Honeywell

Application Module Implementation Guidelines

AM12-510

Honeywell

Implementation
Application Module - 2

Application Module Implementation Guidelines

AM12-510
Release 510
11/95

Copyright, Trademarks, and Notices

About This Publication

This publication provides guidelines for the implementation of Application Modules, including the allocation of AM memory, adding units to AMs, moving units between AMs, the AM Redundancy option, recovery from certain error conditions, obtaining AM processing-schedule information, AM checkpointing, the AM Redundancy option, custom software, CL extensions, and background CL.

This publication supports TDC 3000X software release 510.

> Change bars are used to indicate paragraphs, tables, or illustrations containing changes that have been made to this manual effective with release 510. Pages revised only to correct minor typographical errors contain no change bars.

About This Publication

This publication provides guidelines for the implementation of Application Modules, including the allocation of AM memory, adding units to AMs, moving units between AMs, the AM Redundancy option, recovery from certain error conditions, obtaining AM processing-schedule information, AM checkpointing, the AM Redundancy option, custom software, CL extensions, and background CL.

This publication supports TDC 3000X software release 510.

> Change bars are used to indicate paragraphs, tables, or illustrations containing changes that have been made to this manual effective with release 510. Pages revised only to correct minor typographical errors contain no change bars.

Table of Contents

Table of Contents

INTRODUCTION
Section 1

This introduction summaries the Application Module (AM) implementation tasks, lists publications that you will refer to, to implement AMs, and describes implementation dependencies.

1.1 SUMMARY OF AM IMPLEMENTATION TASKS

1.1.1 Engineering Personality Activities

The Engineering Personality activities listed below may be affected by the implementation of an AM or must be used to implement an AM.

See 1.1.2 for references to instructions for each of these activities.

Activities named in `THIS TYPEFACE` are activated by targets on the Engineering Personality Main Menu.

- `UNIT NAMES`—The process units defined for each AM data point are established in this activity.
- `AREA NAMES`—The area name and descriptor for any units with AM points that are assigned to an area are established in this activity.
- `LCN NODES`—The AMs on the LCN and redundant AM partners are defined in this activity. Also, this activity has a role in the allocation of AM memory and the implementation of custom software and background CL programs.
- `VOLUME CONFIGURATION`—The AM checkpoint volume, &5np, is established in this activity. This volume must have adequate storage space to accommodate the checkpoint data for all AMs on the LCN. Please refer to Section 6, AM Checkpoints, in this publication before you configure any AM checkpoint volumes.
- `APPLICATION MODULE`—AM data points are built in this activity.
- `PICTURE EDITOR`, `FREE FORMAT LOGS`, `BUTTON CONFIGURATION`—Any of pictures, logs, and buttons built by these activities can access AM points, once the points are built and loaded.
- `HM HISTORY GROUPS`—AM data point values for which continuous history is to be collected are defined in this activity, by assigning them to specific HM history groups.
- `AREA DATA BASE`—This activity defines how and where data for data points, including AM data points are used and displayed in a given process area. The area database is the database loaded into a Universal Station, so that database defines the process area monitored and controlled through the US.

- Control Language (CL)—Use this language to prepare user-written programs to accomplish data acquisition and control functions that cannot be accomplished by the standard AM data points and algorithms.

1.1.2 Other AM Implementation Activities

- Estimating AM memory used by AM data points and AM memory allocation—AM performance can be enhanced through these activities. Information and instructions for them are in Section 2 of this publication.

- Adding and moving AM process units—As your process changes, your unit allocations to AMs may need to change. Information and instructions for these activities are in Section 3 of this publication.

- Scheduling the processing of AM points for best performance—If you spread out the point-processing load in your AMs, you will attain better performance and reduce the chance that processing of a point will be delayed because of a processing overload. Information and instructions for displaying and printing of AM point scheduling information are in Section 5 of this publication.

- Establishing AM checkpoint volumes and considering the impact on them of AM options—Section 6 of this publication provides guidelines for establishing the size of AM checkpoint volumes, the effect of AM options on the size of those volumes, the effect of reserved AM memory on those volumes, and information about saving AM checkpoints on removable media (cartridges or floppies).

- Implementing redundant AMs—For applications that require high reliability, AMs can be implemented as redundant partners. Section 7 of this publication provides information for the implementation and use of redundant AMs.

- Implementing Background CL and custom software—You can prepare CL programs that run in the background on an AM, that is programs that run when the AM is not busy with point processing or other activities. Also, custom software, usually custom application software can be provided by Honeywell for your AMs. Section 8 of this publication provides guidelines for the implementation of these functions.

1.1.3 References for Engineering Personality Activities

- UNIT NAMES and AREA NAMES

  Network Form Instructions in the *Implementation/Startup & Reconfiguration - 1* binder

  Network Data Entry in the *Implementation/Startup & Reconfiguration - 1* binder

- LCN NODES

  Network Form Instructions in the *Implementation/Startup & Reconfiguration - 1* binder

  Network Data Entry in the *Implementation/Startup & Reconfiguration - 1* binder

  Sections 2 and 8 in this publication.

- VOLUME CONFIGURATION

  Section 6 of this publication and Section 7 of the *Engineer's Reference Manual*.

- APPLICATION MODULE

  Application Module Control Functions in the *Implementation/Application Module - 1* binder

  Application Module Algorithm Engineering Data in the *Implementation/Application Module - 1* binder

  Application Module Parameter Reference Dictionary in the *Implementation/ Application Module - 2* binder

  Data Entity Builder Manual in the *Implementation/Engineering Operations - 1* binder

- NETWORK INTERFACE MODULE

  Logic Manager Control Functions in the *Implementation/Logic Manager* binder

  Logic Manager Parameter Reference Dictionary in the *Implementation/Logic Manager* binder

  Data Entity Builder Manual in the *Implementation/Engineering Operations - 1* binder

- PICTURE EDITOR, FREE FORMAT LOGS, BUTTON CONFIGURATION

  Instructions for these activities are in binders 1 and 2 of *Implementation/ Engineering Operations*

- HM HISTORY GROUPS

  HM History Group Form Instructions in the *Implementation/Engineering Operations - 1* binder.

  Data Entity Builder Manual in the *Implementation/Engineering Operations - 1* binder

- AREA DATA BASE

  Area Form Instructions in the *Implementation/Engineering Operations - 1* binder

  Data Entity Builder Manual in the *Implementation/Engineering Operations - 1* binder

- CONTROL LANGUAGE (CL)

  CL/AM publications in the *Implementation/Application Module - 2 & 3* binders

  Data Entity Builder Manual in the *Implementation/Engineering Operations - 1* binder

1.1.4 Engineer's Reference Manual

The *Engineer's Reference Manual* in the *Implementation/Startup & Reconfiguration - 2* binder is a reference and guidelines manual that is useful in all implementation activities.

1.2 AM IMPLEMENTATION DEPENDENCIES

Figure 1-1 shows which Application Module implementation tasks depend on information entered in other tasks. This figure does not necessarily dictate the order in which the tasks must be completed, but it does show the dependencies that must be satisfied before an AM can be fully operational.

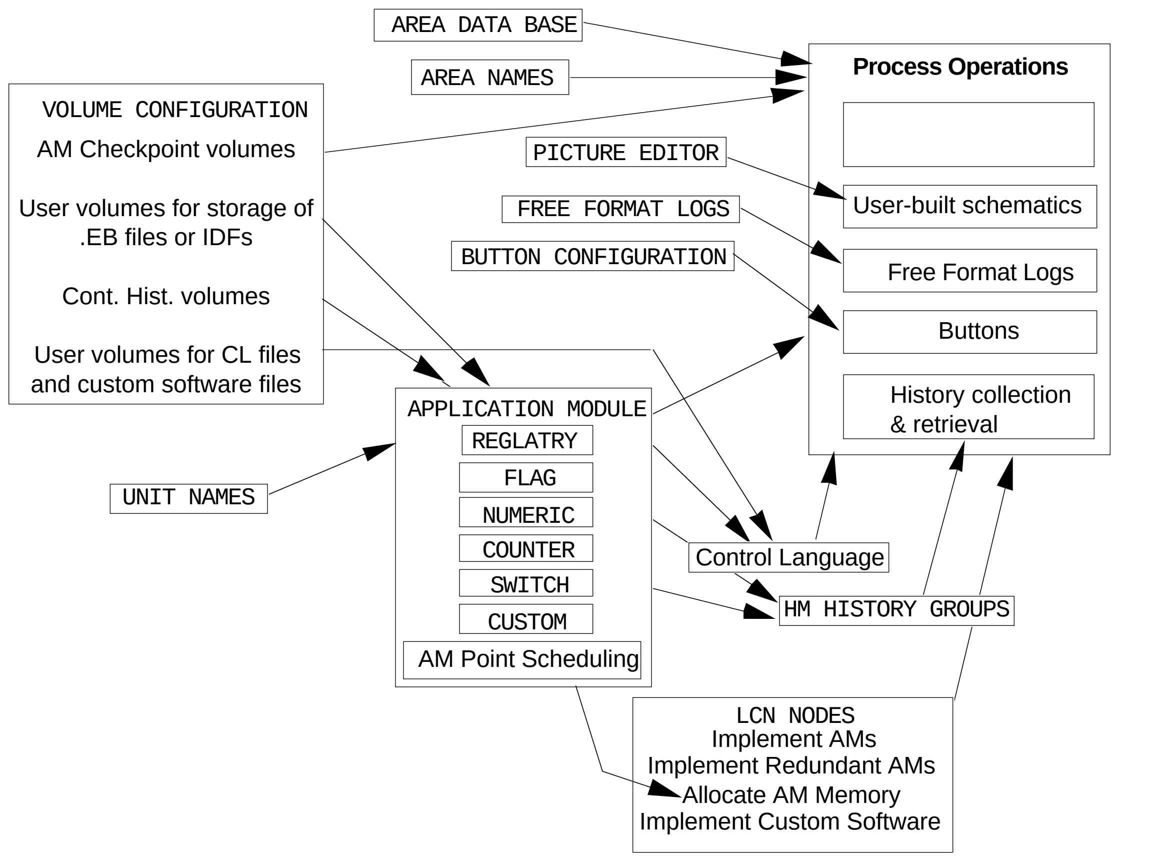


Figure 1-1 — Application Module Implementation Dependencies

1.3 AM PSDP Parameters

Every node on the LCN has a processor Status Data Point (PSDP) that contains a number of parameters. These parameters reflect important performance status and memory-use values.

PSDP parameters have been added to determine the points which take the longest time to execute. These parameters will display and maintain the point names and CL block names that took the longest time to process by the Fast/Slow Point Processors. Another parameter of the AM PSDP is used to switch the collection of the other parameters ON or OFF. The switch parameter will not switch off the saving of the point name and CL block name that is used for the crash statistics.

A list of the these parameters and additional information on the use of the AM PSDP parameters is provided in subsection 22.3, AM PSDPs, of the *Engineer's Reference Manual.*

Section 2 – AM Memory Allocation

2.0 Overview

Introduction

Engineers who design the TDC 3000X system configuration, including the Application Module configuration, can influence the allocation of the AM's memory and thereby affect the AM's point capacity.

This section provides guidelines for the allocation of AM memory and how to interpret the AM processor status data point (PSDP) parameters that are used to monitor AM memory allocation.

Background

Data points in Application Modules require differing amounts of memory, as determined by the point types and the options selected for the points. For example, an AM regulatory point with the following options uses much more memory than a simple flag point with minimal options:

- PV and control algorithms
- CL blocks
- custom data segments

Unlike the NIM and HG, AM points reside entirely in the AM where they are processed.

AM points can fetch data from and store data to other AMs, NIMs, and HGs, requiring memory buffers to store the data until it is used in point processing.

The AM buffers must be made large enough to accommodate the largest amount of data exchanged on a single point-processing cycle. Larger memory buffers leave less memory for data points.

Topics

This section contains the following topics:

2.1 AM NCF Configuration Tasks and Displays

NCF tasks

The following AM configuration is performed through the Network Configuration File (NCF):

- configure custom functions,
- configure background functions,
- specify optional runtime extensions (such as those used when reading and writing to text files resident on the History Module or removable media),
- verify or modify the amount of AM memory available for various tasks and functions,
- specify additional custom software that may be loaded into an AM.

User Memory Allocation Display

One of the major AM configuration tasks that you will need to perform is allocating AM user memory. User memory allocation is the overall purpose of page 2 of the AM's NCF Node Configuration display. Figure 2-1 is an example of NCF page 2.

Figure 2-1 User Memory Allocation Display—NCF Page 2

```
                                                                  20 Mar    04:21:25   1
APPLICATION MODULE NODE                                   PAGE  2 OF  3 | ON-LINE

   NODE     40               USER MEMORY ALLOCATION

FUNCTIONAL ADJUSTMENTS:                                               MEMORY USED(WORDS)
  # BACKGROUND CL TASKS                                    [2    ]            66200
  # CONCURRENT DATA ACCESSES FROM BACKGROUND CL            [2    ]             2100
  BACKGROUND TASK STACK SIZE                               [25000]
  CVB SIZE FOR FAST & SLOW POINT PROCESSORS                [10000]           170000
  REDUNDANCY BUFFER INCREASE (# OF 32KW BLOCKS)            [1    ]            32768
  INCLUDE INTERNETWORK POINT PROCESSOR?     YES NO                            44050
  CVB SIZE FOR IPP                                         [2000 ]
  USER MEMORY RESERVED (# OF 32KW BLOCKS)                  [1    ]            32768
  EXTERNAL LOAD MODULES CODE (ROUNDUP FROM NEXT PAGE)                        608256
  EXTERNAL LOAD MODULES POOL1 (FROM NEXT PAGE)                               120832
                                                                            -------
TOTAL MEMORY FOR FUNCTIONAL ADJUSTMENTS                                     1076974

CURRENT DATA BASE SIZE  (AMMEMTOT)                         136149.0
ROOM LEFT FOR DATA BASE GROWTH                             807422.0

TOTAL USER MEMORY AFTER SOFTWARE OPTIONS  (AMMEMAOP)       2020540.

F1=CHECK     F3=SET OFFLINE    F5=ABORT                           F9=PACK NCF
F2=INSTALL   F4=PRINT
```

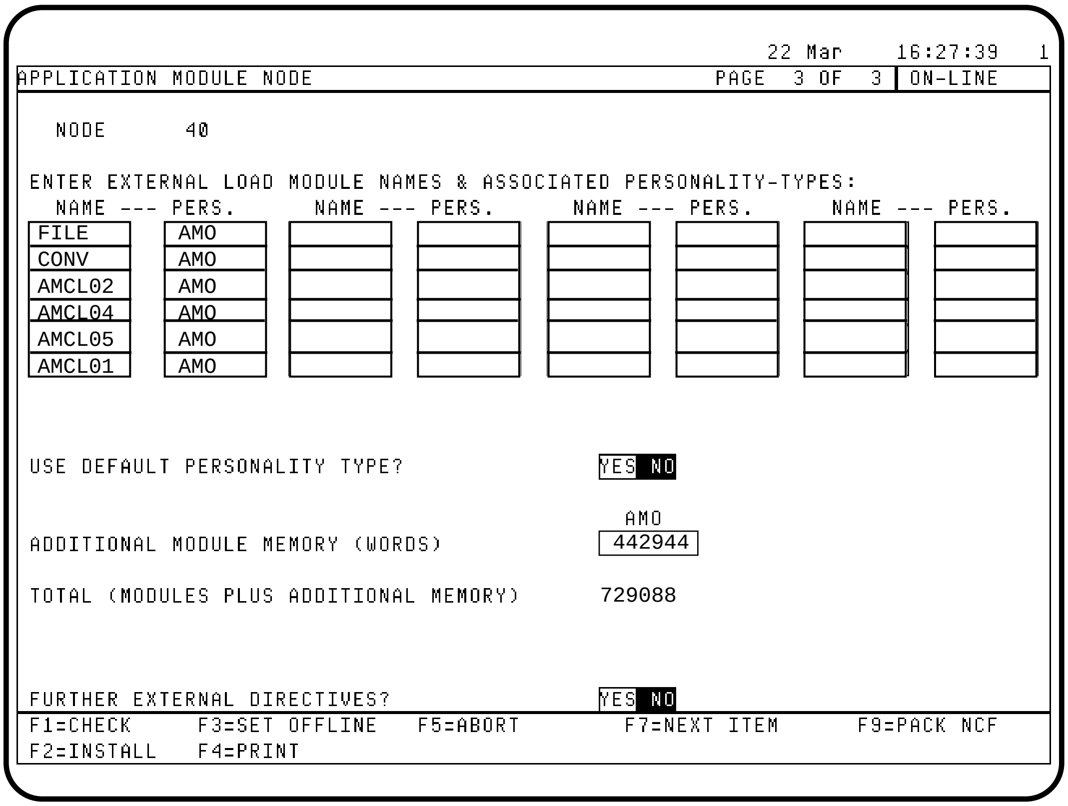
53664

Continued on next page

External Load Module Display

Another major AM configuration task is to specify the optional software (called EXTERNAL LOAD MODULES) that you can load into an AM. Figure 2-2 is an example of that NCF page.

Figure 2-2 External Load Module Display—NCF Page 3



33686

2.2 AM Memory Sizes

Description

The total amount of memory in an AM can range from 3 Mw to 16 Mw (Mw = one million words).

The amount of **user** memory available in an AM (PSDP parameter AMMEMAOP) is a function of the total AM memory size.

Available User Memory

Table 2-1 shows how the AM memory available to the user is related to the total AM memory size.

NOTE: Options such as AM redundancy and K4LCN processor reduce available AM user memory (see Figure 2-3, 2-4 for additional information).

NOTE: Memory mapping is different for an $A^{X}M$.

NOTE: AM's maximum memory used is dependant on the hardware used. For example, the AM memory with:

the HPK2/HMPU processors uses a maximum of 7 Mw,

the K2LCN processor uses a maximum of 7 3/4 Mw, and

the K4LCN processor exists in 4 Mw, 8 Mw , and 16 Mw sizes only.

The available AM user memory with 7 3/4 Mw is 5,786,245 words.

NOTE: The $A^{X}M$ available user memory for the: 6 Mw $A^{X}M$ corresponds to the 3 Mw AM; 7 Mw $A^{X}M$ corresponds to the 4 Mw AM; 8 Mw $A^{X}M$ corresponds to the 5 Mw AM; and 16 Mw $A^{X}M$ is 11,058,224 words.

Table 2-1 Available AM User Memory

Total AM Memory Size	Available AM User Memory (R500)
3.0 Mw	1,016,358 words
4.0 Mw	2,020,545 words (2,009,435 for K4LCN)
5.0 Mw	3,024,731 words
6.0 Mw	4,028,918 words
7.0 Mw (HPK2/HMPU maximum)	5,033,105 words
7.75 Mw (K2LCN maximum)	5,786,245 words
8.0 Mw	6,037,291 words (6,020,211 for K4LCN)
16.0 Mw	14,070,784 words (14,045,430 for K4LCN)

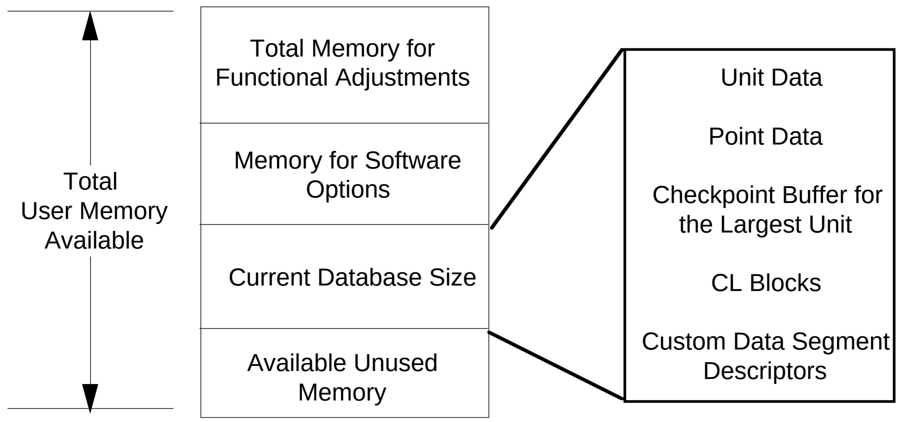
Exceeding available user memory

The total AM memory used increases with each point loaded into the AM and with each CL block linked to a point. If one of these operations results in the total memory used exceeding the total user memory available, the operation is aborted and an error message is generated.

2.3 How the AM User Memory is Organized

Conceptual diagram

The following diagram is a conceptual view of the AM's user memory. The actual structure is not as shown here.



NCF configuration

For purpose of NCF configuration, the AM user memory is divided into the four sections described in Table 2-2.

Table 2-2 Sections of AM User Memory

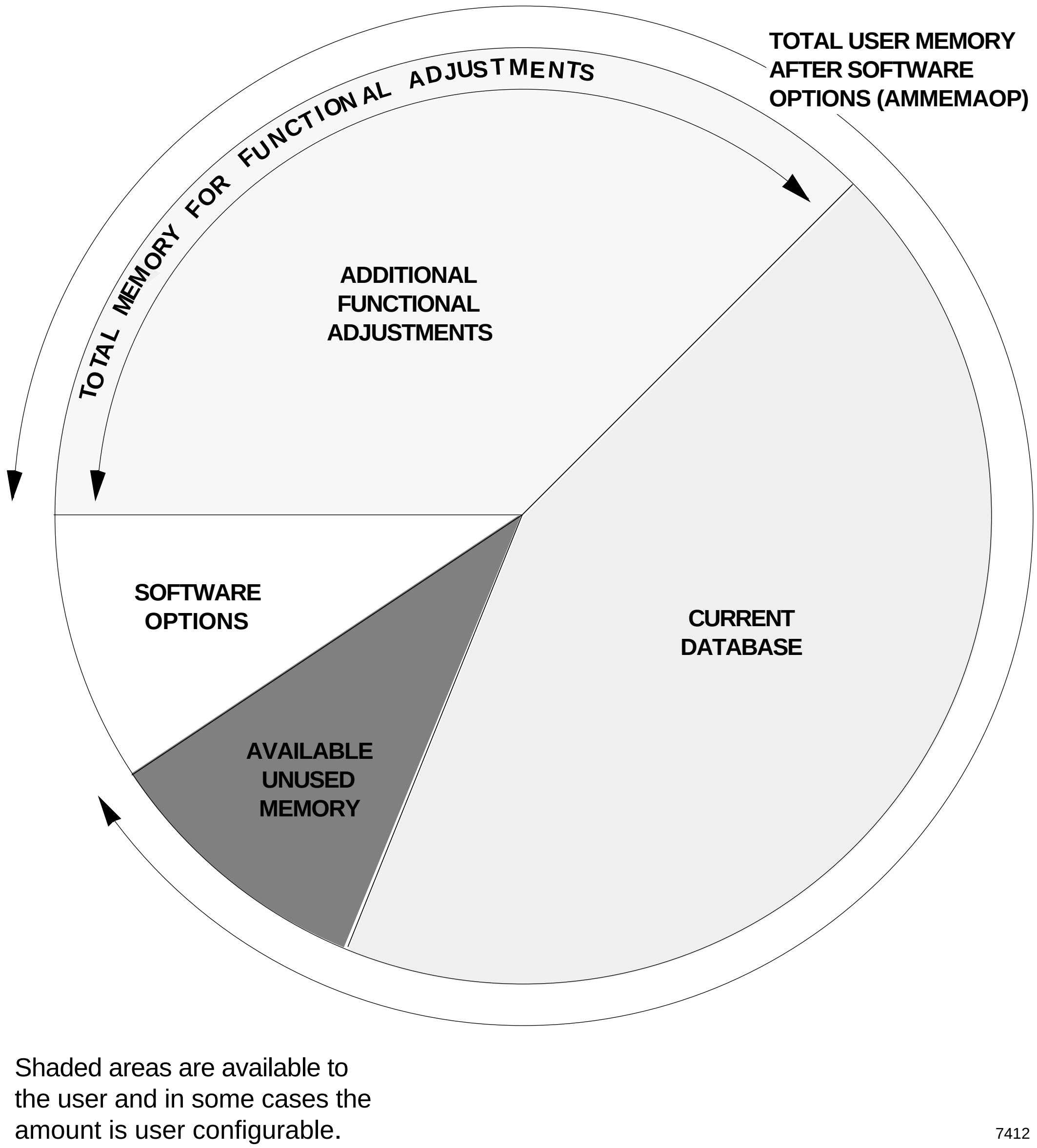
Section	Description
1. Total Memory for Functional Adjustments	Configured on NCF page 2 of the AM's LCN Node configuration. These entries can be changed from the NCF display while the AM is on-line. In order for the changes to take effect, the NCF must then be installed and the AM (or redundant AM pair) restarted.
2. Memory for Software Options	K4LCN Processor Option If the AM uses the K4LCN processor, then space for memory management will be required depending on the amount of memory present on the board. The following K4LCN memory sizes are available: • 4 meg K4LCN requires ≈12,000 words • 8 meg K4LCN requires ≈18,000 words • 16 meg K4LCN requires ≈26,000 words AM Redundancy Option Redundant AMs require the following memory: • 157,000 words (fast/slow processor buffers), plus an additional • 22,000 words if using the AM redundancy Internetwork Point Processor (IPP) option.
3. Current Database Size	Memory required for AM points, CL programs, custom data segments, and checkpoint buffers. This memory is allocated through the DEB and CL compiler while the AM is on-line.
4. Available Unused Memory	Memory that remains unused after the other three sections are allocated—user memory that is available and not allocated. This memory is described in some displays as "room left for database growth.

Continued on next page

Conceptual diagram

Figure 2-3 is a conceptual representation of the AM user memory. As you can see in the diagram, the more memory the user allocates to "Additional Functional Adjustments" in the NCF configuration, the less memory there is available for database.

Figure 2-3 Conceptual Diagram of AM User Memory



2.4 Parameters For Monitoring AM Memory Use

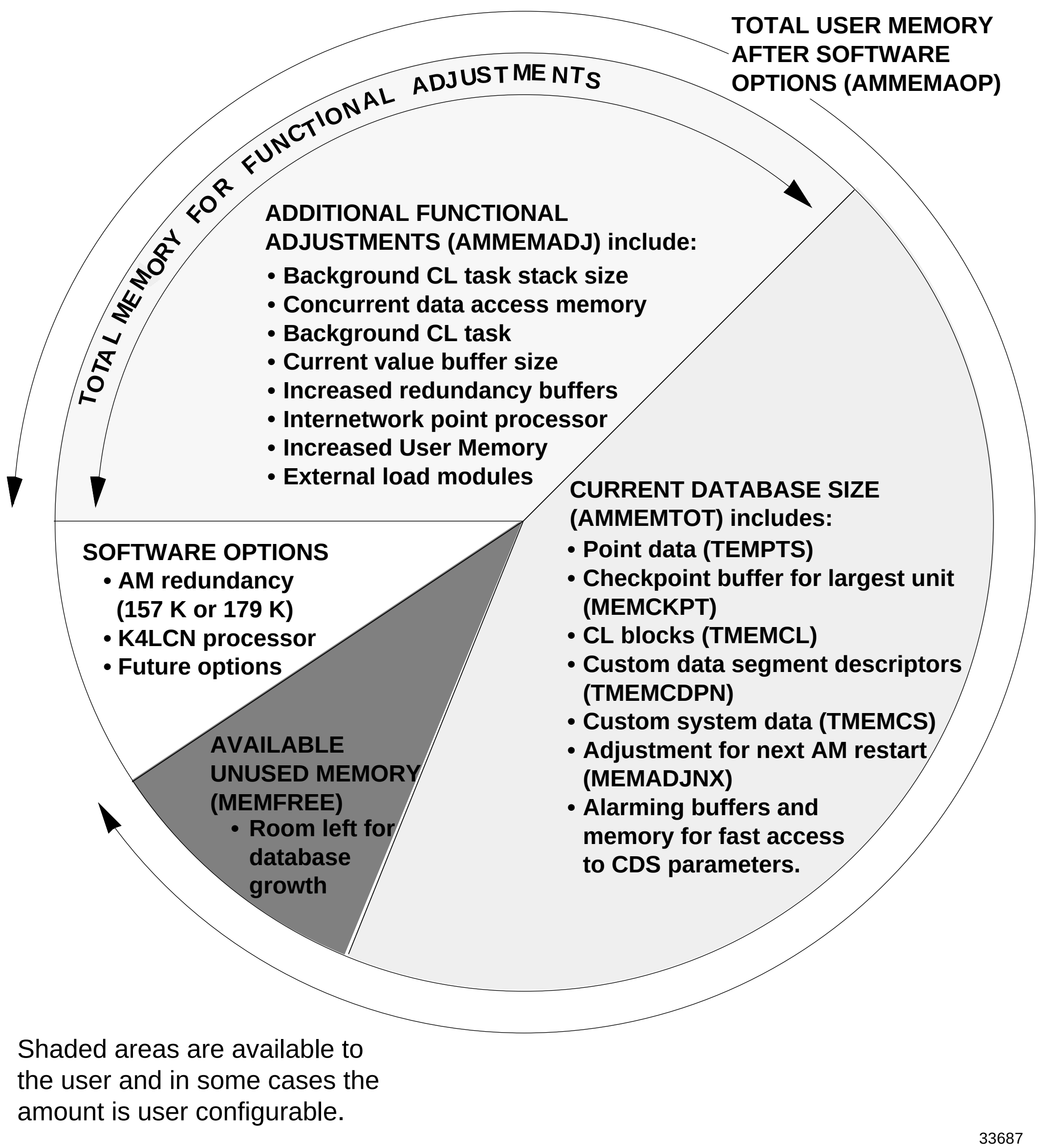
Description

AM Processor Status Data Point parameters are available to monitor AM memory use. Some of the parameters are shown on page 2 of the AM's NCF display—the User Memory Allocation display.

Correlating PSDP parameters to AM user memory

Figure 2-4 correlates the AM user memory with the parameters available on the NCF User Memory Allocation display.

Figure 2-4 Diagram Correlating AM Memory to NCF Parameters



Continued on next page

Memory allocation parameters

Since several of the AM memory allocation parameters can be modified when configuring the AM NCF, you can use the Application Module NCF display to check their values.

The following functional adjustments memory allocation parameters can be viewed on the following NCF display and are described in Table 2-3.

	PSDP Parameter
CVB SIZE FOR FAST AND SLOW PROCESSORS	MEMCVBLM
REDUNDANCY BUFFER INCREASE (# OF 32KW BLOCKS)	AMDATA(48)
CVB SIZE FOR IPP	MIPCVBLM
TOTAL MEMORY FOR FUNCTIONAL ADJUSTMENTS	AMMEMADJ
CURRENT DATA BASE SIZE	AMMEMTOT
ROOM LEFT FOR DATA BASE GROWTH	MEMFREE
TOTAL USER MEMORY AFTER SOFTWARE OPTIONS	AMMEMAOP

Figure 2-5 User Memory Allocation—NCF Display

```
                                                          20 Mar    04:21:25   1
APPLICATION MODULE NODE                               PAGE  2 OF  3  ON-LINE

   NODE     40               USER MEMORY ALLOCATION

FUNCTIONAL ADJUSTMENTS:                                          MEMORY USED(WORDS)
  # BACKGROUND CL TASKS                                 2                  66200
  # CONCURRENT DATA ACCESSES FROM BACKGROUND CL         2                   2100
  BACKGROUND TASK STACK SIZE                            25000
  CVB SIZE FOR FAST & SLOW POINT PROCESSORS             10000             170000
  REDUNDANCY BUFFER INCREASE (# OF 32KW BLOCKS)         1                  32768
  INCLUDE INTERNETWORK POINT PROCESSOR?     YES NO                         44050
  CVB SIZE FOR IPP                                      2000
  USER MEMORY RESERVED (# OF 32KW BLOCKS)               1                  32768
  EXTERNAL LOAD MODULES CODE (ROUNDUP FROM NEXT PAGE)                     608256
  EXTERNAL LOAD MODULES POOL1 (FROM NEXT PAGE)                            120832
                                                                         -------
TOTAL MEMORY FOR FUNCTIONAL ADJUSTMENTS                                  1076974

CURRENT DATA BASE SIZE  (AMMEMTOT)                     136149.0
ROOM LEFT FOR DATA BASE GROWTH                         807422.0

TOTAL USER MEMORY AFTER SOFTWARE OPTIONS  (AMMEMAOP)  2020540.


F1=CHECK     F3=SET OFFLINE   F5=ABORT                          F9=PACK NCF
F2=INSTALL   F4=PRINT
```

53664

Continued on next page

Parameter definition

Table 2-3 describes the Processor Status Data Point (PSDP) parameters that allow you to monitor AM memory use.

Table 2-3 AM PSDP Parameters For Memory Use

Parameter	Description
AMMEMAOP	TOTAL USER MEMORY AFTER SOFTWARE OPTIONS (based on available memory size and software options)
AMMEMADJ	TOTAL MEMORY FOR FUNCTIONAL ADJUSTMENTS
AMMEMTOT	CURRENT DATABASE SIZE—the sum of the following: • Point Data (TMEMPTS) The point size plus eleven words for each point (Reference Subsection 2.12 of this manual.). Also add the space for the Custom Data Segments attached to any points (Reference Appendix B.2.5, in the *CL/AM Reference Manual*). (Default=approximately 1800 words per unit when using the NULL/STARTUP checkpoint.) • Largest Checkpoint Buffer (MEMCKPT) Space for the largest unit—1.03 times the Point Data item for the largest unit. CL blocks, reference lists, and custom data descriptions do not need to be included. (Default=approximately 6000 words when using the NULL/STARTUP checkpoint.) • CL Blocks (TMEMCL) The sizes for these are defined in B.2.3 in the *CL/AM Reference Manual*. Remember that each CL block requires the same space for each unit that uses it. Reference lists require space on each point to which the block is linked. (Default=zero words per unit when using the NULL/STARTUP checkpoints.) • Custom Data Descriptors (TMEMCDPN) The sizes for these are defined in Appendix B, Section 2.5 of the *CL/AM Reference Manual*. Remember that AM custom data descriptions require space only once in a given AM—they are shared among points and units. (Default=approximately 1200 words per unit when using the NULL/STARTUP checkpoint.) • Custom System Data (TMEMCS) • Adjustment for next AM restart (MEMADJNX) • A variable (without a PSDP parameter) for alarm buffers, buffers for fast custom parameters, and the optional extension for fast external CDS fetch (AMCL04). The number is approximately 46000 words plus the memory impact of the optional AMCL04. (Reference G.3 of the *CL/AM Reference Manual*)
MEMFREE	ROOM LEFT FOR DATABASE GROWTH (MEMFREE=AMMEMAOP-AMMEMADJ-AMMEMTOT)

Continued on next page

Table 2-3 AM PSDP Parameters For Memory Use, continued

Parameter	Description
MEMCVBLM	Current Value Buffer (CVB) SIZE FOR FAST AND SLOW PROCESSORS The current size for each of the 10 Fast Point Processor (FPP) and Slow Point Processor (SPP) CVBs (and related buffers). You can change this value so that the size of the CVBs changes the next time the AM is reloaded. Default=34,000. If you change the CVB size in MEMCVBLM , the total FPP/SPP CVB allocation is 17*MEMCVBLM. (Note: CUB allocation is included in total memory for functional adjustments.)
MEMCVBTH	The user-changeable alarm threshold for CVBs used by FPP and SPP. Default=1600. If the portion of an FPP/SPP CVB used by the system becomes larger than the value in MEMCVBTH, an alarm is generated. This alarm indicates that it is possible that the MEMCVBLM value may soon be exceeded. If the MEMCVBLM value is exceeded, prefetch and poststore data is lost, a (simulated) communication error occurs, and another alarm is generated. The first alarm (the MEMCVBTH alarm) indicates that the CVB size needs to be changed, less of the dynamic heap should be used (such as by deleting points, CL blocks, CDSs, and references to other nodes), or the threshold in MEMCVBTH is too low.
MCVBMAXC	The largest amount of memory used by an FPP/SPP CVB during the current hour. This parameter can be changed only by the system. It is used to evaluate the use of dynamic heap during actual operation.
MEMCVBMX	The largest amount of memory used by an FPP/SPP CVB since the last time the AM was loaded or its value was reset by the engineer. It also is used to evaluate the use of dynamic heap during actual operation.
MIPCVBLM	CVB SIZE FOR IPP If the Internetwork Point Processor has been configured in the AM, this parameter sets the sizes for each of the two IPP CVBs (and related buffers). You can change this value so that the size of the CVBs changes the next time the AM is reloaded. Default=18,000. If you change the value in MIPCVBLM, the total IPP CVB allocation is 9*MIPCVBLM.
MIPCVBMX	The largest amount of memory used by an IPP CVB since the last time the AM was loaded or its value was reset by the user. Used to evaluate use of dynamic heap during operation.
REDBFZ	This parameter for the secondary AM returns the redundancy buffer size.

Continued on next page

PSDP point name

The PSDP point name is $PRSTSnn, where nn is the AM's LCN node number. For example, to read the value of parameter MEMCVBMX in node forty, you would enter `$PRSTS40.MEMCVBMX`.

REFERENCE—Section 22 of the *Engineer's Reference Manual* defines additional AM PSDP parameters.

REFERENCE—You can see AM PSDP parameter values (and you can change the values in certain parameters) by *building a schematic* that accesses the point. For one way to do this, see the example of "a display to read and store in system variables" (abstract ACC_RSS_1), in the following manual:

Actors Manual
Implementation/Engineering Operations - 2 binder.

Also, these AM PSDP parameters can be viewed and modified from some of the toolkit displays such as DATACHNG. See the following manual:

"System Performance Displays"
Engineer's Reference Manual
Implementation/Startup & Reconfiguration - 2 binder.

2.5 AM User Memory Allocation—NCF Display

Description

The AM's User Memory Allocation NCF display shows the memory totals indicating how much AM memory the user currently has allocated:

Total Memory for Functional Adjustments (AMMEMADJ)	1076974
+ Current Database Size (AMMEMTOT)	136149.0
+ Room Left for Database Growth (MEMFREE)	807422.0
= Total User Memory After Software Options (AMMEMAOP)	2020545.

The lower part of the display summarizes how you have used the memory available to you. (The AMMEMAOP value shown on the display may differ slightly from the above calculation, due to rounding. The difference is not of any great concern.)

Example

Figure 2-6 is an example of the AM's User Memory Allocation NCF display.

Figure 2-6 User Memory Allocation Display with Highlighted Areas

```
                                                                  20 Mar    04:21:25   1
APPLICATION MODULE NODE                                    PAGE  2 OF  3  ON-LINE

    NODE     40                USER MEMORY ALLOCATION

 FUNCTIONAL ADJUSTMENTS:                                        MEMORY USED(WORDS)
   # BACKGROUND CL TASKS                               2                    66200
   # CONCURRENT DATA ACCESSES FROM BACKGROUND CL       2                     2100
   BACKGROUND TASK STACK SIZE                          25000
   CVB SIZE FOR FAST & SLOW POINT PROCESSORS           10000               170000
   REDUNDANCY BUFFER INCREASE (# OF 32KW BLOCKS)       1                    32768
   INCLUDE INTERNETWORK POINT PROCESSOR?     YES NO                         44050
   CVB SIZE FOR IPP                                    2000
   USER MEMORY RESERVED (# OF 32KW BLOCKS)             1                    32768
   EXTERNAL LOAD MODULES CODE (ROUNDUP FROM NEXT PAGE)                     608256
   EXTERNAL LOAD MODULES POOL1 (FROM NEXT PAGE)                            120832
                                                                         --------
 TOTAL MEMORY FOR FUNCTIONAL ADJUSTMENTS                                  1076974

+   CURRENT DATA BASE SIZE  (AMMEMTOT)                 136149.0
+   ROOM LEFT FOR DATA BASE GROWTH                     807422.0

=   TOTAL USER MEMORY AFTER SOFTWARE OPTIONS  (AMMEMAOP) 2020540.

 F1=CHECK     F3=SET OFFLINE   F5=ABORT                        F9=PACK NCF
 F2=INSTALL   F4=PRINT
```

33688

Continued on next page

Functional adjustments

The user can make entries that affect the total for functional adjustments. The user's entries allocate or "use up" the remaining available user memory. The larger the value configured for any particular functional adjustment entry, the less memory is available for other functional adjustment entries.

As shown below, if you add up all the numbers in the right hand column of the User Memory Allocation display, you will notice that they equal the total for functional adjustments.

# Background CL tasks	66200
# Concurrent Data Accesses From Background CL	2100
CVB size for Fast or Slow Point Processors	170000
Redundent Buffer Increase (# of 32KW Blocks)	32768
Include Internetwork Point Processor? YES	44050
User Memory Reserved (# of 32KW Blocks)	32768
External Load Modules Code(Roundup From Next Page)	608256
External Load Module POOL1 (From Next Page)	120832
Total Memory For Functional Adjustments	1076974

ATTENTION

ATTENTION—If network configuration is performed on-line with an AM that is in the "OK" processing state, the memory allocation of your AM is checked to ensure that your allocation does not exceed the memory available in your AM. A warning message appears if available memory is exceeded.

If the configuration is performed off-line, the configurator does not "know" what the available AM memory is. It is possible for you to configure an incorrect amount off-line and not be able to start your AM because the amount of total user memory has been exceeded.

Refer to Table 2-1 if you configure your AM off-line.

2.6 NCF Functional Adjustments—Background CL

Description

The background CL function is part of the standard AM product. CL programs that run in background will execute only if there is available AM processor time.

The first three entries for the functional adjustments relate to the background CL capability in your AM:

- # Background CL Tasks
- # Concurrent Data Accesses From Background CL
- Background Task Stack Size

The entries default to zero in the NCF, implying that you must make an entry if you plan to use background CL programs. If you accept the entries of zero, that means you do not plan to use background CL.

Number of Background Tasks

Because a background task may be delayed by a request from data in another module (for example, SET A = TIC21243.PV, where TIC21243 is a tag in a Process Manager), and because a background task is allocated only 50 milliseconds to execute when processing time is available, it is possible that the task will not run to completion. For other points to execute their background tasks, more than one background CL task should be configured.

The actual number of background tasks to configure is a tradeoff between memory and whether your CL blocks get executed. If the number is too small, the task queue will fill up and your background CL may not execute as quickly as desired; if the number is too large, you may not have sufficient memory for your other functions.

To use the background CL function, you must enter a minimum of 1. To determine if you need a larger number, consider the following:

- The amount of AM memory available.
- The number of background programs you are writing.
- Whether any software packages require background CL.

ATTENTION

Software packages such as SPQCII. require a minimum number of background tasks. Be sure to include the amount specified for your packages in your calculations.

Continued on next page

Concurrent data accesses from background CL

After determining how many background tasks are needed, you then determine the entry for concurrent data accesses from background CL. This entry specifies the total number of background CL initiated off-node data access requests or tasks that can be in progress at a given point in time.

The following calculation determines how much internal AM memory it will take to support additional programs to fetch/store off-node point**.**parameter data (data access):

> # concurrent data accesses from background CL[BKGDAREA] * 1050 words

Background task stack size

The Background Task Stack Size NCF entry specifies the size of the runtime stack for all the background CL tasks:

Range = 700 to 32000 words (32 kW)

Default = 25000

Typical Entry = 25000

Changing the background stack size requires an on-line NCF change, then an AM restart for it to take effect.

With foreground CL programs, the stack size is fixed at 15 kW, but dynamically allocated. In background CL, the stack size is adjustable by you through the NCF, but during runtime it becomes a fixed allocation.

Determining adequate stack size

The Processor Status Data Point parameter BKGSTACK returns the background stack size. The value, however, is the value you configured in the NCF display. To determine whether your stack size is adequate you need to monitor whether any stack overflows occur.

Calculation to determine background CL memory

The memory calculation for background CL is as follows:

background CL tasks [BKGTASKS] * background task stack size [BKGSTACK] + 16200

The 16200 is the AM internal memory it takes to support the background tasks (2000) plus memory support for CDS read/write (14200).

REFERENCE—Section 7 of this manual provides additional information on background CL in redundant AMs.

2.7 NCF Functional Adjustments—CVB Size (point processors)

CVBs

Prefetched and poststored data is held in Current Value Buffers (CVBs), which are the dynamic portion of heap memory. Users can change the size of the CVBs, based on an estimate of the prefetched and poststored data that is required on any processing cycle. If the CVBs are bigger, the dynamic area is bigger and less room is left for static data.

A functional adjustment entry in the AM NCF display is used to change the CVB size for fast point processors (FPPs) and slow point processors (SPPs).

Fast point processor

A point that is scheduled to run on the current processing cycle and that requires data from another node causes the fast point-processor to prefetch the data one cycle earlier (IPP off-node data is fetched on the current cycle). If the same point is to store data in another node, the data is poststored by the fast and IPP point-processor on the next cycle.

Areas of memory

The AM memory includes these basic areas:

- Fixed—Contains programs and functions and changes only on software releases.
- Stack—Memory assigned to run AM tasks. Most of this is assigned when the AM is released, however, some stack space is assigned from heap at AM startup (background CL/IPP).
- Heap—The remainder of the AM's memory. User memory is located in Heap. (The maximum amount of memory available for the heap is fixed, but the actual amount in use can dynamically change.)

Dynamic heap

The dynamic part of memory is heap that is used for LCN communications and when functions are executed in the AM.

The dynamic portion of heap is used by the system when points or CL blocks fetch or store values from or to another node.

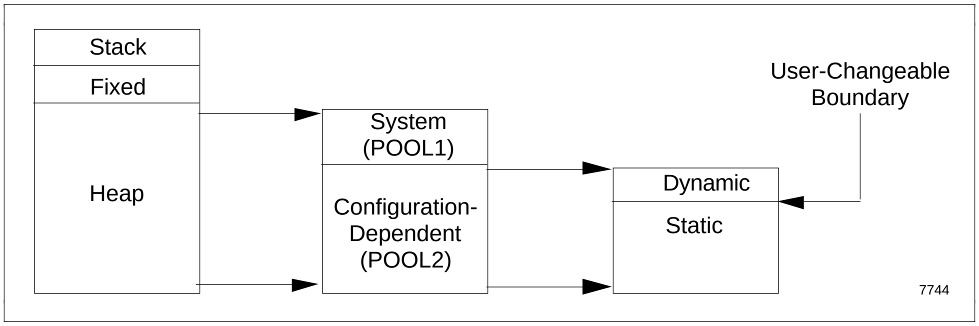
The “system” portion of heap is reserved for the system and is not available to users. The remainder of heap is configuration dependent, and consists of an area for storage of dynamic data and an area for static data.

Continued on next page

Conceptual overview of memory

The following figure provides a conceptual view of the entire AM memory, including the heap.

Figure 2-7 Conceptual Overview of AM Memory Including Heap



Estimating CVB Size

There are various ways to estimate how many points you can build and the current value buffer size needed.

2.8 NCF Functional Adjustments—Redundancy Buffer

Redundancy Buffer

In an AM redundant pair, the data from the primary AM is transferred to the secondary AM through a buffer called the redundancy buffer.

This buffer size has a base value of 122880 words. Unless an buffer overload occurs, the size of this buffer does not need to be increased.

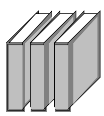
Estimating buffer size increase

In the event of a redundancy buffer overload, an AM crash or resynchronization (copying the primary AM's database to the secondary) may occur.

You need to determine whether the redundancy buffer size specified in the NCF should be increased to prevent an overload of the buffer.

AM Redundancy

AM Redundancy is a purchased option for AMs with 68020 HMPU/AMR microprocessors, and 68040 K4LCN/EAMR. AM Redundancy includes hardware and software that allow a secondary AM to back up the operation of the primary AM. To do this, all of the application data in the primary AM is transferred to the secondary AM, where an exact copy of the data is maintained to be used if the primary AM should become inoperative.



REFERENCE—Section 7 of this manual provides additional information on redundancy buffer overloads and the redundancy buffer size.

2.9 NCF Functional Adjustments —CVB Size For IPP

IPP

The AM can access data from one LCN to another through the Network Gateway. If internetwork access is desired with foreground programs, then the internetwork point processor (IPP) is required.

Configuration

To implement the IPP, the following configuration is required:

- The AM must be configured with the the internetwork capability. Selecting YES for the internetwork point processor (IPP) NCF configuration permits internetwork data access from assigned points.
- Some amount of user memory must be allocated for the IPP current value buffer.

2.10 NCF Functional Adjustments—User Memory Reserved

Description

You can reserve memory for future software options with this NCF entry.

REFERENCE—Subsection 6.2.1 shows you how to reserve AM memory for future options.

2.11 NCF Functional Adjustments—External Load Modules

Definition

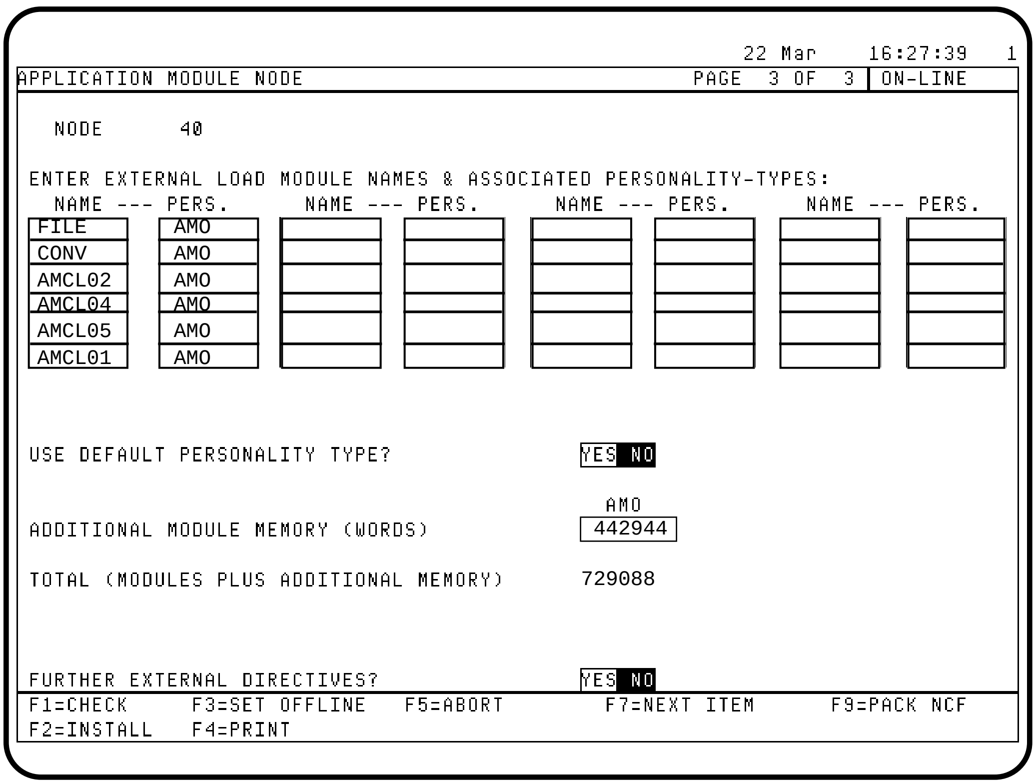
Two lines in the User Memory Allocation of page 2 of the NCF display apply to software options (External Load Modules) defined on NCF page 3. When loaded into the AM, External Load Modules become part of the loaded AM personality.

Code entries

The sum of just the External Load Module code entries on NCF page 2 equals the `ADDITIONAL MODULE MEMORY` (in Figure 2-8 this value is 729088). External Load Modules become part of the configuration-dependent portion of heap (POOL2), as indicated by this NCF entry on page 2 of the NCF display:

`EXTERNAL LOAD MODULES CODE (ROUNDUP FROM NEXT PAGE)`

Figure 2-8 External Load Module Display—NCF Page 3



33686

POOL1

This line on NCF page 2 is part of the total memory for functional adjustments:

`EXTERNAL LOAD MODULES POOL1(FROM NEXT PAGE)`

The above line means that part of the available user memory is going to be used by the system portion of the heap (POOL1). The value is automatically calculated by the system, and is done in order to have enough memory to run your External Load Module software options.

REFERENCE—Section 8 of this manual provides additional information on External Load Modules.

2.12 How To Estimate AM Memory

Requirements

To determine how many points can be built in an AM, you need to know these three things:

- The total amount of user memory available.
- The number of units configured for the AM (can be determined from your configuration data).
- The average size of each type of point.

Estimating size of point

Table 2-4 provides information needed to estimate the average size of each AM point. The table provides only an estimate. Each point has many different memory segments that are counted as "fixed;" and many optional items have more than one segment.

ATTENTION

ATTENTION—You can save considerable time and minimize calculation errors by using a spread-sheet program on a personal computer to do the several calculations described in this section.

CAUTION

CAUTION—If more than fifteen adjacent units are configured with only one point built in each unit, the AM can “hang” while trying to display points. This can happen when using either ORGANIZATIONAL SUMMARIES under the System Menu, or LIST ENTITIES IN MODULE under the DEB.

The best solution is to have more than one point per unit. Alternatively, have units with one point interleaved with units that have more than one point.

Continued on next page

Table 2-4 AM Point-Size Estimator

Point Type	Segment	Size in Words	
Flag	Fixed	151	
Numeric	Fixed	147	
Timer	Fixed	149	
		General Inputs	13+ no_gi*12
		General Outputs	4+ no_go*12
		Prefetch/Poststore	4+ no_offnode*15
		Minimum=170	
Counter	Fixed	224	
		General Inputs	13+ no_gi*12
		General Outputs	4+ no_go*12
		Prefetch/Poststore	4+ no_offnode*15
		Minimum=245	
Regulatory	Fixed	193	(see note 2)
	Ratio and Bias	15	
	Target Value	10	
	PV Filter	6	
	PV Base	41+ no_P*3	
	Setpoint	12	
	Control Base	53+ no_X*3	
	General Inputs	13+ no_gi*12	
	General Outputs	4+ no_go*12	
	Control Inputs	4+ no_ci*10	
	PV Inputs	4+ no_pi*11	
	Control Outputs	3+ no_co*13	
	Prefetch/Poststore	4+ no_offnode*15	
	CL	4+ no_CL*7	(see note 1)
	PV Algorithms		
	DataAcq	3	
	MidOf3	4	
	HiLoAvg	4	
	GenLin	56	
	Totalizer	16	
	FlowComp	EqA=15, EqB=21, EqC=23, EqD=23, EqE=25	
	Summer	EqA=7, EqB=7 + no_p*2 (23 wds, max)	
	MulDiv	EqA=15, EqB=35, EqC=35	
	SumProd	EqA=23, EqB=35	
	VdtLdLag	EqA=41, EqB=101, EqC=101, EqD=101	

Continued on next page

Table 2-4 AM Point-Size Estimator, continued

Point Type	Segment	Size in Words
Regulatory, continued	Control Algorithms	
	Pid	29
	PidErfb	31
	PidFF	36
	LeadLag	20
	AutoMan	9
	RatioCtl	11
	MulDiv	23
	RampSoak	64
	IncrSum	26
	OrSel	4
	Switch	4
	CL	3
	Summer	EqA=7, EqB=7 + no_x*2 (15 wds, max)
	Misc. Segments	
	Switch	4+ no_swh*29
	Switch Alarm	15
	CDS	4+ no_cds*2 (see note 1)
	Pkgname	2+ no_pkg*4
		Minimum=193
Custom	Fixed	
	CL	4+ no_cl*7 (see note 1)
	Prefetch/Poststore	4+ no_offnode*15
	CDS	4+ no_cds*2 (see note 1)
	PkgName	2+ no_pkgs*4
		Minimum=153
Switch	Fixed	143
	CL	4+ no_cl*7 (see note 1)
	Prefetch/Poststore	4+ no_offnode*15
	Switch Segment	4+ no_switch*29
	Switch Alarm	15
	CDS	4+ no_cds*2 (see note 1)
	PkgName	2+ no_pkgs*4
		Minimum=205

Note 1: This table does not include the sizes of CL blocks and custom data segments. These are provided in B.2 "CL/CDS Capacities" in the *CL/AM Reference Manual.*

Note 2: AM regulatory points are computed differently than the other points. All items after fixed size are optional.

Continued on next page

Table 2-4 AM Point-Size Estimator, continued

LEGEND:

no_x: Number of x inputs
no_p: Number of P inputs
no_gi: Number of general-input connections (8 max)
no_go: Number of general-output connections (8 max)
no_ci: Number of control-input connections (8 max)
no_co: Number of control-output connections (8 max)
no_offnode: Number of references to parameters in another node, rounded up to the nearest multiple of 3. Off-node control-output connections take 5 off-node references in the prefetch/poststore segment. A point in a different unit in the same node must be considered as an off-node point.
no_switch: Number of Switches — Minimum of 1
no_cl: Number of CL blocks
no_cds: Number of CDSs
no_pkg: Number of CL packages
EqA – EqE: Equation A, B, C, D, or E

Example

As an example of the use of Table 2-4, consider a regulatory point with a Summer PV algorithm using equation B and two PV inputs, a PID control algorithm, and one control output. The average size is as follows:

Fixed			193
PV Base	= 41 + 2*3	=	47
Control Base	= 53 + 1*3	=	56
PV Input	= 4 + 2*11	=	26
Control Output	= 3 + 1*13	=	16
PV Summer	= 7 + 2*2	=	11
Pid			29
Setpoint			12
	TOTAL		390 words

If all connections are to points in another node (off node):

Prefetch/Poststore	=	4 + (5 + 2, rounded up to a multiple of 3)*15
	=	4 + 9*15 = 139.
390 + 139	=	529 words.

2.13 Optimizing CVB Size

Purpose

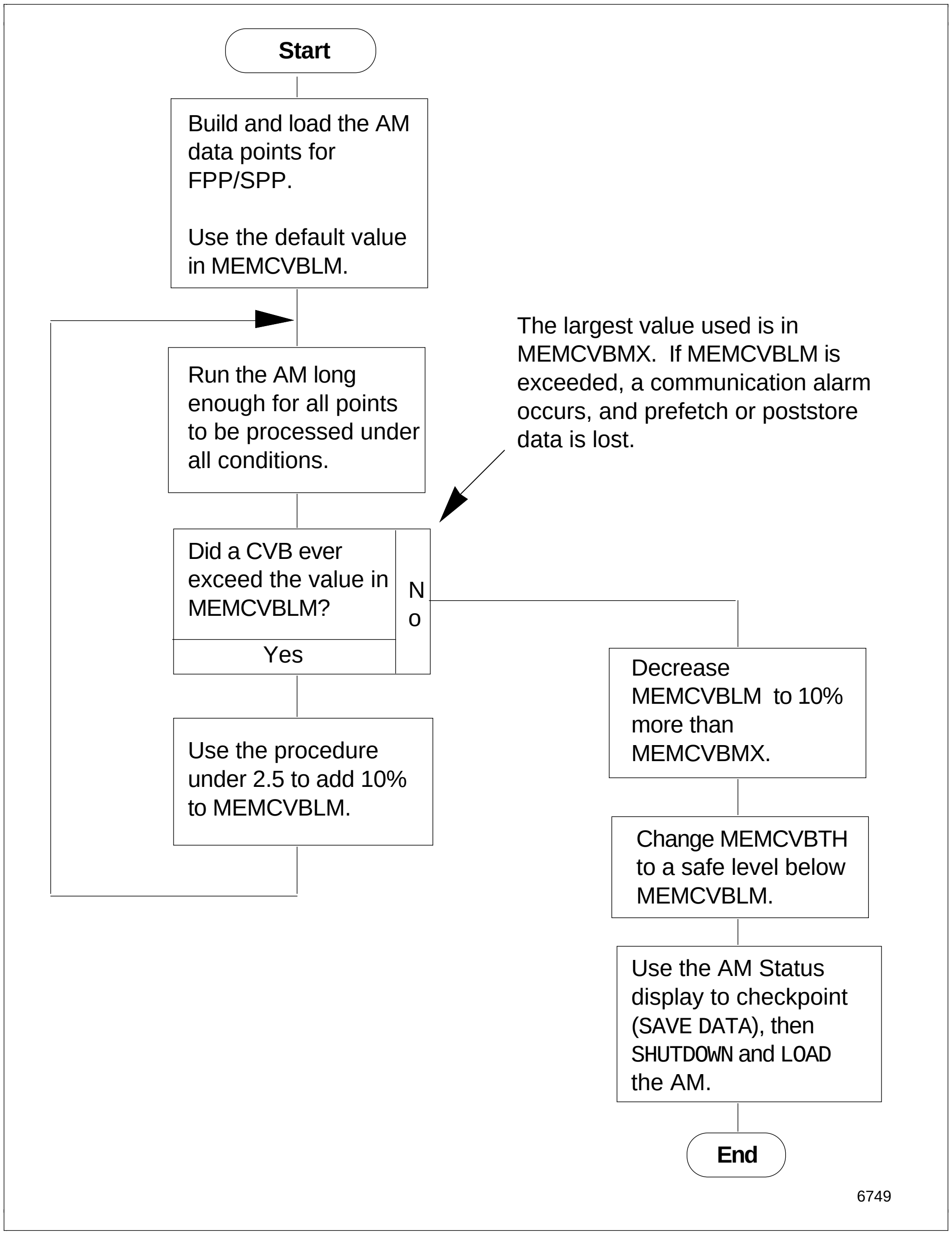
If there was some assurance that the greatest use of FPP/SPP CVB memory that could ever occur in an AM had already occurred and that value was in MEMCVBMX, you could set MEMCVBLM to that value, change MEMCVBTH to a value below MEMCVBLM by some safe amount, checkpoint, and reload the AM. This is not likely though, because points and CL structures can be added at any time. They are two methods for estimating the best value for the CVB size. The value is then entered into the AM's NCF display.

Method 1

One way to estimate the best value for the FPP/SPP CVB size is to build the database, load and run the AM long enough for all points to be processed, then evaluate the memory-allocation data. Figure 2-9 shows this process.

Continued on next page

Figure 2-9 A Method for Estimating Memory-Allocation Values



ATTENTION

ATTENTION—If you try to increase the value in MEMCVBLM for an AM whose static heap is as full with configuration data as it can be, the new value is rejected. In such a case, you have to delete units, points, or CL structures, increase MEMCVBLM, checkpoint, and reload the AM.

Continued on next page

Method 2

Another method for estimating the maximum CVB size is to evaluate the points that refer to points in other nodes. Table 2-5 shows the number of CVB words required for each type of prefetched and poststored data.

Table 2-5 CVB Words vs Data Types

Prefetched or Poststored Data Type	**CVB Words Used** (see Note 1)	
Inputs related to control-output connections	15	
PV fetched by a control or PV-input connection	8	
Real Value input or output by CL, control, or general I/O	4 (single value)	2*n + 4 (array) (see Note 2)
Enumeration value	9 (single value)	2*n + 4 (array) (see Note 2)
Parameter ID	3	
Entity ID	6 (single value)	4*n + 6 (array) (see Note 2)
Boolean value	3 (single value)	n + 3 (array) (see Note 2)
Integer value	3	
Time value	5 (single value)	3*n + 5 (array) (see Note 2)
String value	3 + 1 per 2 characters	

Note 1. For any output (poststore), add five words to the result.

Note 2. CL prefetches an entire array; n represents the number of elements in the array. For array poststores, individual elements are stored.

Continued on next page

Method 2 example—single point estimate

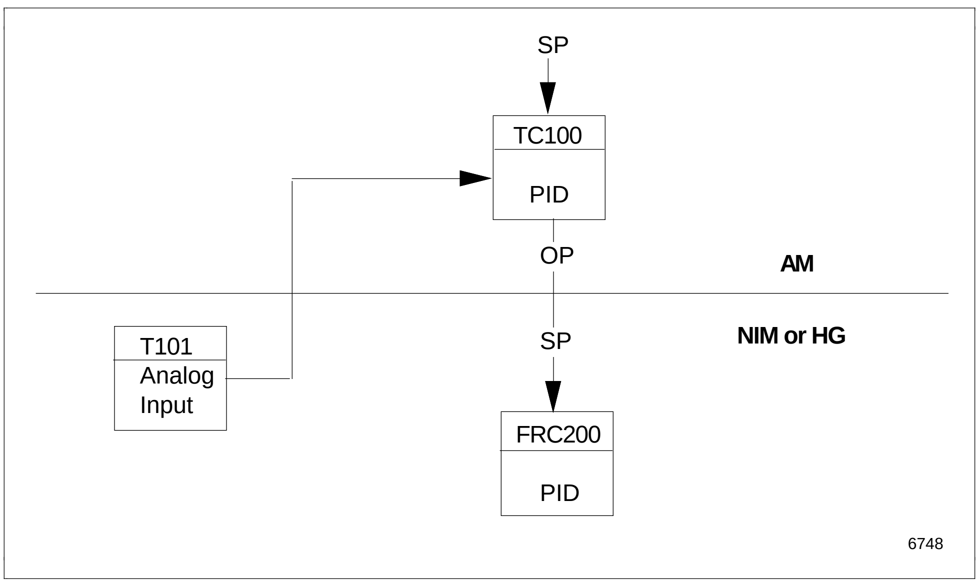
Consider the following processing example of the AM point in Figure 2-10:

- If TC100 is scheduled to be processed once each second, its PV input is prefetched from T101 and stored in a CVB one cycle (one-half second) before TC100 is processed. Because TC100's OP value can be initialized by FRC200, additional information is prefetched from FRC200 and stored in the CVB.
- One cycle after TC100 is processed, its OP value is poststored in FRC200 after having been stored in the CVB; therefore, the CVB has to hold the prefetched data until the next time TC100 is processed, while it is also holding the poststored information from TC100's last processing pass.

Now we apply the values in Table 2-4 to this example:

Inputs related to the control-output (OP) connection	= 15
The PV input	= 8
The real-number output (OP) (5 added because it is an output)	= 9
CVB words for this point	**= 32**

Figure 2-10 Example of Estimating Memory-Allocation Values



Continued on next page

Method 2 example—estimate for entire AM

To estimate the maximum CVB needed for the entire AM, evaluate the AM process database, as in the following example.

If the fully configured AM is to do 150 prefetches and poststores for each processing interval, and 30% of these are inputs related to control-output connections, 50% are PV inputs, and 20% store real-number outputs, then the CVB needed is calculated as follows:

$$CVB = 150*(30\%*15 + 50\%*8 + 20\%*9) = 1545 \text{ words.}$$

This value does not include unusual processing, such as process-special and CL fetches and stores caused by some alarm condition. So to be safe, MEMCVBLM might be set to 2000 words.

ATTENTION

ATTENTION—Remember that data transfers within the same AM do not require prefetches and poststores, so they do not use CVBs.

ATTENTION

ATTENTION—Increasing the CVB size reduces the static heap available for configuration data. Decreasing the FPP/SPP CVB size increases the risk of threshold (MEMCVBTH) alarms, and worse, it increases the risk of losing FPP/SPP prefetched and poststored data. Decreasing the IPP CVB size increases the possibility of IPP processing overruns.

If you have ample memory for your data points and CL structures and if you never experience FPP/SPP CVB alarms and have few or no IPP CVB overrun alarms, there is no need to change the CVB size.

Continued on next page

Procedure

Table 2-6 describes the procedure to change the CBV size.

Table 2-6 Procedure to Change CVB Size

Step	Action
1	Specify a backup NCF pathname on the default volume pathnames display.
2	On the Engineering Main Menu, select LCN NODES RESULT: The LCN Node Configuration display appears.
3	Select the node number of the AM whose MEMCVBLM and/or MIPCVBLM value is to be changed (may need to press [PAGE FWD]) RESULT: The Application Module Node display appears.
4	Select MODIFY NODES. RESULT: Page 2 of the Application Module Node display appears.
5	To change MEMCVBLM, key in a new value in the port labeled `CVB SIZE FOR FAST & SLOW POINT PROCESSORS`, and press [ENTER]. To change MIPCVBLM, key in a new value in the port labeled `CVB SIZE FOR IPP`, and press [ENTER]. RESULT: The AM Node display reappears with the new value in blue (cyan). If there are errors in the value you entered, they are noted in red. The values in the MEMORY USED column also change.
6	Hold [CTL] and press [F1] to run the checker. RESULT: The “Modify Node” display appears with instructions for installing and loading the change you just made.
7	Follow the instructions, including shutting down and reloading the AM). RESULT: PSDP parameters MEMCVBLM and MIPCVBLM assume the current values, and the memory reserved for CVBs changes accordingly.

2.14 AM Database Size

Description

The AM database size is determined by the following:

- Point Data,
- Largest Checkpoint Buffer,
- CL Blocks, and
- Custom Data Descriptors.

Example

Assume that the average number of words per point is 300, including a factor for shared and unshared CL blocks and custom data descriptions (This assumes some regulatory control points, some simpler points, and a few points with little CL.)

Further assume for the purpose of the calculation of the checkpoint snapshot buffer size, that all units in the AM are the same size.

Then the following list indicates the number of points that would fit into a 3 Mw AM as a function of the number of units:

Number of Units	Total Number Of 300-Word Points*	Number of Units	Total Number of 300-Word Points*
1	732	11	1239
2	972	12	1236
3	1113	15	1219
4	1150	16	1218
5	1188	30	1165
6	1212	45	1082
9	1239	62	980
10	1240		

*Assumes an equal number of points per unit.

ATTENTION

ATTENTION—Clearly, the above example is a simplification. The number of points that can fit into an AM depends on the following additional factors:

- The average complexity of the points, including CL and custom data use.
- The number of units (small impact).
- The size of the largest unit (large impact).

ADDING AND MOVING AM PROCESS UNITS
Section 3

This section provides guidelines for adding units to an AM, and for moving units from one AM to another.

3.1 HOW TO ADD A NEW UNIT TO AN AM

WARNING

If more than fifteen adjacent units are configured with only one point built in each unit, the AM can “hang” while trying to display points. This can happen when using either ORGANIZATIONAL SUMMARIES under the System Menu, or LIST ENTITIES IN MODULE under the DEB.

The best solution is to have more than one point per unit. Alternatively, have units with one point interleaved with units that have more than one point.

New units are added to an AM by adding the unit in Network Configuration (`UNIT NAMES` and `LCN NODES`), and then restarting the AM with the revised NCF, using the checkpoint last saved for the AM as the data source. To add a unit to an AM, you must delete the node (AM) and then add it back with the desired changes. Refer to *Network Data Entry,* Table 7-25 to delete an AM and *Network Data Entry,* Table 7-24 to add an AM.

When an AM is started the first time, before any checkpoint files with user data are available, you use the null-checkpoint volume supplied by Honeywell. This creates the node directory for this AM (&Znn), but no unit directories (&Fuu) exist yet. The &Fuu directories and checkpoint files for configured units are created the first time a checkpoint is saved for the AM.

Use the following steps to add the new unit directory (the AM's checkpoint volume will have to be large enough to accept the new directory—see Section 6). In these steps,

`uu` is the unit index. The index is the position you configured on the Unit Names Display, not your Unit ID. For example, if BB was the third unit configured on the Unit Names Display, the index is 03 (for `uu` = 100, enter `00`).

`np` is the node pair number for the HM that contains the node directory (&Znn) for the AM.

1. Use the Command Processor to create the checkpoint directory for the new unit. This is the form of the command line:

```
CD NET>&5np> &Fuu
```

2. Copy the master checkpoint file for this unit from the &AMC cartridge or floppy.

   ```
   CP $Fn>&AMC>AM0uuMAS.CP NET>&Fuu>= -V
   ```

3. NULL checkpoints are write protected on the distribution media. They must be unprotected or the checkpoint will fail.

   ```
   UNPROT NET>&Fuu>*.*
   ```

4. Reconfigure the network to establish the new unit (see *Network Data Entry* in the *Implementation/Startup & Reconfiguration - 1* binder). Then use the AM Status display to shutdown and reload the AM, using the last checkpoint saved for this AM as the data source for reloading (use any one, or the restart options—hot, cold, warm, or no processing). The remaining checkpoint files for the new unit are established in your new &Fuu directory.

3.2 HOW TO MOVE UNITS FROM ONE AM TO ANOTHER

WARNING

Moving units requires caution. For example, if you use AMCL05, the AM Extension for Off-Logical-Node-Access, and subsequently move a unit containing any of the involved references off-physical-node (to another AM), the AM will not be able to perform the references. At startup, you will get one or both of the following status messages indicating the units with invalid references.

```
Bad Direct References on unit: xx    (most likely message)
Bad Indirect References on unit: xx
```

You may also get a WARNING node status, depending on the nature of the reference. If the situation is not corrected, you will get configuration errors at run time. If this error occurs, you will have to unlink the CL, relink the CL, checkpoint, and reload the AM.

To move a unit from one AM to another AM, you must delete the unit from the first AM and then add the unit to the second AM. To add or delete a unit from an AM, you must delete the node (AM) and then add it back with the desired changes. Refer to *Network Data Entry,* Table 7-25 to delete an AM and *Network Data Entry,* Table 7-24 to add an AM.

If you move a unit from one AM to another, the old checkpoint files can be used to reload the AMs. Use the procedures that follow.

On a system without HMs—After the system is restarted with the new NCF, when you reload the AM from a removable medium, a prompter requests the checkpoint medium that contains the "new" unit. You should then mount the cartridge or floppy that has the checkpoint from the AM where the unit previously resided and continue the loading process.

On an HM-based system

- **Case 1**: The original AM checkpoint volume and the new AM checkpoint volume reside on the same physical HM.

  Action: None.I

- **Case 2**: The original AM checkpoint volume and the new AM checkpoint volume reside on different physical HMs. Note: For this case, be **sure** that you delete the unit from the original AM—the system does not check for duplicate unit numbers in AMs that have checkpoint volumes on different HMs.

  Action: Use the Command Processor to move the unit's checkpoint directory from the original AM's checkpoint volume to the new AM's checkpoint volume (the new AM's volume will have to be large enough to accept the new directory—see Section 6). Use steps 1 through 5, below, to do this.

> **WARNING**
>
> In the following procedure, be very sure that you use the correct unit index (uu). If you enter a wrong number, you might move the wrong checkpoint or you might delete the wrong checkpoint.

In the following steps,

`uu` is the unit index. The index is the position you configured on the Unit Names Display, not your Unit ID. For example, if BB was the third unit configured on the Unit Names Display, the index is 03 or 003 (for `uu` = 100, enter `00`).

`np` is the node pair number for the original HM, and `mp` is the node pair number for the new HM.

1. Use the AM Status Display to do a demand checkpoint (`SAVE DATA`) to a cartridge or floppy.

> **WARNING**
>
> The next step deletes the HM copy of the checkpoint (database). Until you restore the checkpoint on the new HM (step 5), you will be completely dependent on the checkpoint copy on your cartridge or floppy. For additional security, make a backup copy of the checkpoint on another cartridge or floppy, before going on to step 2.

2. On the original HM, delete all the files in the directory with the checkpoint for this unit. This is the form of the command line:

```
DL NET>&Fuu>*.*
```

3. Delete the directory from which you just deleted the files.

```
DD NET>&Fuu>
```

4. Create a checkpoint directory for the unit on the new HM.

```
CD NET>&5mp> &Fuu
```

5. Copy the checkpoint from the removable medium to the new directory on the new HM.

```
CP &Fn>&Fuu>*.* NET>&Fuu>=
```

NOTE

If different versions of the same custom data segment (CDS) exist in two different units in the same AM (because one of the units has been moved into this AM), on startup, the points with the older segments are alarmed, and those segments are deleted. If this occurs, to reestablish a valid database, you must rebuild the points affected, reattaching the proper segment.

USER-CORRECTABLE "SOFTWARE" ERRORS
Section 4

This section provides guidelines for recovering from errors reported in the real-time journal by AM software tasks.

There are several errors that can occur while an AM is being loaded or while it is on-line. The error messages contain the word SOFTWARE, but these errors are not problems in the software. They are conditions that can often be corrected by the user.

4.1 ERROR MESSAGE FORMAT

The error messages are real-time journal messages. They are printed (journalized) on a printer in this form:

```
(time)(node type/number)(task name)SOFTWARE(three hex nos.)(4 decimal nos.)
```

In most cases, an auxiliary status message containing additional information appears on the Auxiliary Status display and also is journalized preceding the error message.

4.2 INTERPRETATION AND REMEDIES

Table 4-1 — Errors and Remedies

Task Name	Leading Decimal Digits	Meaning	Remedy
B$IPP B$EVD B$NST L$CLL data_access B$PRSTS	419 121 195 or 309 29, 41, or 45 14, (integer),0, 13501 224	AM ran out of memory during a load.	Add memory to the AM (see Section 2) or change Network Config. to reduce the number of units.
B$NST	336 and the last decimal digit is 4	Master file for a unit is missing from the HM during a load.	Determine the missing file and copy from a cartridge or floppy to the HM.
B$NST	336 and the last decimal digit is 5	Time out while waiting for a volume to be mounted during a load.	Start over
L$CLL	37 or 38	File required for a load is missing.	Determine missing file and copy to startup medium (cartridge, floppy, or HM).

(Continued)

Table 4-1 — Errors and Remedies (continued)

Task Name	Leading Decimal Digits	Meaning	Remedy
L$CLL	49	Invalid CL catalog encountered during a load.	Start over using valid catalog.
B$RED_TASK	239	Primary-Secondary personality revision/ version mismatch	Restart both primary and secondary with same personality version/revision.
B$RED_TASK	240	Redundancy hardware failure	Replace AMR board in primary AM, or replace AMR board in secondary AM.
B$RED_TASK	241	LCN failure	Refer to *LCN Guidelines*. in the *LCN Site Planning & Installation* binder.
(Any Task)	252	Redundancy hardware failure	1st check AMR board in secondary AM, then check AMR in primary AM.
B$RED_TASK	245	Redundancy hardware failure on start up	1st check AMR board in secondary AM, then check AMR in primary AM.
B$RED_TASK	347	Primary and secondary memories do not compare.	1st check AMR board in secondary AM, then check AMR in primary AM.
B$RED_TASK	238	Resynchronization during secondary startup	Try the startup again.
B$RED_TASK	250	Secondary buffer over-flowing repeatedly	Add to redundancy buffer size (see 7.1.3.1).
Data_Access	265	Primary switchover during secondary startup	Restart both primary and secondary.

(Continued)

Table 4-1— Errors and Remedies (Continued)

Task Name	Leading Decimal Digits	Meaning	Remedy
B$NST or B$RED_TASK	266	Primary and secondary memory mismatch	Check memory boards in primary and secondary.
B$RED_TASK	296	Repeated redundancy bus errors	Check AMR board in second-dary AM.
B$RED_TASK	242, 1, -5, 17	Secondary ROM full error	Check AMR board in second-dary AM.
B$NST	389	Illegal node config., 68000 microprocessor used in one of the redundant AMs.	If the boards are different, replace the older microprocessor with an HMPU or K4LCN board.
B$NST	335, and the last two decimal digits are 19	File for unit refer-enced in master file missing during a load.	Retry, use backup checkpoint, or call Honeywell.
B$PRSTS	226	Point-processor overrun.	AM is overloaded. Either reduce the load by setting points inactive, or reconfigure the AM with fewer units.
B$PRSTS	227	CL block in an infinite loop or point-processor overrun	If you can't determine which CL block has an infinite loop, call Honeywell. If you have a processor overrun, see the remedy above.
data_access	26,0,0,13705	During startup, a CDS's rev. code doesn't match an existing CDS.	Call Honeywell.
data_access	16,(intgr.),0,13503 or 4,(integer),0,1309 or 4,(intgr.),0,13098	Unrecoverable file error during a load	Copy DSD files from $ASY floppy to HM and retry. (For a system with no HMs, retry with a new &ASY cartridge or floppy.)

(Continued)

Table 4-1— Errors and Remedies (Continued)

Task Name	Leading Decimal Digits	Meaning	Remedy
data_access	15,(intgr.),0,13502 or No DSD (integer) or 4,(intgr.),0,13096 or 4,(intgr.),0,13099	DSD file required for a load is missing.	Copy DSD files from $ASY floppy to HM and retry. (For a system with no HMs, retry with a new &ASY cartridge or floppy.)
data_access	15,(intgr.),0,23903	File DUPLICAT.SP missing from &ASY vol.	Reload a US with Operator Personality or Eng. Personality to recreate the missing file.
data_access	16,(intgr.),0,23905 or 16,(intgr.),0,23907 or 18,(intgr.),0,23909	Unrecoverable file error during a load.	Delete the DUPLICAT.SP file from &ASY and reload a US with Operator Personality or Eng. Personality to recreate DUPLICAT.SP on &ASY.
data_access	4,(intgr.),(intgr.), 29306	File DUPLICAT.SP is wrong version	Delete the DUPLICAT.SP file from &ASY and reload a US with Operator Personality or Eng. Personality to recreate DUPLICAT.SP on &ASY.

OBTAINING AM SCHEDULE INFORMATION
Section 5

This section provides instructions for obtaining a report that you can use to analyze AM point scheduling.

5.1 INITIATING THE AM SCHEDULE DUMPER

An AM Schedule Dumper is available to provide you with a printed or displayed report that provides cycle-loading and scheduling information for the data points in an AM unit. The report shows loading information for each processing cycle and it shows the processing order (schedule) for each point on each processing cycle. You can use the information in the report to determine if you should rearrange processing schedules or if you need to adjust processing-cycle loading. Refer to 2.2.1 in *Application Module Control Functions*, in the *Implementation/Application Module - 1* binder, for information on AM point-processing schedules and processing-cycle loading.

The dumper places the report in a file in a user-specified volume on an HM. You can use the Command Processor Print command to display or to print the content of the report.

Do this to dump the schedule information for a unit in an AM:

1. In the following PSDP parameter, store a complete pathname that points either to an existing file, or to a file you want to create, on one of your HM user volumes or directories; (see 22.1 in the *Engineer's Reference Manual* for a method for using a schematic display to do this):

   `$PRSTSnn.AMSCHDMP(uu)`

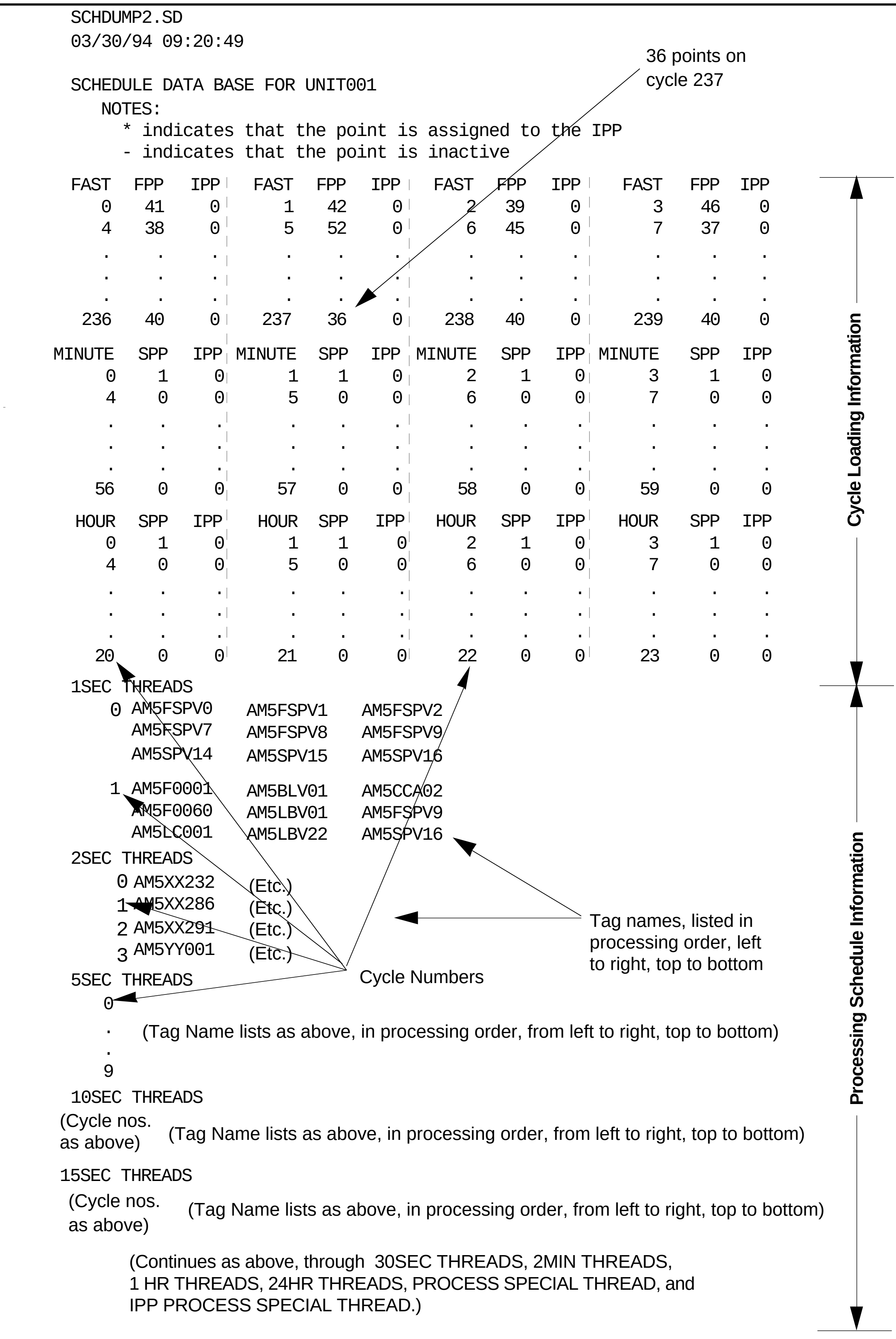
   where `nn` is the AM node number and `uu` is the unit index number (not the unit name). You can see a list of the unit index numbers and the unit names through the `UNIT NAMES` pick on the Engineering Main Menu. Use 00 as the unit index number to dump the schedule information for all units.

   The pathname is in this form (schedule dumps to a removable medium are not allowed).

   `NET>usrv>file.zz`

2. As soon as the pathname is stored, the schedule for the unit named begins to be dumped into the user file. An operator message is generated if any errors are detected during the dump operation, and a message is generated when the dump is complete. You can see these messages by pressing the MSG SUMM key on the Operator's Keyboard of a US that is running the Operator Personality and has an area to which the AM unit is assigned.

3. To display or print the schedule information execute a Command Processor Print command (see 5.21 in *Command Processor Operation*, in the *Implementation/Engineering Operations - 1* binder). Figure 5-1 shows the form and content of the schedule report.



```
SCHDUMP2.SD
03/30/94 09:20:49

SCHEDULE DATA BASE FOR UNIT001
   NOTES:
     * indicates that the point is assigned to the IPP
     - indicates that the point is inactive

 FAST  FPP  IPP | FAST  FPP  IPP | FAST  FPP  IPP | FAST  FPP  IPP
    0   41    0 |    1   42    0 |    2   39    0 |    3   46    0
    4   38    0 |    5   52    0 |    6   45    0 |    7   37    0
    .    .    . |    .    .    . |    .    .    . |    .    .    .
    .    .    . |    .    .    . |    .    .    . |    .    .    .
    .    .    . |    .    .    . |    .    .    . |    .    .    .
  236   40    0 |  237   36    0 |  238   40    0 |  239   40    0

MINUTE SPP  IPP |MINUTE SPP  IPP |MINUTE SPP  IPP |MINUTE SPP  IPP
    0    1    0 |    1    1    0 |    2    1    0 |    3    1    0
    4    0    0 |    5    0    0 |    6    0    0 |    7    0    0
    .    .    . |    .    .    . |    .    .    . |    .    .    .
    .    .    . |    .    .    . |    .    .    . |    .    .    .
    .    .    . |    .    .    . |    .    .    . |    .    .    .
   56    0    0 |   57    0    0 |   58    0    0 |   59    0    0

 HOUR  SPP  IPP | HOUR  SPP  IPP | HOUR  SPP  IPP | HOUR  SPP  IPP
    0    1    0 |    1    1    0 |    2    1    0 |    3    1    0
    4    0    0 |    5    0    0 |    6    0    0 |    7    0    0
    .    .    . |    .    .    . |    .    .    . |    .    .    .
    .    .    . |    .    .    . |    .    .    . |    .    .    .
    .    .    . |    .    .    . |    .    .    . |    .    .    .
   20    0    0 |   21    0    0 |   22    0    0 |   23    0    0

1SEC THREADS
  0 AM5FSPV0   AM5FSPV1   AM5FSPV2
    AM5FSPV7   AM5FSPV8   AM5FSPV9
    AM5SPV14   AM5SPV15   AM5SPV16

  1 AM5F0001   AM5BLV01   AM5CCA02
    AM5F0060   AM5LBV01   AM5FSPV9
    AM5LC001   AM5LBV22   AM5SPV16
2SEC THREADS
  0 AM5XX232   (Etc.)
  1 AM5XX286   (Etc.)
  2 AM5XX291   (Etc.)
  3 AM5YY001   (Etc.)
5SEC THREADS
  0
  .   (Tag Name lists as above, in processing order, from left to right, top to bottom)
  .
  9
10SEC THREADS
(Cycle nos.
as above)   (Tag Name lists as above, in processing order, from left to right, top to bottom)
15SEC THREADS
(Cycle nos.
as above)   (Tag Name lists as above, in processing order, from left to right, top to bottom)

(Continues as above, through 30SEC THREADS, 2MIN THREADS,
1 HR THREADS, 24HR THREADS, PROCESS SPECIAL THREAD, and
IPP PROCESS SPECIAL THREAD.)
```


Figure 5-1 — Schedule Report

5.2 INTERPRETING THE AM SCHEDULE PRINTOUT

The Schedule Data Base printout is organized by units. For each unit, the information is divided into two major parts.

- Cycle Loading Information—tabular information that shows the number of points that will be accessed on each processing cycle.
- Point Scheduling Information—lists that show the individual points that are assigned to each of the processing threads by frequency and by cycle.

5.2.1 Format of the Cycle Loading Information

There are three sets of cycle loading tables. One each for FAST, MINUTE, and HOUR processing. The cycle processing tables are further organized into four sets of three columns each.

Columns headed “FAST” contain cycle numbers
0-239 for two minutes of 1-second to 2-minute FPP processor threads, and
0-23 for two minutes of 5-second to 2-minute IPP processor threads.

Columns headed “MINUTE” contain cycle numbers
0-59 for one hour of 1S-minute to 1-hour SPP processor threads, and
0-59 for one hour of 5-minute to 1-hour IPP processor threads.

Columns headed HOUR contain cycle numbers
0-23 for 24 hours of 8-hour to 24-hour SPP or IPP processor threads.

The columns headed “FPP” or “SPP” contain the counts of total number of points of that type that are processed on that cycle. For example, the first entry for FPP in the “FAST” cycles portion shows the total number of 1-second to 2-minute points for FPP Cycle 0, while the first entry under SPP in the “MINUTE” cycles portion shows the total number of 1S-minute to 1-hour points for SPP Cycle 0.

The columns headed “IPP” contain the counts of total number of points of that type that are processed on that cycle. For example, the first entry for IPP in the “FAST” cycles portion shows the total number of 5-second points to 2-minute points for IPP Cycle 0, while the first entry for IPP in the “MINUTE” cycles portion shows the total number of 5-minute to 1-hour points for IPP Cycle 0.

5.2.2 Format of the Point Scheduling Information

The point scheduling lists consist of three columns of point names listed in processing order from left to right and top to bottom. They are organized by threads (1-second through 24-hour types) and by cycle within the thread type.

For example, the 1-second threads are subdivided into cycles 0 and 1, while the 5-second threads are subdivided into cycles 0 through 9 for FPP points and cycles 0 and 1 for IPP points. FPP and IPP points are both listed, but IPP points are differentiated by a “*” prefix. The process special thread and IPP process special thread show snapshots of the thread contents at the time the dump is requested, and may contain points for units other than the unit requested.

Note that only those cycles that have points assigned will be shown on this printout.

5.2.3 Additional Information About AM Point Scheduling

The automatic scheduling of point processing with multiple units in the same AM is independent on a per unit basis. With multiple units where each unit has one AM point scheduled at a one-second frequency, all of these points would execute on the same cycle (zero). The reason for this separation of unit scheduling is that AM unit databases can be moved to or from other AMs and this separation is required to maintain the unit database order.

The schedule dumper is a tool that tells you how many AM points are processed during a one-half second (fast processor), five-seconds (internetwork processor/IPP), and a minute or hour (slow processor/IPP). The BEFAFT parameter option of CYCLE can be used so NORMCYCL parameter can specify the processing cycle.

The NORMCYCL definition in the AM Parameter Dictionary shows the base number of cycles for FPP is 240 within a two-minute period, 24 cycles for IPP within the same two-minute period, and two cycles for SPP with the same two-minute period. The faster FPP points repeat their periodic scheduling within this 240 cycles (for example: the 120 one-minute cycles would repeat twice within the two-minute period).

The SPP one-minute points repeat twice within the 240 FPP cycles, the SPP two-minute points execute once within this 240 cycles and the larger SPP periods would process only a portion of their points within this 240 cycles. The SPP point scheduling is spread across the one-half second periods during the FPP two minutes and will remain on a particular one-half second cycle until a point rebuild or scheduling change happens. You cannot easily determine the SPP points processed at each FPP one-half second.

There is a time relationship between SPP and FPP because the FPP processor executes the scheduling to initiate point processing for both FPP and SPP a one-half second before the actual processing occurs.

Both SPP and FPP scheduling starts at the same absolute time (AM reload time), however, the actual execution of processing (algorithms and CLs) on FPP and SPP points does not occur at the same time. You can tell the AVERAGE points per second SCHEDULED on SPP and FPP by looking at parameter PRAVGC, however, this is not the same as the actual points which completed processing in the same one-half second. What points get scheduled by FPP to a particular one-half second can change for reasons other than point rebuilding and may affect the processing average (such as: process special PPS requests and point activation PTEXECST status).

The faster IPP points repeat their periodic scheduling within these same FPP 240 cycles (for example: the 12 one-minute cycles would repeat twice in a two-minute period). IPP point scheduling is independent of FPP and SPP scheduling (the cycles are not synchronized).

There are Data Entity Builder (DEB) AM point building tools that allow users to override the AMs automatic assignment to the least loaded cycle in the schedule (BEFAFT, BEFAFTID, NORMCYCL). The NORMCYCL option has a one-half second resolution for FPP, a five-second, one-minute, and one-hour resolution for IPP, and a one-minute and one-hour resolution for SPP. For the SPP cycle assignment the AM automatically assigns the SPP points to the same one-half second cycles used by FPP.

You can relate cycle PSDP parameter values to clock time. The PRMCYCL, PRMXCYCP, PRMNCYCC, and PRMNCYCP start counting when the AM is loaded. the values reset to zero when the AM is shutdown and reloaded, however, the values continue to count on a failover to a redundant AM. The absolute time for the AM load can be seen in the Event History for AM node load or can be displayed on the AM NODE "Status Detail" display page but is not available as a parameter.

You should be able to attach absolute processing cycle to a real system time by

- dividing the cycle value by 7200 (the number of cycles in an hour),
- then add the quotient (number of hours) to the AM reload time hours,
- convert the cycles remainder to minutes (divide by 120 [the number cycles in a minute]), and
- then add to the reload time minutes.

Subtracting these quotient and remainder times from current time should also approximate the AM reload time within the hour. The AM system cycle counter is continuously updated and would roll over at 4294967295 cycles (or 24855 days). The PRMNCYCC, PRMNCYCP, and other PSDP values are converted to a 24-hour value (or 172800 cycles).

5.2.4 AM Overrun Diagnostic Display (SMCC)

A display called "AM OVERRUN DIAGNOSTIC" is provided in the SMCC main menu. This display may be called if an AM has failed and it will show the point name and CL block name (if applicable) that the AM was processing at the time of the failure.

In addition, new PSDP parameters have been added to determine the points which take the longest time to execute. Separate lists will be maintained for the fast and slow point processors. The longest running CL block name along with its asociated point will also be saved. Another parameter of the AM PSDP is used to switch the collection of the other parameters ON or OFF. The switch parameter will not switch off the saving of the point name and CL block name that is used for the crash statistics.

A running AM may be diagnosed by using the AMDIAGNS display from the Toolkit set of schematics directory TLK1. Additional information on the use of this Toolkit is provided in the "System Performance Display" section of the *Engineer's Reference Manual*.

Additional information on the use of the SMCC AM Overrun Diagnostic display is provided in the "Using SMCC Displays" section of the *Maintenance Test Operations Manual*.

AM CHECKPOINTS
Section 6

This section provides guidelines for establishing AM checkpoint volumes, determining options available, and tells how to reserve user memory for future options.

6.1 ESTABLISHING AM CHECKPOINT VOLUMES

Like other checkpoint volumes, AM checkpoints can exist on removable media (cartridges or floppy disks) and on HMs. General guidelines and information on checkpointing are provided in Section 21 of the *Engineer's Reference Manual*.

The table labled "HM Volume Size Estimator" in Section 7 of the *Engineer's Reference Manual* defines how to estimate the size of AM checkpoint volumes. A significant factor in the size of such volumes is the amount of user memory in the AM. You should note that the user memory sizes in this table are expressed in kb rather than words, and therefore are approximately twice the values given under Section 2 of these *Application Module Implementation Guidelines*. Also note that each AM checkpoint volume on an HM must be large enough to simultaneously store the checkpoints for all the AMs it serves.

If you add memory to an existing AM, you must use the table in Section 7 of the *Engineer's Reference Manual* to re-estimate the AM checkpoint volume size, and if the existing volume is not large enough, you will have to use Network Configuration to reconfigure the HM volumes (see *Network Data Entry* in the *Implementation/Startup & Reconfiguration - 1* binder).

Usually, demand checkpoints (requested through the SAVE DATA target on the AM Status Display) for a single AM with up to 3 Mw of memory can be saved on floppy diskettes, but checkpoints for 4 Mw AMs, particularly if they have a single, large unit, may not fit on floppies. It is also possible that cartridges will be required for 3 Mw AMs, especially if there is only one unit in the AM and the predominant use is for CL programs. For storage of demand checkpoints, cartridge disk drives should be made available in such situations, and for any AMs with larger than 3 Mw memories.

6.2 OPTIONS AND AM CHECKPOINTS

Purchased options for AMs are AM memory size (2, 3, and 4 Mw) and AM Redundancy (see Section 7 of these *Application Module Implementation Guidelines*).

If the memory size of an existing AM is increased, unless the HM volume that is to store checkpoints for than AM is already sufficiently large, you will need to use Network Configuration to reconfigure the volume size. The calculation of the volume space (volume &5np) needed for an AM checkpoint is defined in the table labled "HM Volume Size Estimator" in Section 7 of the *Engineer's Reference Manual*.

Adding the AM Redundancy option to an existing AM reduces available user space by 157,000 or 179,000 words (see Section 2.8). If this option is added without adding memory, you may not be able to reload the checkpoint data for this AM, because there is insufficient user memory. To avoid this problem, you can reserve user memory for future options as described in 6.2.1.

6.2.1 How to Reserve User Memory for Future Options

PSDP parameter RESRVMEM (for more information about PSDPs, refer to Section 22 in the *Engineer's Reference Manual*) is used to change the amount of reserved memory. Initially, this parameter contains 0, which specifies no reserved user memory. Do this to reserve user memory in blocks of 32 kw:

1. On the Engineering Main Menu, select `LCN NODES`. The LCN Node Configuration display appears. Note that a backup NCF pathname is required.

2. Select the node number for the AM whose RESRVMEM value is to be changed (you may need to press PAGE FWD). The Application Module Node display appears.

3. Select MODIFY NODES. Page 2 of the Application Module Node display appears (Figure 2-3).

4. To change RESRVMEM, key in a new value in the port labeled `USER MEMORY RESERVED (# OF 32KW BLOCKS)`, and press ENTER. The Application Module Node display should reappear with the new value in blue (cyan). If there are errors in the value you entered, they are noted in red. Note that the values in the MEMORY USED (WORDS) column also change.

   The value entered will allocate reserved memory as follows:

0 = no reserved memory	5 = 160 kw to be reserved	10 = 320 kw to be reserved
1 = 32 kw to be reserved	6 = 192 kw to be reserved	11 = 352 kw to be reserved
2 = 64 kw to be reserved	7 = 224 kw to be reserved	12 = 384 kw to be reserved
3 = 96 kw to be reserved	8 = 256 kw to be reserved	
4 = 128 kw to be reserved	9 = 288 kw to be reserved	

   Note that the sum of the value in RESRVMEM and the value AMDATA(48) (see 7.3.1 in these *Application Module Implementation Guidelines*) cannot be larger than 12. If this AM doesn't have the redundancy option, AMDATA(48) contains 0.

5. Hold CTL and press F1 (F1=Check). A Modify Node display appears with instructions for installing and loading the change you just made. Follow those instructions (they include shutting down and reloading the AM). The PSDP parameter RESRVMEM assumes the current value, and the memory reserved for CVBs changes accordingly.

6.2.2 How to Discover the Reason for Checkpoint Errors

The AM PSDP parameter CPFMERR contains the secondary status code from a File Manager error during a checkpoint. See heading 4.10 in the *Control Language/Application Module Reference Manual* for a complete list of File Manager error codes.

NOTE

If the AM is shut down during a checkpoint cycle, it will go to the FAIL state rather than to the QUALIFIED state (in order to ensure a clean checkpoint termination).

AM REDUNDANCY OPTION
Section 7

This section provides guidelines for installing and configuring, loading, and the operation of redundant AMs.

AM Redundancy is a purchased option for AMs with AMR/HMPU, EAMR/K4LCN Redundancy and processor boards only. AM Redundancy includes hardware and software that allow a secondary AM to back up the operation of the primary AM. To do this, all of the application data in the primary AM is transferred to the secondary AM, where an exact copy of the data is maintained to be used if the primary AM should become inoperative.

When a hardware fault is detected in the primary AM, the secondary AM takes over within approximately five seconds. No external accesses to the AM are lost during the transfer (failover).

When valid data is stored in the secondary AM, the primary and secondary are said to be synchronized. Communication errors and processing overloads are rare, but should they occur, the AM pair is automatically resynchronized. An automatic memory checker compares the data in the secondary with that in the primary and should they not compare the secondary fails, which generates an alarm.

7.1 INSTALLING AND CONFIGURING REDUNDANT AMs

Instructions for installing and interconnecting redundant AMs is provided in Appendix A of the *LCN System Installation* manual in the *LCN Installation* binder. Each AM in a redundant pair includes an AM Redundancy board (AMR) that connects to the HMPU board or an EAMR extended redundancy board (EAMR) that connects to the K4LCN board through a ribbon cable that is visible at the front of the chassis. The AMR boards in each AM are interconnected by two ribbon cables that are visible at the rear of the chassis.

AM Redundancy requires the following memory in redundant AMs: 157,000 words (fast/slow processor buffers), plus an additional 22,000 words for the AM redundancy Internetwork Point Processor (IPP) option if used.

You define the redundant AM pairs in the LCN Node Configuration activity in Network Configuration (see *Network Form Instructions* and *Network Data Entry* in the *Implementation/Startup & Reconfiguration - 1 binder).*

7.2 LOADING OF REDUNDANT AMs

The AM Status Display is used to load, shut down, and reload both AMs. First load the primary AM using the appropriate type of load (hot, cold, warm, or no processing). Then load the secondary. For the secondary, use any of the load options (which you use, makes no difference). When redundancy in in effect, the primary's status is `OK` and the secondary's status is `BACKUP`.

> **NOTE**
>
> The EAMR redundancy board must be installed before loading a K4LCN AM node that is configured as redundant through the LCN Node Configuration. The AM load will abort and produce a status notification message if the K4LCN AM node is loaded without the required redundancy hardware.

7.3 OPERATION OF REDUNDANT AMs

Unless an overload occurs, a properly configured redundant AM pair (see 7.1) with appropriate status runs without further intervention.

Occasional errors may occur that cause loss of synchronization for a few seconds while the secondary AM's data is being refreshed. The value in PSDP parameter REDINOP (redundancy in operation) is False during resynchronization. PSDP parameter RESYNCS(1) contains a count of all resynchronizations since the secondary AM was last started. Other values in the RESYNCS array define the causes of resynchronization. These parameters are defined under 22.3 in the *Engineer's Reference Manual*.

Under some conditions, a redundancy buffer that stores data being transferred to the secondary can be overloaded. A normal, temporary overload causes resynchronization, but the following conditions cause a heavy overload, which may cause the secondary AM to fail (crash).

- DEB reloading of points with CL blocks with many references to other physical nodes. Approximately 5000 such references while normal processing is taking place for other points can cause a heavy overload and failure of the secondary AM.

- Storage of a large number of values by a CL block or several CL blocks on one cycle. Storage of approximately 6000 "on point" values to CDS parameters combined with normal point processing could cause an overload.

If more than five overloads occur within ten minutes, the secondary AM fails (crashes).

7.3.1 Overloads and Redundancy Buffer Size

You may be able to reduce the number of redundancy buffer overloads (see 7.3) by increasing the size of the redundancy buffer, which decreases user memory (for more information, refer to 2.2).

The default value for redundancy buffer size is 122,880 words. The current size is in PSDP parameter REDBFZ in the secondary AM.

PSDP parameter AMDATA(48) in the secondary AM contains an integer that represents the amount of additional redundancy buffer currently in use. Initially the value in the parameter is 0. Do this to increase size of the redundancy buffer (in blocks of 32 kw):

1. On the Engineering Main Menu, select `LCN NODES`. The LCN Node Configuration display appears. Note that a backup NCF pathname entry is required.

2. Select the node number for the AM whose RESRVMEM value is to be changed (you may need to press PAGE FWD). The Application Module Node display appears.

3. Select MODIFY NODES. Page 2 of the Application Module Node display appears (Figure 2-2).

4. To change AMDATA(48), key in a new value in the port labeled `REDUNDANCY BUFFER INCREASE (# OF 32KW BLOCKS)`, and press ENTER. The Application Module Node display should reappear with the new value in blue (cyan). If there are errors in the value you entered, they are noted in red. Note that the values in the MEMORY USED (WORDS) column also change.

   The value entered will add buffer memory as follows:

0 = 0 kw to be added	11 = 352 kw to be added
1 = 32 kw to be added	12 = 384 kw to be added
2 = 64 kw to be added	13 = 418 kw to be added
3 = 96 kw to be added	14 = 448 kw to be added
4 = 128 kw to be added	15 = 480 kw to be added
5 = 160 kw to be added	16 = 512 kw to be added
6 = 192 kw to be added	17 = 544 kw to be added
7 = 224 kw to be added	18 = 576 kw to be added
8 = 256 kw to be added	19 = 608 kw to be added
9 = 288 kw to be added	20 = 640 kw to be added
10 = 320 kw to be added	

   Note that the sum of the value in AMDATA(48) and the value RESRVMEM (see 6.2.1 in these *Application Module Implementation Guidelines*) cannot be larger than 12.

5. Hold CTL and press F1 (F1=Check). A Modify Node display appears with instructions for installing and loading the change you just made. Follow those instructions (they include shutting down and reloading the AM). PSDP parameters AMDATA(48) and REDFBSZ assume the current values, and the memory reserved for the redundancy buffer changes accordingly.

7.4 BACKGROUND CL IN REDUNDANT AMs

When switchover from primary to secondary occurs in redundant AMs, there is a significant difference between background and foreground CL programs. Runtime software has built-in protection for foreground CLs. It provides a smooth transition during switchover, protecting database integrity and preserving the sequence of program actions. <u>Background CL does not have the same protection.</u>

After a switchover from primary to secondary, all running background CLs restart from the beginning. CL stores that were done in the primary may not have taken place in the secondary before switchover, and if this is a problem, the user must provide protection. For example, assume a primary background CL is in the middle of a sequence of stores to on-node parameters when switchover occurs. The stores are done in real time in the primary, and into a list for the secondary, where they are performed after a time lag. Therefore, without specific programming by the user, there is no way to know how many, if any, of the parameters were stored in the secondary's database. Another potential problem is that when a switchover forces a background CL to restart at the beginning, sequence-critical functions in the background program may be forced out of the required sequence.

If your process would be harmed by loss of data and disruption of sequence that could occur if your background CL programs restart as a result of switchover, you have two choices. You must either implement the critical programs in foreground CL, or you must include logic in the background programs to protect the system if switchover occurs.

The following is one possible technique that you can adapt and use to program for redundancy in background CL. In a Redundant AM CL program, the primary can communicate synchronization data to the secondary through CDS parameters. This synchronization data can be a code that indicates to the secondary where it should resume execution after a switchover. If a switchover occurs, the running background programs will restart. There is a CL Built-In function, BKG_Switchover_Restart (described in the CL/AM Reference Manual), that can be used by a background CL program to determine if it started as a result of switchover. The program can use this function to establish that a switchover occurred, and then the program can test the synchronization data in the CDS parameters to determine at what point in the CL it should resume execution.

As an example, consider the situation mentioned above where a background CL program includes a sequence of on-node stores. To provide switchover protection, you could have the program set the code “3” in a CDS parameter just before the sequence of stores is started, and change the code to “4” when the sequence is completed. After a switchover restart (determined from the BKG_Switchover_Restart function), the program can check the code. If the code is “3,” it is known that the switchover occurred during the sequence of stores and that is it possible that some were completed while others were not. The secondary would resume execution at the beginning of the sequence of stores. If the code is “4,” the program would resume after the sequence of stores. Using this technique, the primary can pass the location of “safe” restart points to the secondary during execution of the program.

Another technique that may be helpful to you is the Multiple_Move_Parameter function that is provided in the custom set AMCL01. This function allows you to do multiple stores to a particular unit without interruption. For information on the Multiple_Move_Parameter function, refer to subsection F.4 of the CL/AM Reference Manual. Subsection F.4.4 discusses data integrity issues.

CUSTOM SOFTWARE, CL EXTENSIONS, AND BACKGROUND CL
Section 8

This section provides guidelines for the installation and loading of AM custom software, CL extensions, and background CL blocks.

AM custom software is optional software, purchased from Honeywell, and provided as external load modules. CL extensions consist of a set of subroutines that enable the movement of data between CL/AM blocks and History Modules. Background CL blocks are CL blocks that, while inserted at a defined place in the point-processing structure, run in the "background" and unlike all other CL blocks, are not required to complete execution before processing of the next AM point can begin.

Documentation for AM custom software, including information about use of AM resources, is provided by the custom software developers. For more information about background CL blocks, refer to Section 4 in *Application Module Control Functions* in the *Implementation/Application Module - 1* binder. For more information about CL extensions, refer to Appendix C in the *Control Language/Application Module Reference Manual* in the *Implementation/Application Module - 2* binder.

8.1 DIRECTORIES REQUIRED FOR EXTERNAL LOAD MODULES AND CL EXTENSIONS

The following Honeywell-provided directories contain files required to support AM custom software in external load modules and to support CL extensions:

- &CUS—contains files FILE.LO and CONV.LO, which are CL-extensions files. This directory may also contain external load modules for AM custom software such as AMCL01 or AMCL02.

- &CLX—contains the set-definition file FILE.SF, which describes to the CL Compiler, the added CL routines. This directory may also contain set-definition files for AM custom software such as AMCL01 or AMCL02.

&CUS and &CLX can be on an HM or on removable media (cartridges or floppies). If on an HM, they must be in a volume, usually a user volume, that has sufficient storage capacity. The standard files that provide and support the CL extensions are provided in directories &CUS and &CLX on the software media provided by Honeywell. These directories are provided on the following media:

- Application Module Cartridge &CR3

- Floppy diskette 51152058, CL File R/W Option Packaging (&AMS).

If you need to use the CL Extensions, you should copy the content of directories &CUS and &CLX to an HM, and you should enter `NET>&CUS>` in the `EXT LOAD MODULE` port on the Engineering Personality's Modify Default Volume Pathnames display.

With only CL Extensions files in the two directories, they occupy memory as follows:

- &CUS; 464 sectors = 116 kb
- &CLX; 33 sectors = 8.3 kb
- A combined total of 124.3 kb

If &CUS contains custom software, the documentation provided with that software advises you of the size of directory &CUS. The guidelines for user-volume configuration in Section 7 of the *Engineer's Reference Manual* recommend the inclusion of enough space in at least one user volume to accommodate directories &CUS and &CLX.

8.2 OVERVIEW OF THE INSTALLATION OF EXTERNAL LOAD MODULES

External load modules are provided separately from the TDC 3000^X System software. They typically contain software that is sold and developed by Honeywell IAC as custom software. The major installation steps are as follows:

1. Have the external load modules available on removable media (cartridges or floppy diskettes) or load them into an HM A sufficiently-large user volume must exist on the HM to accept these modules. The modules are stored in a directory in that volume whose name typically is &CUS. You can add this directory to a volume with the Utilities' Create Directory (CD) command (for more information, refer to 5.6 in *Command Processor Operation* in the *Implementation/Engineering Operations - 1* binder).

2. In the Modify Default Volume Pathnames display, enter a pathname that points to the volume that contains the external load modules. The default pathname is `NET>&CUS>`.

3. Enter data related to the external load modules in the node configuration displays for the nodes that are to use them. This modifies the NCF workfile.

4. Install the modified NCF.

5. To make the modified NCF effective, shut down (if necessary) the affected nodes and (re)load them.

This installation is normally accomplished on the system with the affected nodes, with the Configurator in the on-line mode. You can use the Configurator in the off-line mode to create a work file for another system, but integrity checks that are accomplished in the on-line mode are not accomplished in the off-line mode.

More detail about installation external load modules and background CL routines is provided in the remainder of this section.

8.3 NODE CONFIGURATION AND EXTERNAL LOAD MODULES PAGE

On the Engineering Main Menu, select `LCN NODES`, then page forward as necessary to find and select a node. Page 1 of the Node Configuration display appears.

The PAGE FWD and PAGE BACK keys move forward or backward through the multiple pages of the Node Configuration display for the selected node (two pages for most nodes, three pages for AMs). To select another node, hold CTL and press PAGE BACK. Then, on the Select Desired Node display, select another node.

If you are adding a node not yet known to the NCF, you can enter appropriate data on any page of the Node Configuration display. If you are modifying an existing node, to get to page 2 , select `MODIFY NODE`. You can go to page 1 but you cannot make changes on that page (it is already configured).

Figure 8-1 provides a simulation of the external load modules page with notation that shows the help messages that appear in the prompter region at the bottom of the page when you place the cursor in one of the ports and press HELP. To cancel the help message, press CANCEL.

For application and customs software, the names of the external load modules, applicable personalities, and information about additional module memory values are provided in the documentation provided with the custom software. For Background CL Routines, refer to Section 4 in *Application Module Control Functions* in the *Implementation/Application Module - 1* binder.

To enter data, select a port, key in the data, and press ENTER. Necessary calculations then take place and totals are displayed. When you are satisfied with the data, hold CTL and press F1 to start the Checker. After a successful check,

- if the Configurator is in the on-line mode, the Installation Instructions display for the selected node appears (Figure 8-3). You can then install the modified NCF by holding CTL and pressing F2. Then follow the installation instructions to make the modified NCF effective.

- if the Configurator is in the off-line mode, the current display remains. You can then copy file NCF.WF from directory &ASY onto a removable medium and install that NCF on the target system..

8.4 NODE CONFIGURATION, AM USER MEMORY ALLOCATION PAGE

Figure 8-2 shows a simulation of AM Node Configuration page 2, the AM User Memory Allocation page, that shows the help messages that appear when you place the cursor in one of the ports and press HELP.

8.5 LOADING OF EXTERNAL LOAD MODULES

After you have made the external load modules available on a storage medium (typically in a &CUS directory in a user volume), and in LCN Node Configuration you have defined external load modules to be loaded in an AM, each time the AM is loaded with the personality associated with one or more of the external load modules, those modules are included in software that is loaded.

As the external load modules are loaded, checks are made to ensure that the module versions are valid, that the personality type is correct, and that interdependencies between modules are satisfied. If errors are found, the load is incomplete or it fails. Information about such errors can be found on the node status displays and in the Auxiliary Node Status for the node, which is accessed through the Event History Retrieval display.

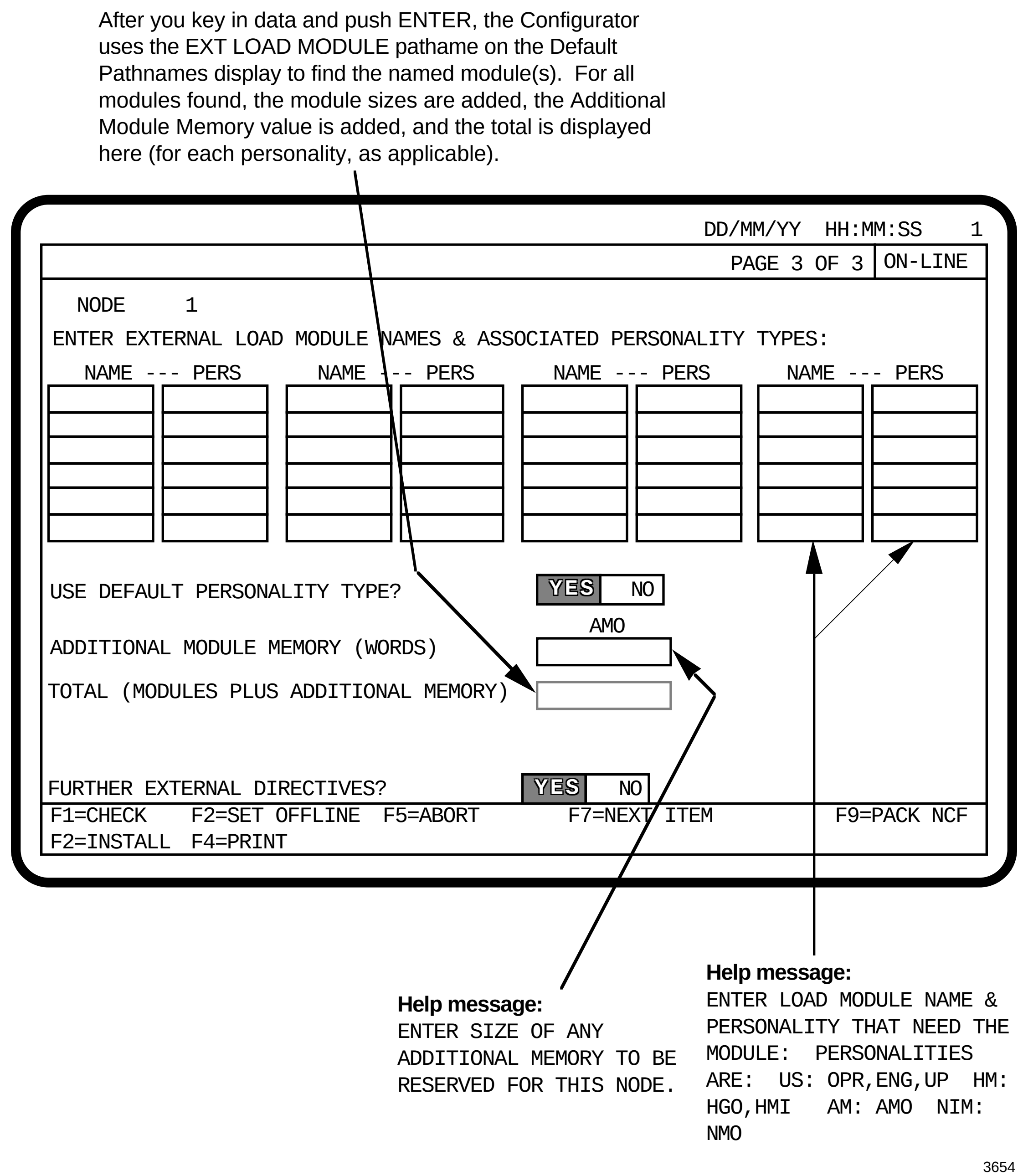


Figure 8-1 — Node Configuration, External Load Module Page

Viewing Help Messages

Place the cursor in a port and press HELP. The Help message appears in the prompter region at bottom of the display.

① ENTER # CL BACKGROUND TASKS. RANGE 0-10. MEMORY USED INCLUDES STACK SIZE. NOTE: (MEMORY ALLOTS 16200 + # CL BACKGROUND TASKS * CL BACKGROUND TASK SIZE WORDS)

② ENTER # CONCURRENT DATA ACCESSES FROM BACKGROUND CL. RANGE 0-4. MUST = AT LEAST 1 IF # BACKGROUND TASKS >0. NOTE: (MEMORY ALLOTS 1050 WORDS * # CONCURRENT DATA ACCESSES)

③ ENTER CL BACKGROUND TASK SIZE (IN WORDS). DEFAULT IS 25000. RANGE IS 700-32000.

④ ENTER CURRENT VALUE BUFFER SIZE FOR FAST OR SLOW POINT PROCESSOR (IN WORDS). DEFAULT IS 2000. RANGE IS 2000-32700.

⑤ ENTER # ADDITIONAL 32KW BLOCKS IN REDUNDANCY BUFFER. RANGE 0-20. (REDUNDANCY BUFFER INCREASE + USER MEMORY RESERVED) <=20

⑥ SELECT YES TO PERMIT INTERNETWORK DATA ACCESS FROM ASSIGNED POINTS. NOTE: (MEMORY ALLOTS 9 * CVB FOR IPP + 21505 IPP OPTION MEMORY)

⑦ ENTER CURRENT VALUE BUFFER SIZE FOR IPP (IN WORDS, RANGE IS 2000-32700).

⑧ ENTER # 32KW BLOCKS RESERVED FOR FUTURE USE. RANGE = 0-12. (REDUNDANCY BUFFER INCREASE + USER MEMORY RESERVED) <=20

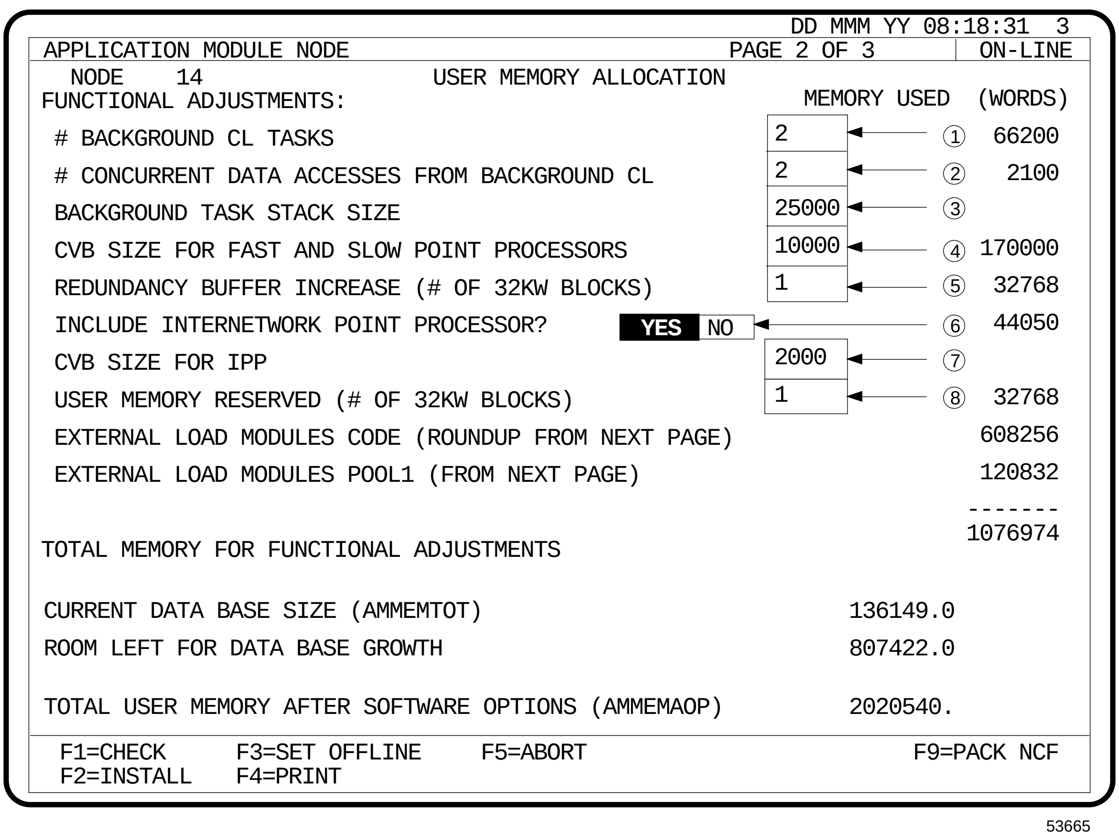


Figure 8-2 — Node Configuration, AM User Memory-Allocation Page

```
                                                    18 Aug 89  15:45:29
                                                      PAGE 1 OF : MOD US 01

1. INSTALL THE CHANGES IN NCF (F2)


THE FOLLOWING STEPS OUTLINE THE PROCEDURE FOR LOADING THE CHANGES
INTO THE NETWORK.  YOU MAY INSTALL OTHER CHANGES BEFORE PERFORMING
THESE STEPS.


Loader Directive changes affect only the node being modified.
Complete the next step to install the new directives.

2. For this node:
    a. SHUTDOWN the node
    b. LOAD the node using the NCF installed in the previous step.




                             F5=ABORT
F2=INSTALL   F4=PRINT
```

3657

Figure 8-3 — Node Configuration, Installation Instructions Page

Index

Index

READER COMMENTS

Honeywell IAC Automation College welcomes your comments and suggestions to improve future editions of this and other publications.

You can communicate your thoughts to us by fax, mail, or toll-free telephone call. We would like to acknowledge your comments; please include your complet name and address

BY FAX: Use this form; and fax to us at (602) 313-4108

BY TELEPHONE: In the U.S.A. use our toll-free number 1*800-822-7673 (available in the 48 contiguous states except Arizona; in Arizona dial 1-602-313-5558).

BY MAIL: Use this form; detach, fold, tape closed, and mail to us.

Title of Publication: **Application Module Implementation Guidelines** Issue Date: **11/95**

Publication Number: **AM12-510**

Writer: **Maria Nelson**

COMMENTS: ____

____

____

____

____

____

____

____

RECOMMENDATIONS: ____

____

____

____

____

____

____

____

NAME ______________________________ DATE ________________

TITLE ____

COMPANY ____

ADDRESS ____

CITY ____________________ STATE __________ ZIP ________________

TELEPHONE ________________ FAX ____________________

(If returning by mail, please tape closed; Postal regulations prohibit use of staples.)

Communications concerning technical publications should be directed to:

Automation College
Industrial Automation and Control
Honeywell Inc.
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

FOLD FOLD

From: ______________________

____

____

NO POSTAGE
NECESSARY
IF MAILED
IN THE USA

BUSINESS REPLY MAIL

FIRST CLASS **PERMIT NO. 4332** PHOENIX, ARIZONA

POSTAGE WILL BE PAID BY ....

Honeywell

Industrial Automation and Control
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

Attention: Manager, Quality

FOLD FOLD

Cut Along Line

Additional Comments:

Honeywell

Industrial Automation and Control
Honeywell Inc.
16404 North Black Canyon Highway
Phoenix, Arizona 85023-3099

Helping You Control Your World