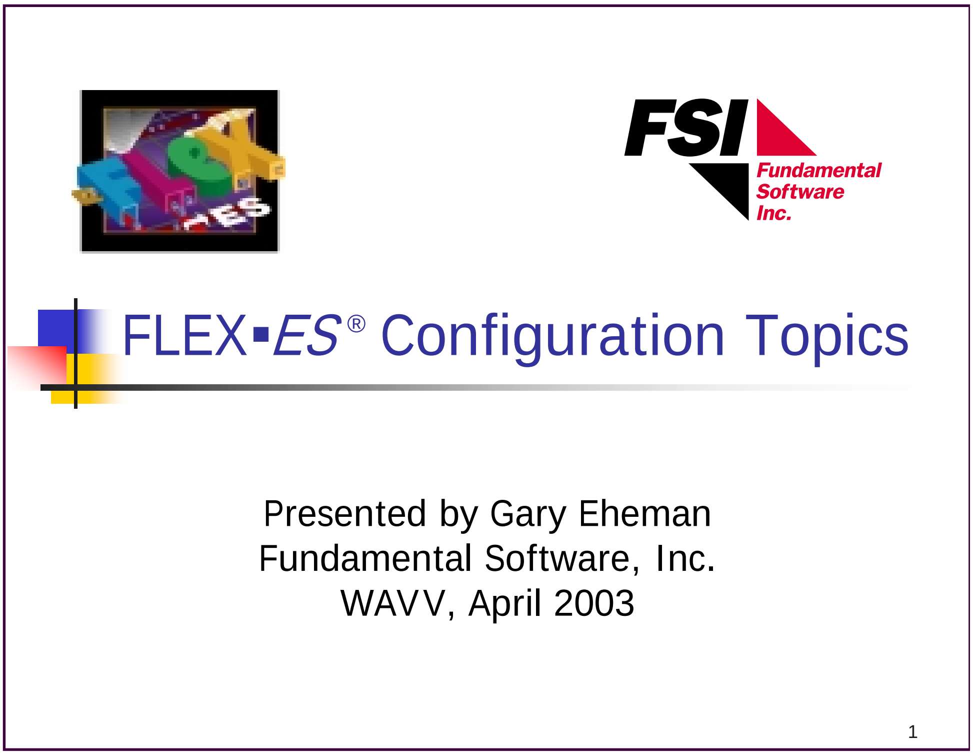
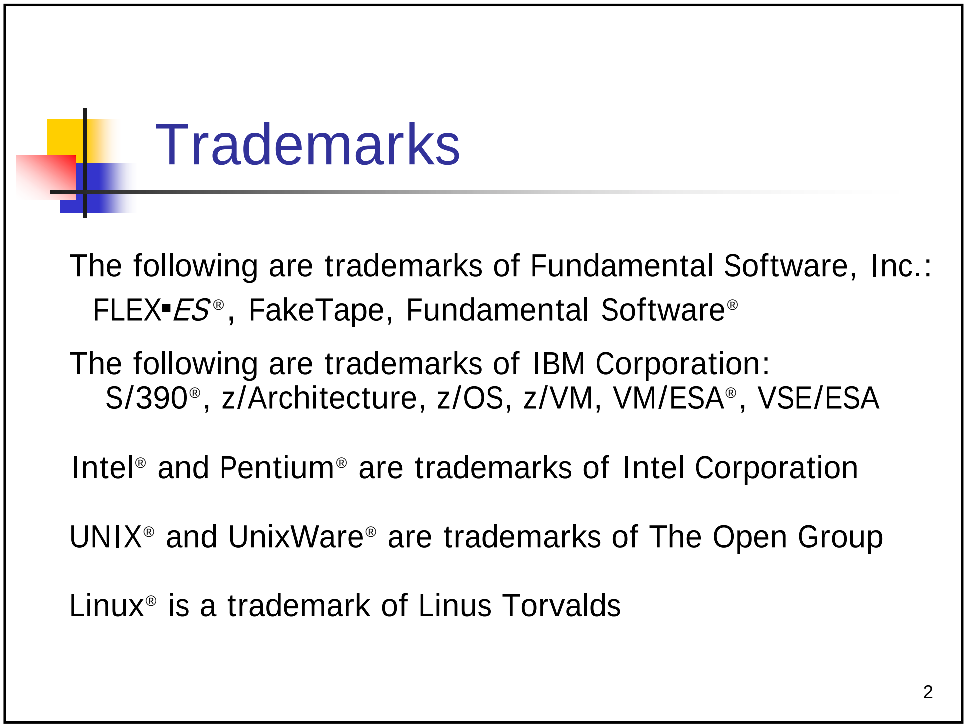


FLEX▪*ES*® Configuration Topics

Presented by Gary Eheman
Fundamental Software, Inc.
WAVV, April 2003



Trademarks

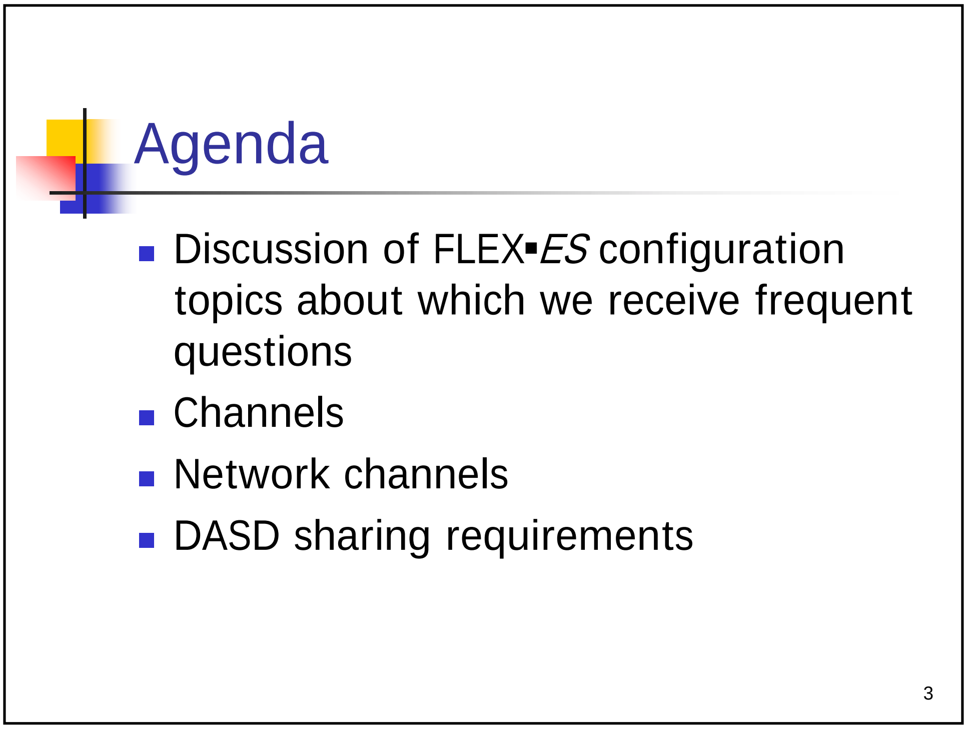
The following are trademarks of Fundamental Software, Inc.:
FLEX▪*ES*®, FakeTape, Fundamental Software®

The following are trademarks of IBM Corporation:
S/390®, z/Architecture, z/OS, z/VM, VM/ESA®, VSE/ESA

Intel® and Pentium® are trademarks of Intel Corporation

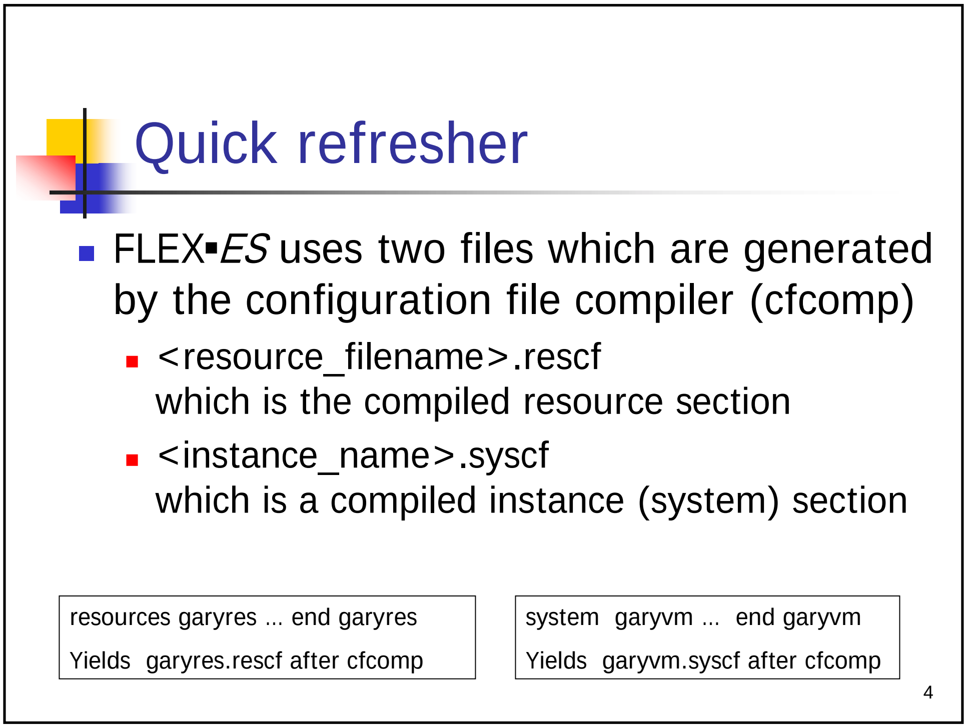
UNIX® and UnixWare® are trademarks of The Open Group

Linux® is a trademark of Linus Torvalds



Agenda

- Discussion of FLEX▪*ES* configuration topics about which we receive frequent questions
- Channels
- Network channels
- DASD sharing requirements

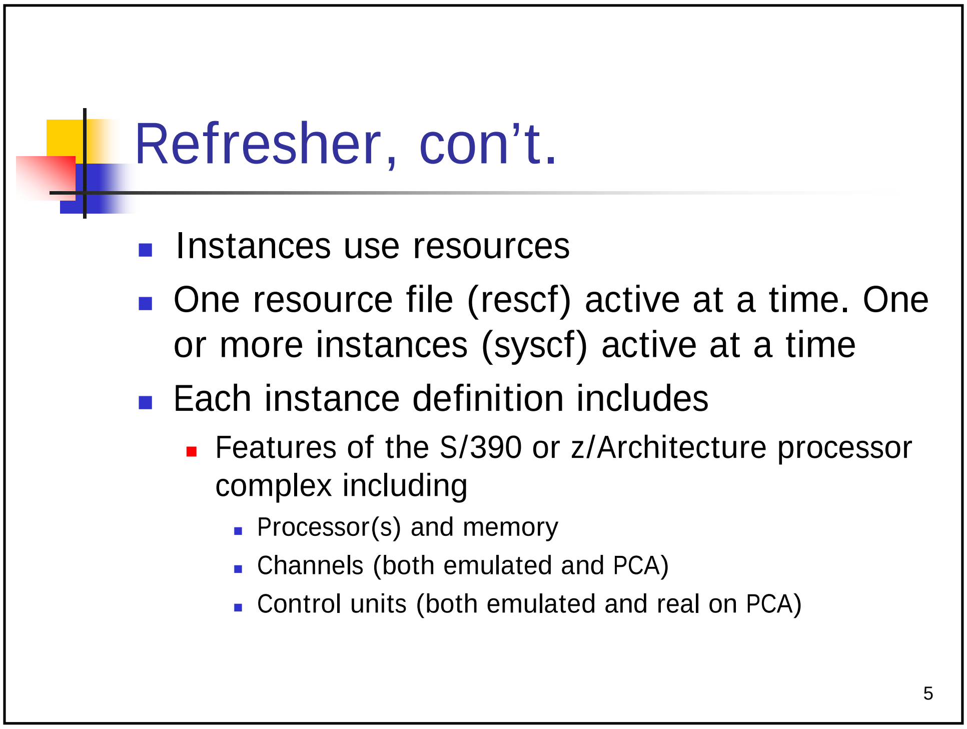


Quick refresher

- FLEX▪*ES* uses two files which are generated by the configuration file compiler (cfcomp)
  - <resource_filename>.rescf
    which is the compiled resource section
  - <instance_name>.syscf
    which is a compiled instance (system) section

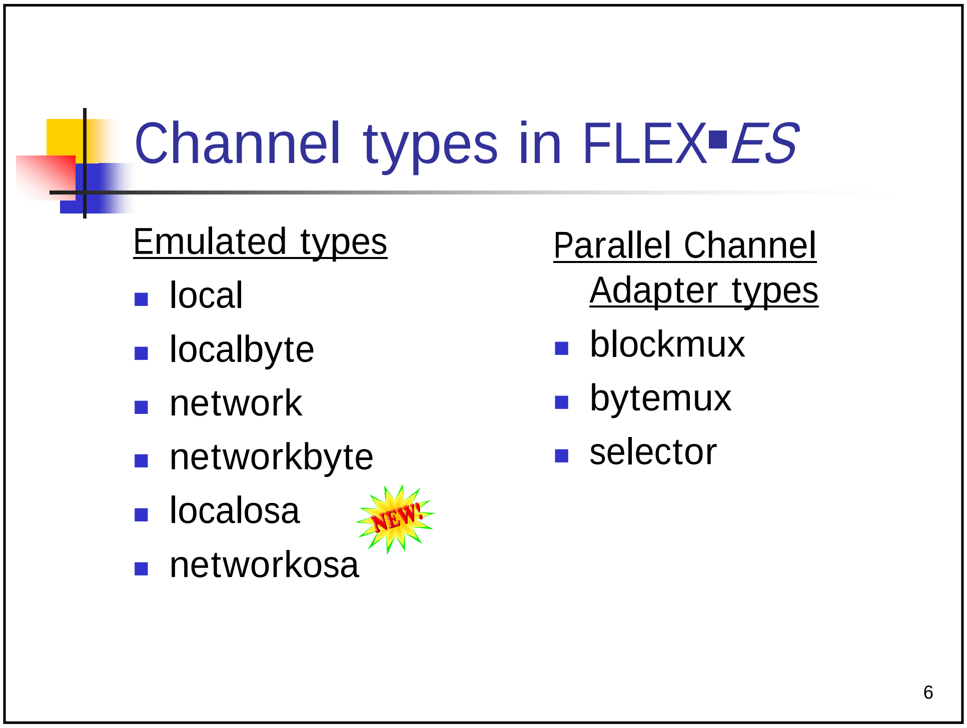
resources garyres ... end garyres
Yields garyres.rescf after cfcomp

system garyvm ... end garyvm
Yields garyvm.syscf after cfcomp



Refresher, con’t.

- Instances use resources
- One resource file (rescf) active at a time. One or more instances (syscf) active at a time
- Each instance definition includes
  - Features of the S/390 or z/Architecture processor complex including
    - Processor(s) and memory
    - Channels (both emulated and PCA)
    - Control units (both emulated and real on PCA)



Channel types in FLEX▪*ES*

Emulated types

- local
- localbyte
- network
- networkbyte
- localosa
- networkosa

NEW!

Parallel Channel Adapter types

- blockmux
- bytemux
- selector

Define the channels you need

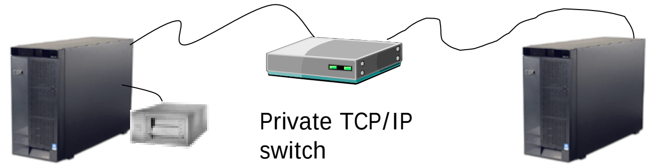
- Channel paths are flexibly configured for each system instance

```
system  garyvm:
channel (0) local # channel path 0 is local block multiplexor
channel (1) localbyte # channel path 1 is local byte multiplexor
channel (8) network # channel path 8 is network blockmux
channel (9) localosa # channel path 9 will have osa devices on it
...
cu devad(0x440,2) path(9) resource(garyosa)
...
```

What is a network channel?

- *Network* and *networkbyte* channels are used to access resources served by FLEX▪*ES* on a second server. The hosting servers' TCP/IP is used to establish the channel path
- Performance is limited by the speed of the network connection
- Production customers who exploit this capability typically use private (isolated for this purpose) network switches between the servers

Network channel example



system1
192.168.1.1

Private TCP/IP
switch

system2
192.168.1.2

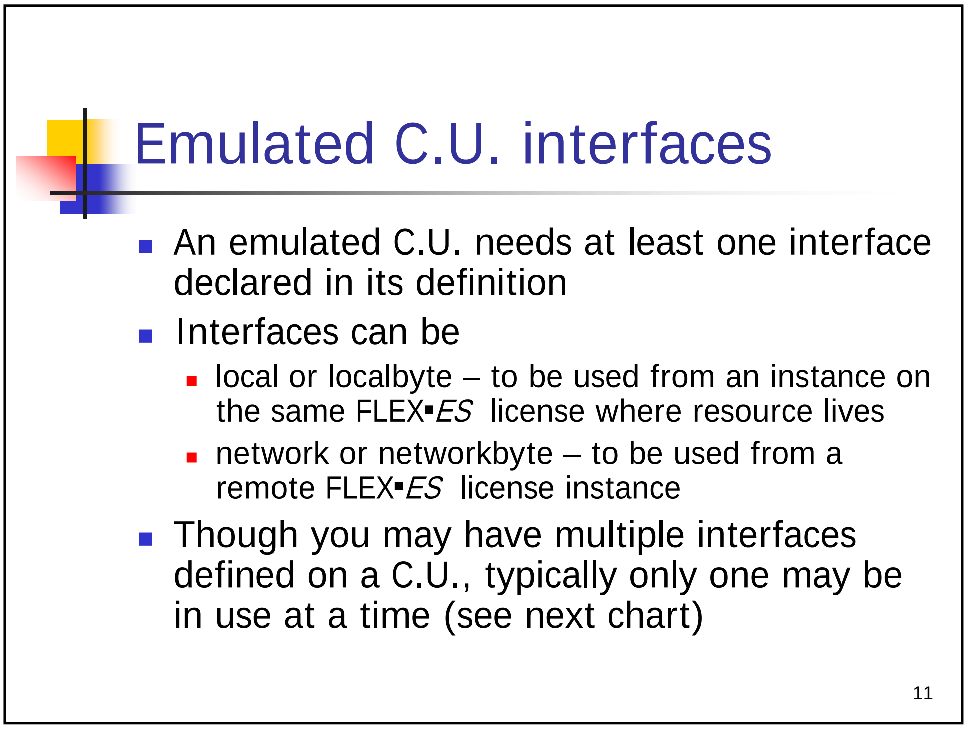
```
resources  myresources:
local_tape: cu 3490
interface local (1)
device (00) 3490-E OFFLINE
end local_tape
remote_tape: cu 3480
  interface network (1)
  device (00) 3480 /dev/rmt/ctape2
end remote_tape
end myresources
```

```
system vsetest:
...
channel (3) network
cu devad(0x180,1) path(3)
        resource (system1:remote_tape)
...
```

VSETEST, running on server system2, sees an emulated 3480 tape drive at address 180 that is physically SCSI connected to server system1

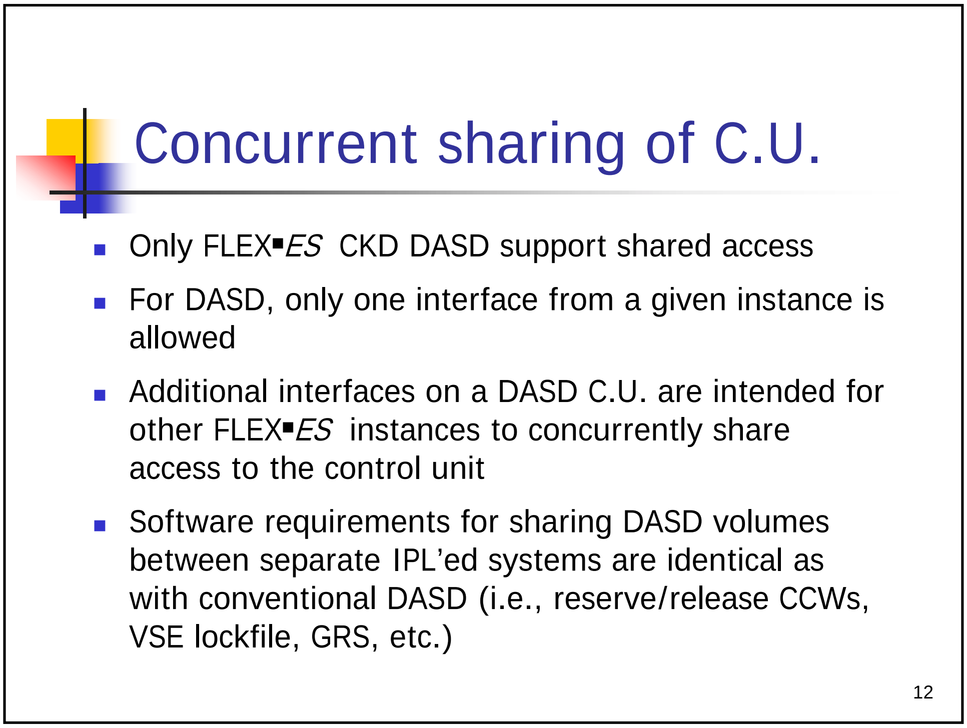
Network resource facts

- If the individual resource or resource manager (resadm) on the owning system is terminated without softly terminating access on the remote system, then I/O CC=3 will be presented
- To display status of remote resources, use *"-h <hostname>"* option of resadm
  - resadm -h system1 -r



Emulated C.U. interfaces

- An emulated C.U. needs at least one interface declared in its definition
- Interfaces can be
  - local or localbyte – to be used from an instance on the same FLEX▪*ES* license where resource lives
  - network or networkbyte – to be used from a remote FLEX▪*ES* license instance
- Though you may have multiple interfaces defined on a C.U., typically only one may be in use at a time (see next chart)



Concurrent sharing of C.U.

- Only FLEX▪*ES* CKD DASD support shared access
- For DASD, only one interface from a given instance is allowed
- Additional interfaces on a DASD C.U. are intended for other FLEX▪*ES* instances to concurrently share access to the control unit
- Software requirements for sharing DASD volumes between separate IPL'ed systems are identical as with conventional DASD (i.e., reserve/release CCWs, VSE lockfile, GRS, etc.)

Example shared DASD C.U.

```
resources myresources:
vse_shared_3990: cu 3990     # a meaningful control unit name
interface local(2)           # one each for two separate instances on this
                             # server
interface network(1)  # one for a remote server via network channel
device(00) 3390-3 /dev/rdsk/...
device(01) 3390-3 /dev/rdsk/...
end vse_shared_3990
...
end myresources
```

13

Feature LPAR and LPARNUM

- *feature lpar* causes the S/390 software to recognize an LPAR rather than a basic image
- *lparnum(n)* specification sets LPAR number in CPU serial number. Default is “0”
- Multiple FLEX▪*ES* instances on one license not using unique LPAR numbers will have identical CPU serial numbers
- Unique CPU serial numbers for multiple FLEX▪*ES* instances sharing resources may be required by O.S. software locking mechanisms
- For VM, *feature lpar* causes VM to disable active wait

14

Meeting VSE lockfile requirements

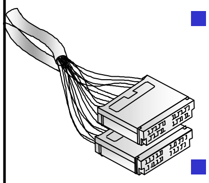
```
system  vseprod:
...
feature lpar
lparnum(1)   # unique lparnum
...
cu (0x300,4) path(1)
        resource(vse_shared_3990)
...
end vseprod
```

```
system vsetest:
...
feature lpar
lparnum(2)   # unique lparnum
...
cu (0x500,4) path(1)
        resource(vse_shared_3990)
...
end vsetest
```

```
flexes-vseprod> display cpuid
CPUID= 0a010099 12460000
```

```
flexes-vsetest> display cpuid
CPUID= 0a020099 12460000
```

Control units attached to Parallel Channel Adapters



- Rules for coding configuration file parameters are similar as for an IOCP on traditional iron
- If migrating from traditional iron to FLEX▪*ES* , let your current IOCP be a source of vital information and clues
- Your focus is the control unit, not the devices behind the control unit

Control unit types 1, 2, or 3

As explained in IBM’s channel publications:

- Type 1 – the control unit can control activity to a single device at a time
- Type 2 – the C.U. controls activity to multiple I/O devices at a time concurrently without losing pending sense information or control of I/O operations
- Type 3 – like type 2, but the C.U. **blocks** I/O to devices on the C.U. upon unit-check from a device until that device accepts a new command (usually sense)

17

When to specify *shared* or *sharedb* option?

- For a Type 1 control unit, specify **shared** option in FLEX▪*ES* definition
- For a Type 2 control unit, *omit* shared or sharedb options in FLEX▪*ES* definition
- For a Type 3 control unit, specify **sharedb** option in FLEX▪*ES* definition

18

But how do I know?

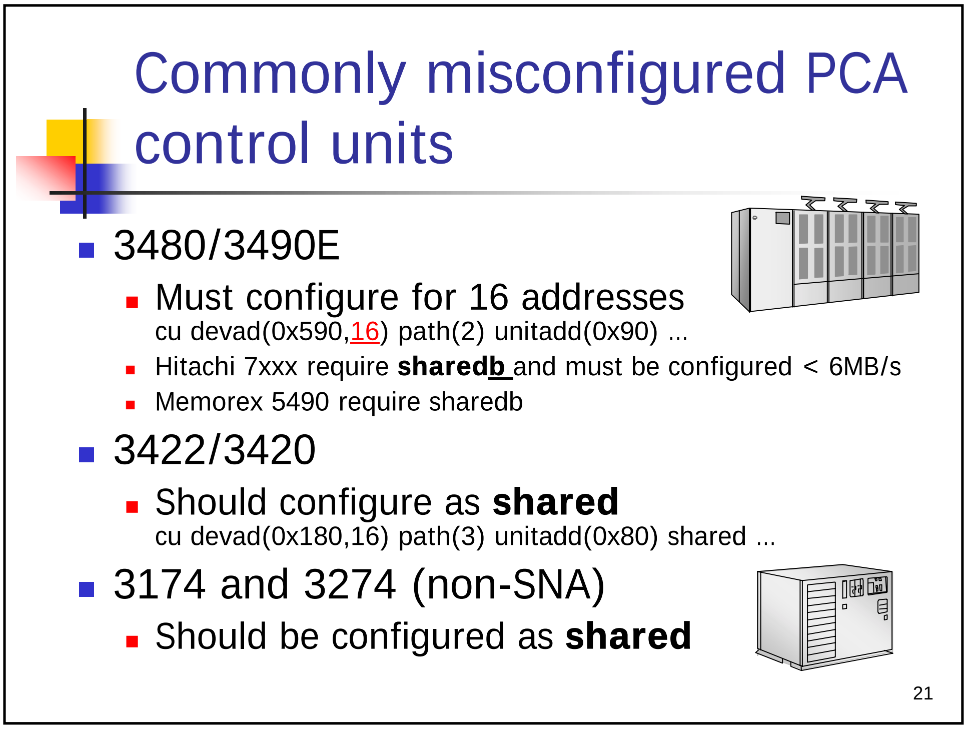
- Refer to the manufacturer’s planning guide for the device as a definitive source
- Refer to the IOCP deck of the processor from which you are migrating for hints
- Refer to tables in appendix of IBM IOCP manuals for common device definitions (SHARED=Y|N|YB). Refer also to footnotes

Configured vs. installed devices

- A C.U. on a PCA might be configured to control more devices than are physically installed
- When defining a PCA-attached C.U. in a FLEX▪*ES* config file, code device count according to C.U.’s configuration
- Configuring fewer will result in a configuration alert on the channel
- Control units on local or network channels specify only defined devices in contrast with PCA-attached C.U.s

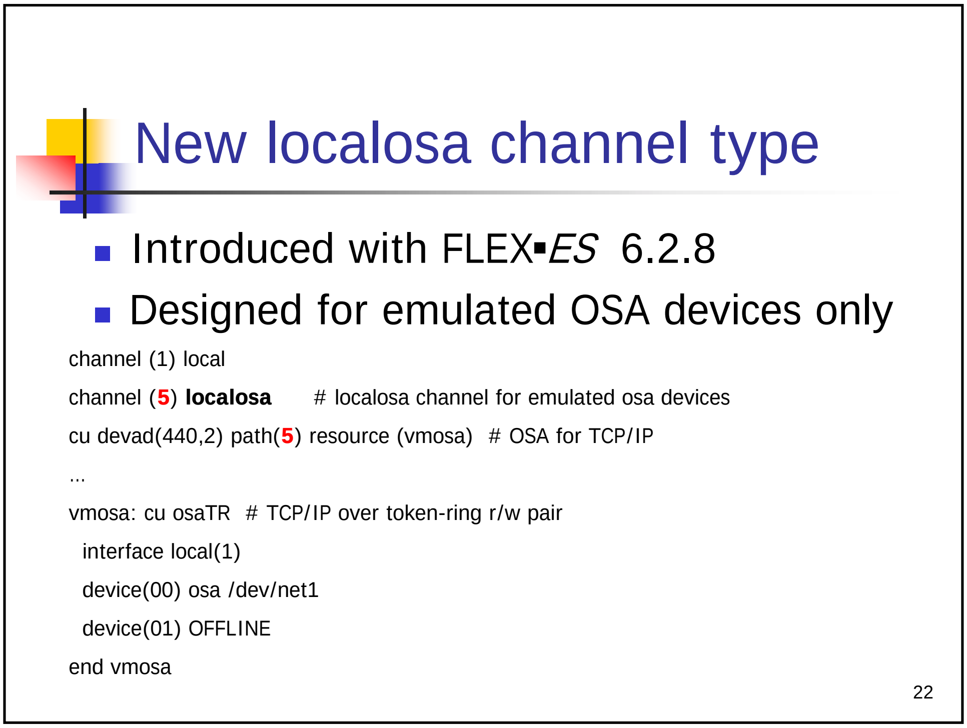
👎
```
cu devad(0x590,4) path(2) unitadd(0x90) streaming(45) # PCA 3490-E C.U.
```

👍
```
cu devad(0x590,16) path(2) unitadd(0x90) streaming(45) # PCA 3490-E C.U.
```



Commonly misconfigured PCA control units

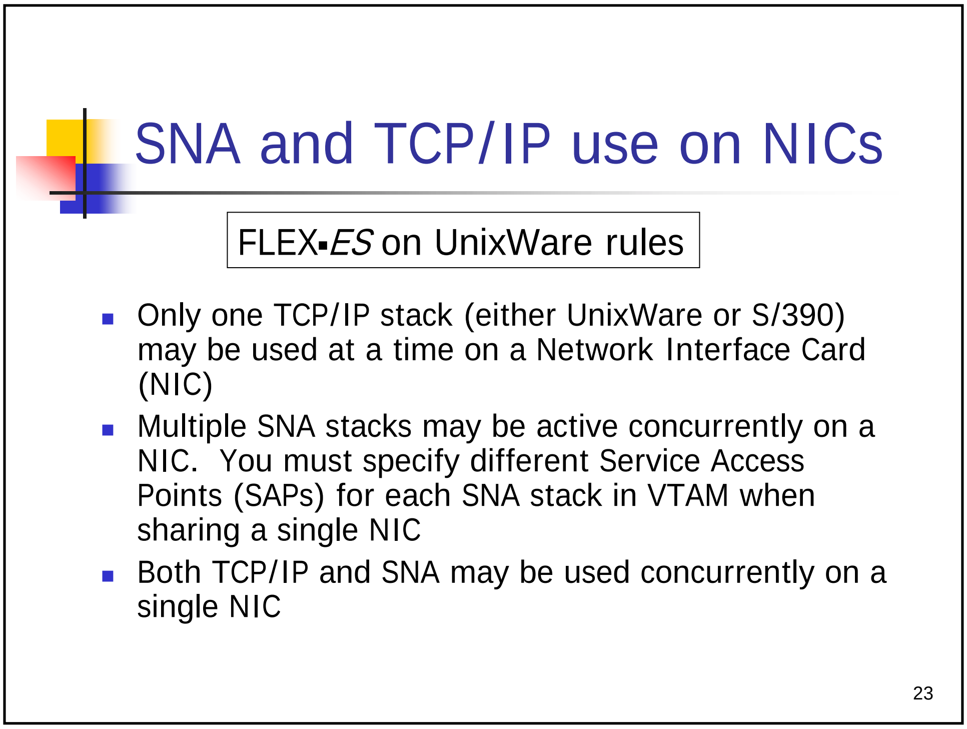
- 3480/3490E
  - Must configure for 16 addresses
    cu devad(0x590,16) path(2) unitadd(0x90) ...
  - Hitachi 7xxx require **sharedb** and must be configured < 6MB/s
  - Memorex 5490 require sharedb
- 3422/3420
  - Should configure as **shared**
    cu devad(0x180,16) path(3) unitadd(0x80) shared ...
- 3174 and 3274 (non-SNA)
  - Should be configured as **shared**



New localosa channel type

- Introduced with FLEX▪*ES* 6.2.8
- Designed for emulated OSA devices only

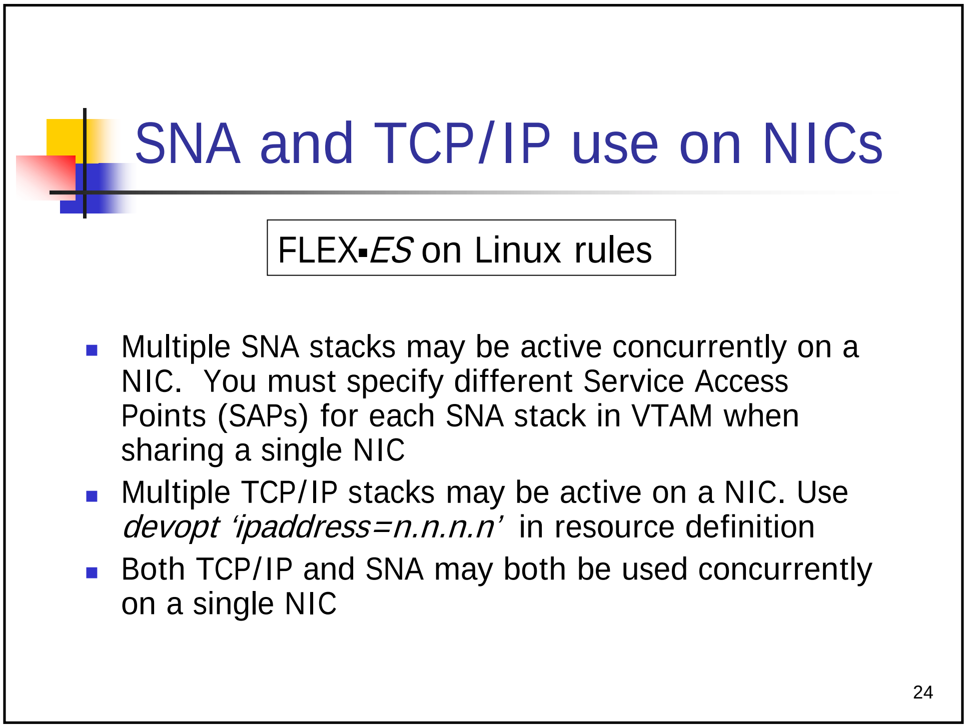
```
channel (1) local
channel (5) localosa      # localosa channel for emulated osa devices
cu devad(440,2) path(5) resource (vmosa)  # OSA for TCP/IP
...
vmosa: cu osaTR  # TCP/IP over token-ring r/w pair
  interface local(1)
  device(00) osa /dev/net1
  device(01) OFFLINE
end vmosa
```



SNA and TCP/IP use on NICs

FLEX▪*ES* on UnixWare rules

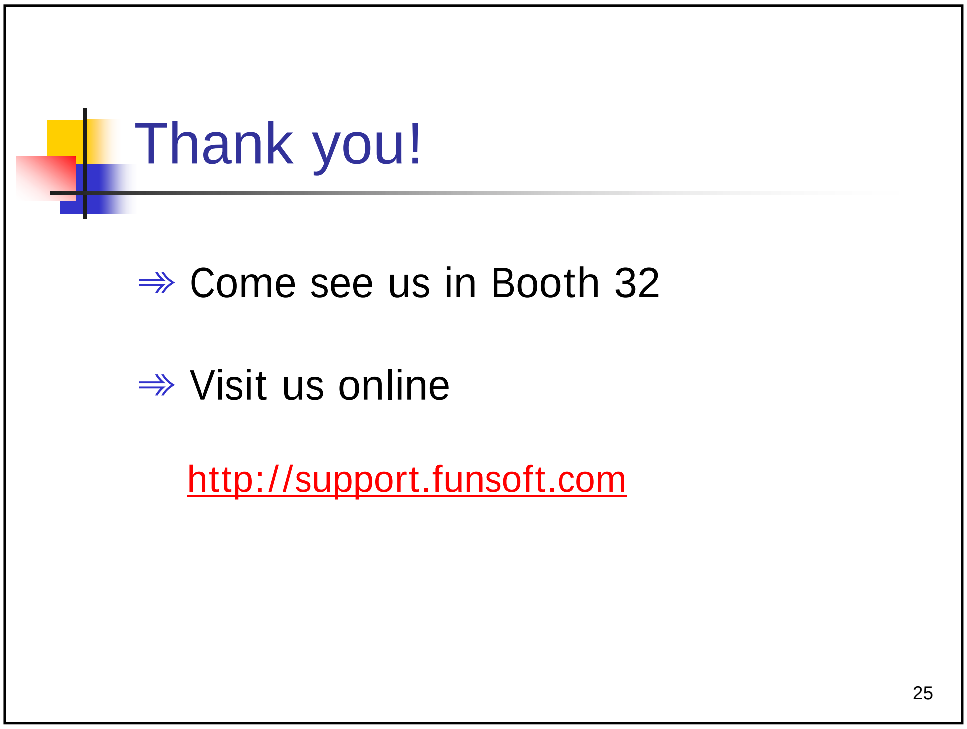
- Only one TCP/IP stack (either UnixWare or S/390) may be used at a time on a Network Interface Card (NIC)
- Multiple SNA stacks may be active concurrently on a NIC. You must specify different Service Access Points (SAPs) for each SNA stack in VTAM when sharing a single NIC
- Both TCP/IP and SNA may be used concurrently on a single NIC



SNA and TCP/IP use on NICs

FLEX▪*ES* on Linux rules

- Multiple SNA stacks may be active concurrently on a NIC. You must specify different Service Access Points (SAPs) for each SNA stack in VTAM when sharing a single NIC
- Multiple TCP/IP stacks may be active on a NIC. Use *devopt 'ipaddress=n.n.n.n'* in resource definition
- Both TCP/IP and SNA may both be used concurrently on a single NIC



Thank you!

⇛ Come see us in Booth 32

⇛ Visit us online

http://support.funsoft.com

25