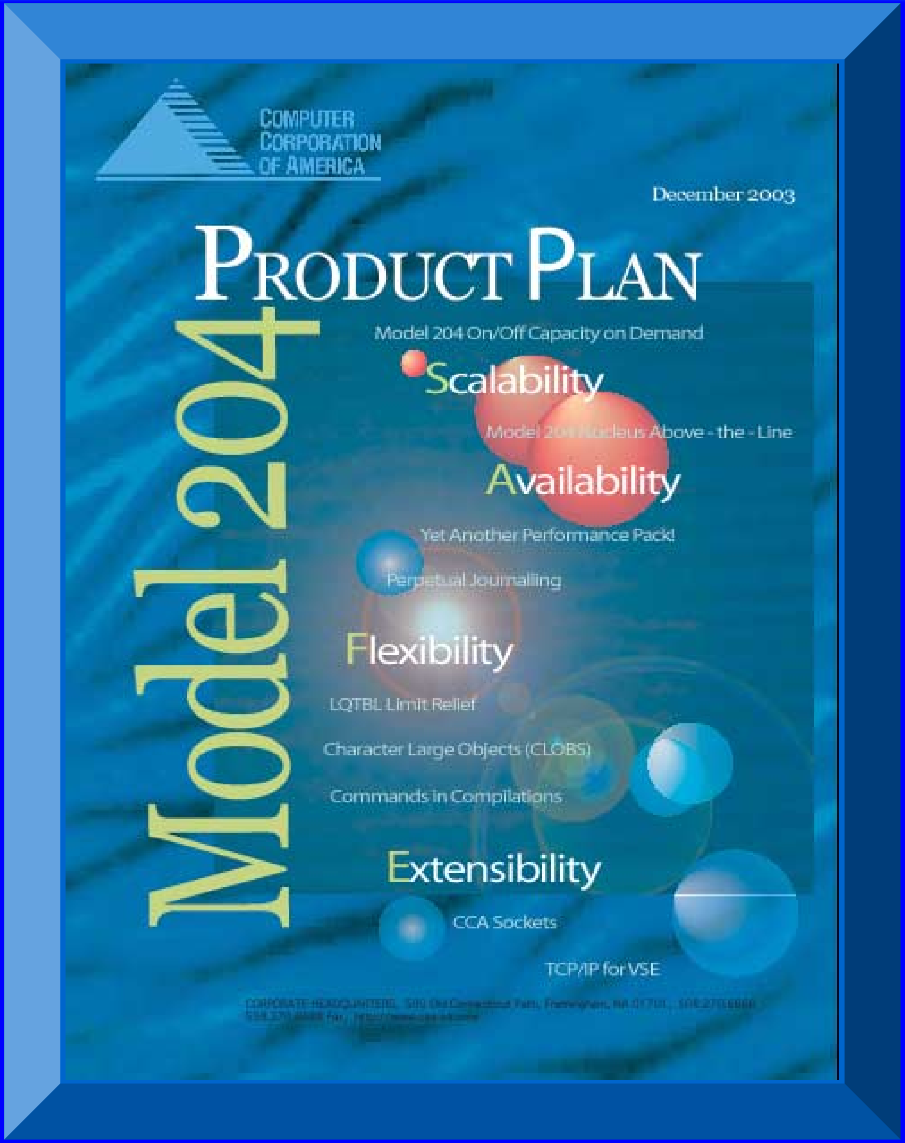
# Model 204 V6R1: Opening New Doors

Gina Scinta
Senior Technical Consultant

COMPUTER
CORPORATION
OF AMERICA

*Sirius User Group*
*Baltimore, MD*
*October 10, 2005*

# Model 204 V6R1



COMPUTER CORPORATION OF AMERICA

December 2003

Model 204

PRODUCT PLAN

Model 204 On/Off Capacity on Demand

Scalability

Model 204 Nucleus Above - the - Line

Availability

Yet Another Performance Pack!

Perpetual Journalling

Flexibility

LQTBL Limit Relief

Character Large Objects (CLOBS)

Commands in Compilations

Extensibility

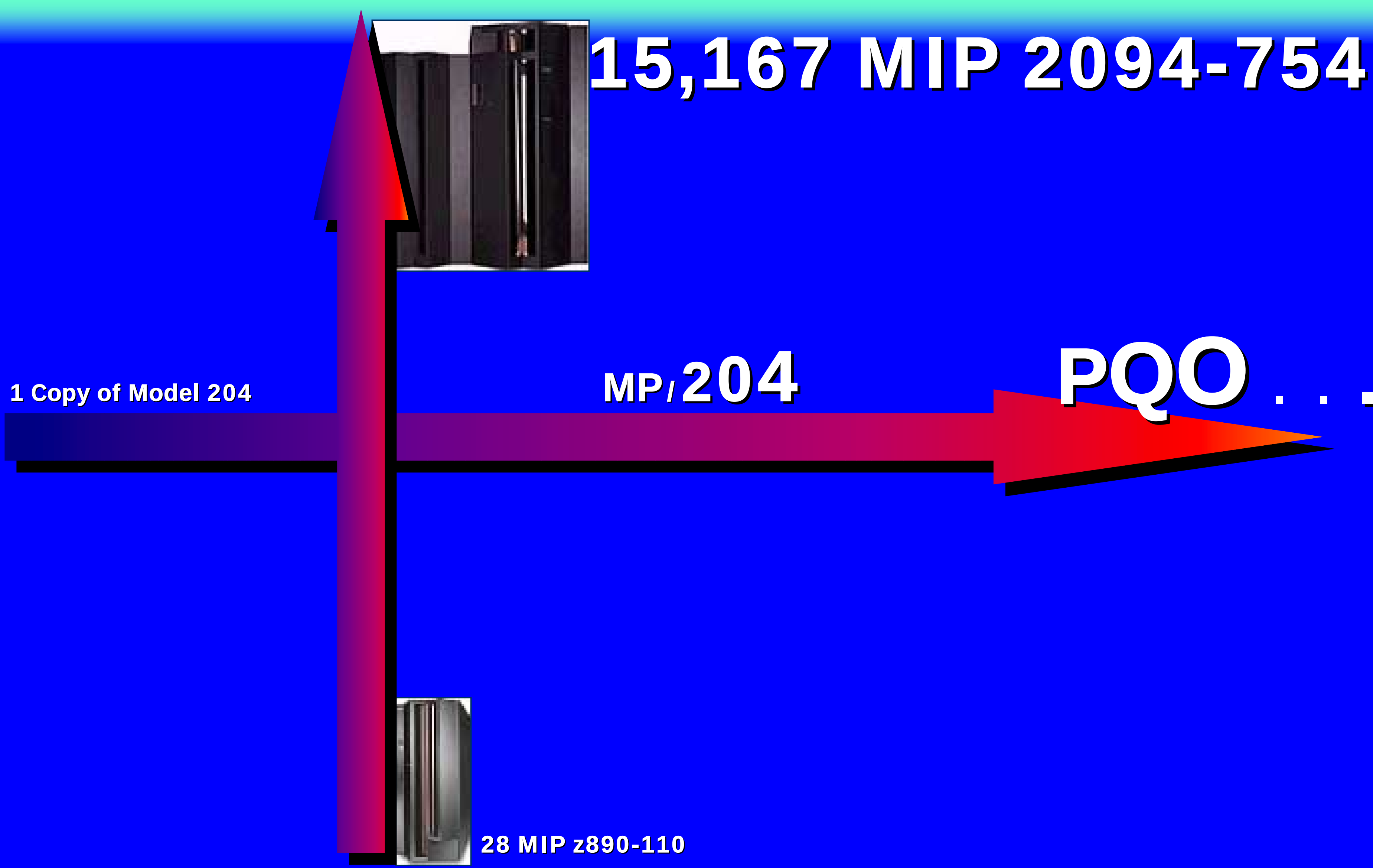
CCA Sockets

TCP/IP for VSE

CORPORATE HEADQUARTERS [illegible] Framingham, MA 01701 [illegible]
[illegible]

- **S**calability
- **A**vailability
- **F**lexibility
- **E**xtensibility

# Model 204 Scalability



15,167 MIP 2094-754

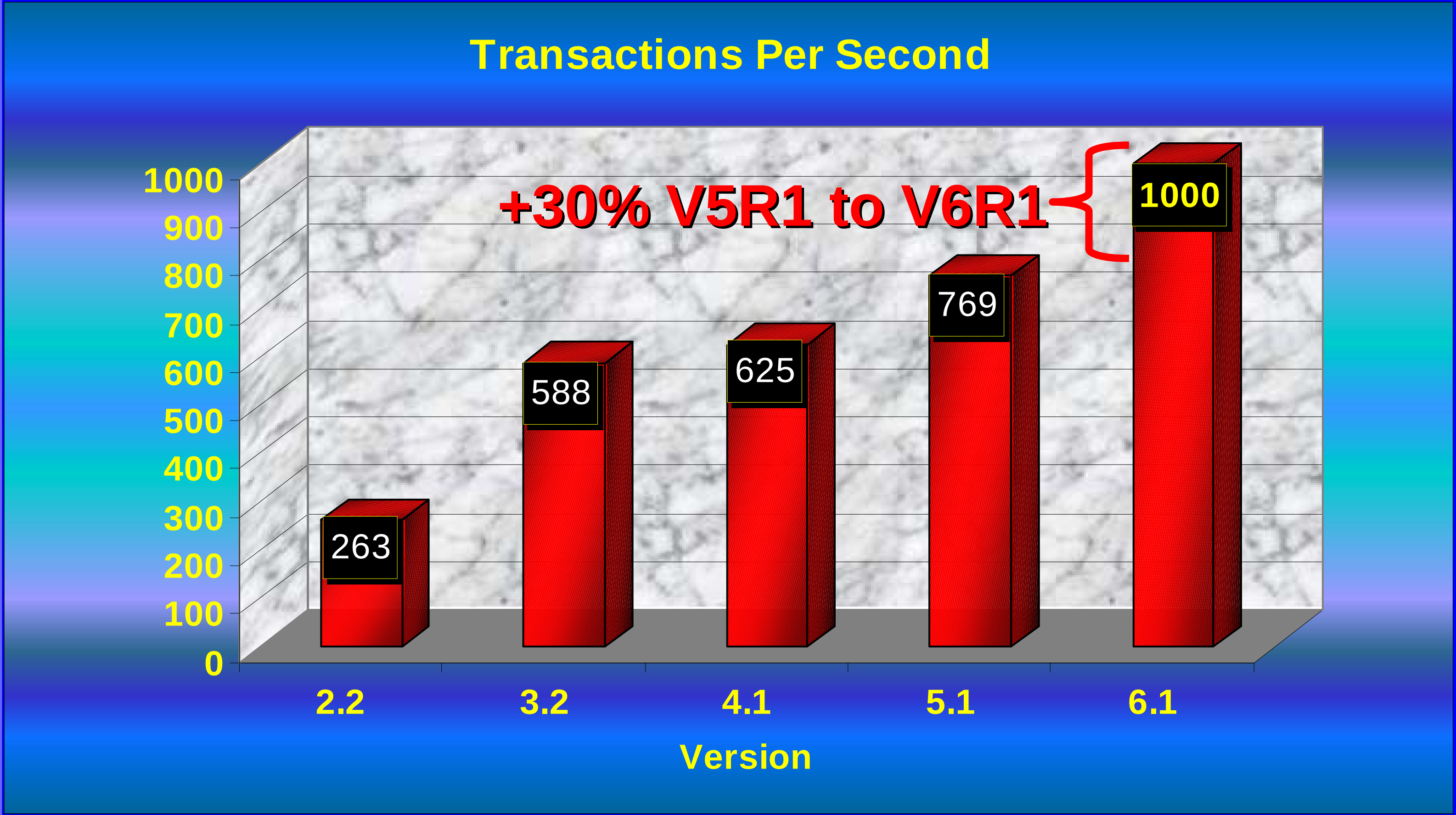
1 Copy of Model 204

MP/204

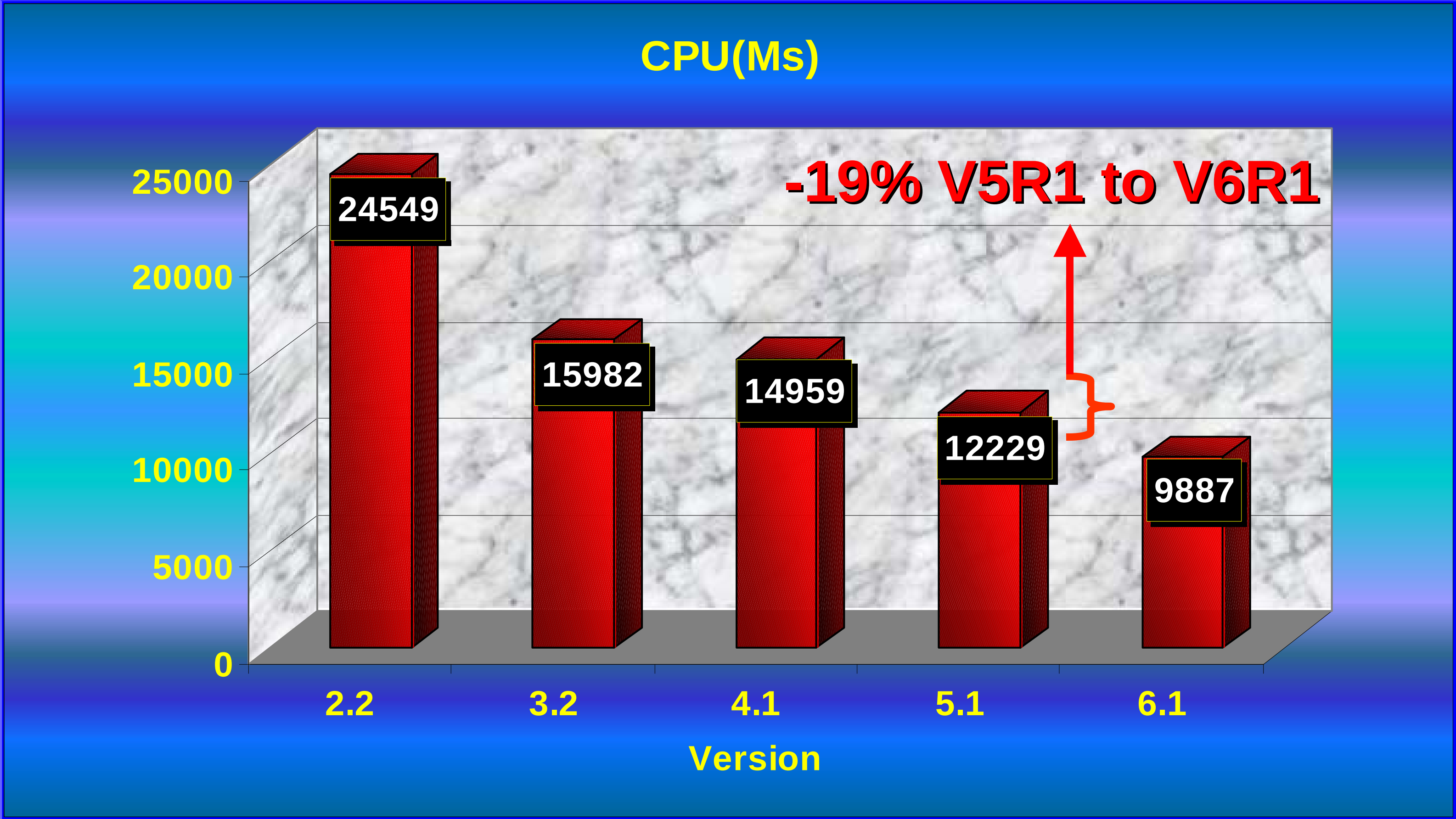
PQO . . .

28 MIP z890-110

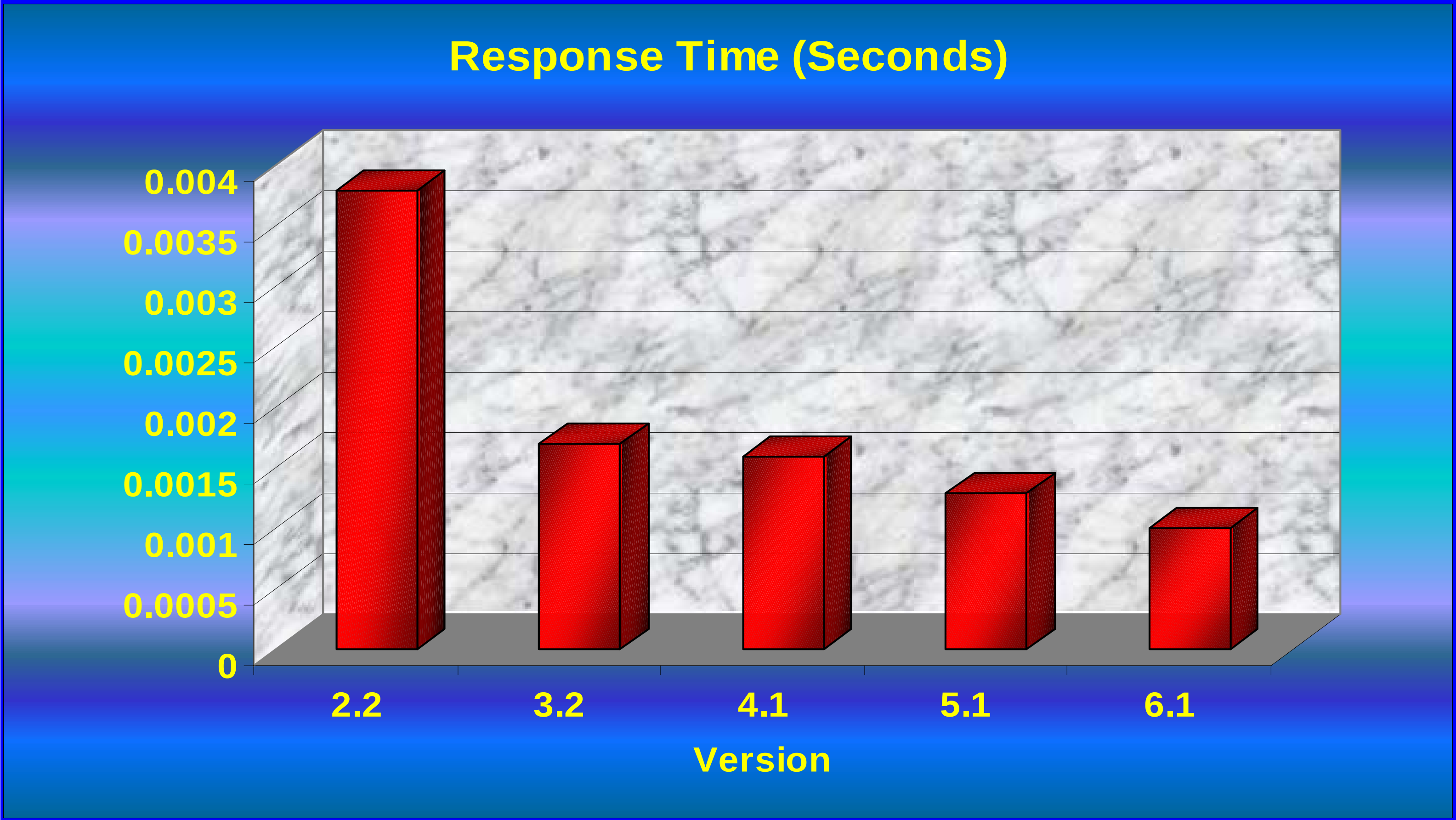
# Doing More Work …



# With Less Resource ...



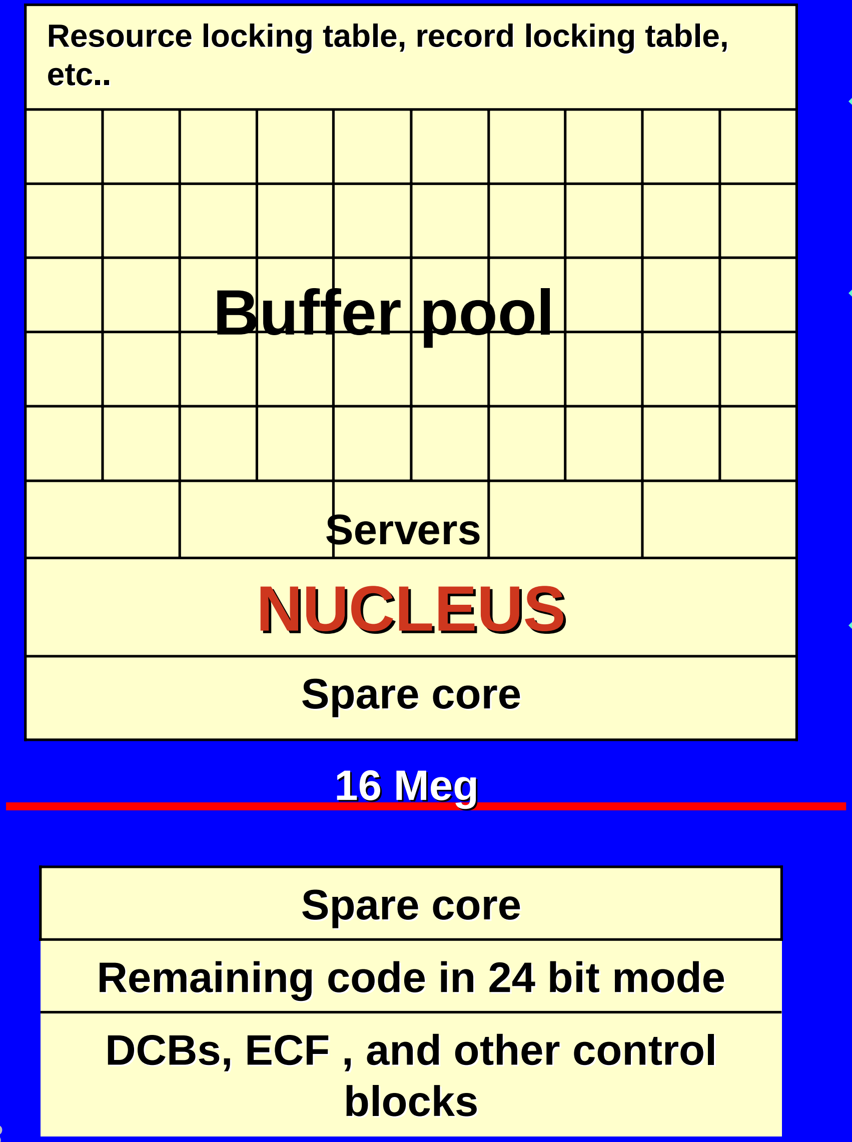
# While Providing Better Service



# Scalability

- Model 204 Nucleus Above the 31-bit Storage line
- MERGJ Enhancements
  - Performance and Multi File Support
- Reduce Journal Size
  - Separation of recovery and audit information
- Performance Pack
  - Delivering further CPU reductions “Out of the Box”
- New MP/204
  - Increased capacity delivered
  - Over 3,000 MIPS from a single region

# Nucleus Above the Line



Resource locking table, record locking table, etc..

Buffer pool

Servers

NUCLEUS

Spare core

16 Meg

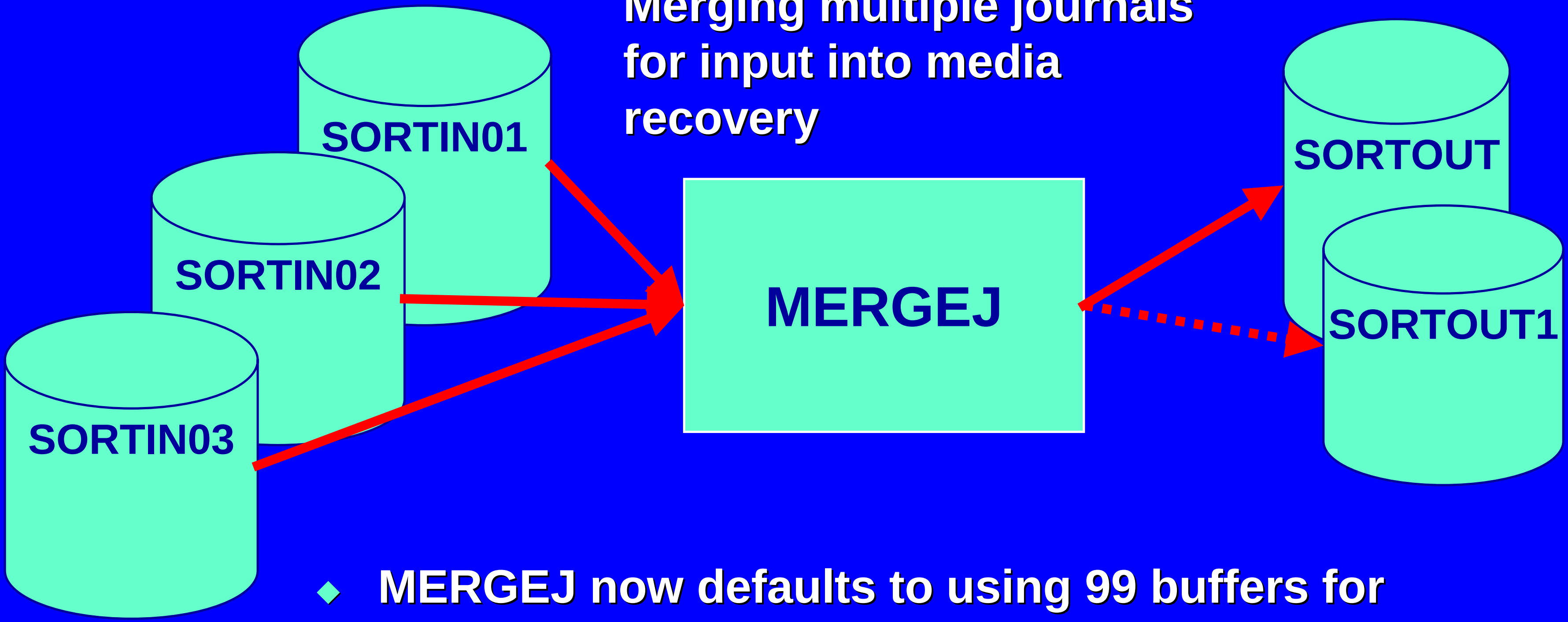
Spare core

Remaining code in 24 bit mode

DCBs, ECF , and other control blocks

- Nucleus can reside above 16 meg line (z/OS only)
- Provides “below the line” storage relief
  - Avoids potential abends due to storage shortages
- Model 204 will use 31-bit addressing in either mode
  - Up to ~ 4 Meg placed above line
    - Some remaining code below the line, e.g. VT75, ULDB

# MERGEJ Improvements

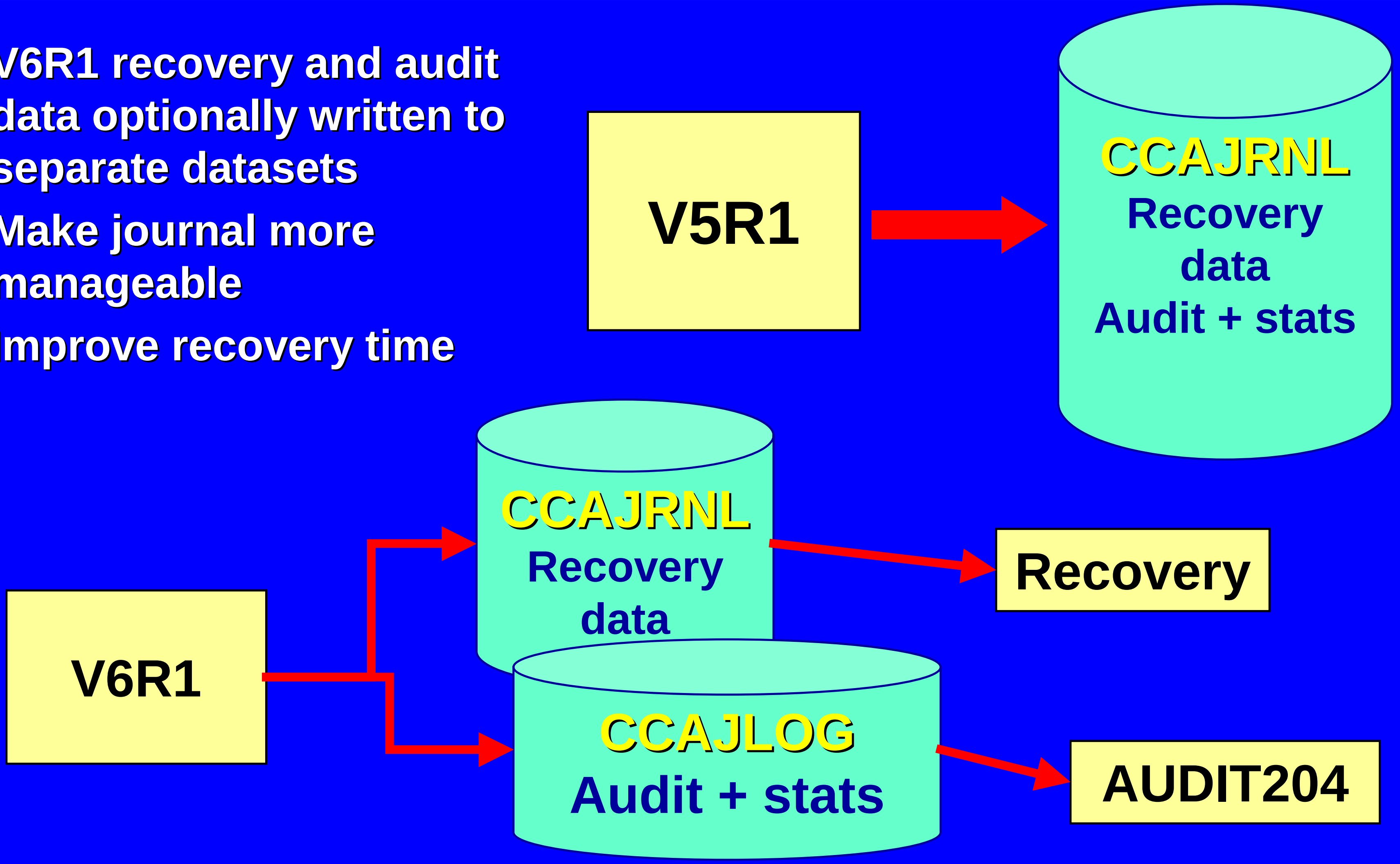


Merging multiple journals for input into media recovery

- MERGEJ now defaults to using 99 buffers for input / output processing
  - *MUCH FASTER !*
- Multiple duplicate output merged journals can be written - up to 9 SORTOUTx datasets

# Reduced Journal Size

- V6R1 recovery and audit data optionally written to separate datasets
- Make journal more manageable
- Improve recovery time



# CCAJLOG

- Can be defined as dataset or stream
  - All stream types supported:
    - Concatenate, parallel, ring or GDG
- NLBUFF used to set buffers for CCAJLOG
  - Buffers written out when full
  - Blksize can vary up to 32K
- Input file to AUDIT204

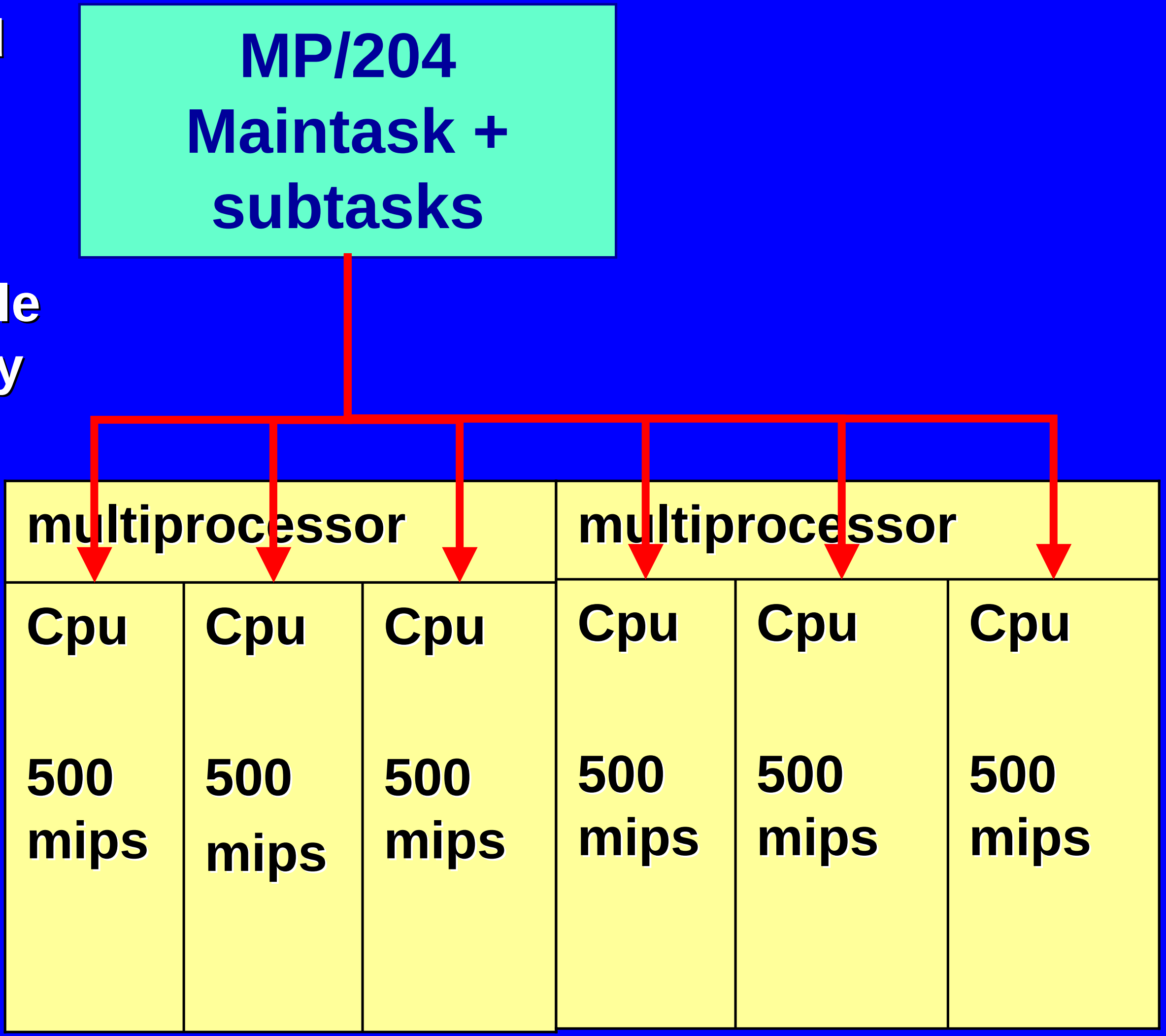
# Performance Pack

- CPU Optimization in the following areas
  - global list / foundset processing
    - Set FASTGLOB=1
  - Optimized internal subroutine transfer (CCALL)
  - Table B field extraction
    - For z/OS only
  - Processing of the TBO constraints log
    - Set CDMINP2X for minimum in-storage page allowance
    - Set CDMAXP2X for maximum in-storage page allowance
  - Scheduler improvements to improve load balancing
    - New defaults for IOSLICE and CPUSLICE

***Estimated 10% Reduction in CPU Utilization!***

# MP/204

- V6R1 will extend offload capabilities to allow equivalent of up to 6 symmetric processors
- Up to 3000 mips available for one MP enabled copy of M204



# Availability

Reliability

Availability

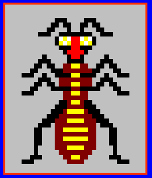
Serviceability

# Software Reliability

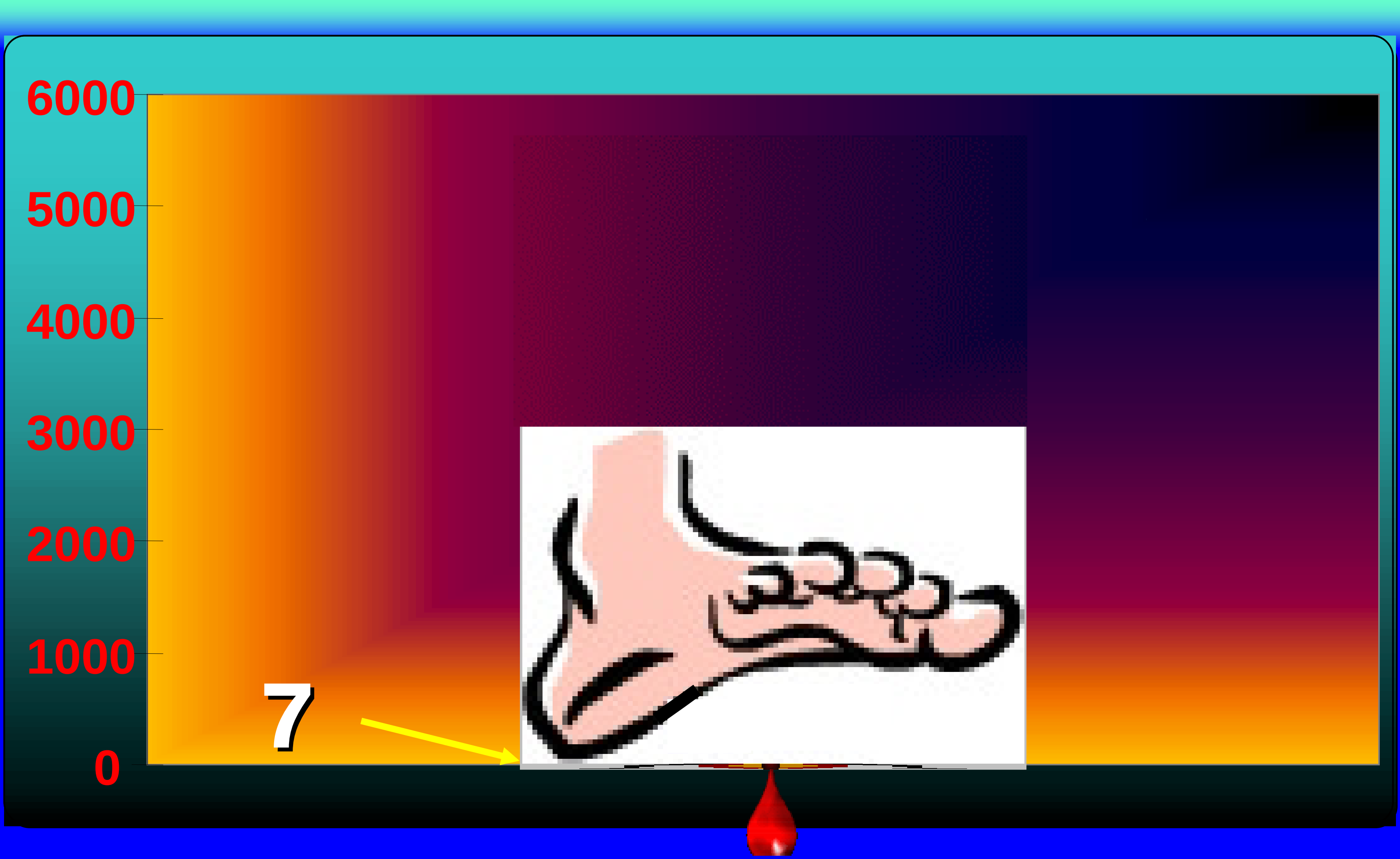
***According to most studies, the average number of software bugs per 1,000 lines of code hovers between five and 20.***

# Model 204 & Reliability?

Model 204 comprises more than 1,000,000 lines of code.

Ergo Model 204 Must Contain at least 5,000 bugs! 

# Model 204 Reliability



October 10, 2005

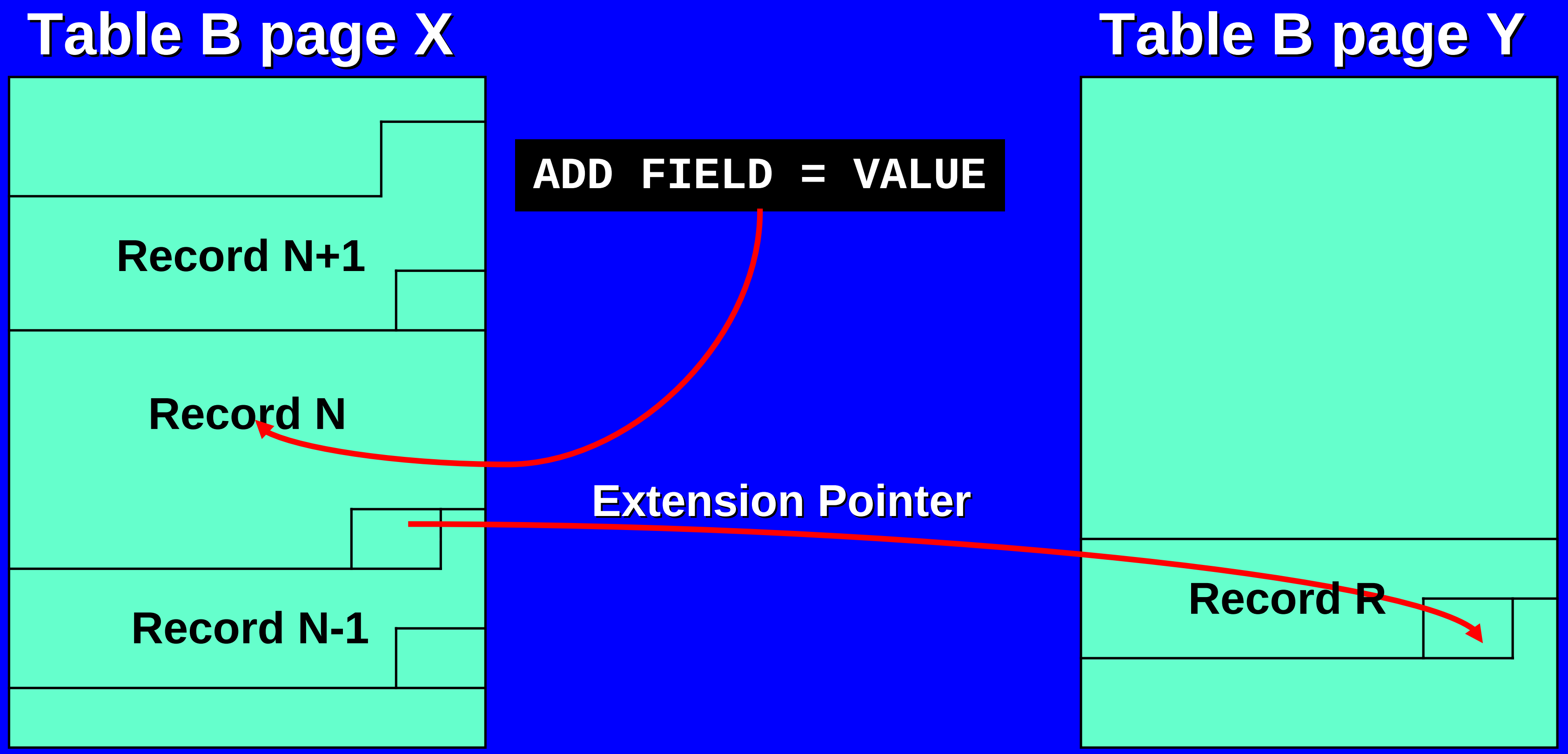
# Availability and Serviceability

## Perpetual/204

- In-flight Partial Database Reorganization
  - Table B Data Compactor
- Perpetual Journaling
  - GDG Support for Checkpoint and Journal Stream Members
- Guaranteed Checkpoint
  - Sub-transaction checkpoints
- Dynamic APSY Subsystem Support
  - Update subsystem definition attributes
  - Refresh subsystem procedures
- Dynamic Extension of Tables B and D

# Table B Data Compactor

- Extension records – why?



**Not enough space to grow record N on page X – extension record R is allocated on page Y**

# Table B Data Compactor continued

- Why extension records are bad?
  - More CPU and I/O time to process a record
  - Resources held for a longer time
  - More buffers used
  - A record number used for an extension record may not be used to store a new record

- Objective: Reduce chains of extension records in unordered files
  - Improves I/O rates to retrieve records
  - Frees up record slots with RRN

# Table B Data Compactor continued

- **COMPACTB command**
  - Reads all records, finds extensions and compacts them
  - Reduces chains of extension records in unordered files
- **Syntax:**

```
[IN file_name] COMPACTB [FROM ssss]
[TO eeee] [FREE nn] [MAXE nn]
```

## Parameters:

FROM ssss - Starting Table B page number, default is 0

TO eeee - Ending Table B page number, default is the last record

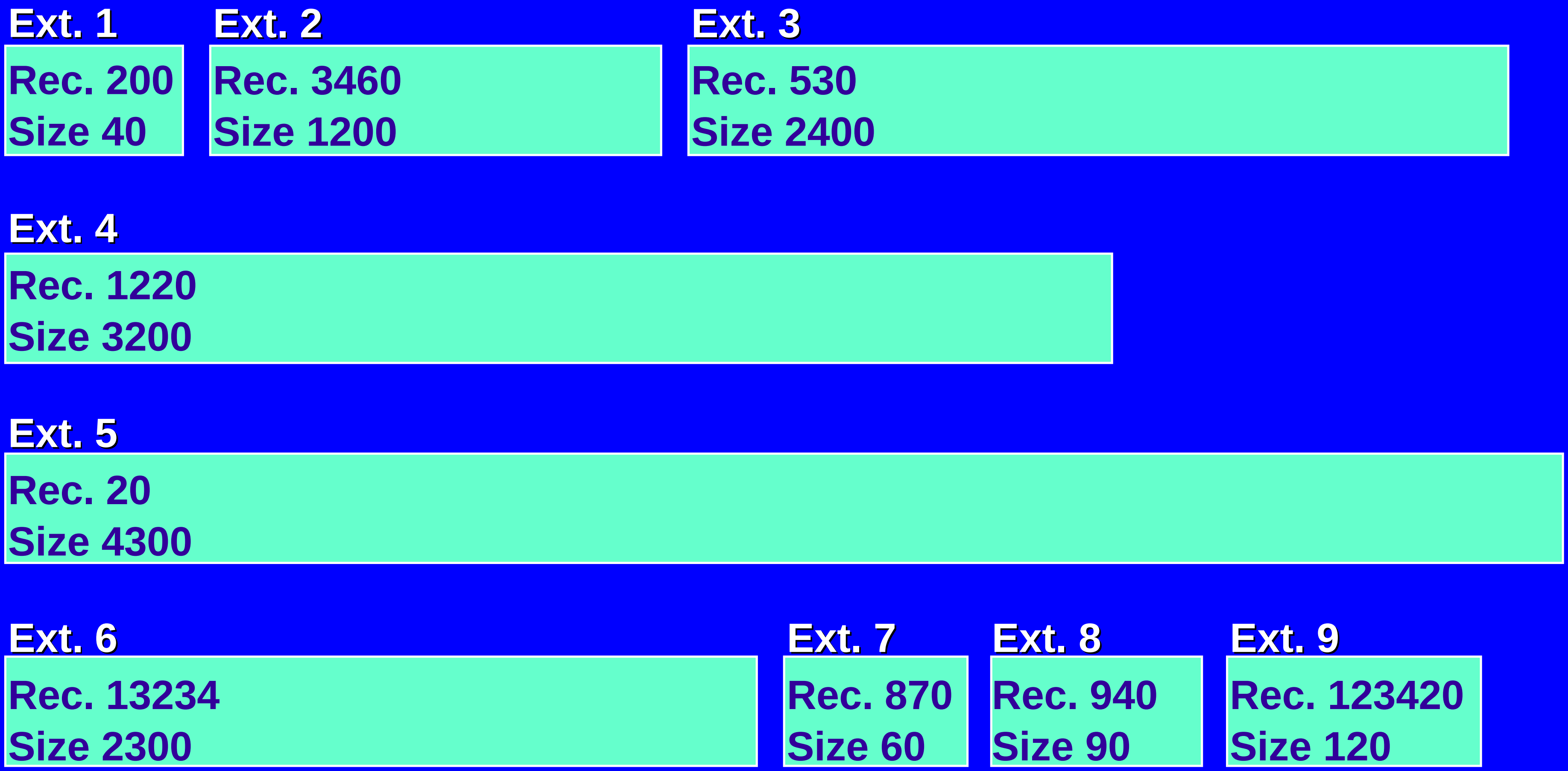
FREE nn - Percentage of unused Table B pages to be used for compaction

MAXE nn - Percentage of page size, maximum extension record size eligible for compaction, default 80%

# Table B Data Compactor continued

## Example: How extensions are compacted

## Record 100



Ext. 1

Rec. 200
Size 40

Ext. 2

Rec. 3460
Size 1200

Ext. 3

Rec. 530
Size 2400

Ext. 4

Rec. 1220
Size 3200

Ext. 5

Rec. 20
Size 4300

Ext. 6

Rec. 13234
Size 2300

Ext. 7

Rec. 870
Size 60

Ext. 8

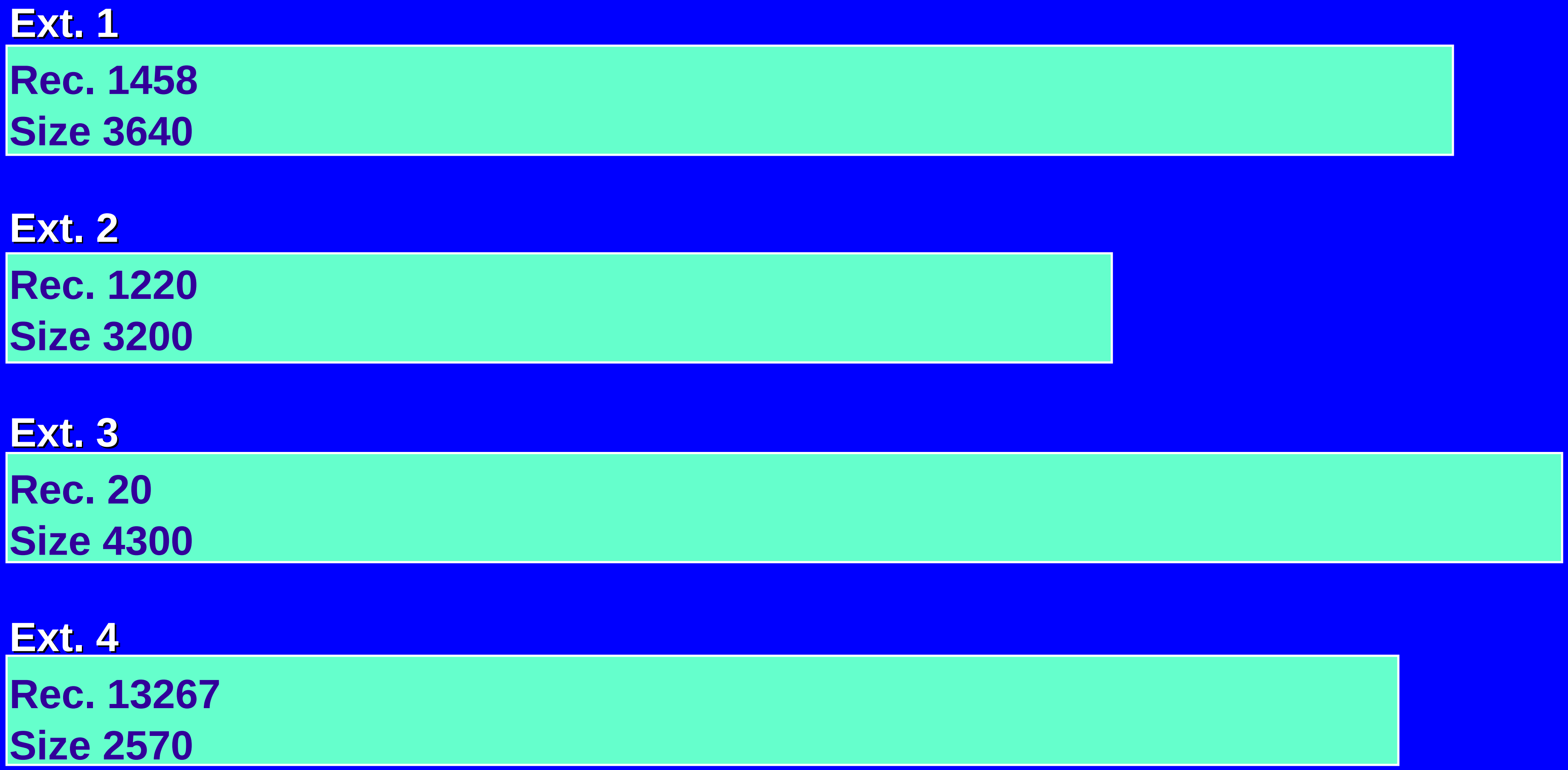
Rec. 940
Size 90

Ext. 9

Rec. 123420
Size 120

# Table B Data Compactor continued

## Results: How extensions are compacted: Record 100 after compaction



Ext. 1

Rec. 1458
Size 3640

Ext. 2

Rec. 1220
Size 3200

Ext. 3

Rec. 20
Size 4300

Ext. 4

Rec. 13267
Size 2570

# BLDREUSE Command

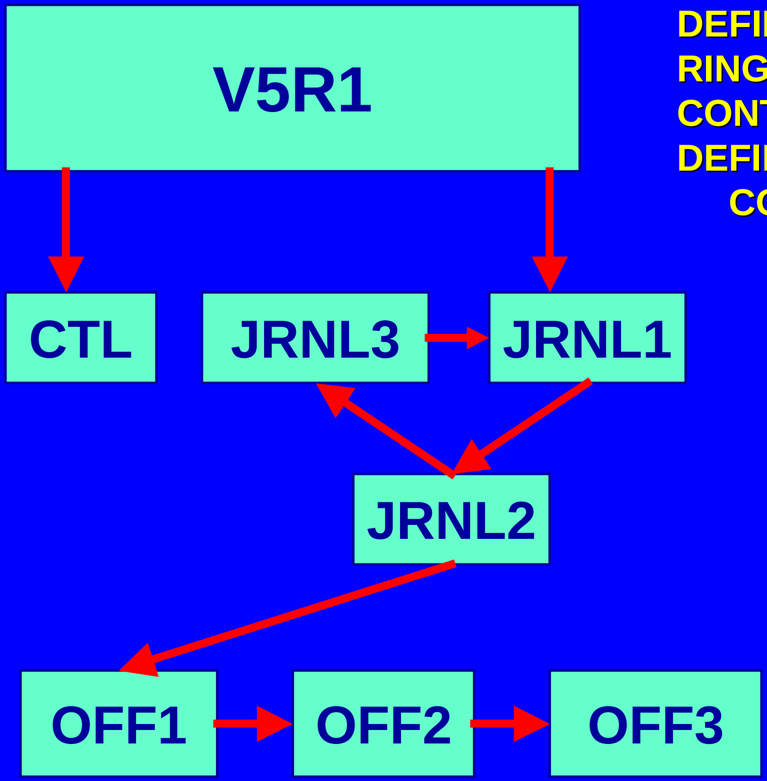
- BLDREUSE – the reuse queue maintenance command
- Two forms:
  - BLDREUSE NEW – to discard the existing queue and build the new one
    - Rebuild the reuse queue after message 1228 “BUG REUSE QUEUE ERROR’
    - Rebuild the reuse queue to remove ineligible pages from the queue
  - BLDREUSE [FROM nn] [TO nn] – to scan all or some records to find pages eligible for reuse
    - Add pages to the reuse queue

# Perpetual Journaling

- z/OS only
- Implements Generation Data Groups (GDGs)
- Virtually unlimited space
- Control file keeps track of active files
- No recovery changes

# Perpetual Journaling continued

## Current ring stream configuration

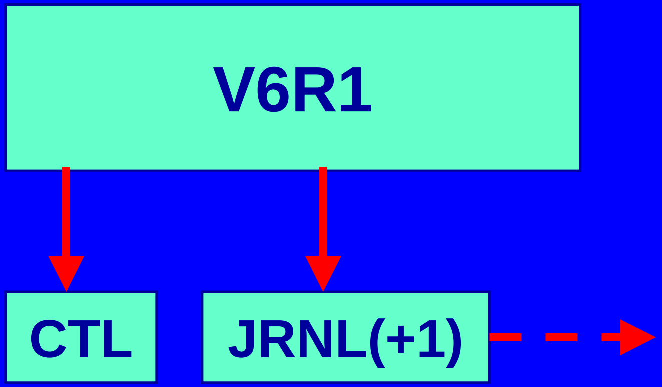


```
DEFINE STREAM CCAJRNL WITH SCOPE=SYSTEM -
RING=(JRNL1, JRNL2, JRNL3) OFFLOAD=OFFTCUR
CONTROL=CTL AUTOOFFLOAD=2 CLOSE=AUTO
DEFINE STREAM  OFFTCUR WITH SCOPE=SYSTEM -
     CONCATENATE = (OFF1, OFF2, OFF3)
```

- Offload stream finite
- Potential for filling offload stream under stress conditions

# GDG Stream

## GDG stream definition …



```
DEFINE STREAM CCAJRNL WITH = -
SCOPE=SYSTEM CONTROL=CTL  -
GDG = JRNL
```

- Removes need for offload stream
  - Avoid potential filling of either ring or offload stream
- As one member fills, next (+1) member is opened
- Control file opened in recovery to track which G000V00 number we should use
- Alternatively, can still define journal as ring, and have offload as GDG
  - Removing need to define finite number of offload members

# Transaction Checkpoints

- What is a transaction checkpoint?
  - Marks point where database is quiescent
  - Is usually ending point for ROLL BACK and starting point for ROLL FORWARD recovery
- What does it mean to “Take a Checkpoint”?
  - No update unit in progress
  - Checkpoint record containing date/time stamp written to: CHKPOINT, CCAJRNL and deferred update datasets
  - Initiated:
    - Automatically at intervals based on CPTIME parameter
    - By user via CHECKPOINT command

# Transaction Checkpoints continued

- Why can’t a transaction checkpoint be taken?
  - When one or more users are in the middle of an update unit transaction

- How do we solve the problem?

***Sub-transaction Checkpoint***

# Sub-transaction Checkpoints

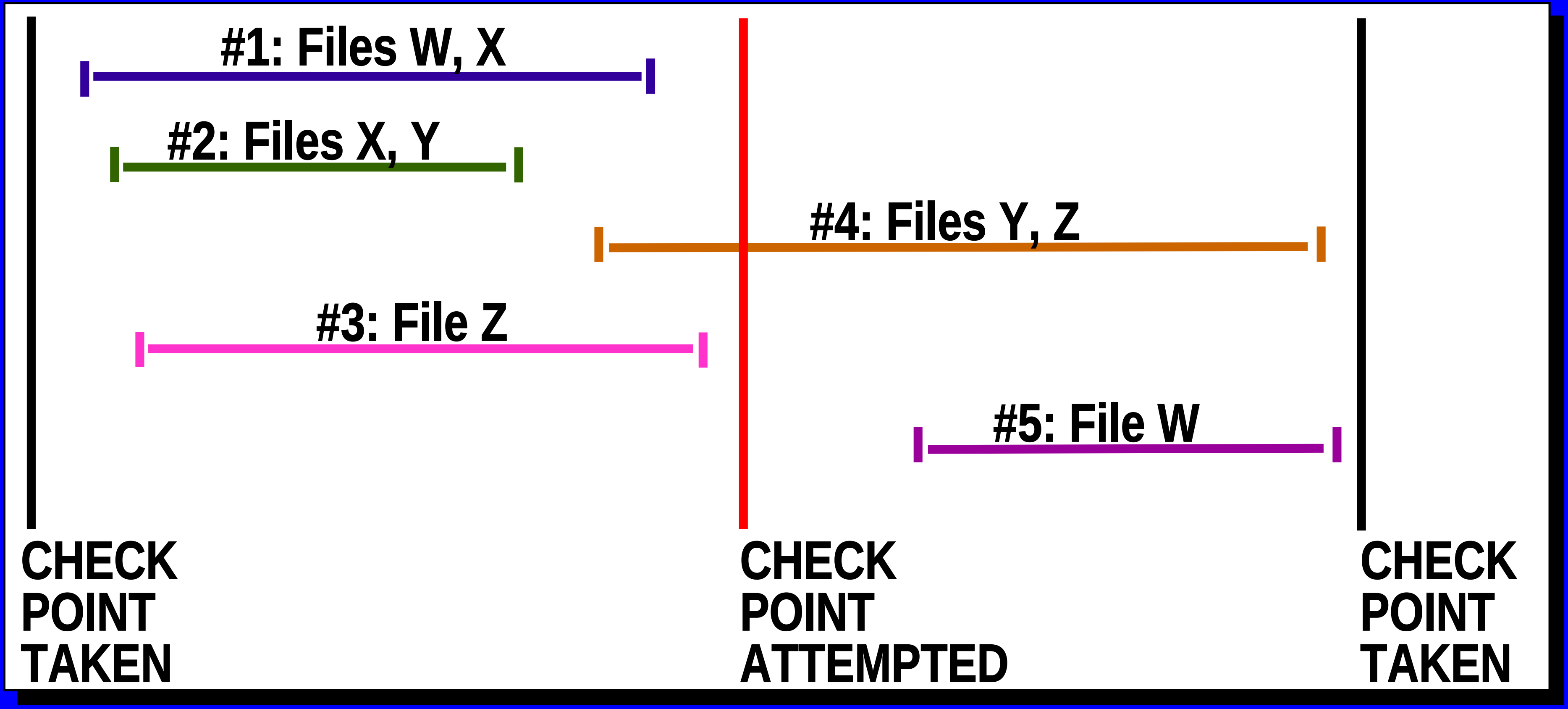
- What is a sub-transaction checkpoint?
  - Checkpoint type taken while updating transactions are in progress and uncommitted
- Eliminates checkpoint timeout situations
- Enabled with CPTYPE=1 parameter
- Additional dataset, CHKPNTS must be defined to job
  - Sized comparable to CHKPOINT dataset

# How It Works in V5R1?

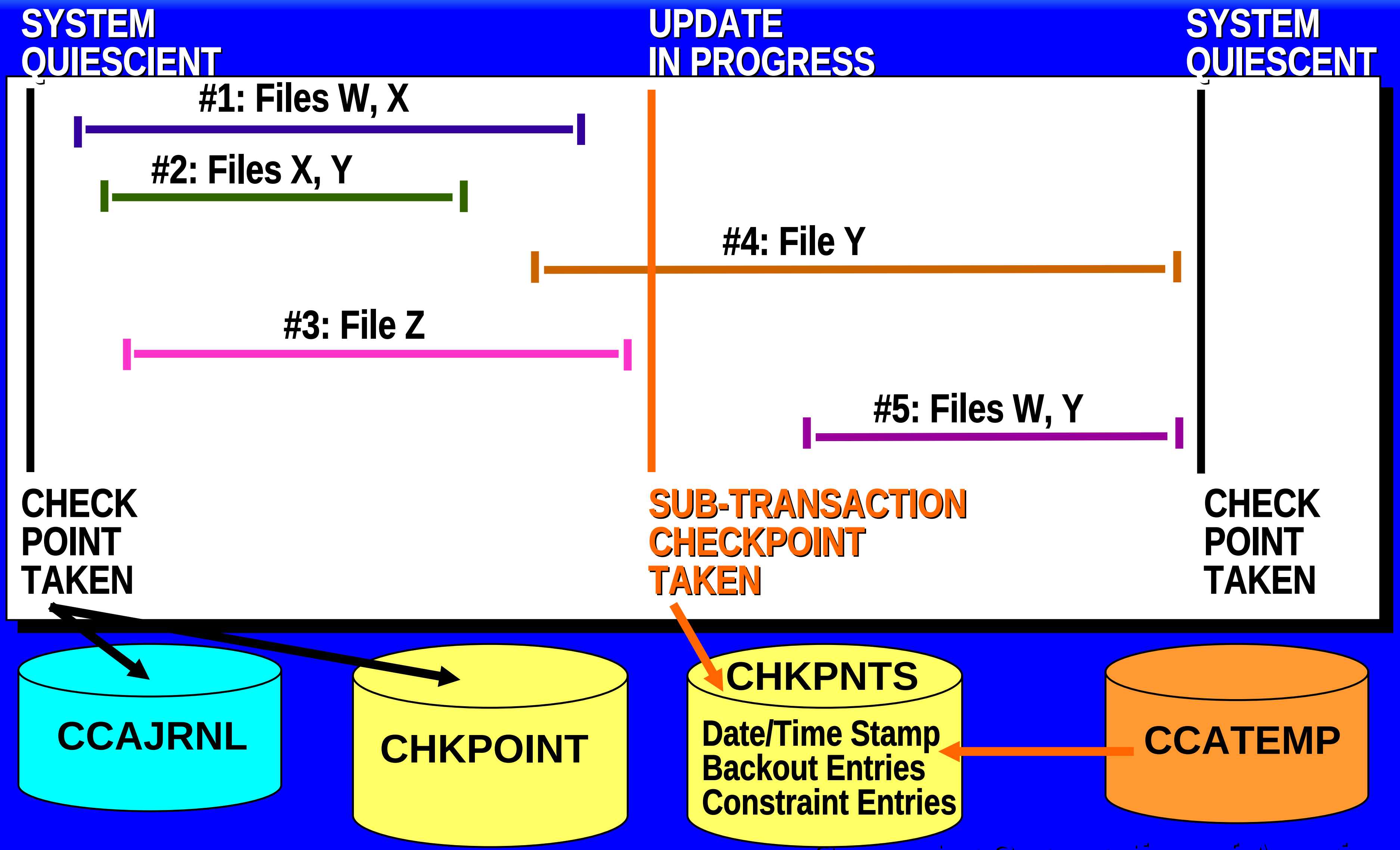
SYSTEM QUIESCIENT

UPDATE IN PROGRESS

SYSTEM QUIESCENT

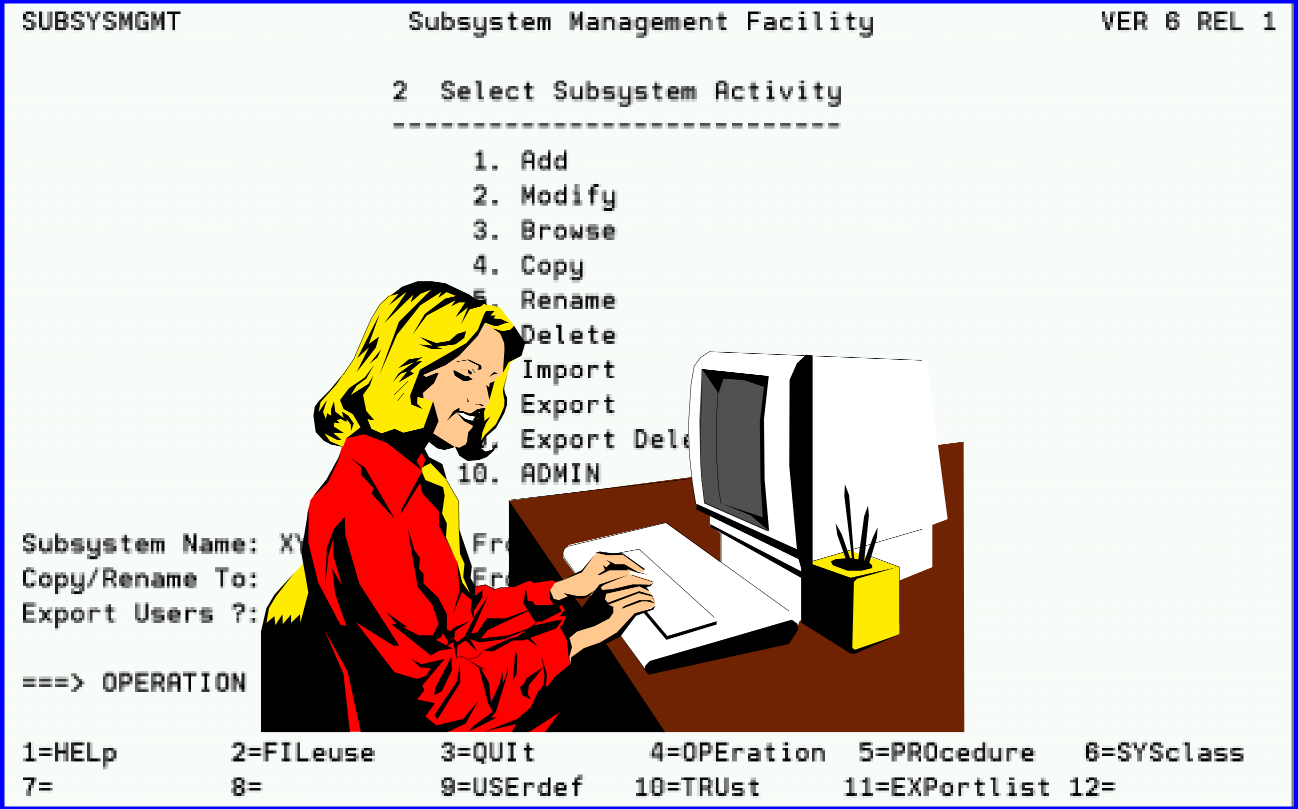


# Sub-transaction Checkpoints



# APSY Changes

- Dynamic APSY attribute changes
- Dynamic refresh of procedure compilations



```
SUBSYSMGMT              Subsystem Management Facility             VER 6 REL 1

                         2  Select Subsystem Activity
                         ----------------------------
                              1.  Add
                              2.  Modify
                              3.  Browse
                              4.  Copy
                              5.  Rename
                                 Delete
                                 Import
                                 Export
                                 Export Dele
                             10.  ADMIN

Subsystem Name:  XY              Fr
Copy/Rename To:                  Fr
Export Users ?:

===> OPERATION


1=HELp        2=FILeuse     3=QUIt       4=OPEration   5=PROcedure    6=SYSclass
7=            8=            9=USErdef   10=TRUst      11=EXPortlist  12=
```

# Dynamic APSY Attribute Changes

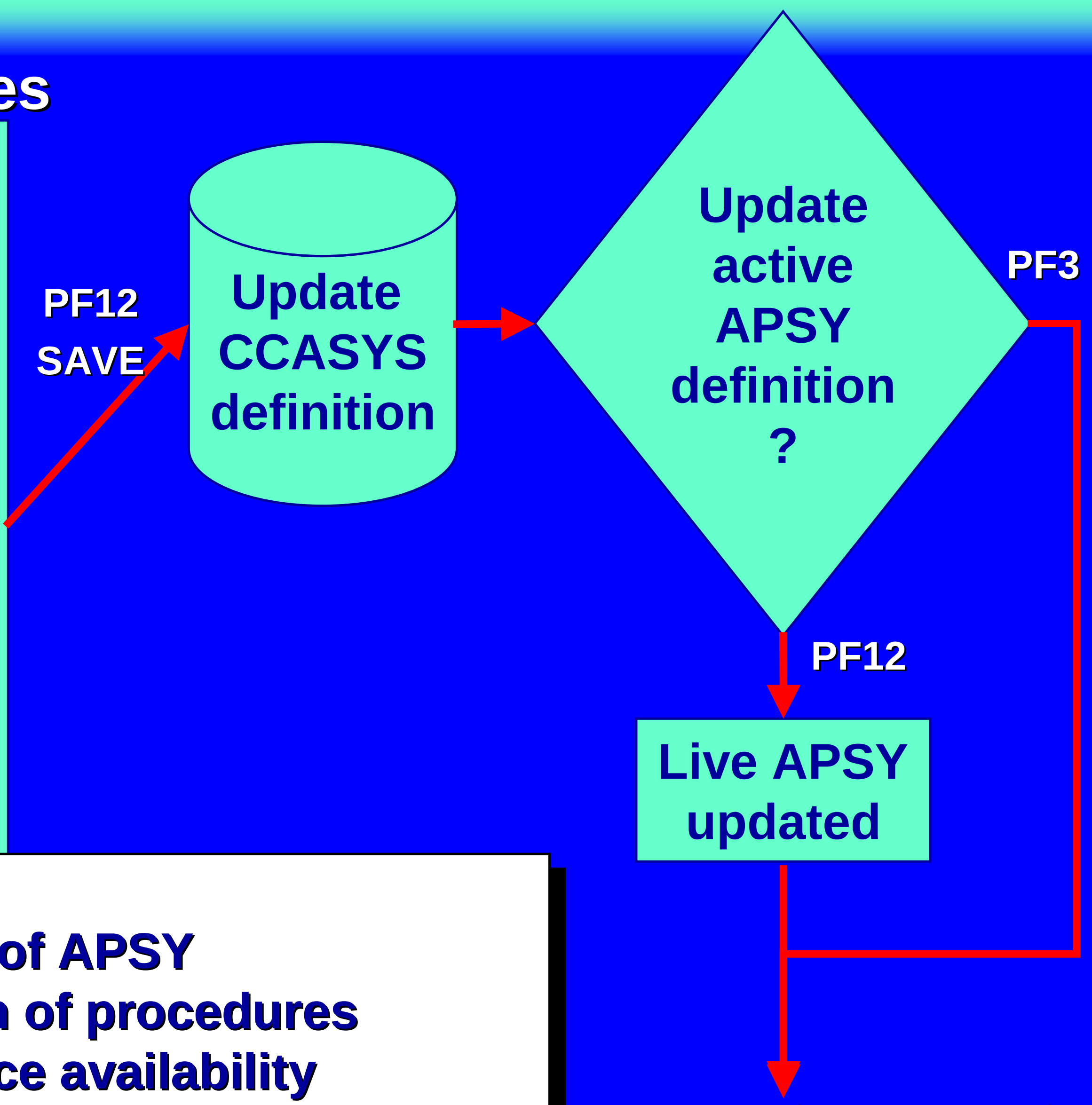
## SUBSYSMGMT Changes

**Operational Parameters**

- Login into Model 204
- Logout of Model 204
- Auto Commit
- Maximum Iterations
- Account
- Disconnect messages
- Informational messages
- Error messages

**Procedure Attributes**

- Login proc
- Error proc



PF12

SAVE

Update CCASYS definition

Update active APSY definition ?

PF3

PF12

Live APSY updated

- **Avoid**
  - **STOP/START of APSY**
  - **Recompilation of procedures**
  - **Improve service availability**

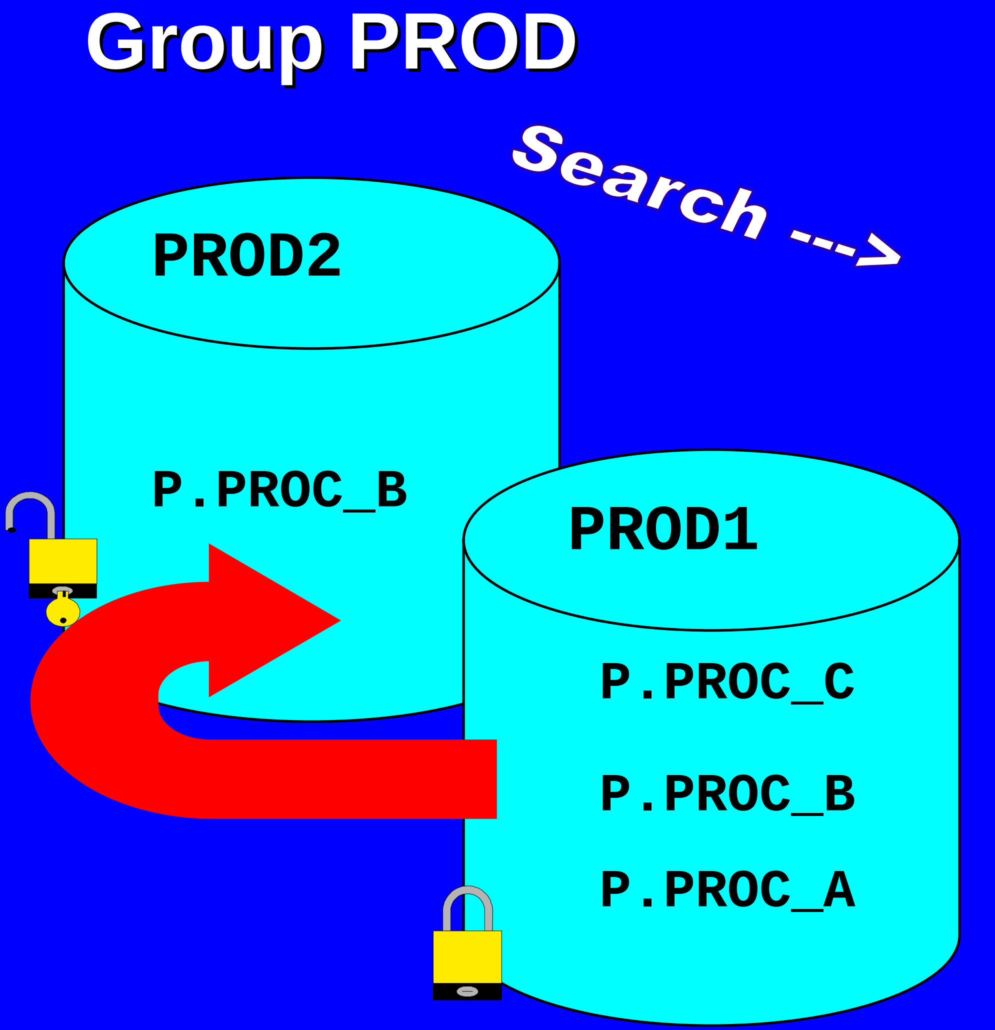
# Changing the Subsystem ... Then

## Prior to V6R1

- Use procfile groups
- Move procedure to unlocked member and change

Note:

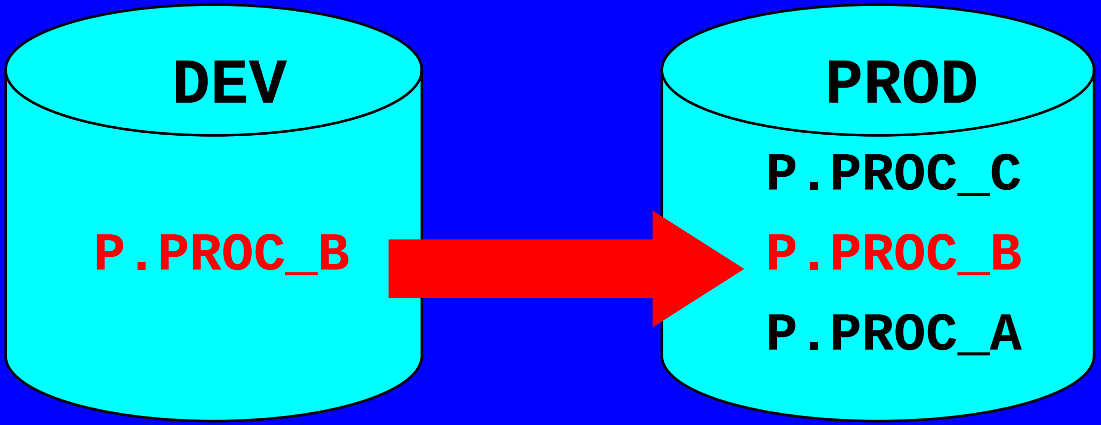
- Unlocked procedure always recompiled



# Changing the Subsystem ... Now!

## Model V6R1

- Refreshes procedures dynamically
- Discards the existing precompilation in CCATEMP
- Precompiles next time through for each SYSCLASS



# REFRESH SUBSYSPROC Command

## Syntax:

```
REFRESH SUBSYSPROC procname –
   IN [ GROUP | FILE] name –
   [ FROM –
   [ {[PERM|TEMP] GROUP} | FILE ] -
   name ]
```

- Copies procedure
  - From FROM filename
  - To IN filename
- Updates APSY in-memory, procedure dictionary
  - Flag procedure as no longer compiled
  - Release CCATEMP/CCAAPSY pages
- Recompiles *procname* when next invoked

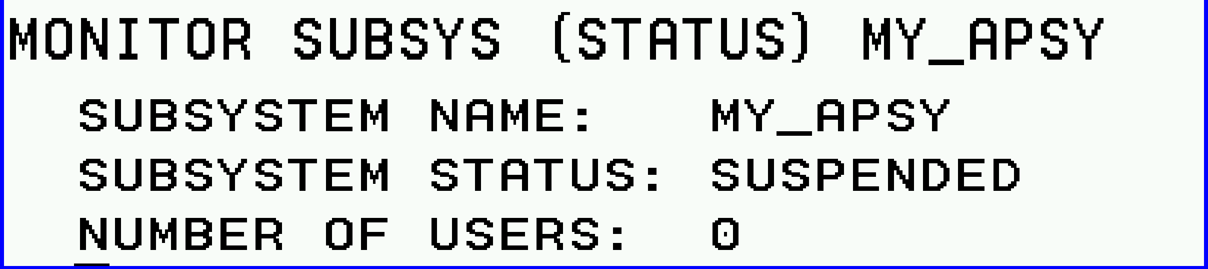
# SUSPEND SUBSYSTEM Command

```
SUSPEND SUBSYS MY_APSY
```

Allows time to refresh a set of logically related procedures

- Preserves precompiled procedures

```
M204.2661: SUBSYSTEM MY_APSY SUSPENDED
-or-
M204.2659: MY_APSY IS SET TO SUSPEND,
REMAINING USERS = <n>
```



```
MONITOR SUBSYS (STATUS) MY_APSY
  SUBSYSTEM NAME:    MY_APSY
  SUBSYSTEM STATUS:  SUSPENDED
  NUMBER OF USERS:   0
```

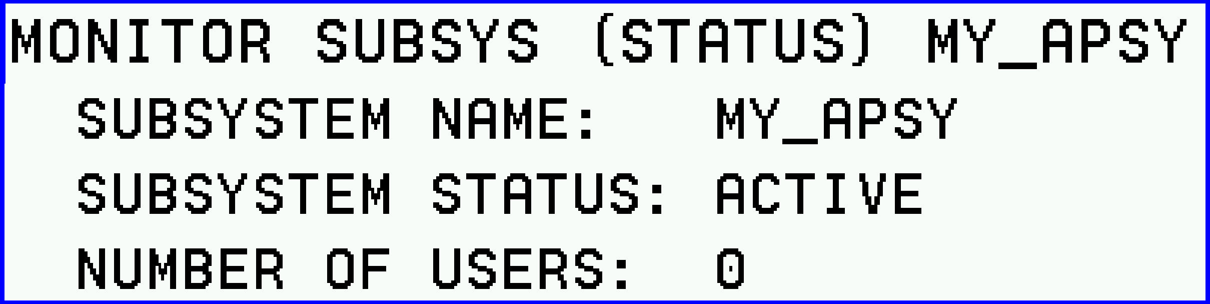
# RESUME SUBSYSTEM Command

```
RESUME SUBSYS MY_APSY
```

Resumes operation of the suspended subsystem

- Subsystem set to Active state

```
M204.2657: SUBSYSTEM MY_APSY RESUMED
```



```
MONITOR SUBSYS (STATUS) MY_APSY
  SUBSYSTEM NAME:    MY_APSY
  SUBSYSTEM STATUS:  ACTIVE
  NUMBER OF USERS:   0
```

# Automatic Increase

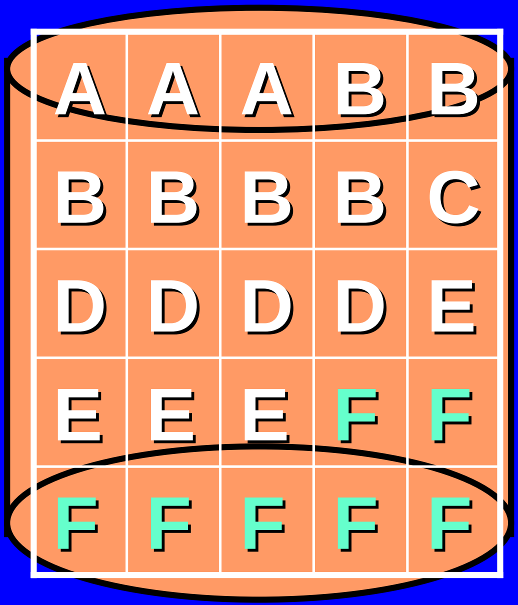
- Auto increase of Tables B and D
  - Occurs when file is opened or when Table B or D becomes full
  - Works similar to the INCREASE TABLEB or TABLED commands
    - Increases a table by adding pages from the free space
  - Controlled by new parameters:
    - BAUTOINC
    - BAUTOTYP
    - DAUTOINC
  - Pages come from FREESIZE
  - For files created in V6R1 only

# Automatic Increase continued

- Parameters
  - BAUTOINC – amount of unused pages that must be maintained in Table B
    - Unused pages are pages after the highest used page (BSIZE-BHIGHPG)
  - BAUTOTYP – Table B auto increase options
    - x’01’ - allow Table B auto increase when table becomes full
    - x’02’ – maximize number of pages available for increase when table becomes full
      - Used when file is opened to add more pages if required

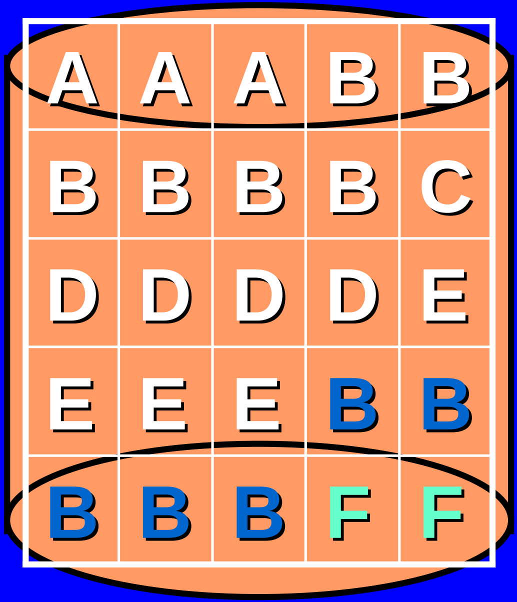
# Automatic Increase Table B

HISTORY



|   |   |   |   |   |
|---|---|---|---|---|
| A | A | A | B | B |
| B | B | B | B | C |
| D | D | D | D | E |
| E | E | E | F | F |
| F | F | F | F | F |

HISTORY



|   |   |   |   |   |
|---|---|---|---|---|
| A | A | A | B | B |
| B | B | B | B | C |
| D | D | D | D | E |
| E | E | E | B | B |
| B | B | B | F | F |

BAUTOINC=5
BAUTOTYP=3

Increase occurs at

1. OPEN HISTORY (physical open)
   Is (BSIZE – BHIGHPG) LT BAUTOINC?

   or

2. TABLEB FULL during EVAL

# Automatic Increase continued

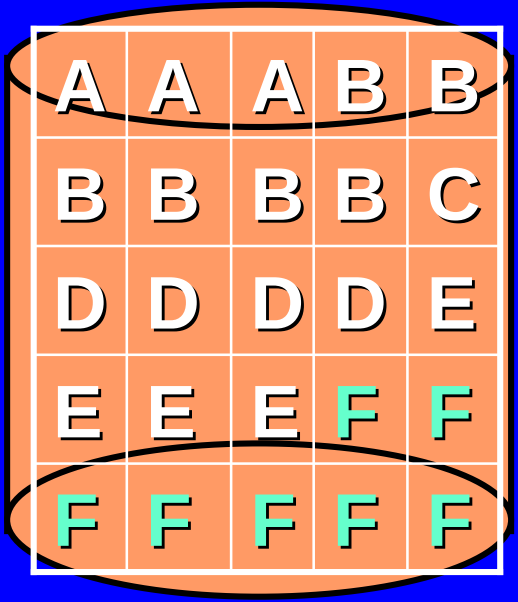
- Auto increase of Table D
  - DAUTOINC - Number of unused pages in addition to reserved pages (DPGSRES) that must be maintained in Table D
  - Unused pages in Table D
    - DSIZE - DPGSUSED
  - Takes place when file is physically opened or when Table D becomes full
    - Table D is increased by the same amount in both cases
  - No messages issued when Table D becomes full and automatically increased

# Automatic Increase continued

- Auto increase of Table D
  - When file is opened Table B is increased first, if FREESPACE is exhausted by the Table B auto increase then no pages are left for Table D auto increase
    - Corresponding message is issued
  - Table D increase is limited by available pages in the FREESPACE
  - If free space is empty after an increase a message is issued
    - Run INCREASE DATASET or DECREASE TABLE command to add pages to the free space

# Automatic Increase Table D

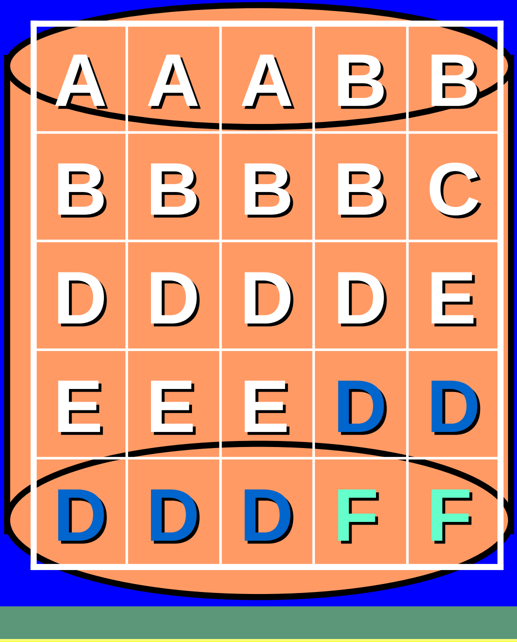
HISTORY



|   |   |   |   |   |
|---|---|---|---|---|
| A | A | A | B | B |
| B | B | B | B | C |
| D | D | D | D | E |
| E | E | E | F | F |
| F | F | F | F | F |

DAUTOINC=5

HISTORY



|   |   |   |   |   |
|---|---|---|---|---|
| A | A | A | B | B |
| B | B | B | B | C |
| D | D | D | D | E |
| E | E | E | D | D |
| D | D | D | F | F |

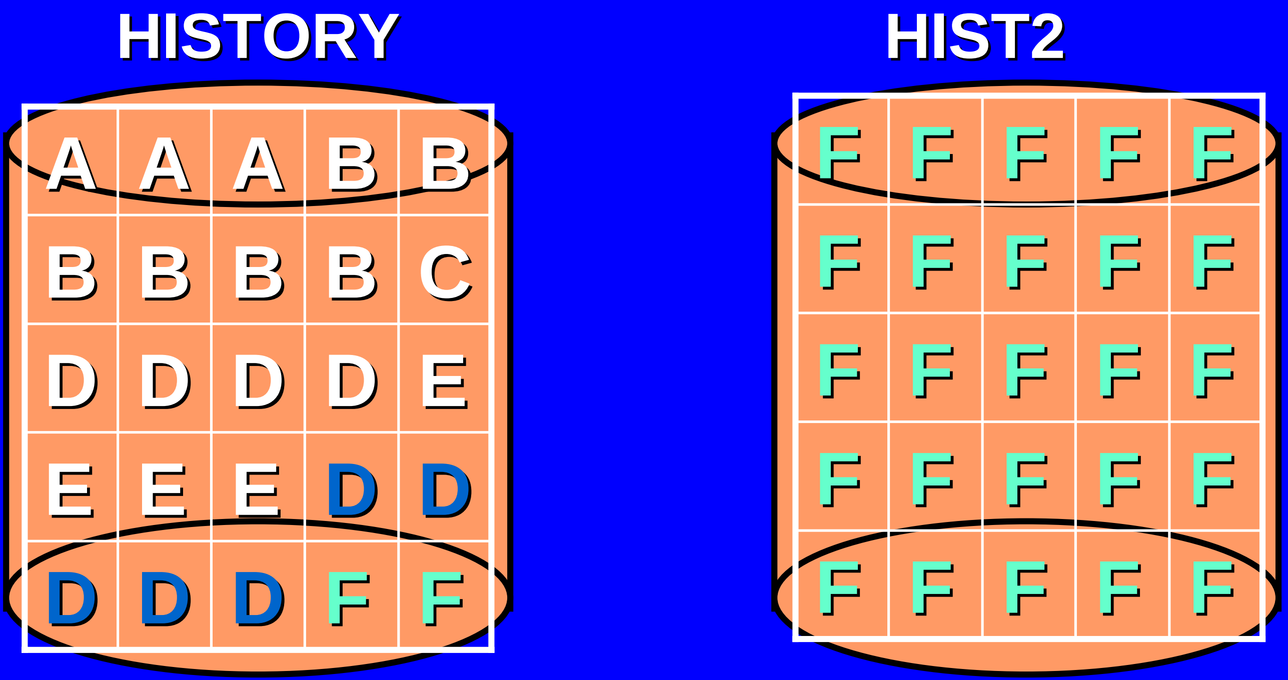
Increase occurs at

1. OPEN HISTORY (physical open)
   Is (DSIZE – DHIGHPG)
   LT (DAUTOINC + DPGSRES) ?
   or
2. TABLED FULL during EVAL

# INCREASE  DATASETS Command

## Need more pages in FREESIZE?

```
OPEN HISTORY
INCREASE DATASETS WITH HIST2
```



HISTORY

| | | | | |
|---|---|---|---|---|
| A | A | A | B | B |
| B | B | B | B | C |
| D | D | D | D | E |
| E | E | E | D | D |
| D | D | D | F | F |

HIST2

| | | | | |
|---|---|---|---|---|
| F | F | F | F | F |
| F | F | F | F | F |
| F | F | F | F | F |
| F | F | F | F | F |
| F | F | F | F | F |

# Flexibility

- INITIALIZE KEEPDEFS Command
  - Preserving field definitions
- CCASTAT Enhanced LOGON Security
  - Password expiration and purging old entries
- Large Object Support
  - New data types – CLOB, BLOB
- Enhanced RENAME FILE Command

# INITIALIZE KEEPDEFS

- INITIALIZE Command Currently Removes…
  - Field definitions and field values (FRV, CODED) from Table A
  - Record data from Table B
  - Index entries from Tables C and D

- New option with V6R1
  - INITIALIZE KEEPDEFS
    - Will preserve field definitions on Table A attribute pages
    - Will check that HASH , SORT and RECORD SECURITY keys are already defined
    - All other functionality the same

M204.0764 END INITIALIZATION: FILE filename FIELD DEFINITIONS PRESERVED

# CCASTAT Password Expiration

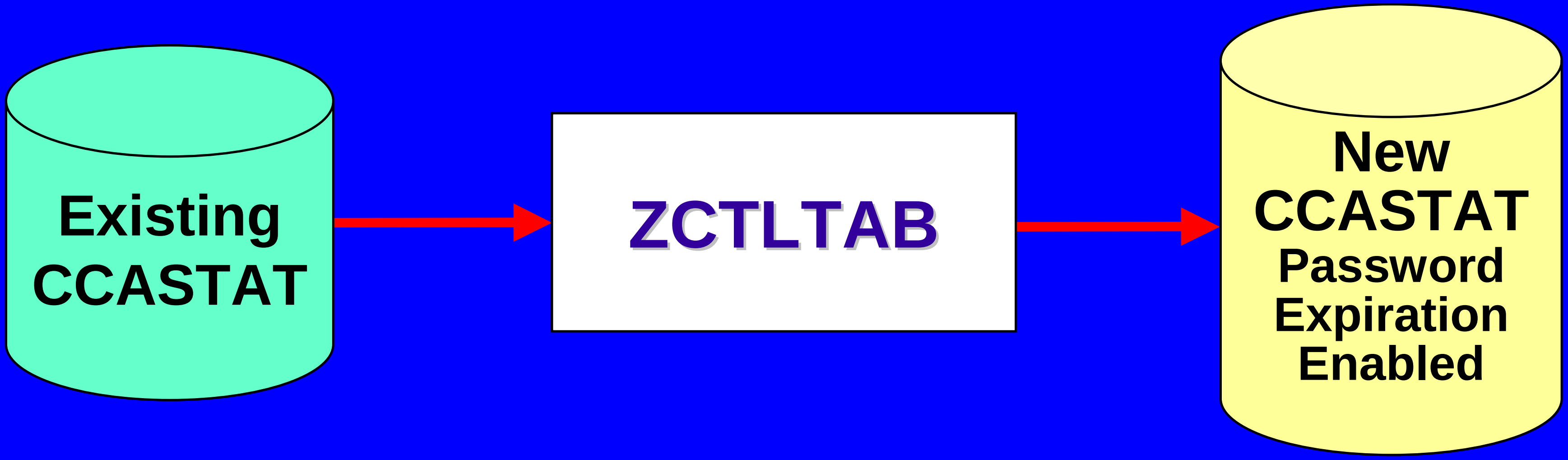
- New optional features for CCASTAT security
  - Brings CCASTAT security closer to security interface facilities
- Tracks Days a Password is Valid
- Revokes passwords
- Suspends user IDs
- Purging CCASTAT Entries
- Defining a password
- Password security rules apply only to LOGON passwords

# Track Days Password Valid

- Password Expiration
  - EXP – number of days a user can login using the same password
  - WARN – number of days prior to password expiration that user receives warning message:

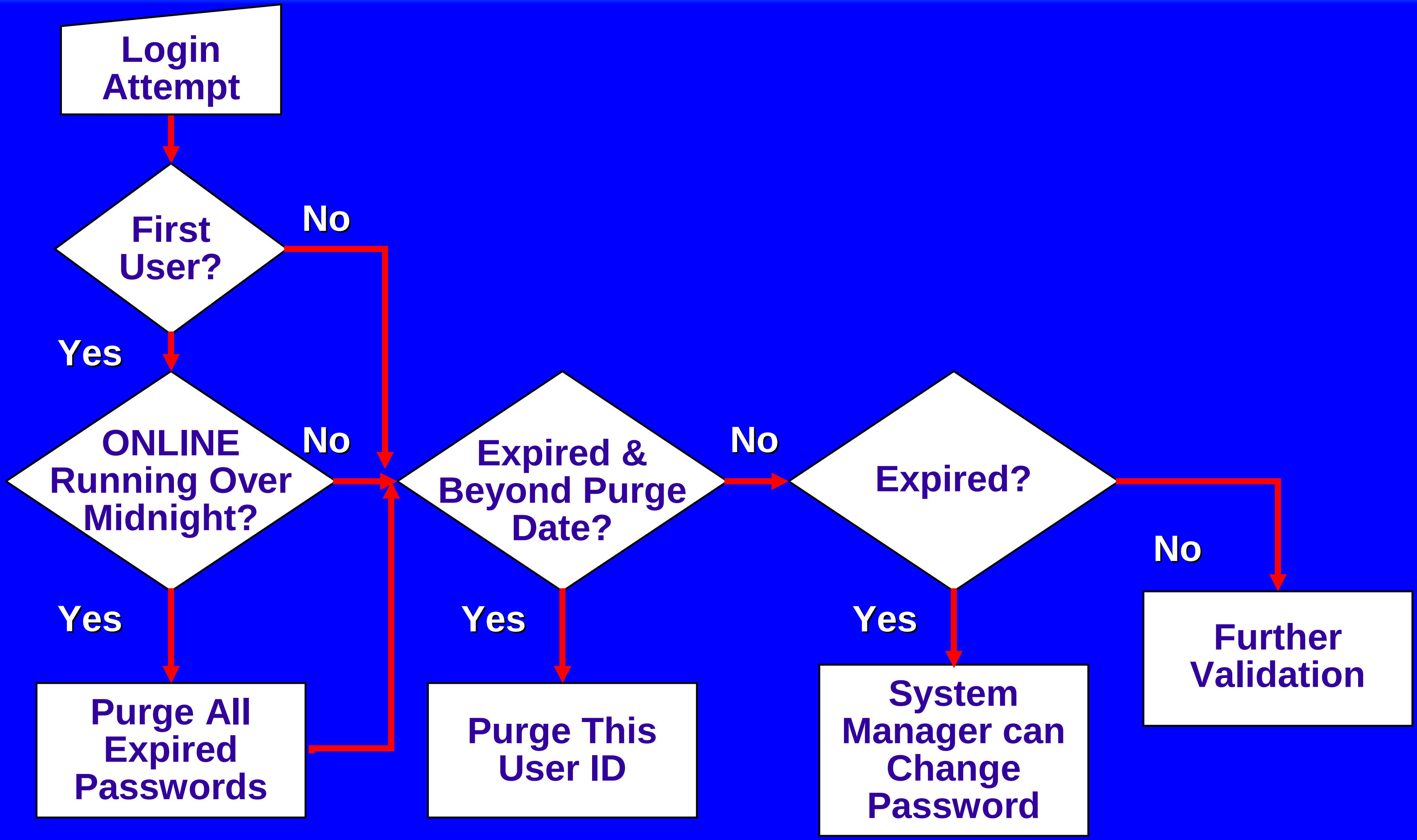
    ```
    ***M204.2634: YOUR PASSWORD WILL EXPIRE IN n DAYS
    ```
- User ID Purging
  - PURGE – number of days an expired user ID is held in suspension in CCASTAT awaiting a new password from the System Manager
  - Password revoked after 3 unsuccessful login attempts
    - May be reinstated before purge date

# ZCTLTAB



- Create new CCASTAT based on contents of existing CCASTAT (optional/recommended)
  - No re-convert capability if you decide to disable
- Used to activate password expiration feature
  - Sets EXP, WARN, and PURGE parameters (days)
  - Rerun ZCTLTAB to reset these parameters

# Purging Entries


Login Attempt
First User?
No
Yes
ONLINE Running Over Midnight?
No
Yes
Purge All Expired Passwords
Expired & Beyond Purge Date?
No
Yes
Purge This User ID
Expired?
No
Yes
System Manager can Change Password
Further Validation

# User ID and Password Maintenance

- LOGLST Command used to:
  - Monitor password expiration

- LOGCTL Command used to:
  - Modify User ID entries in password table
  - Define User IDs
  - Change Password

- LOGON/LOGIN Command used to:
  - Change Password

# Large Objects

- What is a Large Object (LOB)?

.doc

.jpeg

.xls

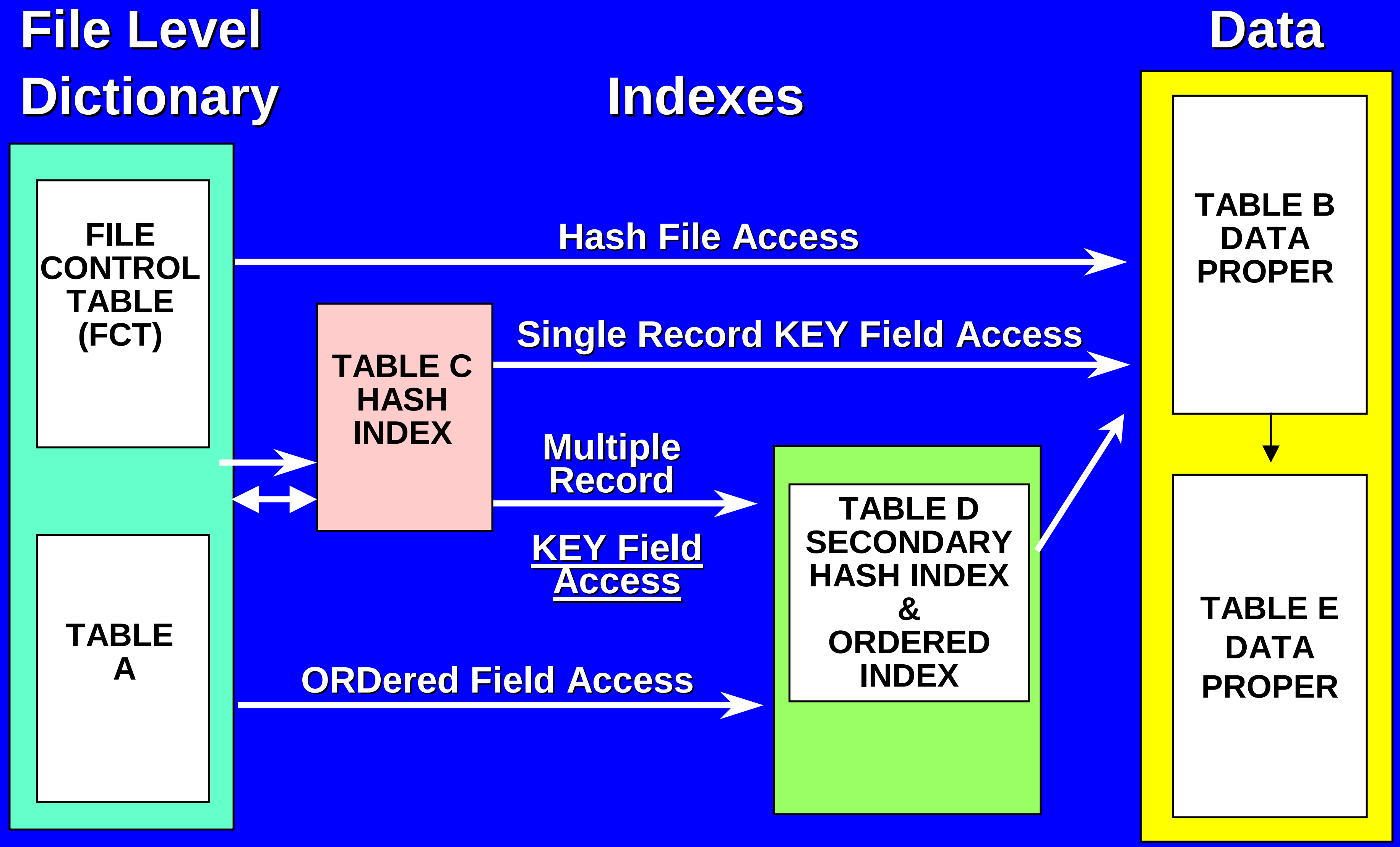
.wav

Stock Charts

Movie Clips

.* (Any combination of bytes from one byte to two gigabytes)

# Logical View of a Model 204 File



# Large Object Field Types

- BINARY-LARGE-OBJECT
  - BLOB
- CHARACTER-LARGE-OBJECT
  - CLOB

DEFINE FIELD EMPJPG (BLOB)

```
D FIELD (ABBREV) EMPJPG
EMPJPG
   (NFRV NKEY   NCOD BLOB NNR VIS UP NORD NUNIQ REPT)
>
```

# File Support for LOBs

```
CREATE FILE INVENTRY
PARAMETER ESIZE=….
……
END
O FILE INVENTRY
IN INVENTRY INITIALIZE
..
DEFINE FIELD ITEM_DESC –
  ( CHARACTER-LARGE-OBJECT | CLOB )
DEFINE FIELD ITEM_IMAGE –
  ( BINARY-LARGE-OBJECT | BLOB )
```

- New Table E for storing large objects
- Files created in V6R1 cannot be opened in prior releases

- New field attributes for LOBs
- All other attributes should default

# Storing LOB Data

```
..
%LEN = 3000
MODIFY MQ_BUFFER SIZE =%LEN
....

WRITE IMAGE TO MQ_BUFFER
%OFFSET = 1
IN INVENTRY STORE RECORD
 ...
 ITEM_IMAGE = MQ_BUFFER, %OFFSET, %LEN
END STORE
..
FD1: IN INVENTRY FD retrieval_criteria
           END FIND
FOR 1 RECORD IN FD1
 ADD ITEM_IMAGE = MQ_BUFFER, %OFFSET,%LEN
END FOR
...
```

Retrieve LOB into MQ buffer

- Reference with BUFFER or MQ_BUFFER

Use STORE RECORD to add Large Object from the buffer

.. Or ADD to existing record

# Retrieving LOB Data

```
...
...
%OFFSET = 10
FD1: IN INVENTRY FD retrieval criteria
    END FIND
FR FD1
  ...
  %LEN = $LOBLEN(ITEM_IMAGE)
  MQ_BUFFER,%OFFSET,%LEN
  READ IMAGE FROM MQ_BUFFER
  ...
END FOR
...
```

Use $LOBLEN to determine LOB length

Move LOB to MQ BUFFER

Read LOB into an image

# Enhanced RENAME FILE Command

- V5R1 introduced RENAME FILE command
- RENAME FILE FILEA  FILEB
- File pages updated directly
  - No need for DUMP, CREATE, RESTORE 64…
  - However, limitations for multi dataset files
    - CREATE FILE FILE1 FROM FILE1, FILE2, FILE3
    - Copy all datasets using IEBGENER
    - RENAME FILE FILE1 FILEA
    - New file consists of FILEA, FILE2, FILE3
      - Page trailers throughout set to FILEA
    - Not possible to ALLOCATE new FILEA alongside the copy of FILE1

# New RENAME Command Options

- Overcome multi dataset limitations and provide more flexibility
  - `RENAME FILE FILE1 FILEA ( FILE2 FILEB ) ( FILE3 FILEC )`
    - Rename all ddnames/dlbls/filedefs comprising a file
  - `RENAME FILE FILE1 FILEA ( FILE3 FILEC )`
    - Rename a file and specific ddnames/dlbls/filedefs
  - `RENAME FILE FILE1 (FILE2 FILEB )…`
    - Rename ddname/dlbl/filedef of a file without renaming the file

# Extensibility

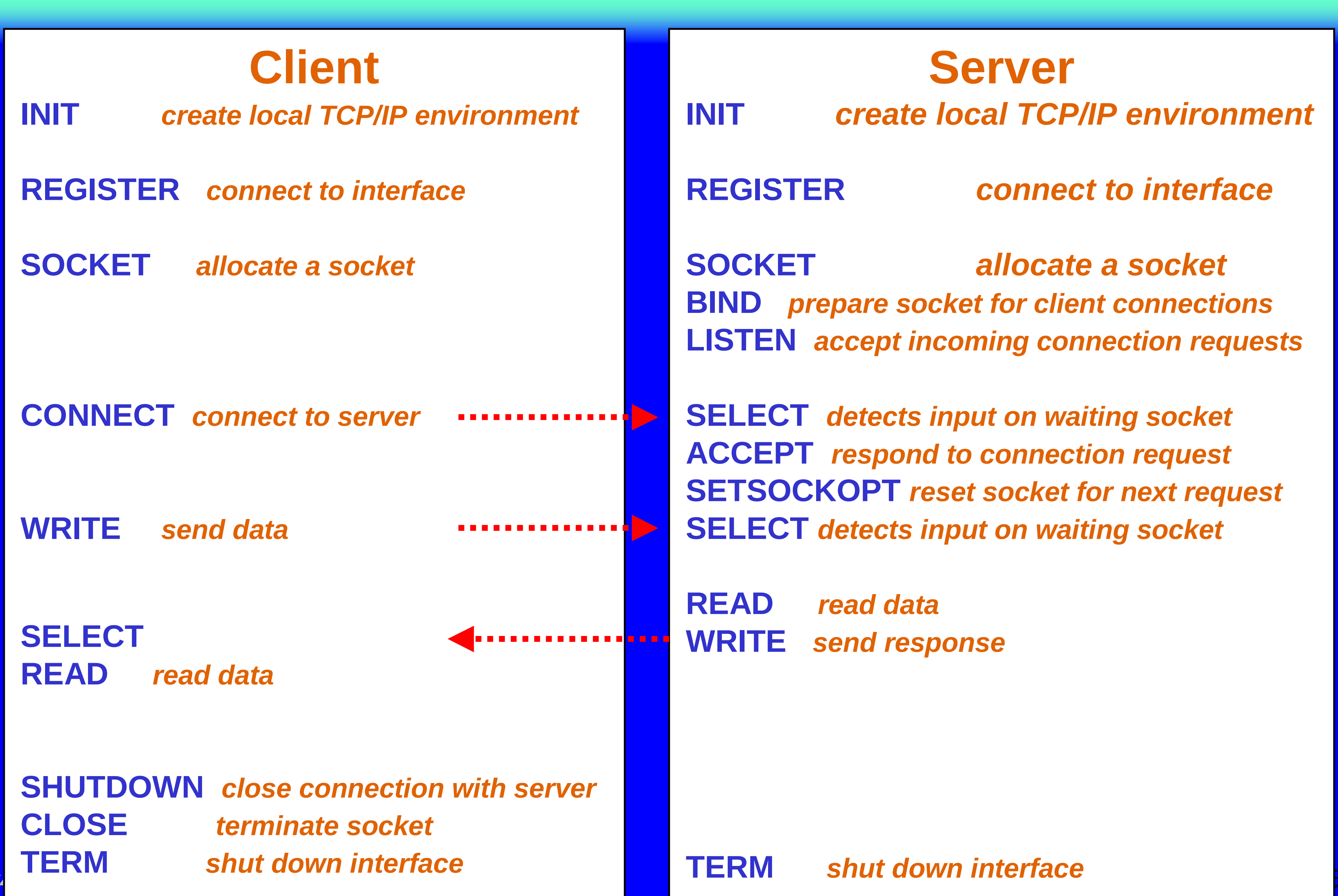
- Native Sockets
  - Use Model 204 as a Web or e-mail server
- Enhanced Java Support
  - Upgraded JDBC for Model 204
    - Integrates Model 204 with Java II Enterprise edition (J2EE) and Web Services

Free!

# CCA Sockets

- TCP/IP Native Sockets Interface
- Provides client – server communications API
  - Allows Model 204 to be deployed as a web or email server
  - Ability to connect to 3rd party software that supports sockets API
- Adheres to Berkeley sockets interface, as implemented by IBM and Microsoft
- Implemented using $SOCKET, $SOCKETX functions
- Calls match standard IBM socket calls

# Client / Server Calls



## Client

INIT create local TCP/IP environment

REGISTER connect to interface

SOCKET allocate a socket

CONNECT connect to server

WRITE send data

SELECT
READ read data

SHUTDOWN close connection with server
CLOSE terminate socket
TERM shut down interface

## Server

INIT create local TCP/IP environment

REGISTER connect to interface

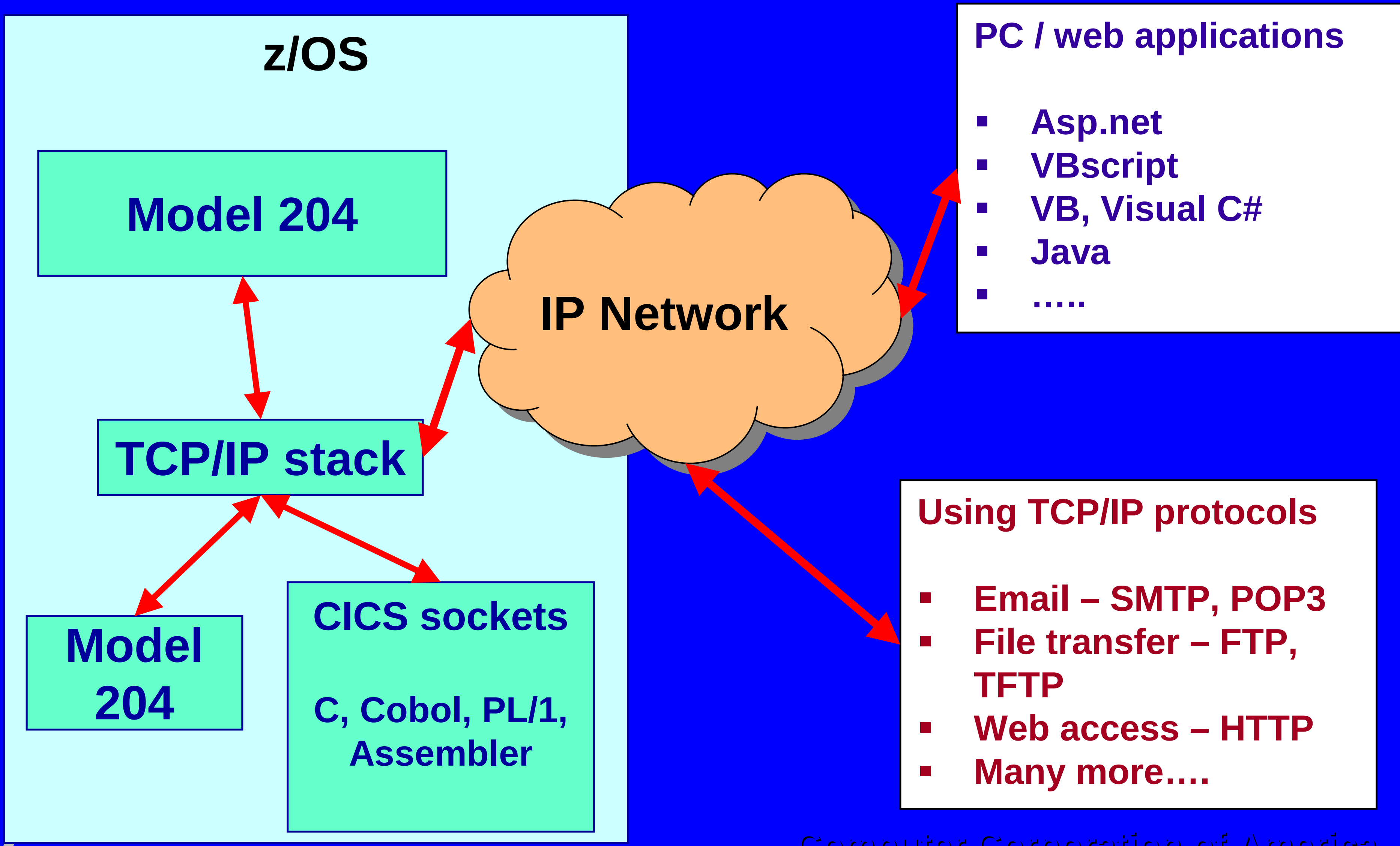
SOCKET allocate a socket
BIND prepare socket for client connections
LISTEN accept incoming connection requests

SELECT detects input on waiting socket
ACCEPT respond to connection request
SETSOCKOPT reset socket for next request
SELECT detects input on waiting socket

READ read data
WRITE send response

TERM shut down interface

# Sockets Connections



z/OS

Model 204

TCP/IP stack

Model 204

CICS sockets

C, Cobol, PL/1, Assembler

IP Network

PC / web applications

- Asp.net
- VBscript
- VB, Visual C#
- Java
- …..

Using TCP/IP protocols

- Email – SMTP, POP3
- File transfer – FTP, TFTP
- Web access – HTTP
- Many more….

# The Birth of Model 204

- October 13, 1965

Model 204 born out of an unsolicited proposal to the Director of the National Security Agency (NSA)

# Model 204 V6R1

Available NOW!

ity

xtensibility