
Debugging Cocoa Applications

General



2002-11-12



Apple Inc.
© 2002 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Cocoa, Mac, Mac OS, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction to Debugging Cocoa Applications 5

Organization of This Document 5

Setting gdb Breakpoints for Cocoa Methods 7

Determining the Retain Count for an Object 9

Document Revision History 11

Introduction to Debugging Cocoa Applications

This topic discusses various ways to debug Cocoa applications.

Organization of This Document

[“Setting gdb Breakpoints for Cocoa Methods”](#) (page 7) tells you how to set a breakpoint for a Foundation or Application Kit method.

[“Determining the Retain Count for an Object”](#) (page 9) tells you how to determine the retain count for an object.

Another source for information on debugging and application performance are the Performance Programming Topics.

Setting gdb Breakpoints for Cocoa Methods

You might find it necessary to set a breakpoint for a Foundation or Application Kit method. This is done by specifying the method name.

In Xcode create a new breakpoint. Type the full method name into the breakpoint like:

```
- [NSException raise]
```

or:

```
+ [NSObject allocWithZone:]
```

If you are using gdb from a Terminal, you can also set a breakpoint using the command line. In this case, you can simply type the method name, such as `raise` or `allocWithZone:`. If more than one class implements the method, gdb will ask you to disambiguate.

Determining the Retain Count for an Object

You might find it necessary to determine what is the retain count for an object.

From the gdb console in Xcode enter:

```
print (int)[theObject retainCount]
```

If you are using gdb from a Terminal, you can also examine the retain count using the command line.

Document Revision History

This table describes the changes to *Debugging Cocoa Applications*.

Date	Notes
2002-11-12	Revision history was added to existing topic. It will be used to record changes to the content of the topic.

