



***Malibu Antenna Control
Custom Derived Function***

May 2013

SYMVIONICS Document SSD-IADS-034

© 1996-2018 SYMVIONICS, Inc.

All rights reserved.



Table of Contents

1. Introduction.....	3
1.1. Overview	3
2. Installation.....	3
3. Instructions for Use.....	4
3.1. Using the ACU Server Function	4
3.1.1. Creating the ACU Server Derived Parameter.....	4
3.1.2. ACU Server Input Arguments and Return Values.....	5
3.1.3. Run-time Operation	6
3.2. Using the ACU Status Function.....	7
3.2.1. Creating the ACU Status Derived Parameter.....	7
3.2.2. ACU Status Input Arguments and Return Values	8
3.2.3. Run-time Operation	9
Table 2-1 IadsMalibuControl DLL Installed Functions	3
Table 3-1 ACU Server Input Arguments	5
Table 3-2 ACU Server Return Values	5
Table 3-3 ACU Status Input Arguments.....	8
Table 3-4 ACU Status Variable Request Numbers.....	8
Table 3-5 ACU Status Return Values	8
Figure 3-1 Derived Parameters in the IADS Configuration Tool.....	4
Figure 3-2 ACU Server Functioning Properly with Heartbeat Shown.....	6
Figure 3-3 Examples of ACU Status Parameters in IADS	7

1. Introduction

This document describes the IADS Malibu Control custom derived function, developed to control and receive status from a Malibu Antenna Control Unit (ACU).

1.1. Overview

The Malibu Antenna Control custom derived function is provided as a Dynamic Link Library (DLL) which contains the two internal functions for interface and control of the Malibu ACU. Both functions use Ethernet via the UDP protocol to send and receive data. Both functions are entered and accessed as regular IADS parameters using the IADS Configuration Tool for entry.

2. Installation

The two functions are contained within the *IadsMalibuControl* DLL. To install them, run the “*regsvr32*” program normally found in the “*c:\Windows\System32*” directory as shown here:

```
regsvr32 IadsMalibuControl <cr>
```

Optionally the IADS Operator Console can be used to distribute the DLL and register these functions at all workstations; this is primarily needed for the *ACUStatus* function because the server function is normally run on a single computer.

Once complete, these functions are now registered and are available for use within the IADS derived equation engine (please see the IADS On-line Help System for additional information using derived equations). The following table shows the Program IDs (ProgIDs) that are used for reference when creating derived functions:

<i>ProgId</i>	<i>Description</i>
<i>IadsMalibuControl.ACUServer</i>	Identifies the derived function that is used to send pointing data to the Malibu ACU via Ethernet
<i>IadsMalibuControl.ACUStatus</i>	Identifies the function used to get status from the Malibu ACU

Table 2-1 *IadsMalibuControl* DLL Installed Functions

3. Instructions for Use

This section will detail how to install and setup each of the derived functions within IADS. Figure 3-1 shows all of the functions used to create the IADS derived parameters using the Configuration Tool. Please note that both of these functions are not available in playback mode unless there exists the same Ethernet connection to the Malibu ACU that was used during the real time operation.

Parameter	DataSourceType	DataSourceArgument
MalibuServer	Derived	ladsMalibuControl.ACUServer(10000, Target1_lat, Target1_lon, Target1_alt*100.0, 1)
MalibuStatus0	Derived	ladsMalibuControl.ACUStatus(4390, 0)
MalibuStatus1	Derived	ladsMalibuControl.ACUStatus(4390, 1)
MalibuStatus2	Derived	ladsMalibuControl.ACUStatus(4390, 2)
MalibuStatus3	Derived	ladsMalibuControl.ACUStatus(4390, 3)
MalibuStatus4	Derived	ladsMalibuControl.ACUStatus(4390, 4)

Figure 3-1 Derived Parameters in the IADS Configuration Tool

3.1. Using the ACUServer Function

The purpose of this function is to send data to the ACU from IADS during a real time test using an Ethernet/UDP communication protocol. It receives Latitude, Longitude and Altitude in floating point format from the test vehicle via the ground station and sends that data to the Malibu ACU. This function is setup in IADS as a derived equation Plugin which can be run from a Client display. However, because it is a continuous process, it is recommended to convert this function to an IAP type for real-time operation. This is done by changing its type to IAP in the “DataSourceType” dropdown in the Configuration tool. The IAP parameter will then run on the Caching Data Server (CDS) machine (note that IAP parameters do not work in IADS playback mode).

3.1.1. Creating the ACUServer Derived Parameter

The ACUServer function is created as an IADS derived using the Configuration Tool (see the example in Figure 3-1).

To create the function:

- 1) In IADS, on the Dashboard click the **Configuration** button.
- 2) Open the **Data** folder, then click **Parameter Defaults**.
- 3) Copy and paste an existing row of data for a parameter that is similar to the one you are creating.
- 4) In the *Parameter* column type **MalibuServer**.
- 5) In the *Data Source Type* column, select **Derived**.
- 6) In the *Data Source Argument* column, type:
IADSMalibuControl.ACUServer(10000, Target1_lat, Target1_lon, Target1_alt*100.0,1)
- 7) Click a save option.

3.1.2. ACUServer Input Arguments and Return Values

This table details the input arguments, which are required in order to properly run this function, If not entered properly; IADS will flag a syntax error.

<i>Input Argument Name</i>	<i>Description</i>
<i>PortId</i>	Identifies the Port ID that the Malibu ACU unit is expecting to receive the data on
<i>Latitude</i>	Vehicle latitude in degrees as a floating point value
<i>Longitude</i>	Vehicle longitude in degrees as a floating point value
<i>Altitude</i>	Vehicle altitude in 1/100 of a meter as a floating point value
<i>ByteSwap</i>	Data order sent to the Malibu ACU, 1 = Big endian, 0 = Little endian

Table 3-1 ACUServer Input Arguments

This table details all return values. The ACUServer function will return negative numbers upon a failure, otherwise and increasing heartbeat value starting at 1 will be returned:

<i>Return Value</i>	<i>Description</i>
<i>-1</i>	The required five arguments were not passed into the function. This most likely occurred because the entry in the "DataSourceArgument" field of the parameter is incorrect
<i>-2</i>	One or more of the input arguments was not passed in as a floating point value
<i>-3</i>	The ACUServer was unable to connect the Malibu ACU on the specified port
<i>0</i>	Normal Operation - Packet not sent
<i>1..n</i>	Normal Operation - Each incremental value denotes a packet sent to the Malibu ACU

Table 3-2 ACUServer Return Values

3.1.3. Run-time Operation

The ACUServer function can run as a Client-side or IAP derived parameter. Initially it may be convenient to run the function as a Client-side derived to verify proper operation. However once complete, it is recommended to switch the type to IAP for real time operation (please see the IADS User Manual for instructions on making IAP parameters). For verification of operation, the ACUServer Parameter can be added to a Stripchart as show in Figure 3-2. Verify operation by looking for an ever increasing integer values. This heartbeat denotes when a packet is sent to the Malibu ACU, which is gated and will occur approximately 2 times per second (please note that that verification of the data received by the ACU will need to be verified within their interface program).

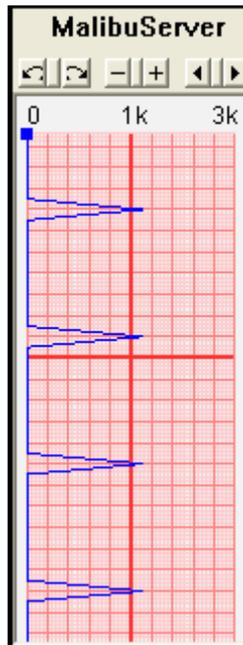


Figure 3-2 ACUServer Functioning Properly with Heartbeat Shown

The ACUServer function expects latitude and longitudinal data in floating point degrees. This data is then converted to integer using the following equation: $(\text{latitude} * 5965232.3555555)$. Altitude is sent as a floating point and uses the value passed into the equation without alteration. Any scaling, for example conversion from feet to meters, will need to be completed within the equation as shown in the example parameter in Figure 3-1.

3.2. Using the ACUStatus Function

The purpose of this function is used to make IADS parameters from the Status values returned from the Malibu ACU for display within the IADS system during real time operation. Data is collected via an Ethernet UDP connection to the ACU on the requested PortId of the ACUStatus derived parameter.

3.2.1. Creating the ACUStatus Derived Parameter

The ACUStatus functions are created as an IADS derived using the Configuration Tool (see the example in Figure 3-1). A separate derived function must be entered for each ACU Status value needed for display (see table 3-4 for a list of the available status variables). Here are two examples of these functions:

Required to receive the ACU Status packet → `IadsMalibuControl.ACUStatus(4390, 0)`

Return Azimuth Pedestal Position → `IadsMalibuControl.ACUStatus(4390, 1)`

To create a status function:

- 1) In IADS, on the Dashboard click the **Configuration** button.
- 2) Open the **Data** folder, then click **Parameter Defaults**.
- 3) Copy and paste an existing row of data for a parameter that is similar to the one you are creating.
- 4) In the *Parameter* column type **MalibuStatus1**.
- 5) In the *Short Name* column type **Az Pedestal Position**.
- 6) In the *Data Source Type* column, select **Derived**.
- 7) In the *Data Source Argument* column, type: **IadsMalibuControl.ACUStatus(4390, 1)**
- 8) Click a save option.

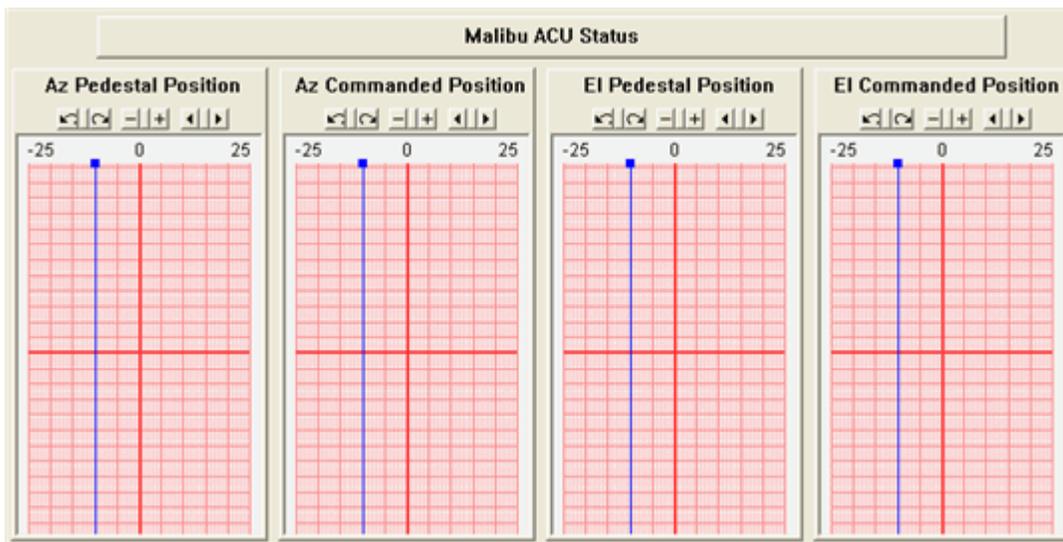


Figure 3-3 Examples of ACUStatus Parameters in IADS

3.2.2. ACUStatus Input Arguments and Return Values

	<i>Description</i>
<i>PortId</i>	Identifies the Port ID that the Malibu ACU unit is sending status data on
<i>Variable Request Number</i>	Integer number that identifies which ACU Status variable to return. The value of (0) is required in order to receive the ACU Status Packet (see table 3.4)

Table 3-3 ACUStatus Input Arguments

<i>Variable request number</i>	<i>Description</i>
<i>0</i>	ACU Status Packet
<i>1</i>	Azimuth Pedestal Position
<i>2</i>	Azimuth Commanded Position
<i>3</i>	Elevation Pedestal Position
<i>4</i>	Elevation Commanded Position

Table 3-4 ACUStatus Variable Request Numbers

<i>Return Values</i>	<i>Description</i>
<i>-1</i>	Internal Error - Memory pointer error
<i>-2</i>	Internal Error - Memory type error
<i>-3</i>	Incorrect number of input arguments were passed in
<i>-4</i>	Input arguments were not passed-in as floating point values
<i>-5</i>	Unable to open the UDP port as requested by the “PortId” argument
<i>-6</i>	Internal Error - Unable to set Socket property
<i>-7</i>	Internal Error - Error with Socket Receive
<i>-8</i>	Packet received is not a Malibu status packet which is identified by the 4 th byte as the ‘B’ character
<i>-9</i>	Successful return code, this code is only returned from the function that passes in (0) as its variable request number
<i>-10</i>	Unknown “WhichVar” argument. Currently the legal values are 1, 2, 3 and 4

Table 3-5 ACUStatus Return Values

3.2.3. Run-time Operation

The ACUStatus function must be called with a variable request number of zero (0) as in order to receive the Status packet from the Malibu ACU and therefore receive status data for any other variable request numbers. During real time operation all the other ACUStatus functions can be use within any IADS display (see Figure 3-3 for an example of using these parameters in a strip chart display).