Honeywell

OpenDDA
Customer Release Guide

DD04-200

Honeywell

OpenDDA
Customer Release Guide

DD04-200
Release 200
11/95

Copyright, Notices, and Trademarks

Revision 04 – October 15, 1996

Honeywell
Industrial Automation and Control
Automation College
2820 West Kelton Lane
Phoenix, AZ 85023

(602) 313-5669

Table of Contents

Table of Contents

Section 1—Introduction

1.1 About This Publication

Purpose

The OpenDDA Customer Release Guide provides

- details on functionality included in this release of Open Data Definition and Access (OpenDDA),
- minimum hardware and firmware revisions and minimum memory sizes,
- any special considerations when using the product,
- directions to installation of the OpenDDA software,
- problem reporting procedures, should any problems occur during the installation, migration, or ongoing use of OpenDDA.

Intended audience

This guide is intended for the system or network administrator's use to properly install OpenDDA.

Release

OpenDDA Release 200 is compatible with A^{X}M Software Release 200 and later.

1.2 Overview

Product description

OpenDDA (Open Data Definition and Access) 200 is a platform-independent software package providing a standard application programming interface. OpenDDA assists in developing and implementing applications with the following qualities:

- perform efficiently,
- read and write varying amounts of heterogeneous data,
- access one or more external data sources,
- can dynamically modify references to external data,
- can migrate, and
- are portable.

Platform

OpenDDA executes on the Application Module^{X} (A^{X}M) providing an open environment for the development and execution of custom control applications created by the customer.

1.3 New and Changed Functionality in this Release

Newfeatures The following table outlines the new features of OpenDDA 200.

Feature	Description
Shared Libraries	Library files have changed from access type to shared type, which conserve disk and virtual memory and facilitate migration.
Remapping CIO 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.
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.
LCN type ENTITY supported	An OpenDDA application can both read and write to a TDC 3000^X parameter of type ENTITY. This allows an application to interact with generic programs in the classic AM.
Dynamic Selection 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.
Hibernation	Allows a CL initiated application to initialize and then hibernate (suspend execution). CL initiated events can activate and terminate a hibernating 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.
Cache	A cache of TDC 3000^X point name references is provided to facilitate superior performance when using Dynamic Selection Sets. A cache Maintenance Tool is provided to maintain the cache.
Array Access Performance Improvement	Data access time has been reduced for reading arrays of greater than 250 elements for all data types except strings and enumerations.
Licensing	This version of OpenDDA contains license software for accessing the OpenDDA executable. YOU MUST FOLLOW THE STEPS TO OBTAIN AND INSTALL A LICENSE KEY BEFORE LOADING OPENDDA. Refer to the License *Installation and Administration A^XM* manual.

Changed functionality

The default value of the Definition (.def) file qualifier STORE_CODE has been changed. The new default is NOSTORE. (The previous value in R100 and R110 was STORE.)

The application's .def file is no longer required at run time; however, in order to change any directives of qualifiers in the .def file, you are still required to have one. If no .def file is present, OpenDDA will attempt to use a system defined .def file in the /opt/DDA/bin directory. Otherwise, OpenDDA default values are used. Please refer to subsection 8.1 of the *OpenDDA Reference Manual* for more information.

Array access performance improvement

The array access performance enhancement in OpenDDA R200 is transparent to the user, other than when monitoring the performance of an application. There are no special functions to call or settings to make to take advantage of this enhancement.

When OpenDDA determines that an array of greater than 50 elements is being accessed for all data types other than strings and enumerations, it uses a more efficient internal call to do the data access.

Obsolete files

The following files are no longer used in OpenDDA (R200 and later):

/opt/DDA/bin/fortran.def

/opt/DDA/bin/c.def

/opt/DDA/lib/dda_lib.a

/opt/DDA/lib/dda_gdx_lib.a

/opt/DDA/lib/dda_gen_lib.a

Default CIO and DEF file names

Beginning with R200, the default CIO and DEF files that OpenDDA looks for are named by appending .cio and .def to the *executable* file name. For example, when you parse myapp.f, the files myapp.cio and myapp.def are generated. In R100 and R110, these are the default CIO and DEF files, regardless of the name of the executable. In R200, if you renamed the "myapp" executable to "demo1", the default CIO and DEF files would be demo1.cio and demo1.def, instead of myapp.cio and myapp.def.

Precompiled source file name change

For release R100 of OpenDDA, the name of the precompiled FORTRAN source code generated by OpenDDA was application_name.src.f. For releases R110 and R200, the file name has been changed to application_name_src.f. You should use the new build script and make files. If you have created any custom build scripts or make files, you will need to make the appropriate changes.

Inconsistency error for INTEGER*2

OpenDDA does not support statuses or values declared as INTEGER*2 or short int. The application will receive an inconsistency error if your HLL variables used with OpenDDA are declared as such.

1.4 Supporting Documentation

Documentation

Before installing and operating OpenDDA 200, you will need to refer to the following manual:

- *License Installation and Administration A^{X}M*

In support of OpenDDA 200, the following documents have been published:

- *OpenDDA User's Guide*
- *OpenDDA Reference Manual*

When OpenDDA is installed on an A^{X}M, the following may also be useful:

- *Application Module^{X} Customer Release Guide*
- *Application Module^{X} System Administration Manual*

1.5 Software Package

Software media

OpenDDA 200 is distributed as a Software Development Kit on a 4 mm DAT tape.

Tape contents

The tape contains an HP-UX *`update`* tape. The software on this tape can be loaded into an A^{x}M or workstation running HP-UX that has a local DAT drive by using the `update` command. It can be loaded into a net distribution server by using the `updist` command.

Additional information on the installation procedure is provided in Section 3 of this guide.

Section 2—Before You Install

2.1 Special Considerations

Load license software

Prior to loading OpenDDA R200, you must follow the following steps to set up the software licensing. Please refer the the *License Installation and Administration A^{X}M* manual for detailed procedures for each step.

- Load Honeywell License Manager software
- Obtain license keys
- Edit the license file
- Set up the license environment variable for all OpenDDA users
- Determine if an options file is required, and, if so, set it up
- Install the license servers

Software loading order

To migrate your A^{X}M to OpenDDA R200, it is important to load software in the following order:

Table 2-1 Software Loading Order

Step	Software to Load
1	HP-UX 9.05 Patches (tape R950731)
2	Optional HP-UX 9.05 Patches (tape R950801) Load only the appropriate filesets—consult the Release Letter
3	Honeywell License software
4	A^{X}M HP-UX software
5	A^{X}M R200 personality
6	OpenDDA R200 software

Load compilers

Prior to using OpenDDA, the system administrator for your A^{x}M or HP-UX system should load the compilers that you plan to use—the Series 700 HP-UX FORTRAN compiler, and/or the HP-UX ANSI C or C++ compiler.

Even though a compiler is not needed for the installation, it is required for building any OpenDDA application on the system.

Default NG

It is highly recommended that you assign a default NG System ID to your TDC 3000^{X} if one is not already defined. The OpenDDA cache bases its Entity names search on the NG as well as the point and parameter names. If the NG ID has not been assigned when OpenDDA begins to use the cache, and it is assigned later, there may be duplicate entries in the cache.

2.2 Hardware Preparation

Installing OpenDDA on a remote node

When developing your OpenDDA applications on a remote node, the minimum level software version of your HP-UX 700 or 800 series should be 9.05 (you can determine this with the command `uname -a`). The processor should be PA-RISC version 1.1 (you can determine this with the `getcontext` command). The system should be equipped with at least 32 MB (Megabytes) of memory. You can verify the amount of memory you have with the `dmesg` command (you must be logged in as root).

Application Module^{x} configuration

To determine the minimum hardware and software requirements for your A^{x}M, please consult the *Application Module^{x} Customer Release Guide*.

Section 3—Installing OpenDDA 200

3.1 Overview

In this section

This section details the procedures for loading OpenDDA 200 from:

- a tape drive onto the local HP-UX node,
- a tape drive onto the netdist server, and
- a netdist server to the Application Module^{X}.

Updist command

The `updist` command prepares a device as a server.

You select filesets from the distribution tape and `updist` loads them into the `/netdist` directory.

Then the `netdistd` daemon is started. `netdistd` must be running in the server for other devices to update from the server.

Update command

You use the `update` command to load filesets/partitions from the server onto the system to be updated.

`Update` maintains a list of the filesets loaded in the directory:

`/etc/filesets`

Removing filesets from the server

You may want to remove filesets from the server before/after an update. The `updist` command loads filesets into the `/netdist/700 directory`.

For each fileset loaded, `updist` creates a source line entry in the main package definition file `/netdist/MAIN.pkg`. The `netdistd` daemon uses this file during the update process. Therefore, when removing files from the `/netdist/700` directory, you must also remove the corresponding source statements from `/netdist/MAIN.pkg`.

Since a fileset involves a tree of directories and files, use the -r option with the remove command.

Example:

```
rm -r /netdist/700/*
```

This command will remove all files and directories in `/netdist/700`.

Reference

Refer to the *man* pages for `netdistd` and `update` (also covers `updist`) for more information.

3.2 Load from a Tape Drive to a Local HP-UX Node

Procedure

The following table lists the steps for loading OpenDDA from a tape drive to a local HP-UX node.

Table 3-1 Load OpenDDA from a Tape Drive to a Local HP-UX Node

Step	Action
1	Login to the host system as: `root`
2	Invoke the UPDATE utility using the following command: `update`
3	From the UPDATE menu, select the option: `Change Source or Destination`
4	From this menu, select the option: `From Tape Device to Local System`
5	Make the following changes and responses for this screen: • Press RETURN in response to the question: `"Create or change the address of /dev/update.src (y or n)" n` • **Source:** Enter: `/dev/rmt/0m` • **Destination:** `/` (The `'/'` default is correct. Do not change this field.)
6	Click Done, or from a character-oriented terminal, press CTL F followed by 4 to return to the main menu
7	From the UPDATE main menu, select the option: `Select All Filesets on the Source Media->`
8	Select the `Start Loading Now...` option.
9	Review the UPDATE instructions and enter "Y" to begin loading or "N" to cancel.
10	When control returns to the prompt, check the `/tmp/update.log` file for errors using the following command: `tail -50 /tmp/update.log | more` • If no errors or warnings were detected (`*error` or `*warning`) in the log file, the OpenDDA for A^{X}M software was loaded successfully. • If errors or warnings were detected, correct them and return to Step 2.

Files needed to link

When developing OpenDDA applications on a remote system, you will require the following files from the A^{X}M, in order to link your application and build an executable file.

- `/opt/TDC_Open/common/lib/libhiber.sl`
- `/opt/TDC_Open/common/lib/liblxs.sl`

The OpenDDA installation process will automatically load these files if they are not already there.

3.3 Installing OpenDDA on the netdist Server

Introduction

The following procedure describes the installation procedures required to install OpenDDA on the netdist server in preparation to load OpenDDA onto your Application Module[X].

Configuring an HP-UX as a netdist server

The following table lists the steps for loading OpenDDA from a tape drive to a local HP-UX node.

Table 3-2 Configuring an HP-UX on a Netdist Server

Step	Action
1	Login to the host system as: `root`

2	Verify the `/netdist` directory exists:
	`ls -al | more`
	If it does not exist, execute the following commands:
	`mkdir /netdist`
	`mkdir /netdist/700`

3	Set your default directory to `/netdist` using the following command: `cd /netdist`

4	Invoke the `UPDIST` utility to setup the network distribution server as follows:
	`updist`
	If you get an error message suggesting you delete the file `/update.lock` and retry, enter:
	`rm /update.lock`
	`updist`

5	Verify the source and destination are the following:
	`Source: Tape Device Destination: Local System`
	`/dev/rmt/0m /netdist`
	If not, then perform the next three steps:
	1. From the main menu, select the option:
	`Change Source or Destination`
	2. Select the `From Tape Device to Local Systems` option
	3. Make the following changes to the source and destination fields:
	• **Source:** Enter the specification of the tape drive `/devt/rmt/0m`.
	• **Destination:** specify the full pathname of the netdist file directory (`/netdist`)

6	Insert the OpenDDA tape into the tape drive. Click Done, or from a character-oriented terminal, press CTL F followed by 4 to return to the main menu.
7	From the `UPDIST` main menu, select the option: `Select All Filesets on the Source Media->`

Table 3-2 Configuring an HP-UX on a Netdist Server (Continued)

Step	Action
8	Select the `Start Loading Now...` option.
9	Review the `UPDIST` instructions and enter "Y" to begin loading or "N" to cancel.
10	When control returns to the prompt, check the `/tmp/update.log` file for errors using the following command: `tail -50 /tmp/update.log | more` • If no errors or warnings were detected (`*error` or `*warning`), continue on to the next step. • If errors or warnings were detected, correct them and return to Step 3. **Note:** `-50` will display the last 50 lines from the end of the log.

11	Check the file `/etc/services` for the line which identifies the netdist service port by entering the following:
	`grep netdist /etc/services`
	You should see a message like this:
	`netdist 2106/tcp # update (1m) network...`
	If not, edit the file `/etc/services` and insert the following line at the end:
	`netdist 2106/tcp # update (1m) network...`

12	Enter the following commands.
	Start the netdist daemon (located in the /etc directory):
	`netdistd`
	Verify that it is running:
	`ps -ef | grep netdistd`
	Result: You should see a line something like this indicating `netdistd` has started and is running:
	`root 2517 1 0 13:27:09 ? 0:00 netdistd`
	The network distribution daemon is now running and ready to load OpenDDA onto your Application Module^{x}.

3.4 Loading OpenDDA from a netdist Server to the A^{x}M

Procedure The following table lists the steps for loading OpenDDA from a netdist Server to your Application Module^{x}.

Table 3-3 Load OpenDDA from a netdist Server to the A^{x}M

Step	Action
1	Login to the A^{x}M as: `root`
2	Invoke the UPDATE utility using the following command: `update`
3	From the UPDATE menu, select the option: `Change Source or Destination`
4	From this menu, select the option: `From Netdist Server to Local System`
5	Modify/verify the source and destination fields. • **Source:** Enter the name of the netdist server and verify the netdist service port number. (To obtain the service port number, refer to Step 11 in Table 3-2.) • **Destination:** / - The default value `'/'` is correct.
6	Click Done, or from a character-oriented terminal, press CTL F followed by 4 to return to the main menu.
7	From the UPDATE main menu, select the option: `Modify/Revise Partitions and Filesets...`
8	Follow the instructions to select the partition described as: `Honeywell OpenDDA for the AxM`
9	Click Done, or from a character-oriented terminal, press CTL F followed by 4 to return to the main menu.
10	Select the `Start Loading Now...` option if not already selected.
11	Review the UPDATE instructions and enter "Y" to begin loading or "N" to cancel.
12	When control returns to the prompt, check the `/tmp/update.log` file for errors using the following command: `tail -50 /tmp/update.log | more` • If no errors or warnings were detected (`*error` or `*warning`) in the log file, OpenDDA was loaded successfully. • If errors or warnings were detected, correct them and return to Step 4. **Note:** `-50` will display the last 50 lines from the end of the log.

3.5 Directories and Files Created by Loading OpenDDA for A^X^M

Installation output

The following are the directories and files generated by the OpenDDA installation procedure.

/opt/DDA

Directory : `/opt/DDA`

```
-rw-rw-rw- 1 axm axm .... DDACache
drwxr-xr-x 2 axm axm .... bin
drwxr-xr-x 2 axm axm .... doc
drwxr-xr-x 2 axm axm .... examples
drwxr-xr-x 2 axm axm .... include
drwxr-xr-x 2 axm axm .... lib
drwxr-xr-x 2 axm axm .... tmp
```

/opt/DDA/bin

Directory: `/opt/DDA/bin`

Files:

```
-rwxr-xr-x 1 axm axm .... c.c
-rwxr-xr-x 1 axm axm .... cmt
-r-xr-x--- 1 axm axm .... dda
-r-xr-xr-x 1 axm axm .... dda_system.def
-rwxr-xr-x 1 axm axm .... dda_template.def
-rwxr-xr-x 1 axm axm .... dda_xstatus
-rwxr-xr-x 1 axm axm .... fortran.f
```

/opt/DDA/lib

Directory: `/opt/DDA/lib`

Files:

```
-r-xr-xr-x 1 axm axm .... libdda.sl
-r-xr-xr-x 1 axm axm .... libddagdx.sl
-r-xr-xr-x 1 axm axm .... libddagen.s1
```

/opt/DDA/include

Directory: `/opt/DDA/include`

Files:

```
-rwxr-xr-x 1 axm axm .... dda_c.h
-rwxr-xr-x 1 axm axm .... dda_c_errors.h
-rwxr-xr-x 1 axm axm .... dda_fortran.f
-rwxr-xr-x 1 axm axm .... dda_fortran_errors.f
-rwxr-xr-x 1 axm axm .... dda_status_include.h
-rwxr-xr-x 1 axm axm .... hgsc_c_errors.h
-rwxr-xr-x 1 axm axm .... hgsc_fortran_errors.f
```

/opt/DDA/examples

Directory: `/opt/DDA/examples`

Files:

```
-rwxr-xr-x 1 axm axm .... alltestc.c
-rwxr-xr-x 1 axm axm .... alltestc.def
-rwxr-xr-x 1 axm axm .... alltestf.def
-rwxr-xr-x 1 axm axm .... alltestf.f
-rwxr-xr-x 1 axm axm .... dda_c_examples.make
-rwxr-xr-x 1 axm axm .... dda_create_c_debug_executable
-rwxr-xr-x 1 axm axm .... dda_create_c_executable
-rwxr-xr-x 1 axm axm .... dda_create_cpp_debug_executable
-rwxr-xr-x 1 axm axm .... dda_create_cpp_executable
-rwxr-xr-x 1 axm axm .... dda_create_fortran_debug_executable
-rwxr-xr-x 1 axm axm .... dda_create_fortran_executable
-rwxr-xr-x 1 axm axm .... ddademo.c
-rwxr-xr-x 1 axm axm .... ddademo.def
-rwxr-xr-x 1 axm axm .... ddademo.f
-rwxr-xr-x 1 axm axm .... dda_fortran_examples.make
-rwxr-xr-x 1 axm axm .... lvrf.def
-rwxr-xr-x 1 axm axm .... event_hiber.c
-rwxr-xr-x 1 axm axm .... readit.def
-rwxr-xr-x 1 axm axm .... readit.f
-rwxr-xr-x 1 axm axm .... read_tank_info.c
-rwxr-xr-x 1 axm axm .... remap.cf
-rwxr-xr-x 1 axm axm .... remap2.edb
-rwxr-xr-x 1 axm axm .... remap3.edb
```

/opt/DDA/doc

Directory: `/opt/DDA/doc`

File:

```
-rwxrwxrwx 1 axm axm size date dda.help
```

/opt/DDA/tmp

Temporary work directory used by OpenDDA.

3.6 Loading OpenDDA on a Workstation

Discussion

When you are loading OpenDDA on a workstation for application development, all of the directories and files covered in subsection 3.5 will be loaded. In addition, there is a directory with two library files that are standard in an A^{X}M that you must have. These two files are required for linking an application.

The OpenDDA installation process checks for the existance of the directory and three library files, and if they are not present, the directory is created and the files are loaded.

/opt/TDC_Open/common/lib

The directory `/opt/TDC_Open/common/lib` will be created if not already present. The following files will be loaded into this directory if not already present:

```
-r-xr-xr-x 1 root bin .... libhiber.sl
-r-xr-xr-x 1 root bin .... liblxs.s1
```

3.7 Adding Users

Assigning axm group

When creating OpenDDA users, be sure to assign each user as a member of the "axm" Group.

> **ATTENTION**
>
> They must be a member of the "axm" group to install and execute an OpenDDA application.

Reference

Refer to 4.3 "Adding and Removing Users" in the *Application Module^{x} System Administration* manual for instructions on adding OpenDDA users.

3.8 Migrating OpenDDA R100 and R110 Applications

Loading software

To migrate your A^{X}M to OpenDDA R200, it is important to load software in a specific order. Refer to "Software loading order" in Special Considerations Before You Install on page 7.

Installed CL-initiated applications

No actions are required for OpenDDA applications installed as CL-initiated tasks in the `/users/axm` directory. After all software is loaded, these applications will execute when the system is brought back up.

Script and make file changes

Because of necessary library name changes, additional libraries to include, and OpenDDA FORTRAN source file name change, the example script files and make files have changed in the `/opt/DDA/examples` directory. If you have copied and modified these files, you will need to review the new files and make any necessary changes to your custom script and make files.

Changes to applications

If any changes are made to your application, you will be required to fully rebuild your OpenDDA application. The following steps should be taken after an application has been modified, parsed, or even relinked.

Table 3-4 Steps to Rebuild an OpenDDA Application

Step	Action
1	Use the Complete Build option (`-cb`) with the `dda` command to recreate the OpenDDA `.set`, `.set_idx`, and `.cio` file.
2	Use the appropriate example script file provided in the `/opt/DDA/examples` directory to compile and link your application.
3	Execute the application to verify functionality.

Section 4—Problem Reporting

4.1 Overview

Introduction

The information provided in this section is intended to assist the administrator and Honeywell's Technical Assistance Center (TAC) in resolving problems encountered when developing and executing applications in OpenDDA.

4.2 Errors While Building an Application

Build application errors

While using any of the command line features in OpenDDA, the most important information is output to the screen when using these functions. The following is a list of files Honeywell's Technical Assistance Center (TAC) may require when determining the resolution to your problem:

- the Application's Definition File (.def)
- the application's file containing both the EDB statements and EXEC DDA statements.

Additional information

Additional information for TAC may be any or all of the following:

- command in use when the error occurs
- a screen dump of the output generated by OpenDDA when the error occurred
- a snapshot of the current processes running on the system
- any script files or make files used when executing the OpenDDA command
- release number of the OpenDDA software and A^{X}M libraries—you can determine these with the commands:

```
what  /opt/DDA/bin/dda
what  /opt/TDC_Open/common/lib/*.sl
```

- the application's .edb (External Data Block) file
- the checksum value of the shared libraries on your system—you can determine these with the commands:

```
sum -p /opt/DDA/lib/*.sl
sum -p /opt/TDC_Open/common/lib/*.sl
```

4.3 Execution Errors

Execution output file

An output file is created by OpenDDA when executing an application. The file name has the format:

`your_application_name.`**dda_out**

This file can contain error and information messages that will assist in debugging your applications.

Output file contents

The **.dda_out** file may contain the following information:

- output generated by the PRINT switches set in your application's Definition file (**.def**)
- output from the application itself if you have written to Standard Output
- error or warning messages from DDA if your application experienced problems during execution.

ATTENTION

Be sure you turn ON the PRINT Values, Statistics, and Messages in the application's Definition file before executing the application, when you are debugging your application.

Section 5—Known Problems

5.1 Outstanding PARs and Known Problems

Enumerated index arrays

OpenDDA cannot access arrays indexed by enumerations.

Reading an array's entity_id parameters

If the entity_id your OpenDDA application is accessing is NULL, the value returned to your application depends on the value status. If the status is good, then your value will be a blank string. If the value status is not good, then the value will be the previous value accessed.

TEST variable problem

The TEST ancillary variable does not function properly with the use of multiple test values and FULL_NAME alias arrays.

The use of the TEST ancillary variable with multiple test values and FULL_NAME alias arrays results in invalid run time read access for the TEST ancillary variable. This only affects the HLL value variable that is associated with the TEST ancillary keyword in the EDB. Only the first test value will be correctly populated into the HLL variable as opposed to all of the test values.

Example:

```
val1[0:3] = 1.0, 2.0, 3.0, 4.0 /REAL
val1_sts[0:3] = INPUT CURRENT (, *.*(*))
val1_fn[0:3] = FULL_NAME
val1_testval[0:3] = TEST
SET_LIST = SET_A;
```

When `SET_A` is read, only the first element of the `val1_testval` array will be populated correctly with the value, 1.0. The values populated for the remaining three array elements will be undefined.

NOTE: The population of 'live' data into `val1` is unaffected.

Workaround:

By using a single test value, proper functionality can be achieved.

TEST MODE problem

The non-sequential duplication of the first test value in a list of test values results in invalid test mode read access.

If multiple test values are used where the first test value is repeated non-sequentially in the list, then improper test mode access will result such that invalid test values are populated.

Example:

```
val1[0:3] = 1.0, 2.0, 1.0, 4.0 /REAL
val1_sts[0:3] = INPUT CURRENT (, PT1.PARAM1(1))
SET_LIST = SET_A;
```

Because the first test value, 1.0, is duplicated as the third test value (non-sequential duplication), upon reading `SET_A` improper values will be returned into the the `val1` array when executing in test mode.

NOTE: Normal ‘live’ data access is unaffected.

Workaround:

Make sure the first test value in the list is unique or do not use test mode.

5.2 Problems Fixed This Release

List of fixed problems

The following PAR and other problems are fixed in this release:

- Accessing the zeroth element of an array on the LCN
- Accessing individual elements of arrays greater than the 257th element on the LCN
- The size of OpenDDA's .set file has been cut in half
- The amount of memory consumed by OpenDDA for processing numerous SETs declared in the application has been reduced
- When the values print switch was turned on in the application's .def file, and the user had a STORE_STATUS defined in the EDB, the STORE_STATUS value was being printed during an EXEC DDA READ command, causing a false error message to be printed in the .dda_output file

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: **OpenDDA Customer Release Guide** Issue Date: **11/95**

Publication Number: **DD04-200**

Writer: **J. Kennedy**

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.)

Honeywell

Industrial Automation and Control
Honeywell Inc.
16404 North Black Canyon Highway
Phoenix, Arizona 85023-3033

Helping You Control Your World