Chico Specification

Microsoft Confidential. Do not copy or distribute. If you need a copy contact Eric Straub.

HIGHLY CONFIDENTIAL

MX 9068493
CONFIDENTIAL

Specification Revision: **Unreleased DRAFT - Revised on 08/20/93 by Eric Straub**

TEX 0012182
CONFIDENTIAL

CMS 00030373

Plaintiff's Exhibit
5595
Comes V. Microsoft

RBC 04751

1. Overview

Chico is a placeholder code name for MS-DOS based on Chicago. This product is being done to ensure we have a competitive option for cost conscious OEMs who don't want to put Chicago on their systems, and to reap revenue from upgrade customers who won't buy Chicago. Anticipated competitors are Novell (DR) DOS and PC-DOS.

Our strategy is to create version of MS-DOS based on Chicago minus GUI with minimal amount of effort. The key differentiating features from Chicago are no Windows GUI support perhaps smaller hardware requirements.[1] Current hardware requirements are defined by Chicago, which is specing a 4 Meg 386 as the minimal configuration as of 7/16/93.

1.1. Assumptions and Non-Goals

Cost of Goods Sold (COGS) is less critical than MS-DOS 6 due to lower volume of product.

We will be as compatible with MS-DOS apps as Chicago. We will not attempt to be more compatible.

In general, we will not be adding any API or services for MS-DOS apps in Chico (other than those services provided by Chicago to MS-DOS apps). However, we may inherit some from Chicago, such as the shutdown API. Chico will not support WIN32 API (other than hooks exposed via the inherited Chicago kernel)

In general, where we don't support a feature of Chicago, this is done by dropping files. If other files (such as the kernel) have hooks for the unsupported feature, we may keep that limited support in order to minimize effort and maintenance.

Wherever possible, when a common file with Chicago (such as Command.com) and Chico, the file should be binary identical. This will minimize testing and maintenance.

1.2. Competitors

Novell DOS 7

As of this writing, Novell DOS 7 is not yet shipping. It has been announced and is expected to ship by the end of CY 93. The major features are:

- Peer-to-Peer networking
- Full-Screen Multi-Tasking
- Stacker Compression (extended with DPMS)
- Cache extended with DPMS
- Uninversal Novell Client including SNMP support
- Remote Install
- Backup and AntiVirus (from Fifth Generation, makers of FastBack)

MX 9068494
CONFIDENTIAL

PC DOS

PC-DOS 6.1 is shipping as of August '93. It is fairly uninteresting. We don't know what they intend to do for their next release.

HIGHLY CONFIDENTIAL

TEX 0012183
CONFIDENTIAL

[1] It is an open at this point whether Chico-DOS will have lower hardware requirements than Chicago.

2. Key Chico Features

2.1. Multi-Tasking MS-DOS VMs

Chico will support multiple MS-DOS sessions, running in seperate VM's. Unlike Chicago which supports running windowed oldapps via WinOldAp, Chico will only run them full screen. The program manager/Task Manager (a new program(s) to be written for MS-DOS 7) will be run full screen in the System VM. In addition, there will be a seperate MS-DOS VM (which is where the initial boot leaves the user) which has the C: prompt. AUTOEXEC.BAT will be executed in that VM (so that any user shell program, etc. is run). The user can switch to the system VM with ALT-TAB. On default startup, there are only two tasks - The system VM and the MS-DOS Prompt VM.

The user can spawn a program in a new MS-DOS VM with the START command from the MS-DOS prompt, or they can select a program via the Program in the System VM and start it from there. If a user simply types a command (like 123) at the C: prompt, it will be run in the same VM.

In Chicago, Alt-Tab in an old-app is noticed by WinOldAp which signals USER to swap the app. We will need to support this functionality somehow since we don't have either of these. Raymond Chen has developed a command shell which will do this, but it is out of date now. It would not be a lot of work for him to make it work again.

From the Program Manager the user can

- create and modify program settings (EMS, XMS, etc.).
- launch programs based on configured program settings

The program information is stored in the registry in a format which is binary compatible with Chicago program information. From the task manager, the user can:

- See a list of all tasks in the system
- Switch to any task in the task list
- adjust system settings for any task (priority, foreground/background operation, etc.)
- terminate any task
- shutdown the system.

It is not yet decided whether the task manager and the program manager will be a single application.

2.2. Long File Names (Protect Mode)

Long File Names (LFNs) will be supported in VFAT, but will only be preserved in real-mode (same as Chicago). Supporting LFN's will be the majority of the work we must do to existing MS-DOS Utilities.

2.3. Plug-N-Play

Chico will support plug-n-play[2] BIOS and devices. Plug-n-Play uses information stored in the registry and information provided by PNP drivers/hardware to build a *hardware* tree which represents all hardware in the system. It then uses the information in the hardware tree to automatically configure devices installed in the system. If conflicts arise, it trys to transparently resolve them. If it cannot, it

[2]Plug-n-Play is defined in the Chicago specification.

provides intelligent options to the user (such as choose between using this device or that device). The hardware tree is built each time the system is started. There are four phases in the boot process: BIOS in control (phase 0), OS in control real-mode (phase 1), Chicago real-mode loader (phase 2), and protect mode (phase 3) There are three components of PNP which interact with the user: Master Installer, Configuration Manager, and part of the System Explorer All three of these components are run during Phase 3. Chico will need to support all of these components, although they parts may be rewritten to work without Windows.

As of 8/13/93 (from Nagara), Chicago plan for PNP support through phase 2 is to have a new file, XXX.SYS, which is loaded by MSDOS.SYS immediately after dblspace.bin, and before config.sys is processed. There will also be a CONFIG$ device which provides the PNP device driver API (get version, get config, lock/unlock config; the lock config will always fail in real-mode). XXX.SYS will have limited access to the registry and look at the buses (PCI, EISA, and ISAPNP) and to collect information about the devices. This information will include which devices are required to boot. XXX.SYS goes away before config.sys processing begins. The information is all passed to CONFIG$ who keeps it conventional memory until XMS becomes available at which time CONFIG$ moves the data into XMS.

Intel has developed a VxD which provides PNP device driver API with Windows 3.1. This VxD uses a plain text file in the root instead of the registry. We have access to source code for this VxD if we want it.

At this point, it appears that the majority of the work to support PnP will be to support the three components mentioned above to work without Windows. The other pieces will fall out of the Chicago work.

Master Installer

Tracy Sharpe is writing the PnP Master Installer The Master installer is started with a WinExec by the configuration manager. It then walks the hardware tree (calls into the configuration manager) to figure out why it was called. When it finds a devnode which needs to be configured, it calls into SETUPX.DLL to configure the node.

According to Tracy, the installer has no UI in it but it does some UI via SETUPX.DLL to find compatible device drivers, configure the device, and load the device driver. In addition, SETUPX.DLL handles things like letting the user decide which device to install when there are more than one choice. Tracy also wrote parts of SETUPX.DLL (RonG also worked on it).

Configuration Manager

Pierre-Yves Santerre wrote the configuration manager.

System Explorer PnP support

Ed Halley is writing the PnP portions of the system explorer.

2.4. Remote Admin

2.5. Universal network client.

Chico will include the Chicago universal network client. This will allow it access MS, Novell, and other network servers.

2.6. Peer file Server

Chico will include a peer file server

2.7. Common PIF Database with Chicago

The Program Information File (PIF) database for ChicoDOS will be binary compatible with the Chicago version. Chicago supports editing this database via WinOldApp system menu. Current thinking for Chico is that the Task Manager will be used to edit the pif settings.

2.8. Communications support

[From RichF]
Objective
Be compatible with the installed base of MS-DOS comm apps

Non-Objective
Providing new communications services

Windows 3.1
Overview
MS-DOS comm apps communicate with a port in one of two ways:

- Writing directly to the hardware by poking the memory location of the port
- Using BIOS Int 14h, which is much slower and less commonly used

Windows does not reliably support high-speed communications in a DOS box because of the overhead of virtualizing the ports. MS-DOS comm apps have always been a compatibility weakness of Windows.

MS-DOS app support
The only MS-DOS comm app support in Windows are basic services for handling communications in a multitasking environment:

- Windows virtualizes the port hardware and buffers incoming data so that comm apps reading data off the port do not drop characters when they are timesliced away (COMMBUFF.386)
- Windows arbitrates device contention when two apps are trying to read or write to the same port (VCD.386)

Configuration issues

- The Ports control panel icon lets users set port settings like baud, flow control, parity, etc., but Keithla says it doesn't really do anything. In any case, MODE can change all the same settings.
- The 386 Enhanced control panel icon has device contention settings the user can change to tell Windows how to handle the case of two apps trying to access the same port
- SYSTEM.INI provides a large number of settings for use in troubleshooting comm problems that can only be changed by editing SYSTEM.INI directly.

Chicago
Overview
Chicago will include VCD.386 and an improved version of COMMBUFF.386 that will greatly improve MS-DOS comm app performance in a VM and lessen the need for users to drop to real mode to run their comm apps.

Necessary files for Chico

- VCD.386
- COMMBUFF.386

Necessary configuration tools for Chico

- Existing MODE

- Direct editing of SYSTEM.INI. The only change from Windows 3.1 is that device contention settings now in the 386 Enhanced control panel icon will have to be changed by editing SYSTEM.INI directly

Open Issues
Chicago contains an entirely rearchitected, layered communications subsystem which is not of interest to MS-DOS apps. As it stands now, COMMBUFF 386 talks directly to the hardware and circumvents the new comm subsystem. However, Keithla is considering changing COMMBUFF 386 so that it no longer talks to the hardware but rather goes through VCOMM.386, the new communications router, through the new port drivers, and then finally down to the hardware.

If this change happens, we will have to ship VCOMM.386 and the standard port drivers, LPT.386 and COMM.386. Also, the new comm architecture anticipates that third-party port drivers will include DLLs that provide a configuration UI. We would either have to support these DLLs, force the user to configure the drivers by editing the registry directly, or not support configurable protect-mode port drivers.

2.9. Printer Support

The Chicago spooler is written in Win32 code. Since this is not supported in MS-DOS, we will not be able to use the Chicago spooler. Printer support for Astro is an open issue at this point.

2.10. International Support

Current thinking is that Chico will inherit all of chicago's international support.

2.11. Registry

Registry is accessed via the Registry VxD. Nagara wrote this code.

2.12. Smaller Edit

Chicago is planning on using a new editor (20K) for their emergency disk. The editor is being written by a teacher at MSU in spare time. The edit engine is based on 3yr old code, the UI is brand new and is not yet alpha. It is limited to 64K file size and has no online help.

3. Chicago Features Not Included In Chico

- Chico will not include any of the following programs:
  - Explorer
  - Windows Applets.
  - resource viewer
  - net watcher
  - hardware viewer
  - system enablers (charmap, vbasic, recorder, system scheduler, and file viewers)
  - mini-word processor/text editor
  - paint replacement
  - cardfile
  - calculator
  - clock
  - games
  - wallpaper
  - multimedia enhancements
  - Workgroup apps
  - terminal.
- WinOldApp and the ability to run MS-DOS apps in a Window.
- Win32 API (other than support which is inherited from Kernel files)
- NT NLS API
- OLE
- Mail and MAPI
- Document management
- FAX API

4. Open Issues

4.1. How do we provide utility and Core functionality?

Chicago provide some basic utilities which are not currently provied in MS-DOS. These utilities are listed in the table below

Utility/Functionality	Chicago Method	Win 3.1 equivalent	Chico Approach
Critical message popup			
Plug-N-Play		N/A	
Access/Edit system Configuration	Explorer	None	
Configuration Manager Conflict Resolution	Configuration Manager		
Device Installation	Master Installer	None	
Control Panel	DannyO owns.	CONTROL.EXE	
Color		CP Color	None
Fonts		CP Fonts	None
Ports		CP Ports	
Mouse		CP Mouse	Chicago will require a mouse VxD for the mouse to work properly. If we provide that, we will also need to provide a means of adjusting the mouse settings in the registry.
Screen Saver/Password protection		CP Desktop	
Keyboard/International		CP Keyboard/ International	
Printers		CP Printers	None
Date/Time		CP Date/Time	Date & Time commands
Network		CP Network	
Device Contention, Virtual Memory, Task Priority		CP 386Enhanced	
Multimedia Drivers		CP Drivers	
Sounds		CP Sound	None

Task Management		TASKMAN E XE	New full screen app in system VM
PIF Editing	done via System menu	PIFEDIT and System menu	Provide via Task Manager
Configuration Editor	New 20K Editor (Alpha code in September) & also via PNP	SYSEDIT	EDIT
Online Registration	JohnPa	none	
Disk Partitioning (FDISK)	WINDISK	none	FDISK
Format		File Manager for floppies. No option for fixed disks	Setup (OEM) and FORMAT
Disk check and repair	Aaron's util?	none	SCANFIX
Virus Scan		none	MSAV/MWAV
Disk Compression	VFAT (BillKru)	none	DBLSPACE.BIN
Disk Compression Setup/Manager		none	DBLSPACE.EXE
Disk Copy/Compare		Copy via FILEMAN; No Compare	DISKCOPY/DISKCOMP
File management		FILEMAN	MS-DOS utilities (move, dir, etc.)
Undelete	recycle bin	none	UNDELETE.EXE (we will need to modify if we need to be compatible with chicago. AndrewCo is developer on wastebasket.
Setup	New Setup program (JohnHe); Generic Installer	Setup	Based on Chicago Setup.
Warn user about dangerous software	WARNING (RichP)	None	
Print Spooling	New 32bit Spooler (LinS)	PRINTMAN	
Backup/Restore	MWBACKUP	None	MSBACKUP
CDROM	CD-IFS	MSCDEX	CD-IFS
Mobv-Win Support			

4.2. Support the Acessibility features provided in Chicago?

4.3. Support long filenames in Real-Mode?

We need to consider the full impact of doing this For example, will we need to rev A.V. Backup, and all other MS-DOS utilities to support?

4.4. Do we support running any version of Windows?

Assuming Novell DOS 7 will support Windows 3.1, seems we will need to support this. Otherwise, users who upgrade from MS-DOS 6.2 to Chico will loose the ability to run any of their existing windows apps.

4.5. Should smaller hardware requirements than Chicago be a design goal?

Novell-DOS 7 could put pressure on us to have lower hardware requirements here.

4.6. Define Setup work.

Setup work depends on some of the other issues (support 286, etc.). Setup will be based on the Chicago setup program.

4.7. Which Virtual Display Drivers (VDD)?

In order to support multi-tasking, these will be required. Which ones?

4.8. Which control files (System.ini, etc.)?

4.9. Screen Saver?

Should we support this?

4.10. Clipboard?

Are we going to support cut/paste between VMs?

4.11. Mouse support

Objectives

- Compatibility with the installed base of mouse-aware MS-DOS apps
- No-footprint mouse driver

Windows 3.1

Overview

MS-DOS apps communicate with the mouse via Int 33h, which is supported in all real-mode mouse drivers including our own MOUSE.COM. Windows apps use Windows APIs which are supported by MOUSE.DRV, Windows' protect-mode mouse driver.

MS-DOS app support

There is no special support in Windows for using a mouse in an MS-DOS app aside from support for a mouse in a windowed MS-DOS app. MS-DOS apps still require a real-mode mouse driver to be loaded in order to use a mouse under Windows.

Configuration issues

None.

- The mouse settings in Windows only affect Win apps. For example, swapping the left and right buttons in the control panel swaps them in Windows apps but not MS-DOS apps running under Windows. Basic Windows mouse configuration settings are stored in WIN.INI, and hardware mouse settings (like, for example, what COM port to use) are stored in SYSTEM.INI.
- Value-addded mouse settings, like the "snap-to" or "wrap-around" settings of the MS mouse driver, are stored in a MOUSE.INI file
- Real-mode mouse drivers come with their own character-mode configuration applets for changing mouse settings in MS-DOS apps. They store all settings in files like MOUSE.INI.

Chicago

Overview

Chicago will provde a new, layered mouse architecture that will support both the Windows mouse APIs as well as the real-mode Int 33h APIs. The layer beneath the Windows kernel is VMOUSE.386, and between VMOUSE.386 and the hardware (typically a com port) will be mouse-specific VxDs such as MSMOUSE.386 or LOGITECH.386.

All Windows mouse configuration information will be held in the registry and modified through the control panel. Mouse configurations set in Windows will affect both Windows and MS-DOS apps running under Windows. Individual mouse vendors will still provide separate configuration applets to provide access to value-added features like snap-to and wrap-around, and will continue to store this additional configuration information in files like MOUSE.INI.

Necessary files for Chico

- VMOUSE.386
- Brand-specific VxDs

Necessary configuration tools for Chico

Although mouse vendors will continue to provide character-mode configuration applets for their real-mode drivers, these applets will, as always, store all their settings in proprietary .INI files. However, since protect-mode mouse drivers will look for some of their settings in the registry and not a .INI file, we will need the Windows mouse control panel tool that controls mouse settings in the registry. The value-added settings can continue to be stored in a .INI file and set by the existing character-mode mouse configuration apps.

4.12. Recycle Bin/Undelete Support

MX 9068503
CONFIDENTIAL

What work will we need to do to Undelete to make it compatible with Chico?

TEX 0012192
CONFIDENTIAL

4.13. Printer Support

We can't use Chicago print spooler because it uses WIN 32 We could potentially get the Snowball print spooler Would still need to do a UI for it.

5. Known Work for Chico

Task Manager

Chico will need some way to manage all of the tasks. The Task Manager will do this.

PNP components to work on Chico

- Functionality provided in the System explorer
- Config Manager
- Master Installer

Updating of MS-DOS Utilities

Utilities	Status	Drivers	Status
Xcopy	Chicago	Display	Leave alone
Move	Chicago	Ansi	Okay
Smartdrv	Chicago	Driver	Okay
Command	Chicago	Emm386	Okay
Fc	Chicago maybe	Himem	Okay
Replace	Chicago maybe	Power	Okay
Subst	Chicago maybe	Ramdrive	Okay
Undelete	Chicago maybe		
Edit	Drop		
Qbasic	Drop		
Append	Drop		
Fastopen	Drop		
Tree	Drop		
Format	Leave alone		
Label	Leave alone		
Debug	Leave alone		
Expand	Leave alone		
Graphics	Leave alone		
Keyb	Leave alone		
Nlsfunc	Leave alone		
Restore	Leave alone		
Setver	Leave alone		
Sys	Leave alone		
Sort	Leave alone		

Key	
Chicago	Chicago will modify
Chicago maybe	Chicago may modify
Drop	Recommend dropping
Leave alone	Recommend no mods for LFNs
Need to update?	Need to discuss if LFN mods required
Okay	No mods required
Update needed	MS-DOS team should modify

Notes
Windows utils dropped
Edit will be replaced

Loadfix	Leave alone
Msd	Leave alone
Memmaker	Need to update?
Chkdsk	Drop
Interlnk/svr	Maybe Okay
Print	Drop or modify to support LFNs
Diskcopy	Okay
Doskey	Okay
Mem	Okay
Mode	Okay
Unformat	Okay
Diskcomp	Okay
Choice	Okay
Attrib	Update needed
FDisk	Update needed
Find	Update needed
More	Update needed
Msav	Update needed
Deltree	Update needed
Dblspace	Update needed
Defrag	Update needed
Scanfix	Update needed
Msbackup	Update needed
Vsafe	Update needed
Help	Update needed