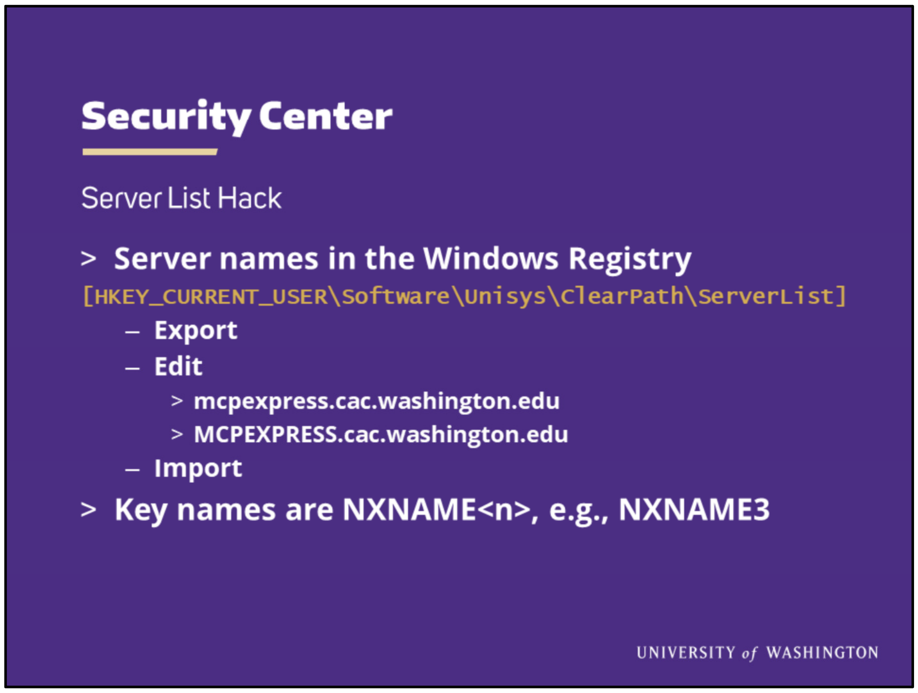
Connected to MCP server "mcpexpress.cac.washington.edu". Unsecure Connection

So now you've entered an FQDN in all lower case. It connects fine, but when you connect, that host name doesn't "stand out" on the Security Center home screen.

OR, you get a new workstation or have to replace your hard drive and none of the server names are in the list.

OR, when you first started using Security Center, you entered the common name and now you enter the FQDN and the list contains both values.

OR, you connected to a test environment and that server no longer exists, but the name is still in the pull down list.



Security Center

Server List Hack

- Server names in the Windows Registry
  `[HKEY_CURRENT_USER\Software\Unisys\ClearPath\ServerList]`
  - Export
  - Edit
    - mcpexpress.cac.washington.edu
    - MCPEXPRESS.cac.washington.edu
  - Import
- Key names are NXNAME<n>, e.g., NXNAME3

UNIVERSITY *of* WASHINGTON

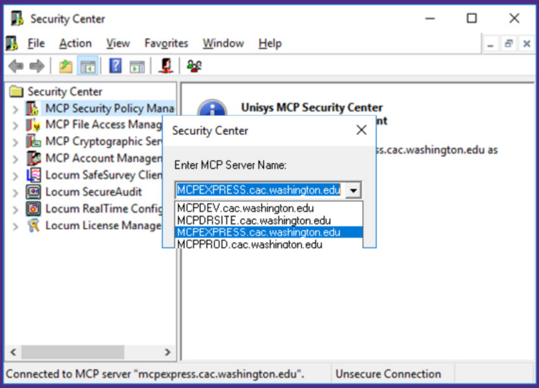
Well, the list of names is stored in the Windows registry in the HKCU\Software\Unisys\ClearPath\ServerList section. Turns out, Security Center, Admin Center, Op Center, etc., all use the same registry list.

By running REGEDIT.EXE, you can delete entries, change values, export the list, edit the list with the "notepad" of your choice, and reimport the list.

The key names for these entries are NXNAME<number>, for example NXNAME3. They are retrieved, starting with NXNAME1, incrementing the number by 1 until the entry isn't found. So if you WERE to edit the list and remove entries, you need to make sure to fixup the KEY NAMES to have sequential numbers.

Security Center

Server List Hack



Security Center

Enter MCP Server Name:

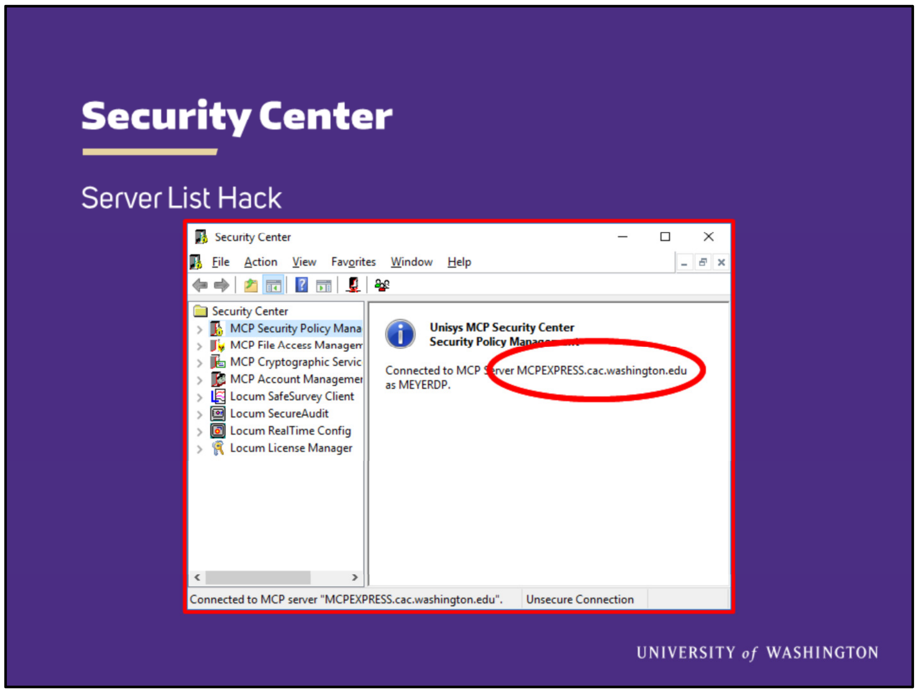
MCPEXPRESS.cac.washington.edu

MCPDEV.cac.washington.edu
MCPDRSITE.cac.washington.edu
MCPEXPRESS.cac.washington.edu
MCPPROD.cac.washington.edu

Connected to MCP server "mcpexpress.cac.washington.edu". Unsecure Connection

UNIVERSITY *of* WASHINGTON

So now, after you've edited the list to remove the "duplicate names", and UPCASED the first node of the FQDN to match the common name, and removed servers that no longer exist … as you can see in this pull down list,



Security Center

Server List Hack

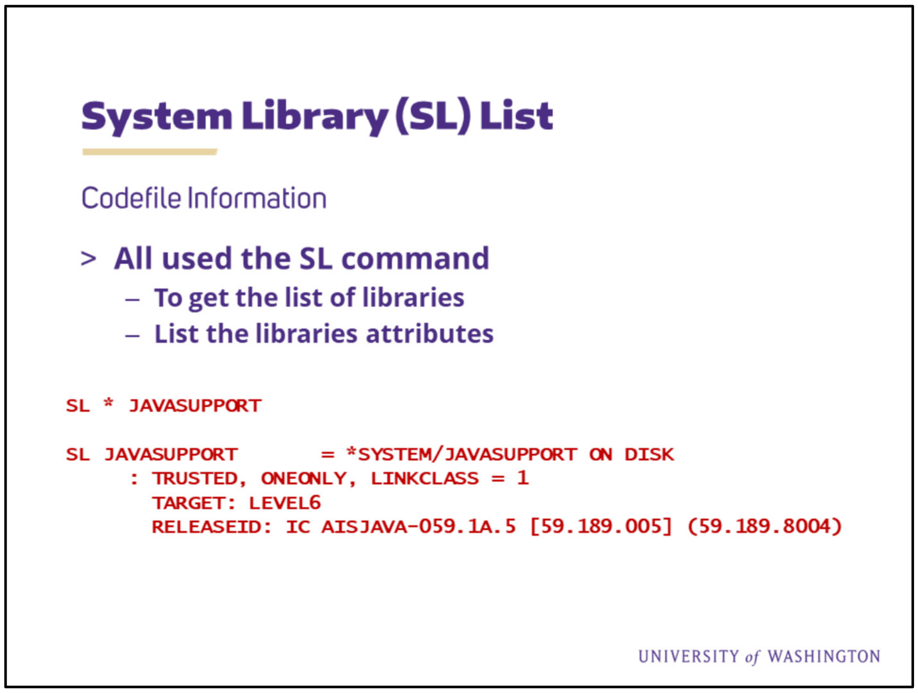
Unisys MCP Security Center
Security Policy Management

Connected to MCP Server MCPEXPRESS.cac.washington.edu as MEYERDP.

Connected to MCP server "MCPEXPRESS.cac.washington.edu". Unsecure Connection

UNIVERSITY *of* WASHINGTON

When you connect to the server, the HOSTNAME does stand out, the domain of the FQDN, which you really don't need to see kind of "disappears" in the home screen.



System Library (SL) List

Codefile Information

> **All used the SL command**
  - **To get the list of libraries**
  - **List the libraries attributes**

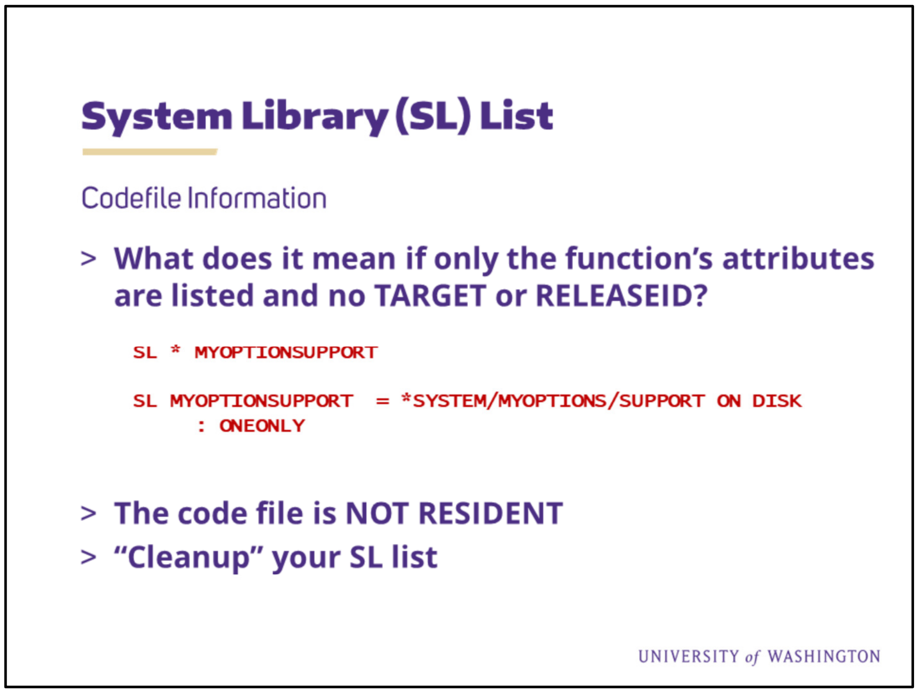
```
SL * JAVASUPPORT

SL JAVASUPPORT       = *SYSTEM/JAVASUPPORT ON DISK
     : TRUSTED, ONEONLY, LINKCLASS = 1
       TARGET: LEVEL6
       RELEASEID: IC AISJAVA-059.1A.5 [59.189.005] (59.189.8004)
```

UNIVERSITY *of* WASHINGTON

Have you ever needed to manage your list of System Libraries and wanted to only have the SLs in place for code files that don't exist?

We all know to use the SL command to get the list of libraries, and we know to use the SL * <function name> to see the attributes of that function, like ONEONLY or TRUSTED

But there's a couple attributes displayed that provide a bit more information: the codefile target level, e.g.,. LEVEL6, and the codefile's RELEASEID value. And this same information is see in the SUMLOG in the LOG CONFIG table.



System Library (SL) List

Codefile Information

> What does it mean if only the function's attributes are listed and no TARGET or RELEASEID?

```
SL * MYOPTIONSUPPORT

SL MYOPTIONSUPPORT  = *SYSTEM/MYOPTIONS/SUPPORT ON DISK
     : ONEONLY
```

> The code file is NOT RESIDENT

> "Cleanup" your SL list

UNIVERSITY *of* WASHINGTON

What does it mean if only the function's attributes are listed, and there is no TARGET or RELEASEID, such as in this example of MYOPTIONSUPPORT?

What it means is the code file is NOT RESIDENT on DISK – maybe you forgot to fully install a new code file. Or maybe you accidently removed a file.

By recognizing the RELEASEID is available via that mechanism, and knowing that information is in the SUMLOG, it gives you a way to know what IC level you were running at a particular point in time.