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 Kohlsaat on 9-Mar-1988 23:48:55

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.

Number: Generated by the AR data base, every AR has a unique number. AR numbers are *never* recycled. ARs are never deleted. The AR number cannot be changed by the user.

Date: The date the AR was originally submitted. This is filled in by the system.

Submitter: The login name of the person who submitted this AR. This is filled in by the system.

Source: 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`.

Subject: 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 **Subject:** field as more details about the true nature of the problem become apparent.

As much as possible, relevent 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.

New	All ARs are generated with a default Status of *New* when submitted. *New* ARs have not been reviewed.
Open	This reviewed AR describes an outstanding problem with released software.
Open/Unreleased	This reviewed AR describes a problem with unreleased software.
Fixed	Problem has been fixed in an Internal loadup. Developers marking ARs as *Fixed* should mark the **In/By:** field according to the release into which the fix is being incorporated. At this time, the developer should also fill in the **Release Note:** field.
Closed	System with fix in it has been tested, documented, & released.
Declined	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 **Description:** field. *Declined* ARs will be reviewed periodically so that old ARs may be re-opened.
Superseded	Another AR already includes the problem described in this one. The **In/By:** field of the superseded AR should include the AR number of the one that supersedes it (ex., 7064), and the begining of the **Subject:** 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 **Description:** such as: "[Supersedes AR #7911.]".
Obsolete	The problem reported is no longer a problem, e.g. the module containing the reported problem is no longer supported.
Incomplete	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 *Declined* in that the request is considered valid, but the AR remains open awaiting more detail.
Internal	This status is used to report problems with internal software.

Wish	This status is usually used to request new features, change of features, Design-Impl or Design-UI .

Problem Type: Defines the type of problem described in the AR. Possibilities for the **Problem Type:** field follow:

Bug	The system does not work as documented.
Design-Impl	The system works, but the internal implementation is wrong. (This type is generally submitted by other developers.)
Feature	Used to indicate a feature request.
Design-UI	The design of the user interface is wrong. This includes problems in the way in which things display, as well as program callable structures.
Documentation	The system works, but the documentation is wrong, unclear, or incomplete. The **System:** and **Subsystem:** fields should reflect the area in which there is a problem with the documentation. The **System:** should not be *Documentation* unless there is a specific problem with the documentation, apart from the system, e.g. "`need better index`".
Performance	The system works, but it is too slow doing the described operation.

Difficulty: A rough estimate of the difficulty of the problem. *This field is to be filled in by developer only.* Categories within **Difficulty:** follow:

Easy	< 1 week to fix
Moderate	< 1 month to fix
Hard	< 6 months to fix
Very Hard	> 6 months to fix
Impossible	can't be fixed

In/By: Used to specify the release for which an AR is/will be fixed or to indicate the number of a superseding AR.

Impact: 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 **Impact:** follow:

Fatal	Causes the system to crash, causes a loss of work, etc. Problem resolution is a requirement for project completion.
Serious	The problem can be worked around but it seriously interferes with work. This type of problem usually requires substantial reimplementation.
Moderate	The problem is tolerable, but clearly a problem, and the responsibility of Interlisp development.
Annoying	The problem is annoying, a minor request for a new feature that

"would be nice".

Minor — May be some dispute about whether it is even a bug, or a very minor feature request.

Frequency: 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. **Frequency:** can be one of:

Everytime — Reproducible every time.

Intermittent — Doesn't always happen.

Once — Saw it happen once.

Priority: The perceived priority of this problem relative to the next release. A submitter may fill in their desired priority when submitting the AR. *However*, *priorities are approved/changed **only** by the Change Control Board.* Four different priorities are possible:

Absolutely — 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 embarassing to Xerox; or 3) Marked *Hopefully* for previous release.

Hopefully — Preferable to be in the pending release, otherwise will be in next release.

Perhaps — Will get implemented if other revisions in same area are completed.

Unlikely — Unlikely to be included in the next release.

System: Subsystem: The category and sub-category of the Xerox Lisp system that is pertinent to this AR. **System:** and **Subsystem:** categories are:

Communications

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
Windows and Graphics	Window System
	Library
	Fonts
	Printing
	Color
	Bitmaps
	Demos
	Menus
	Other
Operating System	Virtual Memory
	Generic File Operations
	DLion Disk
	Daybreak Disk
	DLion Floppy
	Daybreak Floppy
	Dolphin/Dorado Disk
	Processes
	Streams
	Keyboard
	Mouse
	Other
Language Support	Arithmetic
	Compiler, Code Format
	For/If
	Microcode
	Storage Formats/Mgt
	Garbage Collection
	Read and Print
	Stack and Interpreter
	Bootstrapping and Teleraid
	Diagnostics
	Other
Programming Environment	Break Package
	Code Editor
	DWIM
	Inspector
	File Package
	History
	Masterscope
	PSW
	Record Package
	Performance Tools
	Edit Interface
	Exec
	Presentations
	Stepper
	Other
Text	TEdit
	TTYIN
	Lafite

	AR Database
	Other
Common Lisp	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
CLOS	Language
	Browsers
	Methods
	Classes
	Meta Classes
	Other
Port	Other
Maiko	Bytecode Emulation
	Native Code
	I/O System
	Host Integration
	Host User Interface
	Foreign Fn Interface
	Installation Procedure
	Documentation
	Other
LOOPS	Active Values
	Composite Objects
	Objects
	Browsers
	User Interface
	Virtual Copy
	Other
PCE	Monochrome Display
	Color Display
	Keyboard
	Emulated Rigid Disk
	Floppy Disk
	Printer Port
	User Interface

	Programmatic Interface
	File System Interface
	Memory
	Ethernet
	Configuration Tools
	Other
PROLOG	Arithmetic
	Dinfo
	Microcode
	Editor Interface
	Compiler
	Interpreter
	I/O
	Debugging
	Prolog-Lisp Interface
	Other
4045	XLPStream
	Remoteserver
	HQStream
	PSO
	Other
Rooms	Window Types
	Overview
	Suites
	Buttons
	Documentation
	Other
Library	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
BusMaster	Speech
	Color

Other

Documentation

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

Other Software

Installation Utility
Release Procedure
Other

Machine: Disk:

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. **Machine:** and **Disk:** 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)

Lisp Version:

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`.

Microcode Version:

This information may be found by typing `(MICROCODEVERSION)` in an Interlisp Exec or `(il:microcodeversion)` in a Common Lisp Exec.

File Server:

What type of file server, if any, is involved with this problem. The menu contains the following items:

8037
IFS
NS
VAX/VMS - 3 MB

VAX/VMS - 10 MB
VAX/UNIX
Micro VAX/VMS
Other

Source Files: The source files pertinent to the problem being reported in this AR.

Memory Size: 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.

Server Software Version: The version of software running on the server.

Disposition: The record of who has changed which fields of this AR and when it was done. This is filled in by the system.

Release Note: 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."

Description: 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.

Workaround: This field should contain a known procedure to work around the problem until it is fixed. This would generally be a short recipe.

Test Case: 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 **Description:** field, not in the **Test Case:** field. When the problem is *Fixed* the **Test Case:** 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").

Edit-By: The login name of the last person to edit the AR. This is filled in by the system.

Edit-Date: The date of the last change made to the AR. This is filled in by the system.

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*.