

eComStation 1.2 class

David van Enkevort
Alex Taylor
Chuck McKinnis



David van Enckevort

- ➔ David is currently employed at Mensys, where he is involved in the development of eComStation. For eComStation version 1.1 he was the lead in the development of the new installer, which with improvements by others is still the basis of the eComStation Installation CD. Since he bought his first computer he has been active in several OS/2 related groups, and released a couple of programs (ASCII Table, Activity Monitor, xDel, etc.). He currently is board member of the editorial board of the Dutch OS/2 quarterly Draad/2.

Alex Taylor

- ➔ Alex started using OS/2 in university, after being given a copy of Warp 3. On graduating, he started work doing OS/2 LAN support at the head office of a large Canadian insurance company. Since joining the eComStation Development Group, he has added OS/2 programming to his resume.
- ➔ Alex's projects include work on the installer, as well as Icon Themes, the Archive Viewer, the Installation Volume Manager, and various design and concept work.

Chuck McKinnis

- ➔ Chuck retired from IBM in 1992 after 26 years as a systems engineer in large systems and communications. He is a private consultant and the New Mexico Location Manager for Sirius Enterprise Systems Group, an IBM Premier Business Partner and largest IBM eServer reseller in the world. He is the author of the eCS Maintenance Tool, NICPAK, and a contributor to RexxAutoStart. He has been an OS/2 user since the beta days of 1989 and has programming experience with Rexx that goes back to the first release of Rexx by IBM.

What you will see during this class

- ⇒ Explain problems with old components.
- ⇒ Per item theory to explain how it works.
- ⇒ How new components work and how they better help you diagnose problems .

CD Boot Process: emulation mode

- ➔ 'Emulation mode' is the CD boot method MCP1/MCP2 eCS 1.0
- ➔ Works by emulating a bootable floppy disk (via an image file on the CD).
- ➔ BIOS support for this method is often not tested (or implemented) these days.
- ➔ There is a limit on the number/size of drivers that will fit inside image files.
- ➔ No real-time update of drivers (diskette image files are read-only).

Non-emulation boot & MemDisk

- ➔ 'Non-emulation' mode is standard for bootable CDs since Windows 2000.
- ➔ Tested by PC/BIOS vendors.
- ➔ eCS solution uses MemDisk developed by Veit Kannegeiser.
- ➔ Mechanism to update drivers before boot!
- ➔ Can even boot from unsupported USB diskette drive via INT 13 calls
- ➔ More disk images can unpacked (are also compressed!)

MemDisk & CSM implementation

- ➔ INT 13 BIOS calls used to unpack images
- ➔ Boot from RAM drive -- looks like HDD to OS
- ➔ CSM menu (a.k.a. preboot menu)
 - Predefined CONFIG.SYS
 - Based on selections/detection it activates certain drivers/options (for example IDE, SCSI or USB)
 - SET statements in CONFIG.SYS used during install (HWMERGE.CMD)

Diagnose & solve boot problems

- ➔ In preboot menu (introduced in eCS 1.2), edit CONFIG.SYS.
- ➔ In preboot menu press ALT F2/F4
- ➔ Update kernel or driver before boot (via floppy).
- ➔ Online documentation can be accessed before boot

Limitations of IBM's hardware detection

- ➔ Hardware detection hard-coded in DLLs; e.g. selective installer `\os2\dll\mmsniff.dll`, network installer `\ibminst\clbncd.dll`
- ➔ In selective installer: hardware detection for sound and video, present in MCP1, disabled in MCP2. (Source of much trouble?) (`OS2\INSTALL\AUDFSAFE.TXT`)
- ➔ No new hardware support like USB detection

eCS 1.2 hardware detection

- ➔ Matching drivers and vendor ID/chipset ID in flat text file.
- ➔ One database for all detected hardware
- ➔ No longer writes to memory I/O ports to detect legacy hardware.

“Troubleshooting” hardware detection

- ➔ All detect and recognized hardware is in
`\ECS\INSTALL\RSP\MACHINE.CFG`
- ➔ All detected hardware is in
`\ECS\INSTALL\RSP\RAWDATA.000`
- ➔ Support can be added by sending in
`rawdata.000` to support@ecomstation.com

Why a new installer?

- ➔ IBM installer hard coded pages in DLL (how to add a component, for example RSJ ?)
- ➔ User interface not consistent (selective installer and network installer really different look and feel).
- ➔ IBM installer did not have easy-to-use layout.

Installer history: eComStation 1.0

- ⇒ Used a 100% custom install process.
- ⇒ Advantages of this approach:
 - Simpler to use: more consistent layout, few questions asked
 - Fewer reboots required
- ⇒ Problems with this approach:
 - IBM would not support it
 - Not designed with multi-lingual support in mind – very difficult to translate to other languages
 - Monolithic, and dependent on proprietary tools – almost impossible to maintain and extend
 - Never managed to properly integrate certain components (e.g TCP/IP configuration)

Installer history: eComStation 1.1

- ➔ Introduced the current installer design.
- ➔ Based on IBM CID process (Configuration, Installation, & Distribution).
- ➔ Uses a three-layer design:
 - Layer 1: User input (graphical install interface)
 - Layer 2: Middleware (REXX)
 - Layer 3: Individual product installers (CID-enabled executables supplied by vendors)
- ➔ Advantages of this approach:
 - Very easy to extend or change
 - Supported by IBM (since we use their own process)
 - Can fix (or work around) known problems in Layer 2
 - Designed to support multiple languages

eComStation 1.2 Installer

- ➔ Same basic design as eComStation 1.1.
- ➔ Some minor changes to Layer 1 (graphical interface) and Layer 3 (installer executables).
- ➔ Layer 2 (middleware) has been almost completely rewritten.

How the installer works

- ➔ The installation CD boots a minimal Presentation Manager session.
- ➔ Instead of the Workplace Shell, the graphical shell used by PM is the eComStation Guided Procedure program (GUID-PROC.EXE).

GUIDPROC.EXE overview

- ➔ GUIDPROC.EXE is the graphical front-end for the install process.
- ➔ Uses a wizard-like layout.
- ➔ Common keyboard commands:
 - Ctrl + / Ctrl - (enlarge/reduce text size)
 - Shift F3 (open a command prompt window)
 - F5 (view detected hardware – after detection runs)
 - > / < (next/previous page)
 - Ctrl Alt F11 (switch user interface mode: fullscreen vs window)

GUIDPROC.EXE design

- ➔ Every page you see is partially defined by a Guided Procedure Script (GPS) file.
- ➔ GPS files define the text displayed on the page; they also help determine how the user's selections are dealt with.
- ➔ All the GPS files for a specific language are kept in a single subdirectory, which is set at runtime. This makes support for multiple languages very easy.
- ➔ It also allows a great deal of customization, without having to modify the program itself.

Installer stages

- ➔ The graphical install process consists of two stages:
 - **Interactive Stage:**
Prompts the user for installation options, and writes them to corresponding CID response files.
 - **Execution Stage**
Launches the REXX middleware which executes the individual product installers. This occurs in three 'phases' (separated by re-boots).

Interactive Stage (1)

- ➔ The choices offered on the first few pages relate to the behaviour of the installer interface itself (e.g. display style, license agreement, level of expertise desired, etc.).
- ➔ The first choice that actually affects the installation itself is the selection of the target volume.
- ➔ The same page allows you to launch the Installation Volume Manager, where you can create or delete volumes, or install and configure the IBM Boot Manager.

Interactive Stage (2)

- ➔ Next, the selected install volume (a.k.a. ?:) may be formatted at the user's discretion. (Formatting may be mandatory, depending on the current state of the selected volume.)
- ➔ At present, if you want to format additional volumes, you must open a command prompt (Shift F3) and issue the format commands from there.

Interactive Stage (3)

- ➔ Throughout the remainder of the interactive stage, the user makes various selections to identify which eComStation components will be installed, and how.
- ➔ GUIDPROC.EXE processes these selections and uses them to generate CID response files, which are written to the directory **?:\ECS\INSTALL\RSP**

Execution Stage (1)

- ➔ The execution stage begins when you select 'Next' on the final interactive page of the installer. At this point, the installer calls the GPS script **EIENDPH1.GPS**, which launches the actual installation.
- ➔ This script specifies a list of programs and parameters which are to be executed in sequence.
- ➔ These programs perform the actual installation of eComStation.

Execution Stage (2)

➔ EIENDPH1.GPS runs the following programs, in sequence:

- MIGPREP.CMD
 - Checks to see if a migration (upgrade installation) is being performed, and, if so, prepares the target volume.
- PATHSED.CMD
 - Loads the REXXUTIL and ECSRXLIB function DLLs.
 - Writes a file called PATHS.SED containing various path specifications required by the install process.
 - Uses the SED utility in conjunction with PATHS.SED to generate several REXX scripts (DEFRSP.CMD, CATALOG.CMD, MIGRATE.CMD, RESUME.CMD and LCUINST.CMD), which are written to ?:\ECS\INSTALL\RSP.
- DEFRSP.CMD
 - Generates default response files for anything that still requires them. (This mostly applies to Easy Installs.)
 - Performs last-minute housekeeping on response files.

Execution Stage (3)

- **CATALOG.CMD**
 - Generates the 'install catalog' (?:\ECS\INSTALL\RSP\PRODUCTS.CAT), which controls which product installers will be run.
- **MIGRATE.CMD**
 - Performs any final tasks required for migration.
- **SCANINIT.CMD**
 - Checks for a user-provided script (INSTINIT.CMD) and runs it if found.
- **LCUINST.CMD**
 - The main installation procedure. This is a specially customized LCU command file.

Execution Stage (5)

- ➔ When EIENDPH1.GPS runs the last program, the reboot prompt appears. (Everything up to this point is called 'Phase 1' of the installation.)
- ➔ After reboot, the system boots newly installed eComStation volume.
- ➔ The startup files on this volume have been configured to run GUIDPROC.EXE again, but this time it simply executes the script EIENDPH2.GPS (referred to as 'Phase 2').

Execution Stage (5)

- ➔ When EIENDPH2.GPS finishes, the reboot prompt appears again.
- ➔ At present, this is followed by the execution of EIENDPH3.GPS (a.k.a. 'Phase 3'), which performs some final clean-up tasks.
- ➔ (We hope to eliminate Phase 3 in the future.)

Installer Middleware

- ➔ The original intention (before eCS 1.1) was to have EIENDPH1.GPS (etc.) call the individual component install programs directly.
- ➔ However, IBM would not support this: CID installers must be run from REXX scripts!
- ➔ For 1.1 GA, a lengthy REXX script was automatically generated to simply call each install program in sequence.
- ➔ For 1.2, we re-architected this process, based on LCU.

CID and LCU

- ➔ CID is IBM terminology for non-interactive installations. A CID-enabled product can be installed using just an ASCII 'response file'.
- ➔ Virtually every OS/2 component, and most third-party software, is CID-enabled.
- ➔ LCU (the LAN CID Utility) is a standardized method for batch installation of multiple CID products. It is basically a special REXX script (called an LCU command file).

LCUINST.CMD

- ➔ eComStation 1.2 uses an LCU command file called LCUINST.CMD, which is auto-generated under ?:\ECS\INSTALL\RSP.
- ➔ LCUINST.CMD runs the installation routine for each eComStation component which has been selected for installation.
- ➔ Most of these installation routines are themselves REXX scripts, which in turn call the vendor's CID-enabled install executable (with some housekeeping).

Monitoring the installation

- ➔ During the installation, the graphical installer displays messages which indicate which component is currently being installed.
- ➔ For more detailed monitoring, you can press Ctrl Alt F11 to switch the installer to window mode, and move or minimize it. This will allow you to see many of the individual install programs running.
- ➔ You can also use the Window List (Ctrl Esc) to access programs running in the background.

Errors during installation

- ➔ The multiple levels of execution makes it difficult to return meaningful error codes back to the user.
- ➔ Also, installers often don't return detailed error codes at all, but simply record messages in their own log files.
- ➔ Therefore, when an individual install program fails, the displayed error code simply identifies the failing install program. You can use this code when requesting support.

Installation troubleshooting

- ➔ Log files are saved in ?:\VAR\LOG.
- ➔ Always check ECSINST.LOG for error messages first. This should indicate which individual install program failed.
- ➔ Once you know which individual installer failed, check the specific log file(s) for that installer.
- ➔ Ask for help on the eComStation BugTracker (available through ecomstation.com). Provide as much information as possible! Strongly recommended to provide the file INSTLOG.ZIP from ?:\VAR\LOG

Resuming an aborted installation

- ➔ If you are able to fix the error which causes the installation to abort, you can (usually) resume the installation from the point where it stopped. Go to the `?:\ECS\INSTALL\RSP` directory, and run the command
`RESUME`

General problem-solving tips

- ➔ Installer 'stalls' during execution stage.
 - Use Ctrl Alt F11 and move or minimize the installer window. Check for concealed error popups.
 - Check ECSINST.LOG for error messages which might not have caused an abort (use TYPE command).
- ➔ Odd errors (crashes, freezes, cryptic messages) occur at seemingly random points.
 - Rewrite the CD (if burnt from an ISO file – **4x speed max!**), or ask vendor for a replacement
 - Redownload the ISO file (if applicable)
 - Check system RAM
 - Make sure system components are seated properly
 - Make sure fans (CPU & chassis) are functioning.

Multimedia and mininstall

- ⇒ New MINSTALL needed because IBM version:
 - Copies outdated files like genin.dll, clock01.sys, etc.
 - Performs poor logging (mininstall.log)
 - Exhibits bugs in processing scripts (e.g. crashes or quits without an error message)
 - GUI sometimes would hang

Multimedia and IBM classes

- ➔ Does not always handle errors gracefully (e.g. when crashing)
- ➔ Loads multimedia code into shared memory of WPS (Workplace Shell). When multimedia crashes, it will hang the entire desktop.

New MINSTALL and MM classes

- ⇒ CW 0.2.8 classes
 - Loads in private memory, much(!) more stable
 - Logging in ?:\CWMM.LOG when multimedia crash occurs
- ⇒ New MINSTALL has much better logging in mininstall.log.
- ⇒ New MINSTALL GUI can:
 - Reset your configuration
 - Uninstall soundcard drivers
 - Reinitialize your CD player

New in MMeCS 1.2

- New classes that no longer load the MMOS/2 subsystem as WPS code but in private memory. Makes WPS much more stable
- New MINSTALL written from scratch
- New JPEG codec to handle progressive JPEGs
- MJPEG video codec integrated
- OpenMPEG from IBM is integrated
- Quickmotion + Anpo AVI codecs integrated
- Base IBM video playback integrated
- OGG Vorbis and FLAG MMIO codec integrated
- Escape GL screensaver (Open GL 1.1)
- New CD player with FREEDB support
- Many new audio drivers

Why and how migration in eCS 1.2

- ➔ People like to maintain their desktop objects
- ➔ eComStation 1.1 had a migration wizard
- ➔ ECS 1.2 checks WPDesktop entry in OS2.INI
- ➔ IBM instaler does not migrate old
- ➔ CONFIG.SYS: intelligent filtering that runs at the end of the eCS installation.
- ➔ Extensive logging

Why eWorkplace ?

- ➔ eWorkplace is based on Xworkplace
- ➔ Warpcenter bug: hang when desktop restarts. eWorkplace fixes this.
- ➔ Has many safeguards if desktop has trouble starting.
- ➔ Replaces IBM many WPS classes

Questions and short install class

- ⇒ Any questions
- ⇒ Victim machines...?