

# ROM construct AI

A read-only-memory construct artificial intelligence is a sentient computer program formed from the neural scan of a living breathing person. Its first use was in YE 33.

Rather than be dependent on special hardware, as many artificial intelligence are (largely quantum computers or systems with huge amounts of memory or processing power) the ROM Construct is designed to run on pretty much anything available - from starships and androids to home computers and even cybernetic implants: Anywhere the code can execute, the AI will run.

Many of the special abilities of the ROM construct are disabled by design and require a member of the Lazarus consortium to authorize their use.

## History and Background

The result of an obsessed student (one Aiesu Kalopsia)'s attempt to merge psychometrics with AI and computer programming, the ROM construct began life as a very simple software system capable of running very simple emulations of brains - mostly those of mice taken in atomically accurate scanning equipment at her university.

As time went on and the technology became more advanced, the constructs became less about a 100% accurate emulation of a brain and instead replacing many parts in atom-for-atom emulation with software run-times and pre-designed code.

A limitation was found however: by using an emulated neural system (which has a limited number of definable connections) in live memory and not a cold-write system, there would be a limited number of possible neural connections the system was ever capable of - a neural contrast limit, similar to the difference between pixels on a screen, with their arrangement and difference forming meaningful information. As the number of connections approaches a high-point, the difference between information and nonsense becomes blurrier and blurrier - white and black pixels fading to the same shade of gray - until the two cannot be distinguished at all - that screen of metaphorical pixels being one soup of meaningless gray. This is called a ROM-half-life. By design, all ROM constructs literally think themselves to death.

Instead of actually fixing the problem, which the designer didn't have the means to accomplish, she instead designed the AI to be reset to its initial conditions every time it is shut down and re-activated - *it leaves no information on the long-term storage drives or memory systems: The AI is Read-Only and only exists in live memory.* This not only meant that the systems could be quite easy to deal with by exploiting this ability but anything they did could be made deniable: Every re-activation meant they would not remember their accumulated experience, in a sense, 'killing' the accumulated personality.

For example, an ARIA running a ROM-construct of reasonable complexity will think itself to death within 7-months to a year.

# Usage

In theory, this software means in practice, the mentality, judgement, personality and skills of a person can be preserved after death in such a way that they can be exploited and controlled very specifically - ideal for exploring various methods of interrogation or to get the construct to perform a special task.

Also noteworthy is that the construct allows a person to quite literally be in more than one place at a time, though they cannot share memories.

Their nature also means that their de-activation removes all digital evidence of their actions and memories, ideal for plausible deniability.

Also important is that while legally, the source-file is property, quite paradoxically the live-running memory legally counts as a Sentient Intelligence and entails the same legal rights as such.

## Special functions

### Replication

ROM-constructs, given the permission, can split and multi-task in two different methods: Agents and forks.

**Agents** are very simple non-sentient AI which perform very simple tasks - pawns of the ROM-construct. For example controlling a powered-armour or making specific adjustments and reactions to code automatically long after the ROM-construct has left the computer are all ideal tasks for an Agent. Agents can also be made as specialist software which parse and convert information or provide an interface, giving ROM-constructs the ability to write and edit software on the same computer on the machine-code level.

Second is a **fork**: A fork is a replicated AI, either from live memory or re-seeded in a different resolution. If a fork is created on a remote computer of sufficiently similar resolution, the live memory information currently running can be copied and run on the remote system, essentially allowing the AI to self-replicate and expand.

These two functions are typically disabled to ROM constructs and require special authorization from a Lazarus Consortium member.

### Alteration

Special payloads can be planted in the execution code of the AI so when specific conditions are met - either by the computer, the AI, its own memories and experience or special keyphrases. In this way, these special payloads can alter the live-memory of a construct, adding or removing memories, information, experiences, directives and skillsets.

This can include making certain information unknowable to the AI as a form of [agnosia](#), adding or removing extra skillsets or even deleting the Source-File so once shut down, the AI can never be booted again.

## Creation & Activation

In order to create a ROM construct, one must begin by creating a high resolution scan of an individual's brain. This information is then encoded into a special format which abstracts and assesses connections - looking for connections and skipping per-atom emulation, which would be hugely computationally draining to do.

This special file is then compressed via a procedural method - that is, it is constructed from a repeating looped process composed of very few lines of code called a Source-File. This Source-file can take hours, days or even months to initially compress.

When this special Source-File is run, it generates in memory the neural information of the subject which populates specially assigned slots in the AI's processing sequencers and library files - similar to compiling software.

This process can be varied in complexity by removing specific instructions or reducing the level of information complexity, producing either very dense and powerful AI or very portable and fast AI. Important is that while fast AI often lack the full resolution of memories in their more complex cousins, they pack the same capacity for reason, decision making and learning. For this reason, simple ROM constructs can come across as incredibly wise yet impossibly naive - many often forgetting they are even ROM constructs at all because the processes needed to confirm or deny their suspicions - or the suspicions themselves, do not exist!

The source-file itself is so incredibly small that the AI boot-process can actually be used as a viral payload during digital attacks.

Finally, once the AI is properly seeded and running, the excess data is cleared from memory and builds up gradually as the ROM Construct runs normally.

These populated clusters of information can be copied or 'transplanted' from AI to another, though this decreases the constructs execution lifetime significantly than if the original donor of the scan information actually knew the skill themselves. In much the same way, existing information can be 'burned out' of the seed-file itself during its creation, ensuring a ROM-construct does not know specific information that its creator does.

## OOC Notes

This page was originally created on 2012/10/26 09:55 by [Osaka/Osakanone](#).

From:

<https://wiki.stararmy.com/> - **STAR ARMY**

Permanent link:

<https://wiki.stararmy.com/doku.php?id=corp:lazarus:rosa>

Last update: **2024/02/24 07:37**

