

# Marinetti

3.0

## Debugging Guide

*“For the Apple IIGS<sup>®</sup>, the world just got a whole lot closer!”*

Written by Richard Bennett-Forrest  
© 1997-2008 Richard Bennett-Forrest



This revision: 9<sup>th</sup> June 2008

# Contents

Introduction	3
Internal debug tool calls	4
TCPIPSetMyIPAddress	4
TCPIPGetDP	5
TCPIPGetDebugHex	6
TCPIPSetDebugHex	7
TCPIPGetDebugTCP	8
TCPIPSetDebugTCP	9
TCPIPGetUserRecord	10
TCPIPRebuildModuleList	11
Debugging and Testing	12
Internal ONLY constants and equates	13

# Introduction

Marinetti is a TCP/IP protocol suite for the Apple IIGS. It allows applications on an Apple IIGS with System 6.0.1 to connect to any network using the internet protocol, and interact with it.

The Marinetti software is free of charge, and is available from various locations, including the Marinetti Home Page:

<http://www.apple2.org/marinetti/>

Updates to Marinetti and this document are products of the Marinetti Open Source Project.

<http://sourceforge.net/projects/marinetti/>

This document describes internal restricted tool calls accepted by Marinetti. They are restricted due to the nature of their function and the possibility that they could be used for functions outside the bounds of what Marinetti was designed for.

**NOTE: THESE TOOL CALLS MAY NOT NECESSARILY BE INCLUDED WITH THE RELEASE VERSION OF MARINETTI, AND SHOULD THEREFORE BE REMOVED FROM YOUR CODE BEFORE PUBLIC RELEASE.**

Revisions since the last release of this document are in blue, with revision bars on the right hand side of the page, like this.

This documentation refers to and assumes a prior knowledge of the Apple IIGS toolbox. Apple IIGS toolbox reference manuals are available from:

Syndicomm Online Store  
<http://store.syndicomm.com>

**This document is only applicable to Marinetti 3.x and 2.x.**

# Internal debug tool calls

The following tool calls are for setting up Marinetti internals.

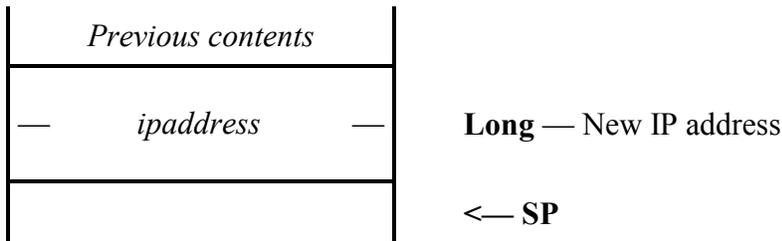
---

## TCPIPSetMyIPAddress \$3836

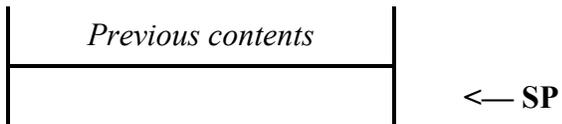
If connected, sets Marinetti's current IP address.

### Parameters

Stack before call



Stack after call



**Errors**                    `terrNOCONNECTION`      Not currently connected to the network

**BASIC**                    `SUB TCPIPSetMyIPAddress (&)`

**C**                            `extern pascal void TCPIPSetMyIPAddress (Long);`

**Pascal**                    `procedure TCPIPSetMyIPAddress (ipaddress: longint);`

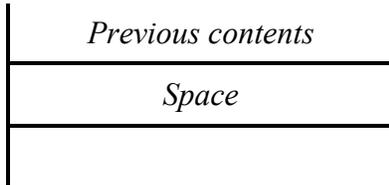
---

**TCPIPGetDP****\$3936**

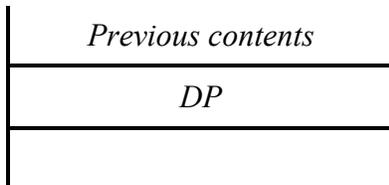
Returns Marinetti's working direct page address.

**Parameters**

Stack before call

**Word** — Space for result**← SP**

Stack after call

**Word** — Marinetti's direct page address**← SP****Errors**           None.**BASIC**           FUNCTION TCPIPGetDP as %**C**               extern pascal Integer TCPIPGetDP (void);**Pascal**         function TCPIPGetDP: integer;

---

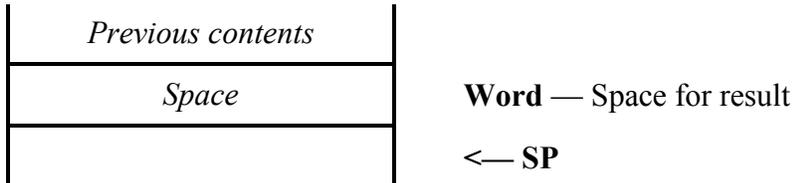
## TCPIPGetDebugHex

## \$3A36

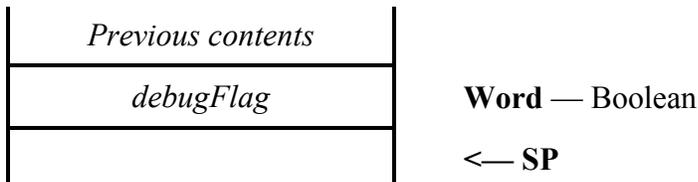
Returns the hex debugging flag, which indicates whether or not Marinetti is currently in hex debugging mode.

### Parameters

Stack before call



Stack after call



**Errors**                None.

**BASIC**                FUNCTION TCPIPGetDebugHex as %

**C**                        extern pascal Boolean TCPIPGetDebugHex (void);

**Pascal**                function TCPIPGetDebugHex: boolean;

debugFlag                The value returned is TRUE (non-zero) if Marinetti is in hex debugging mode, and FALSE (\$0000) if it is not..

◆NOTE: *Debugging is an optional part of the Marinetti, and requires the TCPIP.debug load module to be installed at boot time. The TCPIP.debug load module is a special build of the TCPIP Init that has functionality included to assist with debugging. It can be obtained upon request from the Marinetti Open Source Project team.*

---

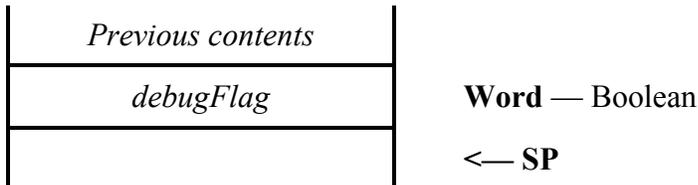
## TCPIPSetDebugHex

**\$3B36**

Sets the hex debugging flag.

### Parameters

Stack before call



Stack after call



**Errors**                None.

**BASIC**                SUB TCPIPSetDebugHex (&)

**C**                        extern pascal void TCPIPSetDebugHex (Boolean);

**Pascal**                procedure TCPIPSetDebugMode (debugFlag: boolean);

*debugFlag*                Pass in TRUE (non-zero) to turn on hex debugging mode, and FALSE (\$0000) to turn it off.

◆NOTE: *Debugging is an optional part of the Marinetti, and requires the TCPIP.debug load module to be installed at boot time. The TCPIP.debug load module is a special build of the TCPIP Init that has functionality included to assist with debugging. It can be obtained upon request from the Marinetti Open Source Project team*

---

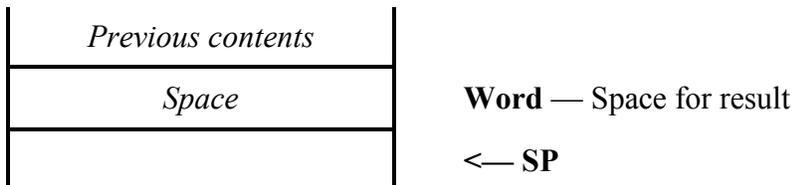
## TCPIPGetDebugTCP

**\$3C36**

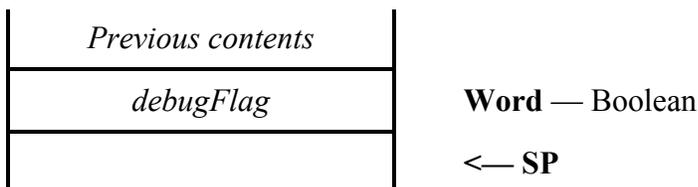
Returns the TCP debugging flag, which indicates whether or not Marinetti is currently in TCP debugging mode.

### Parameters

Stack before call



Stack after call



**Errors**                None.

**BASIC**                FUNCTION TCPIPGetDebugTCP as %

**C**                        extern pascal Boolean TCPIPGetDebugTCP (void);

**Pascal**                function TCPIPGetDebugTCP: boolean;

debugFlag                The value returned is TRUE (non-zero) if Marinetti is in TCP debugging mode, and FALSE (\$0000) if it is not.

◆NOTE: *Debugging is an optional part of the Marinetti, and requires the TCPIP.debug load module to be installed at boot time. The TCPIP.debug load module is a special build of the TCPIP Init that has functionality included to assist with debugging. It can be obtained upon request from the Marinetti Open Source Project team*

---

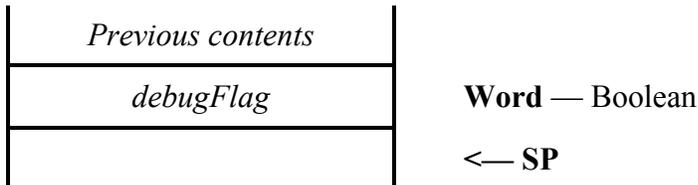
## TCPIPSetDebugTCP

**\$3D36**

Sets the hex debugging flag.

### Parameters

Stack before call



Stack after call



**Errors**                   None.

**BASIC**                   SUB TCPIPSetDebugTCP (&)

**C**                         extern pascal void TCPIPSetDebugTCP (Boolean);

**Pascal**                  procedure TCPIPSetDebugTCP (debugFlag: boolean);

*debugFlag*                Pass in TRUE (non-zero) to turn on TCP debugging mode, and FALSE (\$0000) to turn it off.

◆NOTE: *Debugging is an optional part of the Marinetti, and requires the TCPIP.debug load module to be installed at boot time. The TCPIP.debug load module is a special build of the TCPIP Init that has functionality included to assist with debugging. It can be obtained upon request from the Marinetti Open Source Project team*

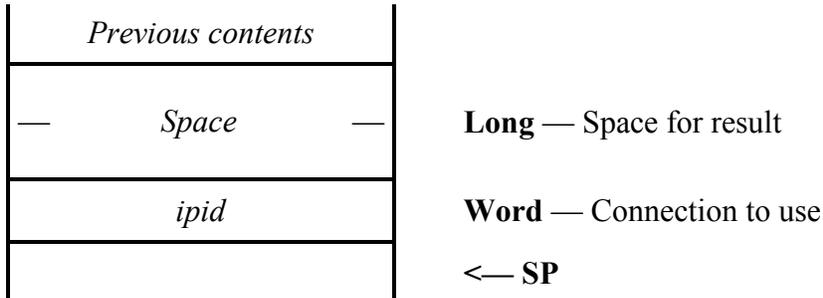
## TCPIPGetUserRecord

\$3E36

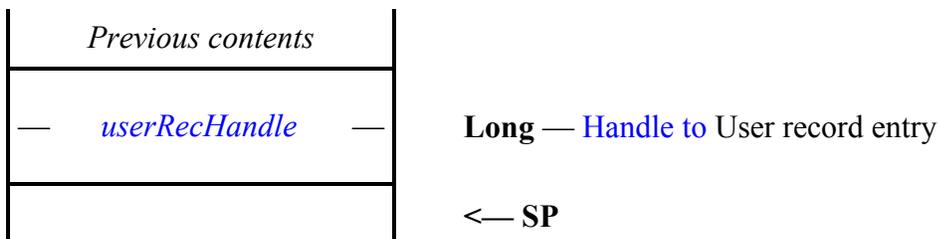
Returns the user record entry for the specified `ipid`. In the case of an allocated `ipid`, this will be Marinetti's internal handle to the user work area for that `ipid`, otherwise it will be `nil`.

### Parameters

Stack before call



Stack after call



**Errors** None.

**BASIC** FUNCTION TCPIPGetUserRecord (%) as `userRecHandle`

**C** extern pascal `userRecHandle` TCPIPGetUserRecord  
(integer);

**Pascal** function TCPIPGetUserRecord(ipid: integer):  
`userRecHandle`;

`UserRecHandle` If a handle is returned, it still belongs to Marinetti, and must not be modified. It is provided for reference only. The record layout is internal to Marinetti, and may vary between releases. The `TCPIPX.H/INT/.PAS/.S` files contain the record layout for the Marinetti version with which it was released.

---

**TCPIPRebuildModuleList****\$4D36**

Empties the internal list of available link layer modules and rebuilds it by scanning \*:System:TCPIP: for valid modules.

**Parameters**

The stack is not affected by this call.

**Errors**               None

**BASIC**               SUB TCPIPRebuildModuleList

**C**                   extern pascal void TCPIPRebuildModuleList (void);

**Pascal**             procedure TCPIPRebuildModuleList;

# Debugging and Testing

## ***Nifty List updates***

Marinetti contains additional tool calls and error codes for debugging, which should be applied to the `NList.Data` file, following the definitions for Marinetti 3.0

The new tool call definitions are as follows:

```
3836 TCPIPSetMyIPAddress(ipaddress/4)
3936 TCPIPGetDP():dp
3A36 TCPIPGetDebugHex():debugFlag
3B36 TCPIPDebugHex(debugFlag)
3C36 TCPIPGetDebugTCP():debugFlag
3D36 TCPIPDebugTCP(debugFlag)
3E36 TCPIPGetUserRecord(ipid): userRecHandle/4
4D36 TCPIPRebuildModuleList()
```

Once the changes have been made, save them back to disk and reboot. You should now be able to issue Nifty List commands against the Marinetti tool calls and error codes. If issuing calls outside of your application, you will most likely need to use Nifty List to issue the `_LoadOneTool(36, 300)` call first.

A file containing the above Nifty List configuration for Marinetti can be found in the Marinetti Open Source Project CVS Repository:

<http://marinetti.cvs.sourceforge.net/marinetti/MOSP/Tools/NiftyList/nl.marinetti>

# Internal **ONLY** constants and equates

There are none.