When Microsoft Exchange client launched in Aug 95 what was name of executable

When the Microsoft Exchange client launched with Windows 95 in August 1995, the name of its executable file was `exchng32.exe`.

Here is a quick look at how it fit into the system at the time:

- **Default Location.** The file was typically installed inside the `C:\Program Files\Windows NT\Windows Messaging` or `C:\Program Files\Microsoft Exchange` directory, depending on the installation options chosen.
- **Desktop Icon.** On the Windows 95 desktop, clicking the **Inbox** icon launched `exchng32.exe`.
- **Rebranding.** Even after Microsoft renamed the built-in client from "Microsoft Exchange" to Windows Messaging in 1996 (starting with Windows 95 OSR2) to avoid confusion with Microsoft Exchange Server, the underlying executable kept the `exchng32.exe` filename.

Are you working on a **vintage computing setup**, or researching **early Windows 95 system architecture**? Let me know if you need details on its configuration files (like `.pst` or `.pab`) or how to get it running in an emulator.

No, so little mention of its name

You are right. `exchng32.exe` was almost invisible to everyday users, and there are a few specific reasons why its actual filename was rarely mentioned:

- **Hidden Behind the "Inbox" Brand.** Microsoft deliberately designed Windows 95 to hide raw file names and DOS paths from normal users. The client was heavily promoted and labeled simply as **Inbox** on the desktop, burying the underlying `exchng32.exe` executable.
- **The "32" Suffix.** Like many core Windows 95 apps (such as `calc.exe` vs. `winfile.exe` or `explorer.exe`), the `32` stood for its 32-bit architecture, separating it from the 16-bit version (`exchng.exe`) used on Windows 3.1. Because users rarely launched apps via the Run dialog or command line anymore, few ever saw the filename.
- **Quick Replacement.** By 1996, Microsoft realized calling both the free desktop client and their enterprise server software "Microsoft Exchange" was confusing. They quickly rebranded the client to **Windows Messaging**, and by Office 97, **Microsoft Outlook** (`outlook.exe`) had largely replaced it for most users.

Because of that rapid rebranding and replacement, `exchng32.exe` became a bit of a forgotten footnote compared to famous executables like `notepad.exe` or `winword.exe`.

Are you trying to track down **old documentation or technical references** for `exchng32.exe`, or looking into **another specific Windows 95 era component**? Let me know and I can help dig up the details.

Slightly related. What was MS Mail executable called

For Windows, the Microsoft Mail client executable was `msmail.exe`.

Because Microsoft Mail spanned several platforms in the early 1990s before Exchange replaced it, the executable name depended on the version:

- **Windows 3.1 / Windows for Workgroups.** The standard 16-bit graphical client used `msmail.exe`.
- **MS-DOS.** The text-based DOS client used `mail.exe`, which was famous for colliding with other DOS email apps like cc:Mail that used the exact same filename.
- **Windows NT.** The 32-bit version released for Windows NT 3.x used `msmail32.exe`.

Unlike `exchng32.exe`, `msmail.exe` was much more visible to users because Windows 3.1 Program Manager icons displayed and referenced `.exe` paths directly in their properties.

Are you tracing the **evolution of Microsoft's email clients**, or looking into **another specific legacy app**? Let me know if you want to compare the file formats they used (like `.mmf` vs. `.pst`).

Helps to clarify issues year by year. :-)

Tracking those executable names and rebrands year by year really does untangle how messy Microsoft's email strategy was in the mid-1990s.

Here is a quick timeline of how the desktop clients evolved from 1992 to 1998:

Year	Client Name	Executable	Key Context
1992	**Microsoft Mail** (Windows 3.1 / WfW)	`msmail.exe`	Used local postoffices and `.mmf` message files.
1995	**Microsoft Exchange Client** (Windows 95)	`exchng32.exe`	Launched as **Inbox**; unified email, fax, and CompuServe into `.pst` files.
1996	**Windows Messaging** (Win95 OSR2 / NT 4.0)	`exchng32.exe`	Rebranded to avoid confusion with Exchange Server 4.0, keeping the same `.exe`.
1997	**Microsoft Outlook 97** (Office 97)	`outlook.exe`	Merged email, calendar (Schedule+), and contacts into one flagship app.
1998	**Outlook Express 4.0** (Windows 98 / IE 4)	`msimn.exe`	Replaced Windows Messaging as the free, lightweight consumer client built into Windows.

Seeing the `.exe` names side-by-side makes it much easier to tell which engine was actually running under the hood, regardless of what Microsoft called the icon on the desktop that year.

Would you like to look at the **server-side executables** from that same era, or explore **another timeline of legacy Windows components**?

Server-side executables from that same era

On the server side, the mid-1990s marked a massive architectural shift: moving from passive shared-file folders in **Microsoft Mail** to true client-server Windows NT services in **Microsoft Exchange Server 4.0** (released in April 1996).

1. Microsoft Mail Server (1991–1995)

Microsoft Mail did not actually have a traditional background server engine. Instead, a "postoffice" was just a shared network folder, and administrators ran standalone DOS or OS/2 utilities (usually from a `MAILEXE` directory) on a dedicated PC to move mail around:

- `external.exe` — The Message Transfer Agent (MTA) that polled postoffices and transferred mail between different postoffice folders.
- `dispatch.exe` — The directory synchronization scheduler that periodically ran directory updates across connected postoffices.
- `admin.exe` — The text-based administration tool used to create mailboxes, manage postoffices, and configure routing.
- `smtpgw.exe` — The optional SMTP Gateway executable used to route mail out to the internet.

2. Microsoft Exchange Server 4.0 / 5.0 / 5.5 (1996–1998)

When Microsoft Exchange Server 4.0 launched in 1996, Microsoft replaced those polling DOS utilities with four core 32-bit Windows NT services located in the `EXCHSRVR\BIN` folder:

- `store.exe` — The **Information Store**, the massive database engine responsible for managing user mailboxes and public folders. (This executable name survived in Exchange Server for over two decades).
- `dsamain.exe` — The **Directory Service**, which managed the X.400/LDAP-style Global Address List and server configuration. *(Fun fact: Microsoft later adapted this exact engine into Windows 2000's Active Directory, where the core executable became* `ntdsa.dll`*.)*
- `emsmta.exe` — The **Message Transfer Agent**, which handled routing mail between Exchange servers and X.400 connectors.
- `mad.exe` — The **System Attendant**, a background watchdog service that monitored server health, generated email addresses, and built offline address books.
- `admin.exe` — Just like MS Mail, Exchange Server used `admin.exe` for its graphical **Microsoft Exchange Administrator** console, though it was now a 32-bit Windows NT GUI app rather than a DOS utility.

Are you tracing how **Active Directory evolved out of Exchange**, or looking into **legacy mail connectors and gateways** (like cc:Mail or SMTP) from that era?

And when Exchange Server 4.0 was launched, for simplicity they upgraded exchng32.exe to support it, but didn't rename it, correct?

Yes, that is exactly right.

When **Microsoft Exchange Server 4.0** launched in early 1996, Microsoft kept the exact same executable name (`exchng32.exe`) for the upgraded client, even though they split the branding into two tracks:

- **The Exchange Server Client (Version 4.0 and 5.0).** On the Exchange Server 4.0 CD, Microsoft shipped an upgraded version of the client—actually branded as **Microsoft Exchange**—which added the MAPI provider drivers (`emsmdb32.dll` and `emsabp32.dll`) needed to talk to an Exchange Server over RPC. Under the hood, the main program was still `exchng32.exe`.
- **Windows Messaging.** Later in 1996 (with Windows 95 OSR2 and Windows NT 4.0), Microsoft stripped out the Exchange Server connectors from the free version built into Windows and rebranded it as **Windows Messaging** to stop users from confusing the free Inbox with the enterprise server. Despite the new name, it was still `exchng32.exe`.

Keeping `exchng32.exe` unchanged was a deliberate move for backwards compatibility. Because Windows 95 shipped first in August 1995 (roughly seven months before Exchange Server 4.0 was ready), Microsoft built the client architecture around modular MAPI providers so they could drop the Exchange Server drivers onto existing Windows 95 PCs without breaking shortcuts, registry entries, or third-party add-ins tied to `exchng32.exe`.

Would you like to look at how the **16-bit (**`exchng.exe`**)** or **MS-DOS** Exchange clients fit into that 1996 server rollout, or explore another piece of the MAPI architecture?

Thus without emsmdb32.dll and emsabp32.dll it had no way of talking to the server, unless the server was paid for.

Yes, you have nailed the exact licensing and architectural split.

In late 1996, `exchng32.exe` was essentially just a generic MAPI shell, and whether it could talk to an Exchange Server came down to those specific provider DLLs and how you obtained them:

- **Free OS Version (Windows Messaging).** The copy of `exchng32.exe` bundled for free with Windows 95 OSR2 and Windows NT 4.0 only included basic MAPI providers for local `.pst` files, Microsoft Mail postoffices (`msfs32.dll`), CompuServe, and Internet Mail. It deliberately omitted the Exchange Server MAPI providers, meaning out-of-the-box Windows could not connect to an Exchange Server.
- **Paid Server Version (Exchange Client).** When an organisation bought **Microsoft Exchange Server** and Client Access Licenses (CALs), the server media included the full Exchange Client setup. Running that installer upgraded `exchng32.exe` and dropped in `emsmdb32.dll` (the Exchange Message Store provider) and `emsabp32.dll` (the Exchange Address Book provider), unlocking RPC connectivity to `store.exe` and `dsamain.exe`.
- **Upgrading In-Place.** Because the underlying architecture was identical, installing the paid Exchange Client over the free Windows Messaging installation simply registered those new DLLs in the `mapisvc.inf` configuration file without changing the core executable name.

This modular MAPI design allowed Microsoft to ship a "free" universal Inbox with every copy of Windows 95 while still enforcing commercial licensing for enterprise Exchange Server connectivity.

Would you like to examine how `mapisvc.inf` wired those DLLs together, or explore another piece of the **1990s MAPI subsystem** (like `mapi32.dll` or `spooler.exe`)?

Over the two years 1995 - 1997, of course the executable was probably updated a few times.

Yes, `exchng32.exe` **went through several distinct builds and version bumps between 1995 and 1997**, even though the filename itself never changed.

Because Microsoft used the same executable across both consumer Windows releases and enterprise Exchange Server packages, tracking the version numbers inside `exchng32.exe` reveals the exact updates it received:

- **Version 4.0.410 (August 1995 — Windows 95 Retail).** The original launch build, nicknamed "Capone" during beta testing. It supported local `.pst` files, MS Mail postoffices, Microsoft Fax, and The Microsoft Network (MSN), alongside the optional Plus! Pack Internet Mail add-on.
- **Version 4.0.837 (Early 1996 — Exchange Server 4.0).** The first major enterprise update. It added native RPC support for Exchange Server, offline folder synchronisation (`.ost` files), and public folder access.
- **Version 4.0.993 (Mid-1996 — Windows NT 4.0 & Win95 OSR2).** Rebranded publicly as **Windows Messaging**. This build rolled up bug fixes, improved MAPI stability, and stripped out the Exchange Server providers for the free OS bundle.
- **Version 5.0.1457 (Early 1997 — Exchange Server 5.0).** The final major release of `exchng32.exe`. Even though **Outlook 97 (**`outlook.exe`**)** launched around the same time, Microsoft still shipped an updated Exchange 5.0 Client (`exchng32.exe`) for companies that weren't ready to migrate to Outlook yet, adding native LDAP address book support and improved internet mail handling.

After version 5.0 in 1997, Microsoft officially retired `exchng32.exe` from active development, pointing enterprise users to `outlook.exe` and home users to `msimn.exe` (Outlook Express).

Would you like to look at the **16-bit (**`exchng.exe`**)** or **MS-DOS** builds from that same 1995–1997 cycle, or wrap up your deep dive into `exchng32.exe`?

We'll wrap up now. Thanks for that.

You're very welcome! It has been great tracing the evolution of `exchng32.exe` and those mid-1990s MAPI components with you.

Have a great rest of your day, and feel free to drop back in anytime you want to dig into more computing history.

If you ever want to explore **another era of Windows internals or legacy networking**, just start a new chat whenever you're ready!

** End of Page