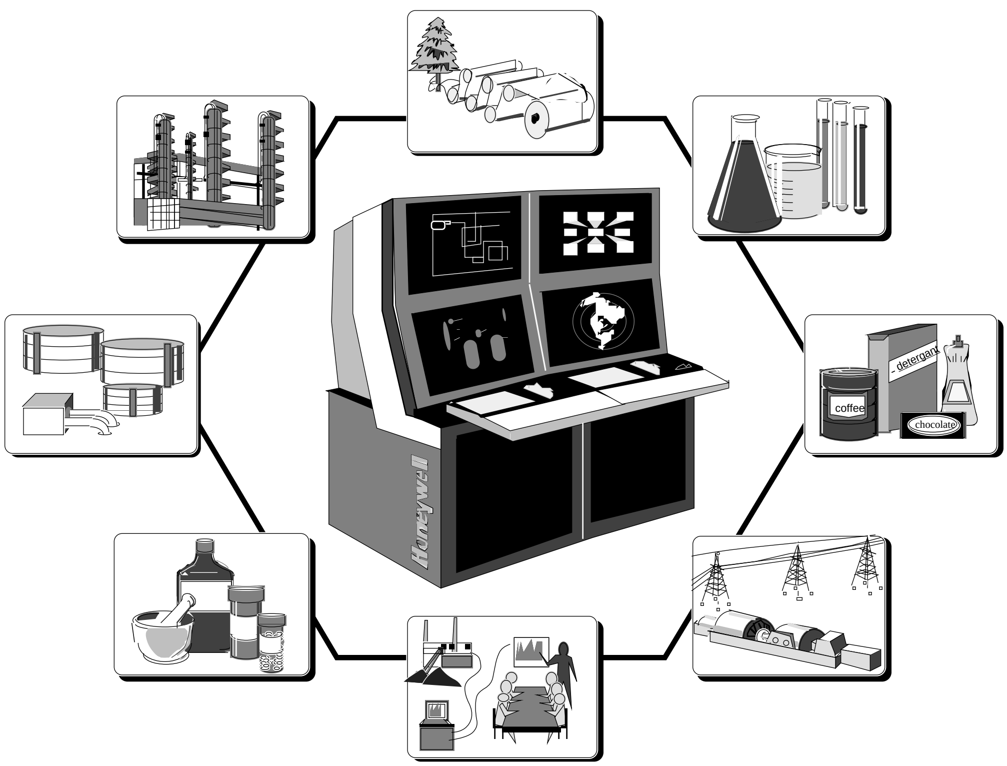
Honeywell

TDC3000^X Application Module^X and OpenDDA Specification and Technical Data

AX03-200
Release 200
1/96



TotalPlant

TDC 3000X Application ModuleX and OpenDDA Specification and Technical Data

Introduction

This publication describes the features in Honeywell's Application ModuleX (AXM) and its primary application enabler, OpenDDA (Open Data Definition and Access).

The AXM is a TDC 3000X advanced applications platform that combines the security necessary for advanced process control and optimization with the power to run complex applications from Honeywell and third-party suppliers. Existing Application Module (AM) functions - point processing, broad Proportional Integral Derivative (PID) algorithm handling, and Control Language (CL) capabilities - are expanded in the Application ModuleX with the addition of a powerful RISC-based UNIX™ co-processor that lets you use a wide range of software applications from a variety of suppliers. The AXM is designed to operate with system software Release 431 or later, and interfaces with all other LCN node types and personalities. OpenDDA is the tool in the Application ModuleX that allows applications to be integrated with LCN data.

Figure 1 illustrates the Application ModuleX dual hardware interface that allows it to serve as a module on the TDC 3000X Local Control Network (LCN) as well as a node on the Plant Information Network (PIN). Accordingly, it can communicate with other modules on single or multiple LCNs; with process-connected devices on Universal Control Networks and Data Hiways; and with PCs, workstations and other devices that reside on the PIN and communicate using industry standard network communications.

Functional Description

Application ModuleX Overview

The heart of the AXM is the tightly coupled integration of workstation technology with the LCN. It utilizes the standard AM as its base functional platform to provide integration and compatibility with the current LCN technology. Additionally, AXM provides the

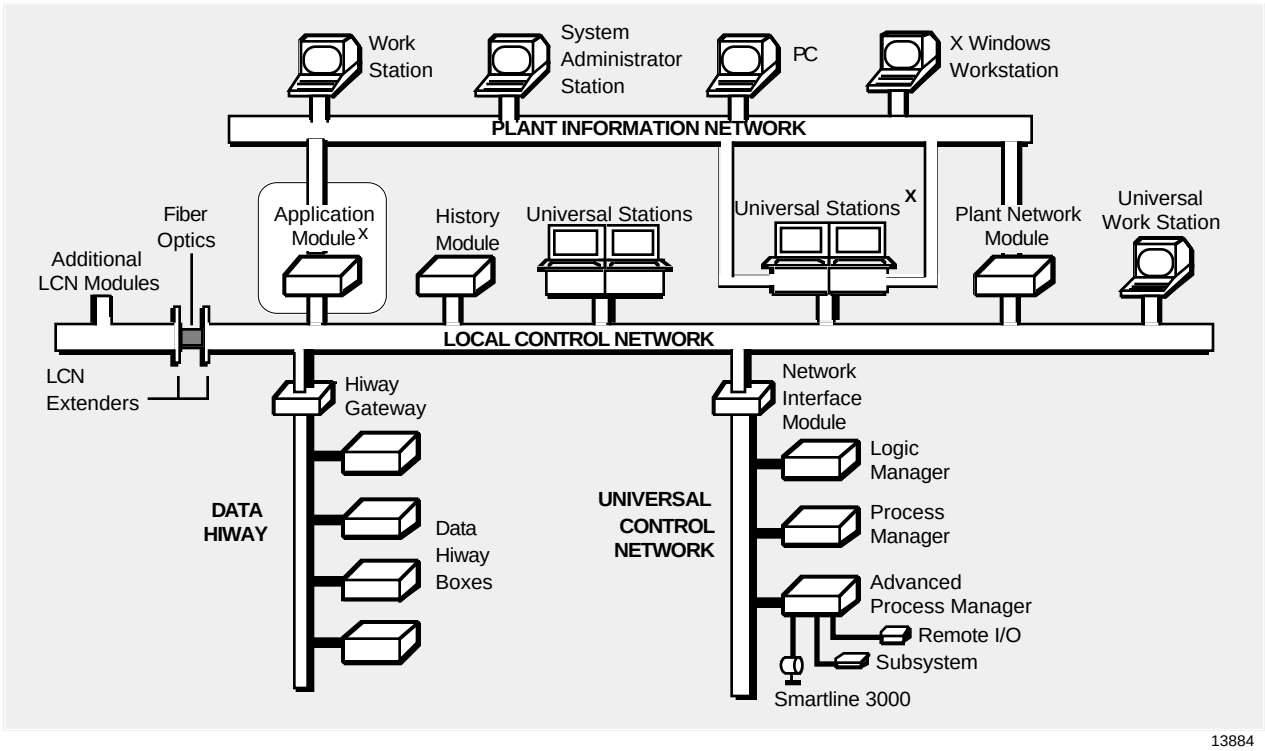


Figure 1 — TDC 3000X System with Application ModuleX

operating environment and security of UNIX to further enhance the total operating environment by leveraging the use of state-of-the-art commercial software.

The UNIX side of the A^{X}M utilizes Hewlett Packard's newest RISC technology and runs HPs standard UNIX operating system: HP-UX (version 9.05). The A^{X}M resides on the LCN and has an AM front end. It has all the standard AM alarming, CL (background and foreground), Custom Data Segment (CDS) data structures, messaging, regulatory control algorithms, and other point-processing capabilities (input/output connections). It also gives the AM some additional features which include the ability to initiate a UNIX program from CL, and the ability to control the types of UNIX programs which can write to LCN parameters through a security switch.

OpenDDA Overview

OpenDDA is a friendly layer to data access primarily focused at the control application engineer. OpenDDA is an application enabler that abstracts data access Application Programmatic Interfaces (APIs) while providing users with a standard method of defining and accessing process data in commonly available languages. It is currently available for use with FORTRAN, ANSI C, and C++ compilers. The application engineer specifies a mapping of LCN point.parameters to program local variables and then uses those local variables in his program. Reading and writing of LCN data can be performed at any point in the program with a simple command. OpenDDA contains a pre-compiler that turns those simple commands into the actual data access calls. OpenDDA also provides a mode for test and debug of applications prior to execution with a "live" process.

User Interface

The Application Module^X does not have a terminal or keyboard attached to it; therefore, it requires some other type of user interface. User interface options include:

- a Universal Station^X
- a System Administration and Development Station
- an X windows station (An X workstation or a PC running X windows software)
- a workstation terminal (Note: Requires DAT drive directly connected to the A^{X}M)

A Universal Station^X is the most complete of the user interface options because it has all the power of an X workstation as well as a view to the AM side of A^{X}M. With this option, if the DAT drive and CD-ROM options for the A^{X}M are not purchased, they must be available on the U^{X}S. A DAT drive is required to backup and restore the UNIX side of the A^{X}M as well as to load software onto the UNIX hard disks. See the Universal Station^X Specification and Technical Data for more details on U^{X}S.

The System Administration and Development Station is a minimum configuration HP workstation recommended by Honeywell if a user does not have a U^{X}S. Use of this station with the A^{X}M is documented in the A^{X}M manual set. See the appropriate section of this document for more details on this product.

An X windows station can provide some of the same functionality as a System Administration and Development Station. If the X station is a non-HP-UX box (i.e., a PC running X windows or a DEC workstation) you will still be able to view some of the graphical interfaces of the A^{X}M, but are not guaranteed that they will all come across (there could be some potential problems with font sizes, etc.).

You will be able to access the A^{X}M at the command line interface level. It is required that you have a DAT drive on your A^{X}M IF the X windows station you are using is NOT an HP-UX workstation. The CD-ROM can be on an HP-UX machine or a PC.

NOTE: If you use a PC-based CD-ROM reader, the HP-UX documentation (which comes on CD-ROM format) can be viewed, but will NOT include the graphical user interface that makes it extremely user friendly. The graphical user interface only works with a CD-ROM reader on an HP-UX machine.

A dumb terminal provides you command line access to the A^{X}M. With command line access you can fully configure and use an A^{X}M, but you will not have use of any of the graphical tools available such as the windows based editor or the graphical System Administration Menu. The necessary cables to connect a dumb terminal are included with the A^{X}M. It is mandatory that you have a DAT drive on the A^{X}M if you are using a dumb terminal as your user interface.

Table 1 — A^{X}M RISC Processor and Memory Features

A^{X}M RISC Board Specifications	64 MHz	100 MHz
Performance		
SPECint92	66.6	100.1
SPECfp92	96.5	137.0
MFLOPS (DP)	25.3	37.8
MIPS	77.7	121.6
Memory management unit		
Virtual memory address	48 bit	48 bit
Instruction TLB and data TLB	120-page entries unified, 16 variable block entries unified, fully associative	120-page entries unified, 16 variable block entries unified, fully associative
External Cache		
Instruction data cache size	256 KB	256 KB
Organization	Direct mapped	Direct mapped
Bus width	64 bits	64 bits
Instruction cache bus peak performance	512 MB/second	800 MB/second
Main memory type	ECC single bit correct, double bit detect	ECC single bit correct, double bit detect
Memory Options	32 MB, 64 MB, 128 MB, 256 MB	32 MB, 64 MB, 128 MB, 256 MB
Main memory bus width	64 data bits w/8 check bits	64 data bits w/8 check bits
Main memory bus peak performance	128 MB/sec (64 byte duration)	133 MB/sec (64 byte duration)
System DRAM technology	60 ns, 4 & 16 MB DRAM	60 ns, 4 & 16 MB DRAM
Memory card sizes	32, 64 MB	32, 64 MB

RISC Processor Features

The Reduced Instruction Set Computing (RISC) board used in the A^{X}M is Hewlett-Packard's Model 743 board which contains their Precision Architecture RISC (PA-RISC) processor. The specific processor is the PA7100LC and it is available in two clock speeds: 64 MHz or 100 MHz.

Both versions of the processor are available in the A^{X}M and both include a built-in high performance floating point co-processor. Memory Options are 32 MB, 64 MB, 128 MB, and 256 MB RAM. Hard disk options include 525 MB, 1.2 GB, and 2.4 GB. Table 1 contains the specific performance data on both versions of the processor.

The I/O for the RISC board includes:

LAN
- Type: IEEE 802.3/Ethernet
- Data rate: 10 MB/sec
- Connector: 15-pin micro D-sub

Networking Products
- NCS
- NFS
- Berkeley 4.3
- TCP/IP
- BSD 4.3 Network Services
- ARPA Services
- DECnet supported as an option

SCSI Interface (1 available)
- Type: SCSI II: single-ended, 8 bit
- Data rate: 5 MB/sec synchronous, 1.5 MB/sec asynchronous with standard cable lengths
- Device limit: 7 devices
- Connector: ALT-1 of SCSI II 50-pin high density

Serial Interface (1 available)
- Type: EIA RS-232-C, CCITT V.24/V.28
- Data rate: 460.8 Kbps (16550A-compatible UART)
- Device limit: 1 per interface
- Connector: 9-pin female micro D-sub (adapter cable is available)

The Operating System that runs on this processor is Hewlett Packard's UNIX (HP-UX). See the software section for specific details of the operating system.

Software Details

There are four key elements of the A^{X}M software: the AM personality, the A^{X}M personality, the HP-UX operating system, and OpenDDA. The standard AM personality is included on Bernoulli so the user has the option to load the A^{X}M as a standard AM. The A^{X}M personality is included on 4 millimeter DDS (DAT) tape and comes pre-loaded on the UNIX hard disk. OpenDDA is included on DAT tape and should reside on the UNIX hard disk during development and be removed and securely stored when not developing. The HP-UX Operating System is included on DAT tape and comes pre-loaded on the UNIX hard disk. The A^{X}M hard disk is shipped with both the UNIX software and the A^{X}M personality loaded on it.

AM Personality

The AM personality is included on a Bernoulli. It includes the standard AM alarming, CL (background and foreground), Custom Data Segment (CDS) data structures, messaging, regulatory control algorithms, and other point-processing capabilities (input/output connections).

A^{X} M Personality

The A^{X}M personality is included on a DAT tape and is a different personality than supplied on the standard LCN media. An operator at a US loads the A^{X}M personality as if he/she was loading an AM using the AUTOLOAD NET command which automatically retrieves the personality from the UNIX hard disk. When AUTOLOAD NET is selected, the user will now have the ability to do a HOT, WARM, or COLD restart. The operator may load the A^{X}M as a standard AM by doing a manual load. The key additional features in the A^{X}M personality, besides all the features of a standard AM, include:

X side Access to any LCN point.parameter

The underlying mechanism that allows transfer of point.parameter information to and from the X side of the Application Module^{X} is included in the A^{X}M personality. This includes queues and priority recognition of data access.

Security Switch

The security switch is a three position switch in the form of a LCN system parameter. The positions are:

- Read Only (No X side applications can write to the LCN)
- CL Initiated Writes Only (Only those X side applications that are scheduled from the AM's scheduler (CL initiation) can write to the LCN.
- Read Write (All UNIX authorized programs can both read and write to the LCN)

Eight Background CL Runtime extensions in AMCL06 external load module

- Execute Task With Wait - Used to initiate an application in the X side. The program uses one application initiation queue slot until it has completed.
- Get Queue Info - Used to find out how many queue slots are available and how many are in use
- Get Queue Slot Info - Used to get status information from a queue slot about a CL-initiated X side program
- Store X Access - Used to change the state of the global X Access Switch
- Initiate Task - Initiates an OpenDDA application capable of hibernating and receiving background CL events
- Activate Task - Sends an activate event to a CL-initiated hibernating OpenDDA application
- Terminate Task - Sends a terminate event to a CL-initiated hibernating OpenDDA application, providing graceful shutdown of the application
- Get Hiber Task Status - Obtains the current status of a CL-Initiated hibernating OpenDDA application capable of receiving background CL events

AMCL06 requires AM memory per Table 2. This memory includes the software, data buffers, and extra space to equalize user available heap with standard AM personality.

HP-UX Operating System

The HP-UX Operating System included with the base A^{X}M software is HP-UX 9.05; delivered on a DAT tape. The A^{X}M hard disk comes pre-loaded with the HP-UX software. It is shipped with maximum security configured; that is, it is configured for NO networking (all network services are set to 'deny'). There is some configuration work required to set up the A^{X}M to run on the PIN. HP-UX 9.05 is the latest version of the HP-UX operating system for HP 9000 Series 700 and 800 workstations. HP-UX conforms to X/Open's Portability Guide Issue 4 (XPG4), Federal Information Processing Specification (FIPS) 151,1, POSIX 1003.1 and POSIX 1003.2. It conforms to AT&T's System V Interface Definition 2 (SVID 2). HP-UX also incorporates selected features from the University of California at Berkeley Software Distribution 4.3 (4.3BSD).

This operating system also comes with a number of 'Ease of Use' functions:

- *HP VUE 3.0*
  HP VUE 3.0 is a simple, intuitive, point-and-click user interface for HP-UX. It includes many features, such as individual workspaces for different tasks, a "dashboard" with commonly used functions, a File Manager that performs common file operations using icons, drag-and-drop, slide-up toolboxes, **a Motif based text editor (VUEpad) - you are not forced to use 'vi'**, and a variety of productivity tools.

Table 2 — AM Memory Used Up by AMCL06

A^{X}M and LCN Release	Memory (K words)
A^{X}M 100 on LCN R430 or 431	499
A^{X}M 110 on LCN R431	840
A^{X}M 110 on LCN R500	2451
A^{X}M 200 on LCN R431	840
A^{X}M 200 on LCN R500	2451

- *System Administration Manager (SAM)*
  SAM greatly eases many common system administration tasks, such as adding users, configuring peripherals, configuring networking, managing processes, and many others. SAM includes both Motif and terminal interfaces.

- *Update and Install*
  Update is a feature that greatly simplifies the procedure of installing and updating both the operating system software and application software.

- *X Windows*
  This is the run-time components for X11R4 and X11R5. These components provide both client and server support for X Window-based applications. HP's X Window system allows applications written in GKS, Starbase, and other graphics libraries to have full access to HP graphics capabilities.

HP-UX also comes with a number of built-in Networking features:

- *TCP/IP Networking*
  TCP/IP-based services in HP-UX include:

  ARPA/Berkeley services-telnet, ftp, rlogin, remsh, sockets, tftp, rwho, finger, gated, BIND, and a number of others. These are the services that are set to 'deny' on new systems (for maximum security).

  These TCP/IP services are supported over FDDI, IEEE 802.3 and Ethernet. Kernel support for AT+T Streams is included in HP-UX RunTime. Streams itself is available as a separate product. Network File Systems (NFS) provides transparent file sharing and distributed applications between multivendor systems.

- *NetStart*
  NetStart provides the capability for a workstation to do a cold boot from a server over the network, and then load enough code from the server to reboot and do a network update of HP-UX.

- *Network Computing System (NCS)*
  NCS supplies a standard Remote Procedure Call (RPC) mechanism for running distributed client/server applications.

- *Other Networking Products*
  There are many other networking products that are available separately for HP-UX. These include:

  OSI products
  X.400
  X.25
  ISDN
  SNA and other IBM connectivity products

HP-UX also allows for Asynchronous Data Communications:

Communication services include the asynchronous multiplexer manager software, which supports CCITT modem communication, BSD job control, block-mode communication, and non-blocking I/O. The asynchronous multiplexer manager supports communication with other operating systems via one or more multiplexer channels, and hardwired or modem links using the uucp capabilities of HP-UX. The uucp commands provide file transfer, remote process execution, and virtual terminal capabilities.

OpenDDA

OpenDDA is the tool used to build A^{X}M applications that need to read and write LCN data. Read and write access of all TDC 3000^{X} point.parameters is available using OpenDDA. OpenDDA provides fourth generation language (4GL) extensions to commonly available programming languages. OpenDDA can be used in a testing/development environment as well as in the execution environment of the A^{X}M.

OpenDDA is available with the HP FORTRAN 90 compiler for 700 Series workstations (HP FORTRAN 90 is FORTRAN 77 compliant), HP ANSI C compiler, HP C++ compiler.

The following list outlines the functionality in OpenDDA:

- External Data Block (EDB) Definition and Parser. The EDB is where the user defines the LCN point.parameters (referred to as external variables) that they want to access and gives them associated local variable names. The user can also associate those external variables with a list (also referred to as a SET) or with many different lists. The user references the external variable in their program by using the assigned local variable name. You can also assign test values to each local variable which are used when OpenDDA is run in the test mode. The EDB is processed by OpenDDA and used to establish a relationship between the local variables and external data values (the external data values can be any LCN point.parameter).

- Generic 4GL-type executable statements which provide a protocol to access and reference external data without coding to the underlying platform's interface. These statements begin with EXEC DDA. The EXEC DDA statements provided are:

  EXEC DDA READ
  Reads in a SET, or ALL sets that were declared in the EDB.

  EXEC DDA WRITE
  Writes out a SET, or ALL sets that were declared in the EDB.

  EXEC DDA INITIALIZE
  This command initializes a communication port in the A^{X}M.

EXEC DDA LOAD STORE CODES
This command loads in the store codes of each external value.

EXEC DDA TERMINATE
Terminates the communication port in the A^{X}M.

EXEC DDA CLEAR CIO
Removes a memory resident CIO file.

EXEC DDA HIBERNATE EVENT
Suspends execution of the application awaiting an event initiated from the background CL block.

EXEC DDA LOAD ALIAS
Loads application variables with the point names, parameter names and array indexes from the memory-resident CIO file.

EXEC DDA LOAD MODIFY_CODE
Loads all application Modify Code variables with a specific Modify Code

EXEC DDA MODIFY CIO
Modifies the data source point names, parameter names, or array indexes for the currently selected memory-resident CIO file.

EXEC DDA READ CIO
Reads a disk-resident CIO file into memory.

EXEC DDA SELECT CIO
Selects a memory-resident CIO file to be the current active one.

EXEC DDA WRITE CIO
Writes a memory-resident CIO file to disk.

- A command line interface to all OpenDDA functions. Included with the command line statements are a number of script files that do the building/compiling/ linking of OpenDDA applications.
- OpenDDA expands the 4GL statements during the building of an application to interface with the underlying data access mechanisms in the Application Module^X.
- An application specific definition file which contains keywords, directives, qualifiers, and sub-qualifiers which allow you to declare operational characteristics about the application. A default definition file is also provided with OpenDDA.
- Support for FORTRAN, C, and C++.
- There are no restrictions placed on the location of the EXEC DDA execution statements. If the EXEC DDA commands are located in a different file, other than the main program, a command line qualifier is available to direct the pre-compiler to the name of the file they are located in. EXEC DDA INITIALIZE must be the first EXEC DDA statement executed.
- Test mode capability. This function provides the ability to use OpenDDA in a stand alone environment without an A^{X}M for the purpose of developing applications. Applications can be developed and tested against test data on an HP 700 Series Workstation running HP-UX 9.05. The applications must be installed on the A^{X}M before they can execute against real TDC data.

The example OpenDDA program on the following page is a simple example of an OpenDDA application. The purpose is merely to illustrate how an OpenDDA application is constructed. There are many options and features of OpenDDA that are not demonstrated in this example.

A^{X}M R200 New Features

Flexible Data Access

Dynamic Selections Sets
This function allows an application to modify the data source point name, parameter name, or array index at run time. This feature allows the programmer to create generic and repeatable programs, reducing development time and associated costs.

Re-map CIO (Control Input/Output) files
This function allows an application to read into memory one or more CIO files, be able to select between them, clear one or more CIO files from memory, and write out a selected CIO to disk.

High Performance Features

Hibernating Applications
This functionality provides the user the ability to force an X side OpenDDA application, initiated by a background CL block, to remain in memory after the first invocation, without requiring subsequent invocations to re-initialize the OpenDDA application.

Applications can resume execution at the appropriate entry point by receiving activate and terminate events from background CL blocks.

Array Data Transfer
The capability to define and access arrays was in OpenDDA R100 and R110; however, the underlying data access mechanism actually accessed each element one at a time. In OpenDDA 200, the underlying data access mechanism accesses real data type arrays of greater than 250 elements as an entire array (arrays of less than 250 elements are still accessed one at a time). This greatly increases the throughput rate of the A^{X}M when accessing large arrays.

Although array data transfer greatly improves performance when accessing large arrays, it is transparent to the user other than when monitoring the performance of the application. There are no special function calls or settings to make to take advantage of this enhancement.

Shared Library
Library files have changed from archive type to shared type, which conserve disk and virtual memory and facilitate migration.

Cache for OpenDDA - Execution time Cache handles to External Data
A cache of TDC 3000^X point name references is provided to facilitate superior performance when using Dynamic Selection Sets.

Cache Maintenance Tool
A cache Maintenance Tool is provided to maintain the cache.

Robustness Features

Data Access Request Retry
Recoverable data access errors are automatically retried.

LCN Access Priority
Low priority non-control data access requests are prevented from interfering with higher priority requests.

OpenDDA Robustness Features
OpenDDA will provide additional robustness in handling errors, when opening channels, when monitoring the priority of applications, on retries for memory constraints, and during initialization. These basic features will even further increase the availability of an OpenDDA application.

CIO File Signature
A signature is bound to every CIO file created and a corresponding one to the associated executable application. When an application attempts to read in a new CIO file at run time, the signature is checked to make certain that it is a valid CIO file for that application. If it is not valid, an error is returned to the application and the offending CIO file is not read into the application.

Inconsistency Error Enhancement
An INCONSISTENCY directive is added to the DEF file. It can specify ABORT or RETURN, which determines the behavior of the system when an inconsistency error is detected.

LCN type ENTITY supported
An OpenDDA application can both read and write to a TDC 3000^X parameter of type ENTITY using the ASCII point name. This allows an application to interact with generic programs in the classic AM.

License Management

A^{X}M R200 is the first Honeywell product to employ our new software license management utility. When a user purchases the base A^{X}M software they will have a license for:

- 1 A^{X}M Personality Node
- 1 On-line version of OpenDDA (Development and Execution version for the host A^{X}M)
- 1 Off-line version of OpenDDA (Development Version that can be used on a PIN computer in the test environment)

It is technically impossible for someone to use software they have not licensed for a particular IP address unless the call Honeywell's World-Wide Software Release Center and add to their license policy (purchase additional licenses). The purpose of license management is to prevent accidental and non-accidental abuse of software licensing policies.

OpenDDA Example Program

```
Description   : A FORTRAN example application that shows the use of
*               several OpenDDA features in one application.
*
*               Scenario:
*
*               We want to read data from 1 of 5 tanks.
*
*               The data collected will be used in some type of
*               database off the LCN.  The data needs to be read on
*               a cyclic bases.  However, we want the application to
*               wait in memory between collections.
*
*               We want to conserve some disk space and memory.  We know
*               the first 3 tanks are read quite frequently.  We therefore
*               decide once these CIO files are create (using DSS), we will
*               save these to disk and keep them in memory when they are
*               used.  The other seven will have their CIO files created
*               each time they are used.
*
*               We stay in memory until the LCN tells us to stop.
*
*               This application will need to be installed on-line, attached
*               to a CL point that sends events on some frequency.
************************************************************************
      PROGRAM READ_TANK_INFO_F
      IMPLICIT NONE
      INCLUDE 'dda_fortran.f'


EXTERNAL DATA BEGIN

TANK.VAL = 1.0 /REAL
  TANK.VS = INPUT CURRENT(,DDAE1TST.GDXRL(1))
  TANK.SC = OUTPUT
  TANK.SS = STORE_STATUS
  SET_LIST = 'TANK';

  DATA1.VAL = 111.0 /REAL
    DATA1.VS = INPUT CURRENT (,*.*(*))
    DATA1.SC = OUTPUT
    DATA1.SS = STORE_STATUS
    DATA1.POINT_NAME = FULL_NAME
    DATA1.ERS = EXT_REF_STATUS
    MODIFY_CODE = MODIFY_CODE
    SET_LIST = 'DATA';

  DATA2.VAL = 222.0 /REAL
    DATA2.VS = INPUT CURRENT (,*.*(*))
    DATA2.SC = OUTPUT
    DATA2.SS = STORE_STATUS
    DATA2.POINT_NAME = FULL_NAME
    DATA2.ERS = EXT_REF_STATUS
    MODIFY_CODE = MODIFY_CODE
    SET_LIST = 'DATA';

  DATA3.VAL = 333.0 /REAL
    DATA3.VS = INPUT CURRENT (,*.PV)
    DATA3.SC = OUTPUT
    DATA3.SS = STORE_STATUS
    DATA3.POINT_NAME = POINT_NAME
    DATA3.ERS = EXT_REF_STATUS
    MODIFY_CODE = MODIFY_CODE
    SET_LIST = 'DATA';
```

OpenDDA Example Program (continued)

```
TANK_POINTS(1,1:3) =
          'XGDX.GDXRL(101)','XGDX.GDXRL(102) 'SIM1' /STRING
  SET_LIST = 'LCN_POINTS';
TANK_POINTS(2,1:3) =
          'DDAE1TST.GDXRL(201)','DDAE1TST.GDXRL(202)', 'SIM2' /STRING
  SET_LIST = 'LCN_POINTS';
TANK_POINTS(3,1:3) =
          'DDAE1.GDXRL(301)', 'DDAE1.GDXRL(302)', 'SIM3' /STRING
  SET_LIST = 'LCN_POINTS';
TANK_POINTS(4,1:3) =
          'DDAREGE1.ULIMIT', 'DDAREGE1.LLIMIT', 'SIM4' /STRING
  SET_LIST = 'LCN_POINTS';
TANK_POINTS(5,1:3) =
          'DDAREGE1.LVLMDL', 'DDAREGE1.UVLMDL', 'SIM5'  /STRING
  SET_LIST = 'LCN_POINTS';

EXTERNAL DATA END

          STRUCTURE /REAL_STRUCT_TYPE/
            REAL*4      VAL             ! data value
            INTEGER*4   VS              ! value status
            INTEGER*4   SC              ! store code
            INTEGER*4   SS              ! store status
            INTEGER*4   ERS             ! external reference status
            CHARACTER*(DDA_FULL_NAME_SIZE) POINT_NAME       ! alias variable
          END STRUCTURE

          INTEGER NUM_CIO_FILES
          PARAMETER (NUM_CIO_FILES = 3)

          CHARACTER*12 CIO_FILES(NUM_CIO_FILES)
            DATA CIO_FILES /'tank1_f.cio',
   -                'tank2_f.cio',
   -                'tank3_f.cio'/
          CHARACTER*20 CIO_FILE_ID(NUM_CIO_FILES)
            DATA CIO_FILE_ID /'tank1',
   -                 'tank2',
   -                 'tank3'/
          LOGICAL*2 CIO_FILE_READ(NUM_CIO_FILES)
          DATA CIO_FILE_READ / .FALSE., .FALSE., .FALSE./
    RECORD /REAL_STRUCT_TYPE/ TANK
          INTEGER*2 CUR_TANK
          RECORD /REAL_STRUCT_TYPE/ DATA1,DATA2,DATA3
          INTEGER*4 MODIFY_CODE           ! modify code variable used for all points
          CHARACTER*(DDA_STRING_SIZE)  TANK_POINTS(5,3)
    LOGICAL*2 POINTS_READ
          DATA POINTS_READ / .FALSE. /

          RECORD /DDA_HIBER_EVENT_TYPE1/ EVT_DATA

          INTEGER*4 IOSTATUS,EVENT_STATUS
          INTEGER NUM_ACTIVATED
          DATA NUM_ACTIVATED / 0 /
          LOGICAL*2 BUILD_CIO

          INTEGER*4 SECS
```

OpenDDA Example Program (continued)

```
    Initializing Application Environment
          EXEC DDA INITIALIZE STATUS = IOSTATUS;

          starting the hibernate loop
800       CONTINUE

           waiting on the event
           EXEC DDA HIBERNATE EVENT, EVENT_DATA = EVT_DATA, STATUS=EVENT_STATUS;

            IF (EVT_DATA.EVENT_TYPE .EQ. DDA_EVENT_ACTIVATE) THEN

              NUM_ACTIVATED = NUM_ACTIVATED + 1

              get the tank number we will be reading
              exec dda read set='tank',status=iostatus;

              CUR_TANK = INT(TANK.VAL)
              PRINT *,'Working with tank #',CUR_TANK

              BUILD_CIO = .TRUE.
              If this is tank 1, 2, or 3, check to see if its cio file is in
              memory
              IF ((CUR_TANK .EQ. 1) .OR.
     -         (CUR_TANK .EQ. 2) .OR.
     -         (CUR_TANK .EQ. 3)) THEN
                 IF (CIO_FILE_READ(CUR_TANK)) THEN
                   if it is in memory, make in the active one

                        EXEC DDA SELECT CIO,ID = CIO_FILE_ID(CUR_TANK), STATUS=IOSTATUS;
                   IF (IOSTATUS .NE. HGSC_VALID) THEN
                         Try to read it off of disk
                    EXEC DDA READ CIO, ID = CIO_FILE_ID(CUR_TANK),
                                         FILE = CIO_FILES(CUR_TANK),
                                 STATUS=IOSTATUS;

                    IF (IOSTATUS .EQ. HGSC_VALID) THEN
                      if the file was found
                      BUILD_CIO = .FALSE.
                           ENDIF
                        ENDIF
                      ENDIF
              ENDIF

              if we need to build the cio file (using DSS) will do it here
              IF (BUILD_CIO) THEN
                make sure we use the cio file read in during the initialize.
                This is the one that we modify each time we need to build one

          EXEC DDA SELECT CIO ,ID='DDA_DEFAULT_CIO_ID',STATUS=IOSTATUS;

                Read the list of LCN points
          EXEC DDA READ SET='LCN_POINTS',STATUS=IOSTATUS;

                    set up point to modify the cio file
                    DATA1.POINT_NAME=TANK_POINTS(CUR_TANK,1)
                    DATA2.POINT_NAME=TANK_POINTS(CUR_TANK,2)
                    DATA3.POINT_NAME=TANK_POINTS(CUR_TANK,3)
                    MODIFY_CODE = DDA_MODIFY
                    EXEC DDA MODIFY CIO ,SET='DATA', STATUS=IOSTATUS

          at this point, the data could be written to disk in order to
          populate PCDE or a speadsheet or what ever.

          if we had to build the cio file and this is tank 1, 2, or 3,
          we will write out the cio file to disk so we will not need to build these again.
```

OpenDDA Example Program (continued)

```
          IF ((BUILD_CIO) .AND.
     -         ((CUR_TANK .EQ. 1) .OR.
     -          (CUR_TANK .EQ. 2) .OR.
     -          (CUR_TANK .EQ. 3))) THEN
            EXEC DDA WRITE CIO, FILE=CIO_FILES(CUR_TANK),
                        OVERWRITE=DDA_OVERWRITE_YES,
                        STATUS=IOSTATUS;
           we do not need to specifiy the id because the cio file we are
           writing is currently the active one.
                  CIO_FILE_READ(CUR_TANK) = .TRUE.


         ELSE IF (EVT_DATA.EVENT_TYPE .EQ. DDA_EVENT_TERMINATE) THEN
                     PRINT *,'Application activated ',NUM_ACTIVATED,' times.'
              ENDIF
            ENDIF

       IF ((EVENT_STATUS .EQ. HGSC_VALID) .AND.
     -    (EVT_DATA.EVENT_TYPE .EQ. DDA_EVENT_ACTIVATE)) GOTO 800


       Terminating Application Environment
        EXEC DDA TERMINATE STATUS = IOSTATUS;

999     CONTINUE
        END
```

Options

Hardware Options:

The A^{X}M can be installed in the console (either new or classic furniture) or the cabinet. If installed in the console, the following options are available:

- DAT drive (4 mm) in the console
- CD-ROM reader in the console

Wherever possible, it is highly recommended that you have a DAT drive directly on the A^{X}M. When A^{X}M is housed in a cabinet, these options are available as non-standard product from our Low Volume factory. Both these peripherals are more sensitive to harsh environments than the standard LCN products. It is only recommended that you choose this option if you are storing your cabinet in a 'computer' room environment.

There are a number of options for the LCN processor used in an A^{X}M:

- K4LCN-8 MW
- K4LCN-16 MW
- K2LCN-8 MW
- HMPU-6 MW

(Note: 6 MW is the minimal acceptable memory for R500 A^{X}M's)

The two options for the RISC processor speed are 64 and 100 MHz.

At least 32 MB of memory is required for the Application Module^X. Memory options include 64 MB, 128 MB or 256 MB. At least one 525 MB hard disk is required for 32 MB and 64 MB A^{X}Ms. 128 MB and 256 MB options require 1.2 GB hard disks or larger. Hard disk options include a 1.2 GB or 2.4 GB hard disk and an additional hard disk which may be either 525 MB, 1.2 GB or 2.4 GB. Hard disk sizes given are for unformatted drives. The primary drives are shipped with the A^{X}M software, HP-UX software and licensing software pre-loaded; the additional drives are shipped as blank formatted drives. OpenDDA is shipped on a DAT cartridge.

Table 2 — Remaining Hard Disk Space after Base Software Load + FORTRAN Compiler when Workstation RAM Memory is:

Unformatted Disk Size	32 B RAM	64 MB RAM	128 MB RAM	256 MB RAM
525 MB	200 MB	130 MB	N/A	N/A
1.2 GB	870 MB	800 MB	640 MB	320 MB
2.0 GB	1.9 GB	2.0 GB	1.8 GB	1.4 GB

Table 2 lists the space left on the drives after the base software (including OpenDDA) and the FORTRAN compiler have been loaded. Note that the higher the RAM memory on the machine, the bigger the swap file you must allocate on the hard disk. The A^{X}M is shipped from the factory configured with a default 90 MB of coprocessor device swap space. This is considered appropriate for 32 MB of RAM. It is recommended that you increase the size of the swap space by 70 MB if you have 64 MB RAM, by 230 MB if you have 128 MB RAM and by 550 MB if you have 256 MB of RAM (over and above the default swap space). It is not required that you do this, but you will not get the full benefit of your extra memory if you don't. Table 2 is assuming you allocate the recommended swap space for a given RAM memory configuration. **NOTE**: you can remove OpenDDA and your compiler if they are not being used (for instance if you are strictly using the A^{X}M for TotalPlant History) to gain some additional hard disk space.

The System Administration and Development Station is an optional device for use with an A^{X}M. See related section for more information.

Software Options

Software options for the A^{X}M are listed below:

FORTRAN compiler - This is one of the compiler options available to build OpenDDA applications. The compiler is the Hewlett-Packard FORTRAN 90 compiler for 700 Series workstations. HP FORTRAN 90 is a superset of FORTRAN 77. It has been specifically tuned for the latest PA-RISC machines (including the processor in the A^{X}M). Programmers can specify the optimization level that best meets desired trade-offs between compile time, code size, and execution performance. HP FORTRAN 90 is fully compliant with these FORTRAN standards:

- Complete implementation of ANSI 77 FORTRAN standard - ANSI X3.9-1978
- Compliant with FIPS FORTRAN 77 specification - FIPS PUB 69-1
- Compliant with MIL-STD-1753 extensions
- Compliant with POSIX FORTRAN 77 specification- POSIX P1003.2/standard
- Subset of FORTRAN 90 ANSI standard (ANSI X3.198-1992) and international standard (ISO/IEC 1539: 1991)

ANSI C Compiler - This is one of the compiler options available to build OpenDDA applications. The compiler is the HP C/ANSI C Compiler for 700 Series Workstations. This compiler is a conforming hosted implementation of the ANSI standard (X3. 159-1989). This product provides a comprehensive development environment and tools, including a symbolic debugger.

C++ Compiler - This is one of the compiler options available to build OpenDDA applications. The compiler is HP's object oriented C++ language for 700 Series Workstations. This is a true compiler based on AT&T's 3.0 version of C++. Optional translation of C code is retained for users who are dependent on this mechanism. Although C++ supports object oriented programming, the OpenDDA portion of your code that is created using this compiler is ANSI C code. This compiler generates native codes for the HP 9000 platforms directly from C++ source code. By eliminating the intermediary transition layer, the new compiler speeds compile time by up to 75%.

PC Data Exchange Software (PCDE) - PCDE is an X side NetDDE server which integrates the LCN point.parameter data with the many Microsoft Windows and PIN-based applications that use either DDE or NetDDE. See the PCDE Specification and Technical Data for more details.

OpenDDA may be purchased by itself to run in test mode on HP 700 Series workstations. The primary use of this will be for the OpenUSE Application Solution Providers to develop interfaces to the A^{X}M for their applications without having to purchase an A^{X}M.

NOTE: OpenDDA running on an HP workstation will NOT read and write TDC data - it will run with test values only. Full testing of a completed OpenDDA interface against real LCN data will require an A^{X}M. The Honeywell Integration Center in TPAC has an A^{X}M available for this type of testing that they offer as a service.

OpenDDA 200 Performance Benchmark Test Results
K2LCN and LCN R431

	READ parameters/sec	WRITE parameters/sec	Total elapsed time for program (sec)
OpenDDA 100 Benchmark I	327	291	13.27
OpenDDA 100 Benchmark II	350	262	13.76
OpenDDA 100 Benchmark III	474	491	1:23.04
OpenDDA 100 Benchmark IV	660	637	1:03.03
ABE 4.1 Benchmark I	146	127	30.72

Honeywell Applications

The current Honeywell applications available for the A^{X}M include:

- RMPCT - Honeywell's Robust Multivariable Predictive Control Technology
- DMCi^X - A Honeywell standard interface product to the DMC package. DMC (Dynamic Matrix Control) is a multivariable control package made by DMC Corporation.
- TPH - Honeywell's TotalPlant History package
- Blend Properties Control (BPC) for Oil Movements & Storage
- Batch History
- Cross Direction Controls II for Paper Machines
- Kamyr Open (Advanced Controls for Kamyr digesters)
- Environmental Solution- A complete environmental monitoring and reporting package using TPH, Versar and Pavilion software.

Third Party Applications

The current third party applications available for the A^{X}M include:

- PI - Oil Systems Plant Information package
- CIM 21- ISI
- G2- Gensym
- InfoPLUS X -Setpoint

There are a number of third parties signed up for the OpenUSE Solution Provider Program that are writing interfaces to the A^{X}M using OpenDDA.

OpenDDA Performance

The current throughput rates for a single OpenDDA application are approximately 350 parameters per second. When running multiple OpenDDA applications simultaneously, you will see an effective throughput of greater than 350 parameters/second.

Application examples of performance follow:

- The A^{X}M version of RMPC runs approximately 50 times faster than the AM version.

- The A^X^M version of DMC with DMCi^X^ runs about 300% faster than the VAX version for a medium sized controller (40x19).

The following performance benchmark testing was done on OpenDDA:

Benchmark I:

A single application accessing values from the local A^X^M node. Total parameters read and written is 2000.

Benchmark II:

A single application accessing values from an 'off-node' AM node on the local LCN. Total parameters read and written is 2000.

Benchmark III:

Ten concurrently executed applications accessing values from a local A^X^M node. Total parameters read and written is 20000.

Benchmark IV:

Ten concurrently executed applications accessing values from an 'off-node' AM node on the local LCN. Total parameters read and written is 20000.

OpenDDA 200 Performance Benchmark Test Results K4LCN and LCN R500

	READ parameters/sec	WRITE parameters/sec	Total elapsed time for program (sec)
OpenDDA 100 Benchmark I	546	542	7.52
OpenDDA 100 Benchmark II	510	495	8.14
OpenDDA 100 Benchmark III	1273	1257	31.85
OpenDDA 100 Benchmark IV	1198	1228	33.15
ABE 4.1 Benchmark I	146	127	30.72

Array Access Performance

Array Size	Unloaded System-Sequential WRITE (Parameters Per Sec.)	Unloaded System-Sequential READ (Parameters Per Sec.)
250	230	754
500	306	989
750	299	1190
1000	326	1309

ABE Benchmark:

The ABE benchmark test was done on a VAX/VMS - PLNM system. The measurements were obtained using ABE 4.1 and CM50S 3.1 with a CNI board running LAT protocol. This test should be compared to the OpenDDA 100 Benchmark II test.

Array Access Performance Results

R200 of the A^X^M provides the capability to access groups of point.parameters of 250 or larger as arrays. The data provided is linear in nature, even at arrays as high as 5000 point.parameters.

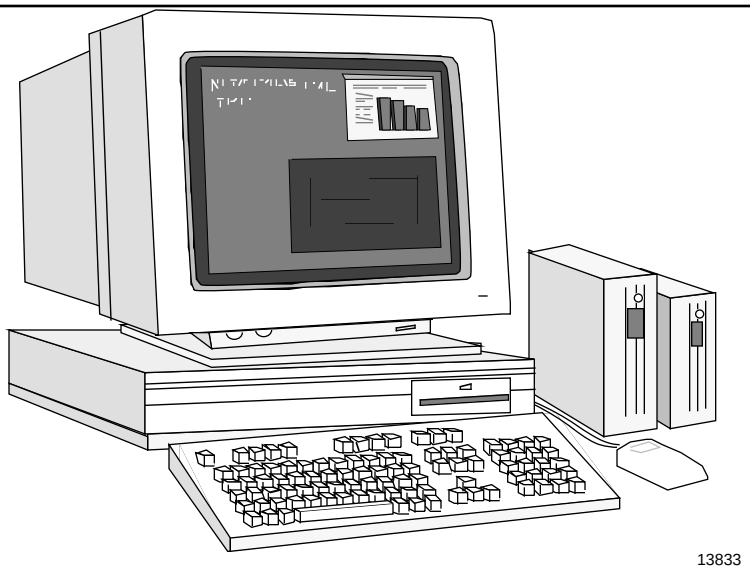


Figure 2 — System Administration and Development Station

Security

There are three main levels of security in the Application Module^X:

No direct writes to the LCN may be done from the network. Direct reads may be done using the PCDE software, but not direct writes. An OpenDDA application must reside in the A^{X}M and actually be doing the writes.

All the UNIX security that comes with HP-UX. HP-UX 9.05 contains the features intended to fulfill the US. Department of Defense Trusted Computer Systems Evaluation Criteria for C2.

This level of security includes an auditing facility for security relevant events, additional file access control lists for better file security, a shadow password file for storing encrypted passwords where only privileged users can read them, and security related documentation for system administrators and users.

The third level of security is in the LCN itself. The security switch is a three position switch in the form of a LCN system parameter.

The positions are:

- *Read Only* (No X side applications can write to the LCN)
- *CL Initiated Writes Only* (Only those X side applications that are scheduled from the AM's scheduler (CL initiation) can write to the LCN.
- *Read Write* (All UNIX authorized programs can both read and write to the LCN)

System Administration and Development Station

This section describes the features of the System Administration and Development Station, which is the minimum recommended configuration for administration of and development on the Application Module^X (A^{X}M). This workstation, made by Hewlett Packard, provides control engineers an environment to build advanced control solutions in a commonly available language that can run on the A^{X}M. By establishing an Ethernet link between the System Administration and Development Station and an A^{X}M, engineers can make full use of the graphical environment of the A^{X}M or create complex control solutions using OpenDDA on the System Administration and Development Station and then have them run on the A^{X}M.

The simplest OpenDDA development environment will be an A^{X}M configured with the System Administration and Development Station in a simple two-node configuration. A more complex OpenDDA development environment would involve attaching the A^{X}M to a Plant Information Network (PIN) that has a System Administration and Development Station or a Universal Station^X (U^{X}S) attached. Both configurations support basic administration and on-going maintenance of the A^{X}M and OpenDDA application development.

Customers having a U^{X}S with DAT drive and CD-ROM reader, or planning to purchase one as part of their entry into the open-systems arena, can use it as their A^{X}M user interface.

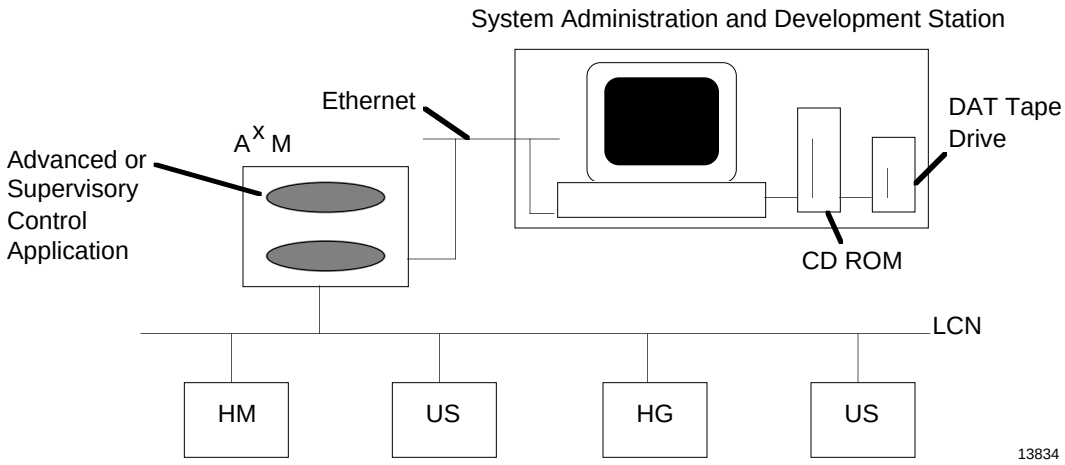


Figure 3 — Usage scenario for the Application Module^X with the System Administration and Development Station

In this case, the System Administration and Development Station is not required as all the administration and development can be done via the U^{X}S and users get the added benefit of having full LCN capabilities coupled with the workstation functionality.

The System Administration and Development Station comes with a DAT drive peripheral and CD-ROM peripheral. Engineers can get on-line help by accessing the entire set of Hewlett Packard manuals on CD-ROM media that comes with the base A^{X}M software. The DAT drive provides users a quick and easy way to backup the hard-disk on the A^{X}M or the System Administration and Development Station itself. The DAT drive can also be used to load new software on the A^{X}M.

Functional Description

Software

Every System Administration and Development Station comes with an HP-UX 9.05 Operating System. Along with the operating System, the workstation comes loaded with the X Window System; i.e., Version 11 Release 5 (X11R5) and OSF/Motif 1.2. The graphical user interface on these workstations is Hewlett Packard's X11 windowing environment (HP-VUE 3.0), which provides users the ability to organize windows with the concept of "workspaces". HP-VUE has a point-and-click editor (Vuepad) which can be used to write OpenDDA applications. This editor is an excellent alternative to the standard UNIX editor (vi).

Usage Scenarios

With both the System Administration and Development Station and the Universal Station^X the engineer can

- build and debug OpenDDA applications on the System Administration and Development Station and then down-load these applications to an A^{X}M.
- load the latest software on the A^{X}M through the DAT tape-drive peripheral.
- use System Administration Manager (SAM) to add users, configure the network, schedule repetitive jobs, perform system shutdowns and other system administration duties.
- backup the A^{X}M hard-drive on a DAT tape.
- get access to the entire set of Hewlett Packard manuals available on CD-ROM media.

As mentioned earlier an Ethernet link between the A^{X}M and the System Administration and Development Station could be in one of two configurations:

1. An end-to-end Ethernet connection between the System Administration and Development Station and the A^{X}M can be established should a user have no PIN or if the user does want to connect their A^{X}M to their PIN. This would provide the user with the graphical tools to administer the A^{X}M and develop OpenDDA applications. The user could use the DAT peripheral for hard-drive backups and software updates while documentation on CD-ROM would provide on-line help.
2. In many cases, the customer will have a PIN to which they would like to connect their A^{X}M and the System Administration and Development Station. Once both the A^{X}M and the System Administration and Development Station are on the PIN, the system will work the same as described in

step 1. There are a multitude of ways for other workstations to communicate with an AXM once it is on a PIN, even though Honeywell documentation only describes how the UXS and the System Administration and Development Station communicate with the AXM. Customers can seek the help of Integration Services to describe communication between an AXM and these other workstations (DECstation with X terminal capabilities, Sun Workstations, etc.).

HP-VUE also comes pre-loaded on the AXM and can be used to support remote X terminals being used for OpenDDA development or system administration (i.e., if the user wished to utilize the HP-VUE windowing environment for OpenDDA development or to administer the AXM, they would run HP-VUE on the AXM and re-direct the display to an X terminal).

X Windows

The X Window System is a sophisticated package of layered software that permits users at a variety of different computer platforms to share information from different applications through an easily customized graphic user interface. Not only can such data be visually shared, but multiple sets of data can be viewed in several "windows" that are opened simultaneously on a single CRT screen.

Additional Features of the System Administration and Development Station

The System Administration and Development Station has the following built-in interfaces to provide its users the best possible performance:

- Instant Ignition, which is a startup option that automatically configures system files on initial system power-up.
- IEEE (802.3) local area networking (LAN)
- External SCSI-2, bi-directional parallel (Centronics)
- Single RS-232C port.

Physical Description

Physical and Electronics Packaging

The AXM can be housed in either the new TDC 3000X furniture or the classic furniture when in the console. It can also be housed in a cabinet. The electronics for the AXM are either provided in a five-slot package or a ten-slot package depending on the LCN processor chosen.

For the K2LCN and K4LCN versions of the AXM, a five-slot package is used. For the HMPU versions, a ten-slot package is used. The Application ModuleX board compliment consists of:

- K2LCN, K4LCN or HMPU processor
- QMEM-4 FOR HMPU CONFIGURATION ONLY - Extra memory for the HMPU board.
- LLCN FOR HMPU CONFIGURATION ONLY - LCN connection for HMPU board.
- WSI-II - Work Station Interface II board with PA-RISC processor and memory boards.
- HDDT - Hard Drive Disk Tray board. Contains up to two workstation hard disks.
- K2LCN I/O or K4LCN I/O- Provides I/O connections for K2 and K4 boards respectively.
- WSI-II I/O - Provides I/O connections for WSI-II board including a serial port and Ethernet connection.
- HDDT I/O - Provides I/O connections for HDDT board including SCSI port.

Specifications

Card File Measurements

Five Card:

Approximate Dimensions		Approximate Weight
Height	7.0”	35#
Width	17.6”	
Depth	20.5”	

Ten Card:

Approximate Dimensions		Approximate Weight
Height	14.0”	45#
Width	17.6”	
Depth	20.5”	

Card File Power Options

Strap-Selected AC-Voltage Options	102-264
Frequency Options	50-60 Hz

Configuration Capability

The number of AXMs per LCN is limited only by the LCN capacity

Specifications of System Administration and Development Station

Central Processor	PA - 7100LC (PA-RISC)	
Clock Frequency	60 MHz	
Performance:	SPECint92	58.1
	SPECfp92	85.5
	MFLOPS(DP)	12.8
	AIM APR II	44.5
Memory and Cache:	Memory Capacity	32 MB
	Instruction Data Cache	64 KB
Monitor	15 inch 1024 x 768	
Multimedia	16 bit CD quality stereo audio JPEG and MPEG decompression	
Mass Storage	525 MB (Internal Capacity)	
Removable Media	1.44 MB Floppy	
User Friendly Features	HP VUE, OSF/Motif, X11 Window Sys, Instant Ignition	
Standard Interfaces	Single-ended SCSI-II, RS-232C, Centronics, PS/2 connectors (2), 2 I/O expansion slots	

Specifications (continued)

Sys Unit Pwr Src Consumption	2.7 A RMS max @ 120 V. 1.2 A RMS max @ 240 V
Auto ranging	90-132 Vac. 98-264 Volts
Line Frequency	47-63 Hz
Maximum power input	110 max Watts
Keyboard - Layout	101/102 Key PC standard
Mouse	3 button
Keyboard and Mouse Interface	Electrical IBM PS/2 compatible
CD ROM Peripheral	600 MB half-height
DAT Tape Drive	2 GB half-height DDS format

CE Conformity

CE Conformity (Europe)	This product is in conformity with the protection requirements of the following European Council Directives: 73/23/EEC, the Low Voltage Directive, and 89/336/EEC, the EMC Directive. Conformity of this product with any other "CE Mark" Directive(s) shall not be assumed. *Deviation from the prescribed procedures and conditions specified in the installation manuals may invalidate this product's conformity with the Low Voltage and EMC Directives.*
Product Classification	Class I: Permanently mounted, permanently connected Industrial Control Equipment with protective earthing (grounding). (EN 61010-1-1993)
Installation Category	Category II: Energy-consuming equipment supplied from the fixed installation. Local Level Appliances and Industrial Control Equipment . (EN 61010-1-1993)
Pollution Degree	Pollution Degree 2: Normally non-conductive pollution with occasional conductivity caused by condensation. (IEC 664-1-1992)
EMC Classification	Group 1, Class A, Industrial, Scientific and Medical (ISM) Equipment. (EN55011-1991; Emissions)
Method of Assessment	EMC: Technical Construction File (TCF) LVD: Technical File (TF)

Copyright, Trademarks, and Notices