CLAN

December 97 NEWSLETTER

J233 StrongARM RISC PC

tomorrow's world, today!

CLAN checks out the power of Acorn's J233

How to install ARM Linux

Will your system work in the Year 2000 ?

'Plug-in' to Underground

Acorn works with Sun to provide Personal Java

Will NC take the 'game console' by storm?

Acorn events in 1998

Don't miss out on your Beta CD

clan@acorn.com

CONTENTS

Please note that all Clan enquiries should now be directed to Acorn on

email
clan@acorn.com

Phone
01223 725925

Mail
Clan Acorn
645 Newmarket Road
Cambridge
CB5 8PB

The clan magazine is a quarterly publication to clan members and developers in the Acorn Community

Contributors include Neil Carson, Dave Walker & Kerri Davies

Designed and produced by Creations
01438 747446

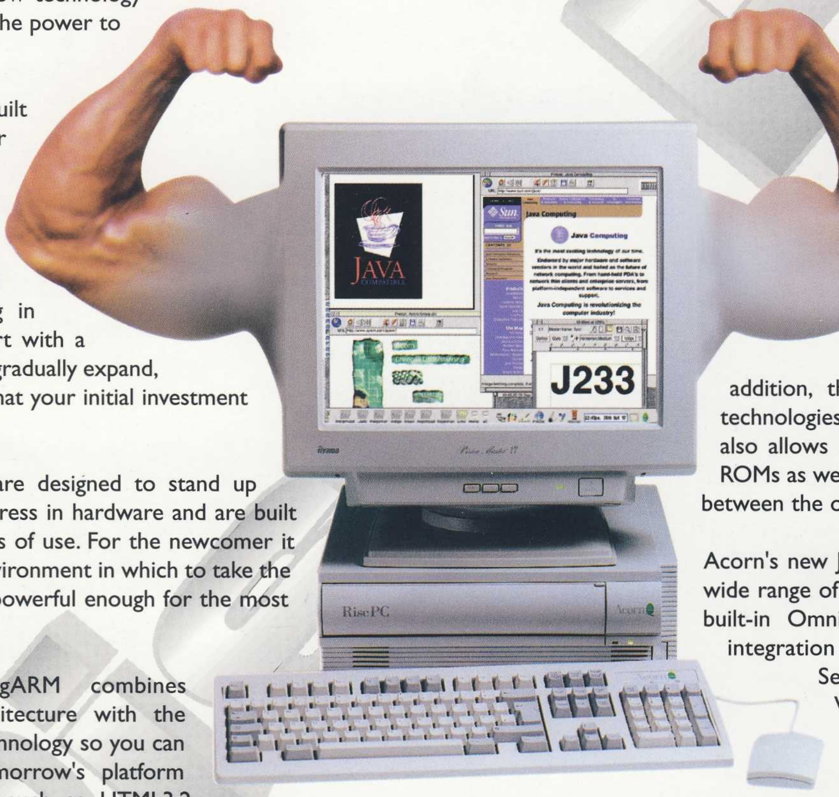
J233 StrongARM

StrongARM tactics from Acorn

Acorn has updated its high performance StrongARM Risc PC so you can make the most of today's networked world. And its modular design, powerful processor, expandable architecture and latest software technology means you will be able to make the most of tomorrow's world as well.

The new J233 version gives you power, flexibility and speed. With the latest StrongARM RISC processor at its heart, the Risc PC will give you levels of performance - now and in the future - which other manufacturers are unable to offer on their own standard priced machines. No matter how technology develops, you will have the power to keep pace.

The Risc PC has been built with a totally modular system. Its innovative mechanical and electrical design means you have virtually unlimited potential for upgrading in the future. You can start with a basic configuration and gradually expand, safe in the knowledge that your initial investment will not be wasted.



StrongARM Risc PCs are designed to stand up against continuous progress in hardware and are built to withstand the rigours of use. For the newcomer it offers a comfortable environment in which to take the first steps, while being powerful enough for the most seasoned professional.

Acorn's new StrongARM combines expandable RISC architecture with the very latest software technology so you can take advantage of tomorrow's platform independent standards such as HTML3.2 and Java. Java gives a new dimension to the use of the Risc PC. Applications and applets written in this emerging platform independent language can be run on any Java capable machine, no matter on which operating system they were written.

The J233 has a x24 speed CD-ROM together with 16-bit sound built in as standard, so there is no need for additional graphics or audiocards. The use of Acorn Replay multimedia architecture enables several audio and video codecs to be played through a simple-to-use interface. Large full colour images can be combined with text, graphics, video and sound.

With 32MByte DRAM and 2MByte VRAM as standard, the J233 has true multimedia capabilities allowing you to access the wide range of resources now available on the Internet. It enables you to make the most out of the massive growth in use of the net. There is easy to use connection to all the main UK Internet service providers, POP3 mail clients enabling you to get online, while the web browser is designed to the HTML3.2 standard and has support for frames and table extensions.

The RISC operating system - RISC OS - is a stable, scalable multi tasking system designed to run from ROM leaving a large amount of the hard disk available for your own data.

Despite giving you access to the vast range of RISC OS software and Java applets, this Risc PC will not lock you out of other platforms. When fitted with an appropriate expansion card, it will access applications on Windows and DOS platforms, up to 586 speeds. In addition, the J233 interconnects with other technologies such as UNIX. Its flexible design also allows sharing of memory, drives and CD-ROMs as well as the transfer of text and graphics between the operating systems.

Acorn's new J233 Risc PC integrates easily into a wide range of different network configurations. Its built-in OmniClient II software gives seamless integration with NFS, UNIX, Windows NT Server, Lan Manager, Windows 95, Windows for Workgroups and OS/2 Server.

Each Risc PC user needs to use their machines in a different way. With RISC OS there are several unique facilities to give you on the fly configuration of screen resolution and memory usage to adapt to specific applications and requirements. The large desktop gives easy access to any application or file. Opening, moving, modifying or copying files and directories is as simple as clicking and dragging. Icons can be pinned to the desktop for easy access and an icon bar lets you control any application that has been loaded.

Acorn's new StrongARM RISC PC is tomorrow's world, today!

NeWS

May we remind you that Acorn is no longer the agency tasked with allocation of filetypes, SWI chunks, OS numbers etc. Since 1995, allocation of these and other system resources and collation of the master allocation lists has been performed by Pineapple Software. Please send all allocation requests (accompanied by output files from !Allocate or plaintext details of the resource you wish allocated, as appropriate) to allocate@pinesoft.demon.co.uk and not to anyone @acorn.co.uk

EVENTS

Acorn Southwest Show Saturday 7th Feb 98

The webbington Hotel, Loxton, Somerset, 10am - 5pm
http://www.argonet.co.uk/users/acornshow
Admission charges: adults £1.50 Clan £1

Current list of exhibitors (as Dec 97)

Acorn Computers Ltd
APDL
Association of Acorn Usergroups
Risc User Group
CJE Micros
CTA Direct
expLAN Computers
iSV Products
Micro Laser Designs
RaspSoft
TBA Software
Topologika Software
Yellowstone

Acorn User
Archimedes World
Bajorasoft
Cambridgeshire Software House
Creative Curriculum
Dalriada Technology
H.S.Software
Jonathan Duddington
Dijas Software
Sherston Software
The ARM Club
Werewolf Software

Akalat Publishing
Argo Interactive
Bristol Acorn
Circle Software
Crow Associates
Electronic Font Foundry
Innovative Media Solutions
LOOKsystems
R Comp
Spacetech Ltd
The Datafile
Xemplar

The ARM Club Open Day

Merton Court School, 38 Knoll Road, Sidcup, DA14 4QU
Saturday 1st March 98
10.00am - 4.00pm

Exhibitors include:

APDL
Eesox
AAUG

Sherton
Spacetech
Datastore

Cumana
The Arm Club

Also featuring a Games Arcade

Theatre Timetable

10.30 - 11.15:

Arthur, Excalibur and Legend.... A Tale

The presentation, given by Brian Richardson of Cambridgeshire Software House, will take the form of a look through some of the interesting contents of their Award Winning CD Rom 'King Arthur'. The talk will also include some anecdotal information as to how CSH manage to put CD Roms of this type together so successfully. 'King Arthur' took two years to research and develop and contains information on over 400 'Arthurian' sites throughout Britain and France. Already voted 'Best Educational Software of the Year' this program reaches out beyond the classroom and many Arthurian enthusiasts talk about it as a major contribution to the Arthurian ideals. Somerset is heavily featured in the software so why not come and listen to one of the UK's leading experts in this type of software and discover for yourself how interesting it can be.

11.30 - 12.15:

Christian Resources

Apart from the use of the Acorn computer's more obvious appeal in DTP work for service-sheets and news-letters, it is ExpLAN's multi-lingual HolyBible software that has provided the springboard for Christian multimedia material to be published for this growing market.

This session will look at the use of HolyBible, both for delivery of specifically Biblical material and also the breadth of third-party resources that are being produced including maps, timelines, quizzes, dictionaries, music and children's stories. Indeed even the Church of England has now published its first Acorn software product in HolyBible's format.

The session will cover general aspects of multi-lingual and multimedia work and will therefore also appeal to those interested in porting material to/from the Acorn platform and licensing issues.

12.30 - 13.15:

News about Xemplar

A representative of Xemplar will be giving the latest news on both present and future developments at Xemplar.

13.30 - 14.15:

Introducing Sibelius

The presentation, given by Jim Nagel of Computer Shopper, will be an introduction to the award winning music processor for the Acorn computer.

14.30 - 15.15:

News about Acorn

Acorn's Chris Cox will talk about both present and future developments at Acorn.

15.30 - 16.15:

Why Internet your school?

The presentation, given by Ian Goodall of Argo Interactive is an introduction of the benefits, that using the Internet can bring.

Wakefield Spring Show

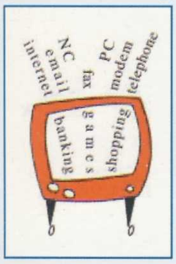
Thornes Park Athletic Stadium, Horbury Road, Wakefield, WF2 8TY
Saturday 16th May - Sunday 17th May 98
Advance Admission: Adults: £2.50
Juniors 6-16 £1.50
Under 6 FREE
Door Admission: Adults: £3
Juniors 6-16 £2
Under 6 FREE
Saturday: 10am - 5.30pm Sunday: 10am - 4.30pm
Advance ticket holders can enter the show 15 minutes in advance.
Advance tickets should be ordered from:-
WACG SHOW 98, 95 Cumbrian Way, Lupset Park, WAKEFIELD, West Yorks, WF2 8JT, ENGLAND
Cheques should be made payable to: WACG SHOW

Exhibitor List (as at 2nd Dec 97)

Wakefield Acorn Computer (User) Group
Acorn User Magazine
Alternative Publishing
Argo Interactive
Cd Computing
Cumana
Davyn Computer Services
Jon Duddington Speech Synthesis
ESP (Expressive Software Projects)
Icon Technology Ltd
Jaffa Software
Raspsoft Business Software
Simtec Electronics
Stuart Tyrrell Developments
Warm Silence Software Ltd
Akalat Publishing (Acorn Publisher magazine)
APDL
Beebug Ltd
Circle Software
Dalriada Data Technology
Desktop Projects Ltd
Eesox
4th Dimension Software
Irlam Instruments Ltd
Norwich Computer Services
Resource
Softease Ltd
TBA Software
Yellowstone
Acorn Computers
Apricote Studios
Cambridgeshire Software Hse.
Clares Micro Supplies
The Datafile
Electronic Font Foundry
Fabis Computing
iSV Products
R-Comp
Sherston Software Ltd
Spacetech Ltd
The ARM Club
Electronic Solutions

The video games console is under threat from Network Computers

Acorn has the vision to define the video games market of the future, and the technology to deliver a blow to the conformity of the video games business. It can achieve point-to-point and multiplayer gaming from a network delivered straight into consumers' homes.



Acorn has the technology to run 16-bit video games from the Internet on a Network Computer (NC). As a result of Acorn's advances in RISC-OS based NCs it is possible for a video games company to download its 16-bit library to NC users. The emulator (approximately 1Mb in size), licensed from Acorn's vast library of technological innovations, enables this downloading via HTML. The NC market in the USA is expected to have an installed base of over one million users by the end of 1998, opening up a potentially enormous market to software companies. The Internet Service Provider (ISP) will supply a hot link on a menu page, and the user ID registers the download andbills the consumer accordingly. As the NC has no internal memory storage, there is no risk of piracy for the software publisher, and for the ISP it provides the ideal revenue stream for a pay-per-play business model. Four emulators can also be produced to run Sega Genesis, Nintendo Game Boy and SNES, along with NEC Turbo Graffix 16 games.

Acorn can deliver technology which allows a home NC linked to a server to download a game to the player on demand. The server can then make point-to-point and multiplayer games routing automatically. Upgrades to the game will involve just one download onto the system, thus increasing the life of the game for the developer and the player. Once written, the game code is transferred directly onto the server, negating the need for a 'packaged product'. The revenue stream is billed on a simple pay-per-play basis, with a royalty paid to the developer.

As the server will accept slaves off the main route into the home, it will be possible to increase the number of players from one feed. This gives the opportunity for multiplayer games in one site and for 'same side' players in a team-based sports game. Moves or shots can be recorded for replay, with view-only access for spectators.

As this technology is flexible, it will allow portable point-to-point gaming via PDAs, and long-term strategy games to be played on the move with the game continued from the point it was left at home on the NC.

Through further advances in Acorn technology, it will be possible for realtime video conferencing to be combined with this game technology, thereby allowing the player to watch the opponents' reactions.

Silicon Graphics Inc. are to integrate Acorn's set-top box technology with their video server technologies, to provide customisable solutions for a wide range of applications including low cost video-on-demand for the small-to-medium enterprise.

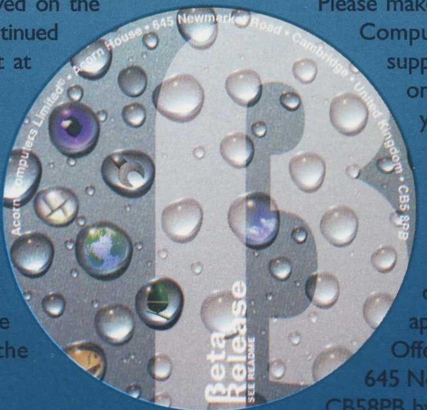
Acorn and Nippon Telephone & Telegraph have developed a prototype videophone, using Acorn's set-top box technology. The two companies are now in the detailed definition phase of the next stage of the project, which is part of a long term alliance between Acorn and one of the world's biggest companies.

TVCentric Java set to sweep the market

Acorn is working with Sun Microsystems to provide Personal Java, a development language for applications which can use either a domestic TV or a conventional monitor. Acorn's TVCentric technology will enable platform-independent Java applications to display high-quality output on standard televisions, without the need for tweaking by application developers. Acorn expects TVCentric Java to become the de facto standard for the convergence of TV and Java technologies. "We are delighted to work with a leader in the consumer Internet appliance area," said Jon Kannegaard, Vice President of Software Products at Sun's JavaSoft Division. "With the adoption of Java on RISC OS now, and the development of Java on Acorn's TVCentric technology, we are sure that we will have a major impact in markets for Internet-enabled products." All Acorn's operating systems, reference and production designs will have deeply embedded support for Java. Java applets will run on consumer devices as freely as they do now on desktop computers. Any Java applet will be able to use Acorn's anti-aliasing font technology, anti-twitter screen displays and advanced graphics rendering without any code changes.

Exclusive to Clan Members only

In response to your requests, the Beta CD is now available to Clan members who were unable to travel to the Acorn World 97 show to collect it. Due to high demand, the Beta CD has now been reprinted and will be despatched to members who complete and return the enclosed disclaimer. A small charge of £2.95 will be required to cover post and packing. Please make cheques payable to Acorn Computers Ltd or alternatively supply your credit card details on a plain piece of paper and your card will be debited upon despatch. Beta CD's will be supplied on a first come first served basis to Clan members who fully complete and return the disclaimer and the appropiate fee to:- Beta CD Offer, Acorn Computers Ltd, 645 Newmarket Road, Cambridge, CB58PB by January 31st 1998.


Acorn Computers Limited · Acorn House · 645 Newmarket Road · Cambridge · United Kingdom · CB5 8PB
Beta Release

HOW TO INSTALL ARM

Welcome to ARM Linux. Here is a quick guide to installing Linux which was missing from the August CD-ROM.

REQUIREMENTS

Machine Specification:
1 ADFS IDE/SCSI/MFM hard disk. 4MB memory minimum, 8MB recommended. RiscOS 3.1 or higher

You should have the following files:
config.arc, copying, install, bootloader.arc, partman.arc, writedisc.arc, root-rpc or root-a5k or root-arc, supplemental and at least 3 other floppy disks containing the base distribution, one blank ADFS formatted disk, two blank MSDOS disks. You should have a machine specific kernel as well. If you do not have any of these, I recommend that you obtain them from ftp.arm.uk.linux.org:/pub/armlinux/distrib before attempting installation.

NOTE: certain IDE/SCSI card manufacturers use a partitioning scheme with their drives. This may be incompatible with Linux, and its use will thus require great care. There should be little problem however if you are only using one partition.

INSTALLATION

1 Read the whole of this installation guide.

2 Set up the ADFS floppy disk.

2.1 Unarchive the following files onto an ADFS floppy disk:
config.arc partman.arc bootloader.arc writedisc.arc

They should produce the following applications respectively:
!LinConfig !PartMan !Linux !WriteDisc

2.2 If the kernel that you downloaded is archived, unarchive it onto this floppy disk, or else copy the kernel onto the same disk.

3 Set up the floppy disks:

3.1 If the root disk or supplemental disks are archived, Unarchive them onto your hard disk, otherwise copy them to your hard disk.

3.2 Run !WriteDisc, and drag the relevent root disk image into the source icon.

3.3 Insert a blank MSDOS disk into the floppy drive.

3.4 Select the destination drive, and click 'Write Disc', followed by 'Write'. (Please see Write Disc's !Help file for more information).

3.5 Label the disk `ARM Linux install disk'.

3.6 Perform steps 3.2 to 3.4 with `supplemental' instead of the root disk, copying onto a second floppy, labelled `ARM Linux supplemental disk'.

4 You should now have three disks setup for Linux. You could repeat 2 and 3 if you wish to make a backup of the installation floppies.

5 Perform this step if you wish to check that the kernel you have will run. (Recommended)

5.1 Ensure that you have saved all data under RiscOS - any unsaved work will be lost!

5.2 Insert the ADFS floppy disk, and press F12 in the desktop.

5.3 Type:
ADFS::0.$.!Linux -bootkernel ADFS::0.$.Kernel Where ADFS::0.$.Kernel is the kernel image you wish to use.

5.4 The kernel will then be loaded and should start up, with messages similar to:
Console: colour A-series...

Installed expansion cards:
Kernel panic: Unable to mount root filesystem OR
VFS: Insert root floppy and press any key.

5.5 If you see either of the last two messages, then the kernel you have is able to run on your machine. If you don't see something similar to the last message, then please check that you have the correct kernel, and if so, please mail a bug report to rmk@ecs.soton.ac.uk.

6 You can hit the CTRL-ALT-DELETE (left ALT!) to reboot the machine back to RiscOS. (Note: please see Known problems for RiscOS 3.1).

7 Backup the hard disk that you wish to use Linux with. If you already have a partitioning scheme (like ICS IDEFS), then backup all partitions on the drive. THIS IS A MUST - you will lose all your data in the next step!

8 Use HForm (or other ADFS initialisation tool) to shrink the size of the ADFS partition on your hard disk. You will have to decide how much space you wish to allocate to Linux, but I would advise a minimum of 250MB if you wish to install everything, and make light use of the system. This should leave you around 100MB free after installation.

To shrink the size of the ADFS partition, when HFORM asks you for the number of cylinders (don't use one of the provided disk shapes!), reduce them. You will have to calculate the size from the sectors, heads and sector size thus:

Cylinders of ADFS = Size of ADFS (bytes)
(sector size * sectors * heads)

9 Prepare the Linux Partitions.
Run !PartMan, and enter the filing system and drive number of the disc you just prepared.

9.1 If the total size, sector size, cylinders/heads/sectors are not correct, then choose 'Geometry' off the Partition Managers menu, and enter the correct values. This is *very* important!

9.2 You should see something similar to:

Linux

HTTP://WWW.ARM.UK.LINUX.ORG/~RMK/HOME.HTML

Partition	Start Sector	EndSector	Size	Type
1	0	590549	288	Filecore
2	590550	836069	121	Free

9.3 Double click on the free space. You should have a dialog box appear, asking to create a new non-filecore partition. The partition type should be 'Linux', and the start sector should be correct. Click on 'Create'.

9.4 An empty window for editing a partition appears. You should close this, and double click on the remaining free space. The editor window reopens with the correct start and end disc addresses.

9.5 You will need to create TWO partitions at least. The second one is a swap partition, and should be around 16 or 20 MB. It can be larger if you require it, but this is a guide only. In order to make room for the second partition, reduce this partition size by 16 to 20 MB, and press return, and select the OK button.

9.6 You should now see something similar to the following in the Partition Editors main window (The following is a guide only):

Partition	Start Sector	End Sector	Size	Type
1	0	590549	288MB	Filecore
2	590550	590551	1KB	Table
3	590552	795109	100MB	Linux Native
4	795110	836069	20MB	Free

9.7 Create the swap partition. Double click on the free space (the partition editing window reopens), select Linux Swap in the partition type using the up/down arrows and click OK. You should now see:

Partition	Start Sector	End Sector	Size	Type
1	0	590549	288MB	Filecore
2	590550	590551	1KB	Table
3	590552	795109	100MB	Linux Native
4	795110	836069	20MB	Linux Swap

9.8 The partitions are setup. All that remains is to save them to disk. Choose 'Save' on Partition Managers menu, and answer the dialog boxes.

9.9 If you wish to check that the information has been saved correctly, re-run Partition Manager and enter the correct filing system and drive. The list should re-appear as it was when you saved it.

10 Restore your previously backed up ADFS drive, and copy all files off the ADFS floppy disk created in step 2 onto the hard drive.

11 Boot your Linux kernel:

ADFS::4.$.Linux.!Linux -bootkernel ADFS::4.$.Linux.Kernel

Where ADFS::4.$.Linux is the path to your Linux directory, and ADFS::4.$.Linux.Kernel is the path to your Linux kernel. When prompted to insert the root floppy, insert the floppy that you wrote in step 3.4, and press a key.

12 You should be presented with the RedHat Linux installation screen. Follow the on-screen instructions.

13 Once the RedHat installer is complete, it will cause the machine to reboot. If everything installed ok, you should be able to install the Linux application properly.

13.1 Run !LinConfig, and click on the Loader icon. A window appears allowing you to set the timeout for the ALT key, and the kernel location. Enter a suitable pause duration into the ALT key timeout. The kernel location is a little more complex:

A kernel location consists of a string like `*ADFS::4.0'. The `*' must be present. It is then followed by the filesystem name `ADFS', case sensitive, a couple of colons `::', the drive number on that filesystem `4', and finally the partition number (0 = first Linux partition on the drive).

Eg. In the above partition tables on ADFS drive 4:
*ADFS::4.0 refers to partition 3, the 100MB linux native partition.
*ADFS::4.1 refers to partition 4, the 20MB linux swap partition.

13.2 The `Partitions' section is not currently supported by the later kernels. All fields should be left blank.

13.3 Once you are satisfied with the results, click 'Save'.

14 You should now be able to run the !Linux application without any extra arguments.

KNOWN PROBLEMS

1) Rebooting to RiscOS under RiscOS 3.1

There is a known problem here. It appears that RiscOS has some static data in memory somewhere that *must* be preserved over *all* resets, except of course power on. The data includes timing information for handling high bandwidth modes and the floppy disk. Since Linux takes over the whole machine, this data gets corrupted. Therefore I recommend that you reboot to the CLI, enter: *FX200 2 and hit the reset button again. This causes RiscOS to effectively perform a power on reset. Sorry, but there appears to be no other easy way round this. It is however, ok to reboot straight back into Linux, so long as you don't default to one of the high bandwidth modes.

Known modes include : 28 (640x480x256) 32 (800x600x256)
Modes that appear ok: 12 (x16) 27 (640x480x16) 31 (800x600x16)

Latest information can be found on http://www.arm.uk.linux.org/~rmk/home.html

NOTE:
No responsibility is taken whatsoever for any damage, loss of data, data being inaccurate, inconvenience caused by use or misuse of these programs. See the GNU GPL paragraph 11 and 12 for exact details.

ALL CLEAR ON THE YEA

WILL THE YEAR 2000 BR

One of the major concerns raised throughout the computing world in the last two years has been concerned with the advent of the year 2000. The reason for this is that a large number of critical systems, especially mainframes, only store the year in a two-digit form - binary coded decimal date formats, as used by COBOL and other similar business-oriented languages, do not cater for a rollover of century. A treatment of dates which results in January 1st 2000 appearing coincident with January 1st 1900 is sure to make for some very interesting computer-compiled stock control and financial reports by the time year 2000 arrives, unless a fix is put in place in time.

Predictions thus far have estimated that the cost of OS, compiler and application modifications to fix the problem on all systems which exhibit it will run to billions of dollars, and a large number of standards organisations have responded by producing definitions of what they consider to be the requirements for a machine to be "Year 2000 conformant".

Acorn has chosen to produce a statement of year 2000 conformity based upon the definition DISC PD2000-1 produced by the British Standards Institute; the BSI definition is reproduced below by kind permission of its authors, the BSI DISC office. At the time of writing our conformity statement has yet to be ratified by our legal department and is thus not legally binding, however it can be considered technically accurate and can thus be usefully disseminated to developers and Clan members. If all goes according to schedule, a legally ratified and binding document will be available by the time of Acorn World 97.

DISC PD2000-1
A Definition of Year 2000 Conformity Requirement

THE DEFINITION

Year 2000 conformity shall mean that neither performance nor functionality is affected by dates prior to, during and after the year 2000. In particular:

Rule 1. No value for current date will cause any interruption in operation.

Rule 2. Date-based functionality must behave consistently for dates prior to, during and after year 2000.

Rule 3. In all interfaces and data storage, the century in any date must be specified either explicitly or by unambiguous algorithms or inferencing rules.

Rule 4. Year 2000 must be recognised as a leap year.

AMPLIFICATION OF THE DEFINITION AND RULES

General Explanation:

Problems can arise from some means of representing dates in computer equipment and products and from date-logic embedded in purchased goods or services, as the year 2000 approaches and during and after that year. As a result, equipment or products, including embedded control logic, may fail completely, malfunction or cause data to be corrupted.

To avoid such problems, organisations must check, and modify if necessary, internally produced equipment and products and similarly check externally supplied equipment and products with their suppliers. The purpose of this document is to allow such checks to be made on a basis of common understanding. Where checks are made with external suppliers, care should be taken to distinguish between claims of conformity and the ability to demonstrate conformity.

Rule 1

1.1 This rule is sometimes known as general integrity. 1.2 If this requirement is satisfied, roll-over between all significant time demarcations (e.g. days, months, years, centuries) will be performed correctly.

1.3 Current date means today's date as known to the equipment or product.

Rule 2

2.1 This rule is sometimes known as date integrity. 2.2 This rule means that all equipment and products must calculate, manipulate and represent dates correctly for the purposes for which they were intended.

2.3 The meaning of functionality includes both processes and the results of those processes.

2.4 If desired, a reference point for date values and calculations may be added by organisations; e.g. as defined by the Gregorian calendar.

2.5 No equipment or product shall use particular date values for special meanings; e.g. "99" to signify "no end value" or "end of file" or "00" to mean "not applicable" or "beginning of file".

Rule 3

3.1 This rule is sometimes known as explicit/implicit century.

3.2 It covers two general approaches:

(a) explicit representation of the year in dates: e.g. by using four digits or by including a century indicator. In this case, a reference may be inserted (e.g. 4-digit years as allowed by ISO standard 8601:1988) and it may be necessary to allow for exceptions where domain-specific standards [e.g. standards relating to Electronic Data Interchange, Automatic Teller Machines or Bankers Automated Clearing Services] should have precedence.

(b) the use of inferencing rules: e.g. two digit years with a greater value than 50 imply 19xx, those with a value equal to or less than 50 imply 20xx. Rules for century inferencing as a whole must apply to all contexts in which the date is used, although different inferencing rules may apply to different data sets.

General Notes

For Rules 1 and 2 in particular, organisations may wish to specify allowable ranges for values of current date and dates to be manipulated. The ranges may relate to one or more of the feasible life-span of equipment or products or the span of dates required to be represented by the organisation's business processes. Tests for specifically critical dates may also be added (e.g. for leap years, end of year, etc). Organisations may wish to append additional material in support of local requirements.

Where the term century is used, clear distinction should be made between the "value" denoting the century (e.g. 20th) and its representation in dates (e.g. 19xx); similarly, 21st and 20xx.

This definition of Year 2000 conformity has been produced with the permission of the authors, the British Standards Institution. It was produced by the BSI committee BDD/1/-/3 in response to demand from UK industry, commerce and the public sector.

R 2000 FRONT!

ING THE WORLD TO A HALT?

ACORN STATEMENT OF YEAR 2000 CONFORMITY

Introduction

This document describes the degree of conformity of the Acorn components listed in Schedule I to the BSI document DISC PD2000-1, "A Definition of Year 2000 Conformity Requirements". It is intended to be read in conjunction with said BSI document.

Rule 1 Compliance

The components listed in Schedule I are conformant to the General Integrity rules 1.1, 1.2 and 1.3 provided that:

* The "current date" is defined as today's date in the Gregorian calendar and no other calendar.

* "Today's date" falls in the range detailed in the Rule 2 Compliance note below.

Rule 2 Compliance

The components listed in Schedule I are conformant to the Date Integrity rules 2.1 through 2.5 inclusive subject to the following limitations:

The baseline reference point for date compliance is the date of adoption of the Gregorian calendar or Gregorian January 1st 1900 AD, whichever is the later. In the case of localisation to other territories which adopted the Gregorian calendar on a date other than the date of the UK adoption of the Gregorian calendar, the baseline reference point for date correctness should be taken as the Gregorian date in that locale on which the Gregorian calendar was last adopted (this takes into account those territories which adopted the Gregorian calendar, renounced it, and then later re-adopted it).

For the systems described in Schedule I only the UK territory is officially supported, although unofficial alternative territory definitions exist.

In reference to the General Notes section of the BSI document, we declare a date ceiling relating to the projected feasible lifespan of the equipment. Dates will cease to be represented correctly in the Gregorian year 2248 AD, hence the chosen declared ceiling for correct date representation is Gregorian December 31st, 2247 AD.

In Rule 2.2, calculation, manipulation and representation of dates is only guaranteed conformant for Gregorian dates in the range above (subject to the Additional Notes below).

In Rule 2.3, only the ROM-based OS software and system hardware detailed in Schedule I is guaranteed to represent process results in a compliant manner (subject to the Additional Notes below).

Rule 3 Compliance

The components listed in Schedule I are conformant to the Explicit / Implicit Century rules 3.1 and 3.2 subject to the Additional Notes below. All components listed in Schedule I store the year in a 4-digit format. Similarly, files written to filing systems which intrinsically support a four year-digit file datestamping system (such as Acorn ADFS) and for which suitable interfacing software (such as RISC OS FileCore) exists will have their datestamp recorded with a four-digit year component.

It is possible to specify years in two-digit form when passing arguments to the system calls SWI "Territory_ConvertTimeStringToOrdinals" (with reason codes 2 and 3) and SWI "OS_Word",15 (with reason codes 15 and 24, which specify that the operation to be performed is a write to CMOS); in the case of these specific calls, dates supplied as arguments and which contain a two-digit year field are treated as follows:

0<year<65: 2000 is added (ie 01-Jan-44 is treated as 01-Jan-2044) 67<year<99: 1900 is added (ie 01-Jan-94 is treated as 01-Jan-1994)

The changeover point below which 2000 is added and above which 1900 is added is 26-Jan-66.00:00:00.

Although this two-to-four year digit conversion metric works consistently as detailed above, Acorn recommends that years are described in four-digit form wherever possible.

Rule 4 Compliance

The components listed in Schedule I are conformant to the 2000-is-a-leap-year rule. To comply with the range set in Rule 2, they also recognise 2100 AD and 2200 AD as not being leap years.

Additional Notes

In the case of Rule 2.2, no recommendations are made for the behaviour of the system clock and the recovery of the real time following power failure in both the main power supply and the battery which powers the realtime clock / CMOS chip during periods when the main power is normally switched off. Should both main and backup power to the real time clock so fail, or CMOS bytes 128 and 129 become corrupted, the clock will be in an indeterminate state when the main power is reinstated; it is very unlikely to retain the correct real time. In this case, the time will need to be reset manually by the user, using commands provided in the ROM images detailed in Schedule I.

SCHEDULE I

RISC OS 3.10 ROM	Image and ARM-based hardware platforms designed by Acorn to run, and running, same
RISC OS 3.11 ROM	
RISC OS 3.50 ROM	
RISC OS 3.60 ROM	
RISC OS 3.70 ROM	
RISC OS 3.71 ROM	

Further information on the year 2000 statement can be found at http://www.acorn/news/year2000/

Browser plugin protocol

Insert Tab A in Slot B...

The World Wide Web is rapidly being extended to offer better support for embedding multimedia data inside Web pages. A well-established mechanism known as "helpers" allows a browser to delegate the display of unsupported data types to other applications. However, the helper application displays this data independently of the browser, usually in its own window.

The idea of a "plug-in" is to integrate the display of such data into the WWW browser's own window. A number of proposed HTML extensions are being promoted, such as <APPLET> (by Sun for Java), <EMBED> (by Netscape) and <OBJECT> (by W3C).

This protocol is described from the point of view of a web browser and plug-in. However it is applicable to any application that needs to embed another application's windows within its own.

Browsers such as Netscape support plug-ins in the form of dynamically loaded code resources (DLLs). On finding data of a type it cannot display itself, the browser seeks a DLL which is capable of handling it. If it finds one, it calls standard entry points in the DLL to get it to display the data in the browser's window.

This model does not fit well with RISC OS systems. RISC OS does not have a standard scheme for DLLs, and the alternative - using relocatable modules - is not practical for very large playback engines (as used in systems such as Java and Director). Therefore plug-ins are implemented as separate tasks, with a special message protocol between the browser and the plug-in to permit communication and control. This message protocol is described in detail below; to cover the operations required of it, it has quite a few message types!

In order to display the data inside the browser's own window, the plug-in needs to be made responsible for updating a certain portion of the browser's work area. This could be done by the browser instructing the plug-in to redraw parts of the window. However, this approach was rejected because it introduces significant differences between a plug-in and a normal application. Instead, we utilise a new facility added to the Nested Window Manager (as covered last issue), whereby windows can be created "inside" a parent window. The Window Manager takes care of event distribution to the plug-in, and also ensures that the "child" window is in a fixed position relative to the work-area of the parent - so the plug-in's display area will be scrolled within the browser window if the user manipulates the browser window's scrollbars.

PROGRAMMER INTERFACE

A plug-in accepts one or more types of data, specified using normal RISC OS filetypes. It is the responsibility of the browser to map MIME types to RISC OS filetypes.

INVOCATION

Having determined the best NC OS filetype for the data, the browser performs the following sequence of actions:-

Broadcast MESSAGE_PLUGIN_OPEN, passing the filename and filetype of the data, and the parent window information. This message also contains an opaque 32 bit value known as the "browser instance handle". This is a word of significance to the browser, and might be different for each instance of a plugin. The plugin must always quote the correct browser instance handle to the browser in subsequent messages.

If a reply MESSAGE_PLUGIN_OPENING is received, an existing invocation of a suitable plugin has agreed to handle the data. The MESSAGE_PLUGIN_OPENING contains an opaque word value, known as the "plugin instance handle", which together with the task handle of the plugin task uniquely identifies the piece of data. The browser remembers both of these values for use in future messages.

If no task responds to MESSAGE_PLUGIN_OPENING, the browser attempts to launch the appropriate plug-in. This is done by looking for an environment variable called "Alias$@PlugInType_xxx" where xxx is the hexadecimal type value. If this variable is not found, no suitable plug-in is available, and the browser regards the attempt to display the data as unsuccessful. If the variable is found, then the browser launches it by calling Wimp_StartTask.

The result of Wimp_StartTask is the task handle of the new invocation of the plug-in. As soon as Wimp_StartTask returns, the browser re-broadcasts the MESSAGE_PLUGIN_OPEN message.

Normally, the plugin accepts this message and replies with MESSAGE_PLUGIN_OPENING, containing a plugin instance handle as described above.

If no reply was forthcoming, the browser assumes that for some reason the plugin was unable to load the data, and it regards the attempt to display this data as unsuccessful. This might be because the data is malformed, erroneous or of an incompatible version to that expected by the plugin, or it might be because of some unexpected eventuality (out of memory, etc). If a detailed failure message is to be issued to the user, it is the responsibility of the plug-in to do this.

If further data of the same type needs to be displayed, either simultaneously or sequentially, then the browser should repeat the whole process starting with the broadcast.

If the plugin replies and so requests then the browser opens a data stream for the initial object being embedded and sends this data to the plugin according to the plugin stream protocol.

GROUND

The plugin examines the contents of the file that was named in the MESSAGE_PLUGIN_OPEN message. This file contains all of the information from the <OBJECT>, <EMBED> or <APPLET> tag, and is used by the plugin to initialise itself. The plugin may have to fetch the contents of more URLs in order to do this; it may get the browser to do this on its behalf by using the URL_ACCESS message.

If during startup the plugin encounters an unrecoverable error it tidies up after itself and send a MESSAGE_PLUGIN_CLOSED message to the browser, setting a flag in the message to indicate that this is due to an error. The message may optionally include an error message for the browser to display.

SHUTDOWN

When the browser wishes the data to be forgotten, for example when the user quits the browser or leaves the current page, the following actions are taken. If multiple pieces of data have been farmed out (to the same or multiple plug-ins) the sequence below is performed for each such piece of data.

Browser sends MESSAGE_PLUGIN_CLOSE directly to the plug-in task, passing the plugin instance handle associated with the data.
Plug-in closes and deletes its window, cleans up state and data, etc.
Plug-in replies with
MESSAGE_PLUGIN_CLOSED.

Plug-in decrements its count of active objects. If the count is zero, it is free to exit if it wishes. A flag in the MESSAGE_PLUGIN_CLOSE acts as a hint to the plug-in as to whether the browser would like the plug-in to remain running or not, but the plug-in does not have to honour this if it does not want to.

PLUG-IN DEATH

If the browser receives a MESSAGE_TASK_CLOSEDOWN, it checks to see whether the exiting task was a plug-in that was currently displaying data on behalf of the browser. If so, all data being displayed by that plugin is marked as undisplayable. The Window Manager has already deleted the child window(s) associated with the task. The NCBrowser will not issue any error in this case, however other possibilities are relaunching the plug-in or reporting the exit to the user.

BROWSER DEATH

If the plugin receives a MESSAGE_TASK_CLOSEDOWN, it checks to see whether it is displaying data on behalf of the exiting task. If so, it deallocates any state or data associated with that task, and reduces its reference count by the correct amount. If the reference count reaches zero (i.e. the dead task was the only task using the plugin), then the plugin may exit if it wants to.

WINDOW EVENTS

The Window Manager's nested window mechanism addresses all subwindow positioning issues automatically. If a browser window is closed then the subwindow is removed from view, and is reinstated when the parent window is reopened. If the browser window is scrolled, the Window Manager ensures that the plug-in window stays at the same position relative to the browser's work area. If necessary, it repositions the subwindow and clips it if it has scrolled partially or entirely out of view.

Repositioning is done by the Window Manager without sending Open_Window_Request events to the plug-in.
If the plugin receives a keypress or mouse button click that it does not want to handle, it must pass it on to the browser by means of Wimp_SendMessage. It must set the window handle field of the message to the handle of its parent window. Note that this mechanism should be used instead of Wimp_ProcessKey.

If the browser wishes to forcibly resize or reposition the subwindow, it sends a MESSAGE_PLUGIN_RESHAPE message to the plug-in, quoting the plug-in instance handle. The plug-in must honour this request by re-opening itself at the new position. The coordinates in this request are work-area coordinates of the parent window. The parent window handle in this message may be different to the original one. The plugin should be prepared to check for this, and re-create its window as a child of the new parent if necessary.

If the plugin wishes to alter its size, it cannot simply resize its window. Instead it must send a MESSAGE_PLUGIN_RESHAPE_REQUEST to the browser. The browser responds by reformatting the page (if necessary) and then replying with a suitable MESSAGE_PLUGIN_RESHAPE. The plugin must act on this in the normal way.

DATA POINTERS

Many of the strings passed around in this protocol are of unspecified size and may, especially in the case of URLs, be larger than could fit within the body of a wimp message. Therefore they are defined as string_values. These are defined as being either offsets from the start of the message body (if less than 256) or as pointers to data held in shared memory (ie the RMA or a dynamic area). It is always the responsibility of the sender to free the memory used for any such pointers.

The protocol is defined in such a way that there should always be a reply received or the message will be bounced by the Window Manager. In either case it is then safe for the sender to free the memory allocated.

However to avoid memory leaks it is recommended that careful track is kept of such pointers so that they can be freed when a plugin instance is closed.

All strings must be null terminated, but need not start at a word-aligned address.

STREAM PROTOCOL

Some plugins may wish the browser to fetch data from the net for them rather than having to implement their own fetching code. A flexible interface is provided for this.

There are several ways a stream can be instigated, as follows:

The browser wishes to transfer the initial data which launched the plugin The plugin requests some data be fetched for it with URL_ACCESS The plugin requests some data be posted for it with URL_ACCESS The plugin wishes to write directly to a browser window

INITIAL TRANSFER

Browser fills in flags, mime type, stream data and sends STREAM_NEW the plugin returns the same message

* quoting the reference
* filling in the plugin stream instance handle
* updating the stream mode (if necessary).

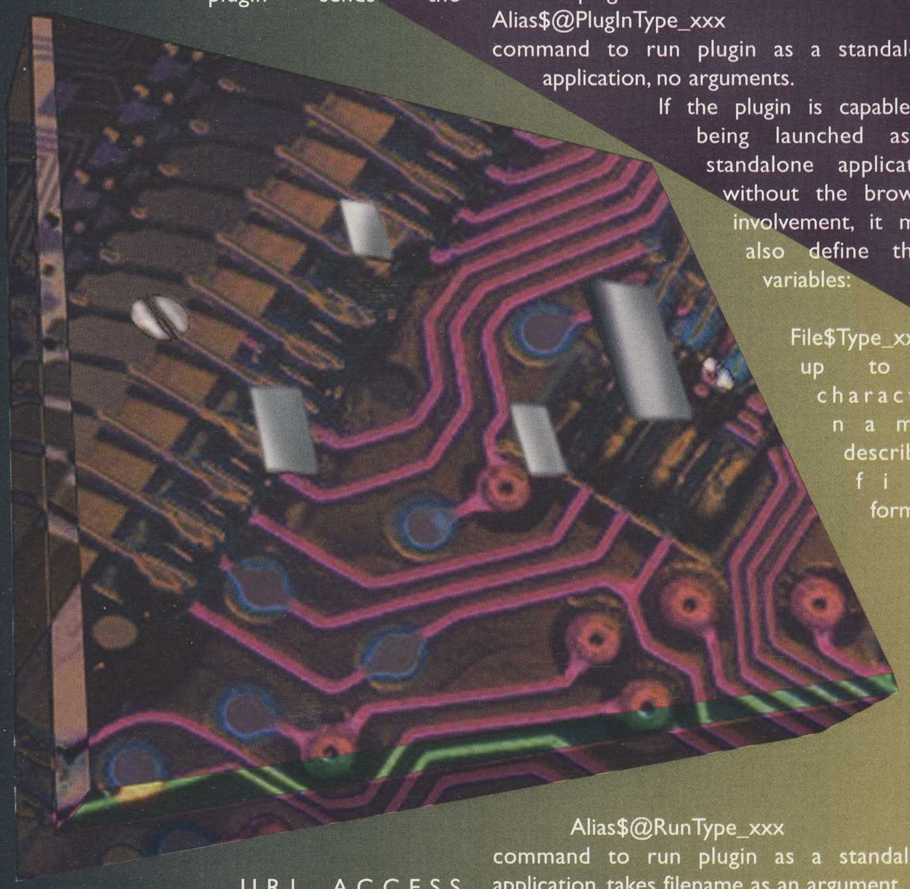
If mode is applicable
browser sends STREAM_WRITE
plugin replies with STREAM_WRITTEN giving the number of bytes that is could process.

UNDER

This is repeated until all data is transferred or an error occurs. Browser sends STREAM_DESTROY with appropriate reason code

PLUGIN REQUESTS DATA BE FETCHED OR POSTED

plugin sends the URL_ACCESS message when data starts arriving we continue as initial transfer.



PLUGIN WRITE TO BROWSER

Plugin fills in mime type, target, plugin stream instance and sends STREAM_NEW
the browser returns the same message

* quoting the reference
* filling in the stream fields the plugin writes data
* plugin sends STREAM_WRITE
* browser replies with STREAM_WRITTEN giving the numberof bytes that it could process.

This is repeated until all data is transferred or an error occurs. Plugin sends STREAM_DESTROY with appropriate reason code.

SYSTEM VARIABLES

For a plugin yyyy whose file type is xxx, the variables which the plugin must set are:
yyyy$Dir
the application directory containing !Boot, !Run etc. files
PlugIn$Type_xxx
name of plugin for browser menu
Alias$@PlugInType_xxx
command to run plugin as a standalone application, no arguments.
If the plugin is capable of being launched as a standalone application without the browser involvement, it must also define these variables:

File$Type_xxx
up to 8 character name describing file format

Alias$@RunType_xxx
command to run plugin as a standalone application, takes filename as an argument.
If the plugin can also be used as a helper application, then this variable must also be set:
Alias$@HelperType_xxx
command to run plugin as a helper application.

For example, a sample !Boot file for a plugin might contain the following:

```
Set Java$Dir <Obey$Dir>
Set File$Type_AE4 Java
Set PlugIn$Type_AE4 Java
SetMacro Alias$@RunType_AE4 /<Java$Dir>.!RunImage -standalone %%*0
SetMacro Alias$@PlugInType_AE4 /<Java$Dir>.!RunImage -plugin %%*0
```

If a file is embedded with APPLET, EMBED or OBJECT then the Alias$@PlugInType_xxx variable is used to start the application.
If a file is pointed to with an anchor (eg <A HREF="applets/myapplet.class">), then the file is downloaded and the Alias$@Runtype_xxx variable is used.

THE OBJECT TAG

Note that plugins can be launched from an OBJECT tag as well as EMBED or APPLET. When this happens, there are some minor differences to the values in the parameter file.

FOR EMBED

The SRC attribute is named DATA

For APPLET

The ALT attribute is named STANDBY.
The CODE attribute is named CLASSID.
The value of CLASSID may not have the .class suffix The value of CLASSID may have a prefix java:

HELPER APPLICATIONS

This same interface is also used for helper applications. Helper applications are very like plugins, except that they open their own windows external to the parent rather than embedded in the parent's window. This means that they are not constrained to close down when the parent window is closed (eg when the browser follows a link to another page), but can still benefit from the communication protocols with the parent. There is a flag in the OPENING message to inform the parent whether a window was embedded or not.

When trying to launch a helper application, the process described in the "Invocation" section is used except that if the initial MESSAGE_PLUGIN_OPENING message is not claimed, the system variable Alias$@HelperType_xxx is used to start the helper task.

HELP PROTOCOL

A plug-in may support the Wimp Help protocol. If it does, then help messages are displayed in the browser status bar (if configured). Messages must be limited to at most 40 characters.

DATA INTERCHANGE

The following new Wimp messages are defined:-
MESSAGE_PLUGIN_OPEN (&4D540)

GROUND

Sent by the browser to create a plugin instance.

R1+16 Message_PlugIn_Open
Common fields
R1+20 Flags
Bit 0: 1 = Open the file as a helper
0 = Open the file as a plugin
Bits 1-31: reserved (must be zero)
R1+24 reserved
R1+28 Browser instance handle (provided by the browser) R1+32 Parent window handle
R1+36 Left (bounding box in parent's
R1+40 Bottom work area co-ordinates)
R1+44 Right
R1+48 Top
R1+52 File type
R1+56 File name (string_value)

The file specified by "Filename" contains a series of parameters in the form of name-value pairs. This data is the list of attributes and parameters from the APPLET, OBJECT or EMBED tag - see their respective definitions. This data is used by the plugin to understand what is being requested of it - see Data formats for details.

If bit 0 of the flags word is set then this is a request to open the file as a helper application. ie external to the parent application. In this case the bounding box (offset bytes 36 to 51) are invalid. The parent window handle may be valid or 0 depending on how the file is launched.

MESSAGE_PLUGIN_OPENING (&4D541)

Sent by the plugin task to say an instance has been created.

R1+12 my_ref field from
Message_PlugIn_Open
R1+16 Message_PlugIn_Opening

Common fields
R1+20 Flags
Bit 0: 1 = plugin can accept input focus
0 = plugin cannot use input focus
Bit 1: 1 = plugin wants code resource fetched
0 = plugin will fetch code resource itself Bit 2: 1 = plugin wants data resource fetched
0 = plugin will data code resource itself Bit 3: 1 = plugin will delete the parameter file itself
0 = parent should delete parameter file now
Bit 4: 1 = plugin has more work to do, keep showing busy indicator
0 = plugin is loaded
Bit 5: 1 = plugin understands PLUGIN_ACTION message beyond the STOP reason code.
0 = plugin understands at most STOP
Bit 6: 1 = plugin task has actually opened a helper window
0 = plugin has embedded itself in the parent window Bits 7-31: reserved (must be zero)
R1+24 Plugin instance handle (invented by the plugin) R1+28 Browser instance handle (copied from Message_PlugIn_Open)

MESSAGE_PLUGIN_CLOSE (&4D542)

R1+16 Message_PlugIn_Close

Common fields
R1+20 Flags
Bit 0: Browser would also like plugin to exit
Bits 1-31: reserved (must be zero)
R1+24 Plugin instance handle to close
R1+28 Browser instance handle

MESSAGE_PLUGIN_CLOSED (&4D543)

R1+12 my_ref field from
Message_PlugIn_Close
R1+16 Message_PlugIn_Closed

Common fields
R1+20 Flags
Bit 0: Plugin will exit after this message Bit 1: This message is NOT in reply to a
Message_PlugIn_Close.
Bit 2: Set if there is an error message at R1+32
Bits 3-31: reserved (must be zero)
R1+24 Plugin instance handle of the closed instance R1+28 Browser instance handle of the closed instance

If flags bit 2 is set:
R1+32 Error number.
R1+36 Zero-terminated message to be displayed by the browser.
(NB This message is always embedded here as the plugin may be exiting).

MESSAGE_PLUGIN_RESHAPE (&4D544)

R1+12 my_ref field from Message_PlugIn_Request (if applicable)
R1+16Message_PlugIn_Reshape

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 Parent window handle
R1+36 Left (bounding box in parent's
R1+40 Bottom work area co-ordinates)
R1+44 Right
R1+48 Top

MESSAGE_PLUGIN_RESHAPE_REQUEST (&4D545)

R1+16 Message_PlugIn_Reshape_Request

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 Width (in OS units)
R1+36 Height

MESSAGE_PLUGIN_FOCUS (&4D546)

R1+16 Message_Plugin_Focus

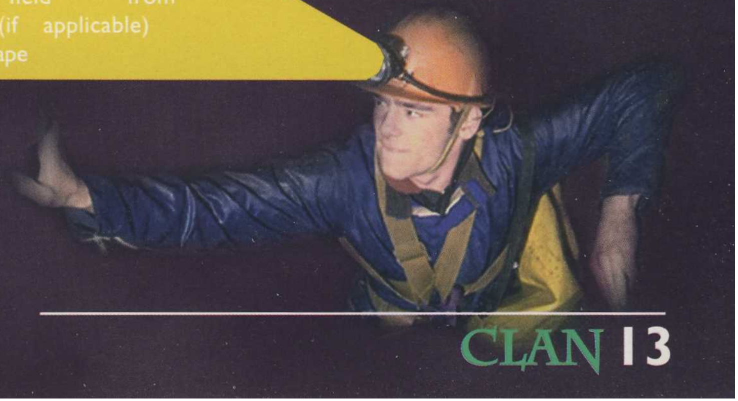
Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle

This message is used to transfer the input focus between a plugin and its parent. It can be sent in either direction. If the recipient cannot or does not wish to accept the focus then it just ignores the message. Otherwise it should acknowledge the message with message type 19 to prevent it being bounced back to the originator.

MESSAGE_PLUGIN_UNLOCK (&4D547)

R1+16 Message_Plugin_Unlock

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 URL for which to unlock the cache file (string_value)



MESSAGE_PLUGIN_STREAM_NEW (&4D548)

R1+16 Message_Plugin_StreamNew
Common fields
R1+20 Flags
Bits 0-3: stream type field
= 0: normal
= 1: seek only
= 2: as file
= 3: as file only
Bit 4: 1 = stream is seekable
Bits 5-31: reserved (must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle

Common stream fields
R1+32 Plugin stream instance handle
R1+36 Browser stream instance handle
R1+40 URL of stream source/dest (string_value) R1+44 end of stream in bytes, or 0 if unknown R1+48 last modified data of URL
R1+52 notify data
R1+56 mime type of URL (string_value)
R1+60 window target (string_value)

MESSAGE_PLUGIN_STREAM_DESTROY (&4D549)

R1+16 Message_Plugin_StreamDestroy

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle

Common stream fields
R1+32 Plugin stream instance handle
R1+36 Browser stream instance handle
R1+40 URL of stream source/dest (string_value) R1+44 end of stream in bytes, or 0 if unknown R1+48 last modified date of URL (in unix time) R1+52 notify data
R1+56 Reason
0 = Stream finished successfully
1 = Stream finished due to an error
2 = Stream finished due to user intervention

MESSAGE_PLUGIN_STREAM_WRITE (&4D54A)

R1+16 Message_Plugin_StreamWrite

Common fields
R1+20 Flags
Bits 0-3: Data type field
= 0: string_value
= 1: anchor
= 2: file handle
Bits 4-31: reserved (must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle

Common stream fields
R1+32 Plugin stream instance handle
R1+36 Browser stream instance handle
R1+40 URL of stream source/dest (string_value) R1+44 end of stream in bytes, or 0 if unknown R1+48 last modified data of URL
R1+52 Notify data
R1+56 Logical offset in stream of data
R1+60 Length of data
R1+64 Data ptr

MESSAGE_PLUGIN_STREAM_WRITTEN (&4D54B)

R1+12 my_ref field from Message_PlugIn_StreamWrite R1+16 Message_Plugin_StreamWritten

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle

Common stream fields
R1+32 Plugin stream instance handle
R1+36 Browser stream instance handle
R1+40 URL of stream source/dest (string_value) R1+44 end of stream in bytes, or 0 if unknown R1+48 last modified data of URL
R1+52 notify data
R1+56 Length of data consumed, < 0 if error

MESSAGE_PLUGIN_STREAM_AS_FILE (&4D54C)

R1+16 Message_Plugin_StreamAsFile

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle

Common stream fields
R1+32 Plugin stream instance handle
R1+36 Browser stream instance handle
R1+40 URL of stream source/dest (string_value) R1+44 end of stream in bytes, or 0 if unknown R1+48 last modified data of URL
R1+52 notify data
R1+56 file name of stream data (string_value)

MESSAGE_PLUGIN_URL_ACCESS (&4D54CD)

R1+16 Message_Plugin_URLAccess

Common fields
R1+20 Flags
Bit 0: 1 = return a NOTIFY message on completion Bit 1: 1 = POST to the URL
0 = GET from the URL
If bit 1 = 1
Bit 2: 1 = POST a file
0 = POST a block of memory
Bits 3-31: reserved (must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 URL to access (string_value)
R1+36 window target (string_value)
R1+40 notify data to be returned
R1+44 length of data to be posted
If bit 2=1
R1+48 file name (string_value)
If bit 2=0
R1+48 ptr to data (string_value)

If the window target is non-null then the URL is fetched to the given window name. Otherwise a stream is opened and data is sent to the plugin.

MESSAGE_PLUGIN_URL_NOTIFY (&4D54E)

R1+16 Message_Plugin_URLNotify

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 URL accessed (string_value)
R1+36 reason for notify
0 = Stream finished successfully
1 = Stream finished due to an error
2 = Stream finished due to user intervention
R1+40 notify data

MESSAGE_PLUGIN_STATUS (&4D54F)

R1+16 Message_Plugin_Status

Common fields
R1+20 Flags (reserved, must be zero)

R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 status message (string_value)
Requests that the parent display some information in its status bar or similar. The message should be reasonably short.

MESSAGE_PLUGIN_BUSY (&4D550)

R1+16 Message_Plugin_Busy

Common fields
R1+20 Flags
Bit 0: 1 = Plugin is busy
0 = Plugin is not busy
Bit 1: 1 = State at offset 32 has some meaning
0 = ignore word at offset 32
Bits 2-31: reserved (must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 State
0: Stop
1: Play
2: Pause
3: Fast Forward
4: Rewind
5: Record

Requests that the parent display some indication of business (eg spinning logo etc.). If the plugin had set the busy bit in its OPENING message then it should send this message with bit 0 clear when it has finished its loading.
This is also used to notify the parent of any state change by the plugin in case it needs to update any user interface.

MESSAGE_PLUGIN_ACTION (&4D551)

R1+16 Message_Plugin_Action

Common fields
R1+20 Flags (reserved, must be zero)
R1+24 Plugin instance handle
R1+28 Browser instance handle
R1+32 New state
0: Stop
1: Play
2: Pause
3: Fast Forward
4: Rewind
5: Record
6: Mute
7: Unmute

This message is used for sending specific commands to a plugin. Not all plugins will understand the commands sent.
The new state sent is the state the plugin should enter. If it is already in that state, then it should ignore the message.
After entering the state, it should send back a BUSY message giving the new state except for the Mute and Unmute actions.

MESSAGE_PLUGIN_ABORT (&4D552)

R1+16 Message_Plugin_Abort

Common fields
R1+20 Flags (reserved)
R1+24 Plugin instance handle
R1+28 Browser instance handle

This message is sent by the browser when the user clicks on the Stop icon (or performs its equivalent). The plugin should stop as much of its activity as possible. Specifically, anything that updates the screen, anything that uses significant CPU time and anything that accesses the network should be suspended.
Note that this message is sent to each plugin instance individually and should be treated as such.

DATA FORMATS

The MESSAGE_PLUGIN_OPEN contains a filename that refers to a file of parameters and attributes. The plugin uses this information to locate the correct data, classes, implementation etc.

The file contains the concatenation of one or more binary records of the form:-

4 bytes type:
0 = terminator (this is the last word in the file)
1 = data from PARAM
2 = URL from PARAM
3 = object ref PARAM
4 = special parameter from browser
5 = data from OBJECT/APPLET
6 = URL from OBJECT/APPLET
4 bytes m = size of record (without header)
(4+n+p) + (4+s+q) + (4+t+r) bytes
4 bytes n = size of name (unpadded)
n bytes Name
p bytes Padding to word boundary
4 bytes s = size of data (unpadded)
s bytes Data
q bytes Padding to word boundary
4 bytes t = size of mime type (unpadded)
t bytes Mime type
r bytes Padding to word boundary

Integers are stored in little-endian order.
Flags (parameters with void value whose presence or absence only is significant) are represented by a parameter of type DATA with zero length.
The parameters include:-
* all the attributes of the OBJECT (or other) element that references this plugin
* all the PARAM elements enclosed within it
* special parameters created by the browser

These parameters are passed exactly as seen in the HTML without any conversions. The data/url/ref distinction is as given in the DTD (for OBJECT attributes) or in the VALUETYPE attribute of the PARAM element.
The plug-in may implement its own URL fetching code, or it may have the Browser fetch URLs on its behalf by issuing a URL_ACCESS message to the browser.
Special parameters are created by the browser (rather than being part of the object element). They are:-

BASEHREF

The full URL of the document containing this object.

USERAGENT

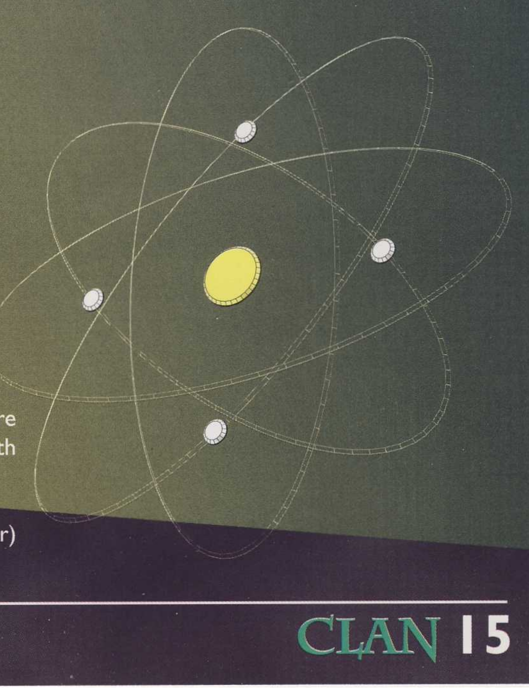
The name of the browser.

UAVERSION

Version number of the browser (user agent) in format x.y. If the plugin needs a specific browser feature, it may refuse to initialise if this version is not high enough.

APIVERSION

Version number of this API in format x.y. Changes in x mean a major incompatible change in formats. If the plugin doesn't understand this version it should refuse to initialise. Changes in y mean some new functionality introduced in a backwards compatible way. ■



TWO GREAT OFFERS FROM

Acorn

Buy any A7000+ or StrongARM Risc PC & get free:

A — MIDI KIT

INCLUDING

49 Note Full Size Keyboard
Evolution MK149
&
MIDI Connect Card
&
MIDI sequencing
software - MIDIWorks lite

OR

B — NINTENDO 64

INCLUDING

Controller
&
2 top selling games

For further information call: Acorn Computer Group plc, Acorn House, 645 Newmarket Road,
Cambridge, United Kingdom CB5 8PB Telephone: 0990 329070 Facsimile: 01223 725435

Terms and Conditions: Offer open while stocks last. Option only available with the purchase of a new StrongARM Risc PC or Acorn A7000+.