

## EVERYTHING YOU WANTED TO KNOW ABOUT THE AR DATA BASE - BUT WERE AFRAID TO QUERY

This document on {Pogo:}<Release Management>Kat>Doc>AR-How-To-Edit.tedit.  
Last edited by Kat Kohlsaas on 9-Mar-1988 15:48:55

### INTRODUCTION

The Action Request data base is the primary vehicle through which the state of Xerox Lisp, including outstanding problems, requested features, and the like, is tracked. Since ARs are the primary channel of communication between the user, customer support, marketing, and development, it is important that the maximum amount of correct information be compressed into each AR. This allows technical information to get to development, and just as importantly, get back out. This process can be facilitated by correct use of the fields of the AR.

### THE AR FORM - THE FIELDS AND WHAT THEY MEAN

The basic component of the AR data base is the individual AR. An AR is the melding of a blank AR form with the data specifying a need. The AR form provides 31 areas, or fields, for the input of information giving a concise summary of the need. A need can be either a problem with the Xerox Lisp system that must be corrected, or a request for a feature that would improve the system if implemented. Since the structure of the AR form must be standardized to allow entry of a wide variety of needs, the data detailing the needs becomes an important component of the AR system. Correct use of the various fields comprising an AR facilitates the exchange of information between the submitter of the AR and the developer who will act upon the AR.

The FillInDefaults option of the left button menu associated with the AR Bug Report Editor title bar will fill in the **Submitter:**, **Source:**, **Status:**, **Machine:**, **Microcode Version:** and **Memory Size:** fields, and will place MAKESYSNAME as well as MAKESYSDATE in the **Lisp Version:** field. Please fill in the **Lisp Version:** field when submitting an AR, either by typing it in or by using FillInDefaults from the pop-up menu. The version of software the bug is being found in is important data.

- |                   |   |
|-------------------|---|
| <b>Number:</b>    | Generated by the AR data base, every AR has a unique number. AR numbers are <i>never</i> recycled. ARs are never deleted. The AR number cannot be changed by the user.  |
| <b>Date:</b>      | The date the AR was originally submitted. <u>This is filled in by the system.</u>   |
| <b>Submitter:</b> | The login name of the person who submitted this AR. <u>This is filled in by the system.</u>   |
| <b>Source:</b>    | The name of the person reporting the problem being documented in the AR. The name or names appearing in this field must give enough information to enable contact if needed, i. e., Doe.PASA or Doe@Berkeley.edu.   |
| <b>Subject:</b>   | A terse summary of the problem, providing both enough information to identify it uniquely and enough keywords for querying. "FOO doesn't work" or "Floppy problem" is not good enough. Think of yourself as a newspaper headline writer: "Attempt to write file when floppy door is open causes awful noise." Implementors may change the <b>Subject:</b> field as more details about the true nature of the problem become apparent. |

As much as possible, relevant keywords should be included in the **Subject:** to facilitate querying on the data base. If the problem relates to a specific package, that package name should be mentioned in the **Subject:**. File names, commands, functions, error messages, etc., are good examples of relevant keywords. For example, rather than "Floppy breaks when using mailfile", a better subject would be "Loading mail file from floppy causes break in FLOPPY.OPEN with error ILLEGAL ARG: 42."

**Assigned To:** The name of the person or persons who took some action based on the AR.

**Attn:** The name of the person or persons responsible for fixing the AR.

**Status:** This field shows the status of the AR. This changes as action is taken on the AR.

<i>New</i>	All ARs are generated with a default Status of <i>New</i> when submitted. <i>New</i> ARs have not been reviewed.
<i>Open</i>	This reviewed AR describes an outstanding problem with released software.
<i>Open/Unreleased</i>	This reviewed AR describes a problem with unreleased software.
<i>Fixed</i>	Problem has been fixed in an Internal loadup. Developers marking ARs as <i>Fixed</i> should mark the <b>In/By:</b> field according to the release into which the fix is being incorporated. At this time, the developer should also fill in the <b>Release Note:</b> field.
<i>Closed</i>	System with fix in it has been tested, documented, & released.
<i>Declined</i>	ARs can be declined for any of a variety of reasons. Perhaps it's a request for feature that is officially "never" going to be implemented (e.g., we think it's a bad idea). Perhaps the bug report is considered spurious (development doesn't think it is a bug). The reason for the AR being declined should be included in the <b>Description:</b> field. <i>Declined</i> ARs will be reviewed periodically so that old ARs may be re-opened.
<i>Superseded</i>	Another AR already includes the problem described in this one. The <b>In/By:</b> field of the superseded AR should include the AR number of the one that supersedes it (ex., 7064), and the beginning of the <b>Subject:</b> field should be edited to include a notation such as: "Superseded by AR #7064". The superseding AR should contain the information contained in the AR it supersedes, with a notation in the <b>Description:</b> such as: "[Supersedes AR #7911.]".
<i>Obsolete</i>	The problem reported is no longer a problem, e.g. the module containing the reported problem is no longer supported.
<i>Incomplete</i>	The information submitted is not enough to take action, i.e., there is not enough information to identify the bug, or the feature request doesn't give enough detail about what is wanted. This is different from <i>Declined</i> in that the request is considered valid, but the AR remains open awaiting more detail.

<i>Internal</i>	This status is used to report problems with internal software.
<i>Wish</i>	This status is usually used to request new features, change of features, Design-Impl or Design-UI .
<b>Problem Type:</b>	Defines the type of problem described in the AR. Possibilities for the <b>Problem Type:</b> field follow:
<i>Bug</i>	The system does not work as documented.
<i>Design-Impl</i>	The system works, but the internal implementation is wrong. (This type is generally submitted by other developers.)
<i>Feature</i>	Used to indicate a feature request.
<i>Design-UI</i>	The design of the user interface is wrong. This includes problems in the way in which things display, as well as program callable structures.
<i>Documentation</i>	The system works, but the documentation is wrong, unclear, or incomplete. The <b>System:</b> and <b>Subsystem:</b> fields should reflect the area in which there is a problem with the documentation. The <b>System:</b> should <u>not</u> be <i>Documentation</i> unless there is a specific problem with the documentation, apart from the system, e.g. "need better index".
<i>Performance</i>	The system works, but it is too slow doing the described operation.
<b>Difficulty:</b>	A rough estimate of the difficulty of the problem. <i>This field is to be filled in by developer only.</i> Categories within <b>Difficulty:</b> follow:
<i>Easy</i>	< 1 week to fix
<i>Moderate</i>	< 1 month to fix
<i>Hard</i>	< 6 months to fix
<i>Very Hard</i>	> 6 months to fix
<i>Impossible</i>	can't be fixed
<b>In/By:</b>	Used to specify the release for which an AR is/will be fixed or to indicate the number of a superseding AR.
<b>Impact:</b>	How seriously does it affect your ability to get work done, value of Xerox Lisp, etc. The items apply to bug reports, but feature requests should be rated along analogous lines. The categories within <b>Impact:</b> follow:
<i>Fatal</i>	Causes the system to crash, causes a loss of work, etc. Problem resolution is a requirement for project completion.
<i>Serious</i>	The problem can be worked around but it seriously interferes with work. This type of problem usually requires substantial reimplementation.

<i>Moderate</i>	The problem is tolerable, but clearly a problem, and the responsibility of Interlisp development.
<i>Annoying</i>	The problem is annoying, a minor request for a new feature that "would be nice".
<i>Minor</i>	May be some dispute about whether it is even a bug, or a very minor feature request.
<b>Frequency:</b>	How reproducible is the problem? If it is not known or is irrelevant to the AR, leave it blank. This is generally only relevant for bug reports. <b>Frequency:</b> can be one of:
<i>Everytime</i>	Reproducible every time.
<i>Intermittent</i>	Doesn't always happen.
<i>Once</i>	Saw it happen once.
<b>Priority:</b>	The perceived priority of this problem relative to the next release. A submitter may fill in their desired priority when submitting the AR. <i>However, priorities are approved/changed <b>only</b> by the Change Control Board.</i> Four different priorities are possible:
<i>Absolutely</i>	A showstopper. The pending release will be held if this AR is not completed. Requirements for this rating are: 1) Work lost with no workaround; 2) Highly embarrassing to Xerox; or 3) Marked <i>Hopefully</i> for previous release.
<i>Hopefully</i>	Preferable to be in the pending release, otherwise will be in next release.
<i>Perhaps</i>	Will get implemented if other revisions in same area are completed.
<i>Unlikely</i>	Unlikely to be included in the next release.
<b>System: Subsystem:</b>	The category and sub-category of the Xerox Lisp system that is pertinent to this AR. <b>System:</b> and <b>Subsystem:</b> categories are:
<i>Communications</i>	NS Protocols NS Filing NS Printing PUP Protocols PUP FTP Grapevine Leaf RS232 VAX Server DEI EVMS/RPC Lisp Servers Clearinghouse TCP/IP Centronics TTYPort Chat

	Chat Interface Pup Chat Driver NS Chat Driver RS232 Chat Driver TTYPort Chat Driver Chat DM2500 Emulator Chat VT100 Emulator NSMaintain Other
<i>Windows and Graphics</i>	Window System Library Fonts Printing Color Bitmaps Demos Menus Other
<i>Operating System</i>	Virtual Memory Generic File Operations DLion Disk Daybreak Disk DLion Floppy Daybreak Floppy Dolphin/Dorado Disk Processes Streams Keyboard Mouse Other
<i>Language Support</i>	Arithmetic Compiler, Code Format For/If Microcode Storage Formats/Mgt Garbage Collection Read and Print Stack and Interpreter Bootstrapping and Teleraid Diagnostics Other
<i>Programming Environment</i>	Break Package Code Editor DWIM Inspector File Package History Masterscope PSW Record Package Performance Tools Edit Interface Exec Presentations

	Stepper Other
<i>Text</i>	TEdit TTYIN Lafite AR Database Other
<i>Common Lisp</i>	Type System Declarations Macros Control Structure Evaluator Symbols/Packages Arithmetic Characters/Strings Sequences Lists Arrays Structures Hash Tables Streams and I/O File System Interface Error System Compiler Tamarin Support Microcoded Operations Common Loops Other
<i>CLOS</i>	Language Browsers Methods Classes Meta Classes Other
<i>Port</i>	Other
<i>Maiko</i>	Bytecode Emulation Native Code I/O System Host Integration Host User Interface Foreign Fn Interface Installation Procedure Documentation Other
<i>LOOPS</i>	Active Values Composite Objects Objects Browsers User Interface Virtual Copy Other

<i>PCE</i>	Monochrome Display Color Display Keyboard Emulated Rigid Disk Floppy Disk Printer Port User Interface Programmatic Interface File System Interface Memory Ethernet Configuration Tools Other
<i>PROLOG</i>	Arithmetic Dinfo Microcode Editor Interface Compiler Interpreter I/O Debugging Prolog-Lisp Interface Other
<i>4045</i>	XLPStream Remoteserver HQStream PSO Other
<i>Rooms</i>	Window Types Overview Suites Buttons Documentation Other
<i>Library</i>	Cash-File CharCode Tables Copyfiles DEdit DatabaseFns FX-80 Printer Support Filebrowser Font Samples GCHax GraphZoom Grapher Hash Hash-File Image Object Interface Kermit Masterscope Browser MatMult Press Printer Support SameDir Sketch

	SysEdit/EXPORTS.ALL Tablebrowser Virtual Keyboards Where-Is Other
<i>BusMaster</i>	Speech Color Other
<i>Documentation</i>	Tools 1108 Users Guide 1186 Users Guide Primer Product Descr/Tech Summary Hardware Installation Guide Programmers Introduction Interlisp Reference Manual Library Package Manual Internal System Documentation Other
<i>Other Software</i>	Installation Utility Release Procedure Other
<b>Machine: Disk:</b>	The value of these fields should be the type of Xerox hardware that is pertinent to this AR, i.e., the machine and disk on which the problem is happening. <b>Machine:</b> and <b>Disk:</b> categories are:
1108	SA1000 (10 MB) SA4000 (29 MB) Q2040 (43 MB) Q2080 (80 MB) T80 (80 MB) T300 (300 MB) Other
1132	T80 (80 MB) Century315 Other
1186	ST212 (10 MB) TM703 (20 MB) TM702 (20 MB) ST4026 (20 MB) Q530 (20 MB) Q540 (40 MB) Micropolis 1303 (40 MB) Micropolis 1325 (80 MB)
<b>Lisp Version:</b>	This field should identify the Xerox Lisp sysout in which the problem occurs (or the feature doesn't occur). The sysout should be identified by the name associated with the release (Koto, Lyric, Medley, etc.,) and/or MAKESYSDATE.



<b>Microcode Version:</b>	This information may be found by typing (MICROCODEVERSION) in an Interlisp Exec or (il:microcodeversion) in a Common Lisp Exec.
<b>File Server:</b>	What type of file server, if any, is involved with this problem. The menu contains the following items: <ul style="list-style-type: none"> <li>8037</li> <li>IFS</li> <li>NS</li> <li>VAX/VMS - 3 MB</li> <li>VAX/VMS - 10 MB</li> <li>VAX/UNIX</li> <li>Micro VAX/VMS</li> <li>Other</li> </ul>
<b>Source Files:</b>	The source files pertinent to the problem being reported in this AR.
<b>Memory Size:</b>	This value is the amount of "real memory", or RAM, in pages. This information may be found by typing (REALMEMORYSIZE) in an Interlisp Exec or (il:realmemorysize) in a Common Lisp Exec.
<b>Server Software Version:</b>	The version of software running on the server.
<b>Disposition:</b>	The record of who has changed which fields of this AR and when it was done. <u>This is filled in by the system.</u>
<b>Release Note:</b>	This field should contain the information to be included in the Release Notes for a given release. It should be release specific, such as: "Medley: In the debugger, the frame inspector window . . ." If a release note isn't required, that should also be explicitly mentioned, example: "Lyric LOOPS: None needed."
<b>Description:</b>	This field should contain a complete description of the problem or request, including any subsequent discussion after the AR submission. If the bug report came via electronic mail, the entire report should be added into this field. In cases where there are a number of electronic mail messages discussing this problem, all messages should be appended into this field.
<b>Workaround:</b>	This field should contain a known procedure to work around the problem until it is fixed. This would generally be a short recipe.
<b>Test Case:</b>	This field should contain a list of the files needed to recreate the problem. Please note that any Common Lisp or Interlisp recipes for reproducing the problem should be in the <b>Description:</b> field, not in the <b>Test Case:</b> field. When the problem is <i>Fixed</i> the <b>Test Case:</b> field should include any appropriate information that can be used to confirm the fix (or a note that a Test Case is not applicable, ex. "N/A").
<b>Edit-By:</b>	The login name of the last person to edit the AR. <u>This is filled in by the system.</u>
<b>Edit-Date:</b>	The date of the last change made to the AR. <u>This is filled in by the system.</u>

#### WHAT HAPPENS TO AN AR AFTER IT IS SUBMITTED?

Change Control Boards have been established for each XAIS product to bring AR priorities more in line with customer needs. The membership of each Change Control Board consists of the Product Development Project Leader, a member of Customer Support, and a member of Release Management. Incoming ARs are reviewed weekly by the appropriate board. At this meeting priorities are assigned for each AR, and other pertinent information, such as who will deal with the AR, is gathered. This information is input to the AR data base and summaries of ARs are generated for each responsible developer.

When a problem is resolved, the **Status:** field of the associated AR is changed to *Fixed*. At this point, the software is incorporated into the Development environment, which is the precursor for the next release. The *Fixed* AR is sent to the Documentation group for incorporation into the appropriate part(s) of the product documentation. When a release of software to customers occurs, all *Fixed* ARs that have been incorporated into that release, software and documentation, are marked *Closed*.