

# Preserving Life

an homage to Hassan's legacy

Roberto Di Cosmo

Director, Software Heritage  
Inria and Université de Paris Cité

September 26th 2025

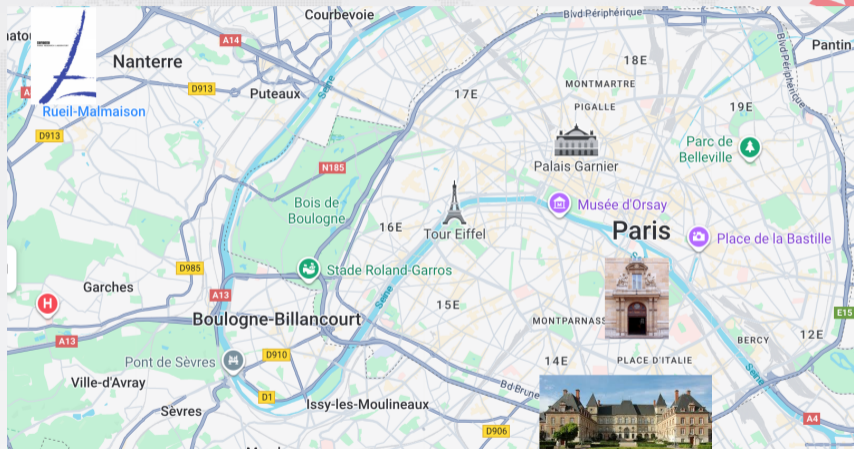


# Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

- 
- 1 Meeting Hassan
  - 2 30+ years later: Software as Heritage
  - 3 Meet Software Heritage
  - 4 Preserving our software heritage
  - 5 Preserving LIFE

# 1992: an Italian in Paris





## An Introduction to LIFE—Programming with Logic, Inheritance, Functions, and Equations

**Hassan Aït-Kaci**


Digital Equipment Corporation  
Paris Research Laboratory  
85, avenue Victor Hugo  
92500 Rueil-Malmaison, France  
hak@prl.dec.com

### Abstract

LIFE (Logic, Inheritance, Functions, Equations) is a programming language with a powerful facility for structured type inheritance. LIFE reconciles styles from functional programming and logic programming by implicitly delegating control to an automatic suspension mechanism. This allows interleaving interpretation of relational and functional expressions that specify abstract structural dependencies on objects. Together, these features provide a convenient and versatile power of abstraction for very high-level expression of constrained data structures.

... l'élément ne préexiste pas à l'ensemble, il n'est ni plus immédiat ni plus ancien, ce ne sont pas les éléments qui déterminent l'ensemble, mais l'ensemble qui détermine les éléments.

GEORGES PEREC *La vie, mode d'emploi*.<sup>1</sup>



# 7

**digital**  
PARIS RESEARCH LABORATORY

## Compiling Order-Sorted Feature Term Unification

December 1993

Hassan Alt-Kaci  
Roberto Di Cosmo

13

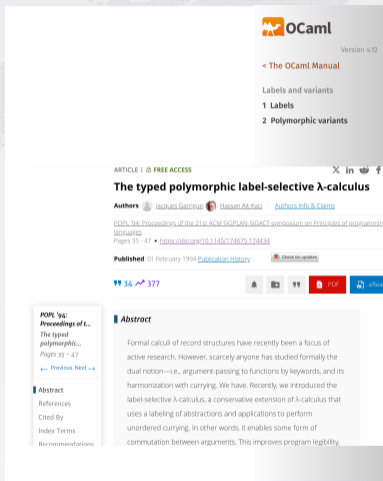
	interact_sort X, person	X: person
	test_feature X, name, X <sub>1</sub> , W <sub>1</sub>	X <sub>1</sub> name - X <sub>1</sub>
	interact_sort X, id	X: id
	test_feature X <sub>1</sub> , first, X <sub>2</sub> , 2, W <sub>2</sub>	X <sub>2</sub> first - X <sub>2</sub>
	interact_sort X <sub>1</sub> , string	X <sub>1</sub> : string
R <sub>1</sub> :	test_feature X <sub>1</sub> , last, X <sub>2</sub> , 2, W <sub>2</sub>	X <sub>2</sub> last - X <sub>2</sub>
	interact_sort X <sub>1</sub> , string	X <sub>1</sub> : string
R <sub>2</sub> :	test_feature X <sub>1</sub> , spouse, X <sub>2</sub> , 1, W <sub>1</sub>	X <sub>2</sub> spouse - X <sub>2</sub>
	interact_sort X <sub>1</sub> , person	X <sub>1</sub> : person
	test_feature X <sub>1</sub> , name, X <sub>2</sub> , 2, W <sub>2</sub>	X <sub>2</sub> name - X <sub>2</sub>
	interact_sort X <sub>1</sub> , id	X <sub>1</sub> : id
	unify_feature X <sub>1</sub> , last, X <sub>2</sub>	X <sub>2</sub> last - X <sub>2</sub>
R <sub>3</sub> :	unify_feature X <sub>1</sub> , spouse, X <sub>2</sub>	X <sub>2</sub> spouse - X <sub>2</sub>
R <sub>4</sub> :	jump W <sub>1</sub>	{skip write code}
	push_cell X <sub>1</sub>	X <sub>1</sub> : person
	set_sort X <sub>1</sub> , person	X <sub>1</sub> : person
W <sub>1</sub> :	push_cell X <sub>2</sub>	X <sub>2</sub> : id
	set_feature X <sub>1</sub> , name, X <sub>2</sub>	X <sub>2</sub> name - X <sub>2</sub>
	set_sort X <sub>1</sub> , id	X <sub>1</sub> : id
W <sub>2</sub> :	push_cell X <sub>2</sub>	X <sub>2</sub> : string
	set_feature X <sub>1</sub> , first, X <sub>2</sub>	X <sub>2</sub> first - X <sub>2</sub>
	set_sort X <sub>1</sub> , string	X <sub>1</sub> : string
	write_test 2, R <sub>1</sub>	
W <sub>3</sub> :	push_cell X <sub>2</sub>	X <sub>2</sub> : string
	set_feature X <sub>1</sub> , last, X <sub>2</sub>	X <sub>2</sub> last - X <sub>2</sub>
	set_sort X <sub>1</sub> , string	X <sub>1</sub> : string
	write_test 2, R <sub>2</sub>	
	write_test 3, R <sub>3</sub>	
W <sub>4</sub> :	push_cell X <sub>2</sub>	X <sub>2</sub> : person
	set_feature X <sub>1</sub> , spouse, X <sub>2</sub>	X <sub>2</sub> spouse - X <sub>2</sub>
	set_sort X <sub>1</sub> , person	X <sub>1</sub> : person
W <sub>5</sub> :	push_cell X <sub>2</sub>	X <sub>2</sub> : id
	set_feature X <sub>1</sub> , name, X <sub>2</sub>	X <sub>2</sub> name - X <sub>2</sub>
	set_sort X <sub>1</sub> , id	X <sub>1</sub> : id
W <sub>6a</sub> :	set_feature X <sub>1</sub> , last, X <sub>2</sub>	X <sub>2</sub> last - X <sub>2</sub>
	write_test 3, R <sub>4</sub>	
	write_test 2, R <sub>4</sub>	
W <sub>6b</sub> :	set_feature X <sub>1</sub> , spouse, X <sub>2</sub>	X <sub>2</sub> spouse - X <sub>2</sub>
	write_test 3, R <sub>5</sub>	
	write_test 1, R <sub>5</sub>	
	write_test 0, R <sub>5</sub>	
W <sub>7</sub> :		

Figure 9 Code for program CSP term (1)

Technical Note No. 7

December 1993

# ... then witnessing a lasting impact



The screenshot shows the OCaml website interface. At the top right, there is a navigation menu with the OCaml logo and the text "Version 4.12". Below this, a list of manual sections is shown: "< The OCaml Manual", "Labels and variants", "1 Labels", and "2 Polymorphic variants". The main content area displays an article titled "The typed polymorphic label-selective  $\lambda$ -calculus" by Jacques Garrigue and Hassan Ajlouni. The article is marked as "ARTICLE" and "FREE ACCESS". It includes a PDF icon and a citation link. Below the article title, there is an "Abstract" section with the following text: "Formal calculi of record structures have recently been a focus of active research. However, scarcely anyone has studied formally the dual notion—i.e., argument-passing to functions by keywords, and its harmonization with currying. We have. Recently, we introduced the label-selective  $\lambda$ -calculus, a conservative extension of  $\lambda$ -calculus that uses a labeling of abstractions and applications to perform unordered currying. In other words, it enables some form of commutation between arguments. This improves program legibility."

An introduction to OCaml

The core language  
The module system  
Objects in OCaml  
• Labels and variants  
Polymorphism and its limitations  
Advanced examples with classes and modules

## Chapter 4 Labels and variants

This chapter gives an overview of the new features in OCaml 3: labels, and polymorphic variants.

### 1 Labels

If you have a look at modules ending in `Labels` in the standard library, you will see that function types have annotations you did not have in the functions you defined yourself.

```
# ListLabels.map;;  
- : f:('a -> 'b) -> 'a list -> 'b list = <fun>  
  
# StringLabels.sub;;  
- : string -> pos:int -> len:int -> string = <fun>
```

Such annotations of the form `name:` are called *labels*. They are meant to document the code, allow more checking, and give more flexibility to function application. You can give such names to arguments in your programs, by prefixing them with a tilde `~`.

```
# let f ~x ~y = x - y;;  
val f : x:int -> y:int -> int = <fun>  
  
# let x = 3 and y = 2 in f ~x ~y;;  
- : int = 1
```

When you want to use distinct names for the variable and the label appearing in the type, you can use a naming label of the form `~name:`. This also applies when the argument is not a variable.

- 
- 1 Meeting Hassan
  - 2 30+ years later: Software as Heritage
  - 3 Meet Software Heritage
  - 4 Preserving our software heritage
  - 5 Preserving LIFE

# Calling for preservation: UNESCO

Experts call for greater recognition of software source code as heritage for sustainable development

6 November 2018



UNESCO, Inria, Software Heritage invite  
40 international experts meet in Paris ...

# Calling for preservation: UNESCO

Experts call for greater recognition of software source code as heritage for sustainable development

6 November 2018



UNESCO, Inria, Software Heritage invite  
40 international experts meet in Paris ...



The call is published on Feb 2019

# Calling for preservation: UNESCO

Experts call for greater recognition of software source code as heritage for sustainable development

6 November 2018



UNESCO, Inria, Software Heritage invite  
40 international experts meet in Paris ...

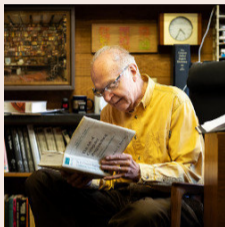
The call is published on Feb 2019

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

<https://en.unesco.org/foss/paris-call-software-source-code>



Communications of the ACM, February 2021



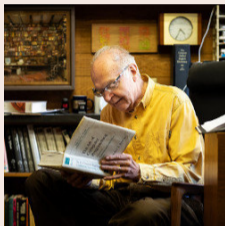
*"Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to."*

*Let's Not Dumb Down the History of Computer Science*

Donald E. Knuth, Len Shustek

<https://doi.org/10.1145/3442377>

Communications of the ACM, February 2021



*"Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to."*

*Let's Not Dumb Down the History of Computer Science*

Donald E. Knuth, Len Shustek

<https://doi.org/10.1145/3442377>

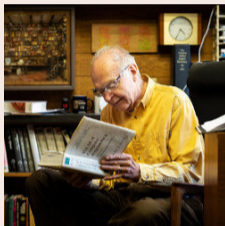
A unique opportunity

most of the creators are still here: we can talk to them!

but the clock is ticking...

We need a universal archive of software source code:

Communications of the ACM, February 2021



*"Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to."*

*Let's Not Dumb Down the History of Computer Science*

Donald E. Knuth, Len Shustek

<https://doi.org/10.1145/3442377>

A unique opportunity

most of the creators are still here: we can talk to them!

but the clock is ticking...

We need a universal archive of software source code: now we have one!

- 
- 1 Meeting Hassan
  - 2 30+ years later: Software as Heritage
  - 3 Meet Software Heritage
  - 4 Preserving our software heritage
  - 5 Preserving LIFE



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

Preserving our heritage, enabling better software and better science for all



## Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

Preserving our heritage, enabling better software and better science for all

### Reference catalog



find and reference all  
software source code



## Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

Preserving our heritage, enabling better software and better science for all

### Reference catalog



find and reference all software source code

### Universal archive



preserve and share all software source code



# Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

Preserving our heritage, enabling better software and better science for all

## Reference catalog



**find** and **reference** all  
software source code

## Universal archive



**preserve** and **share** all  
software source code

## Research infrastructure



**enable analysis** of all  
software source code

One infrastructure  
open and shared



One infrastructure  
open and shared



Largest archive

One infrastructure  
open and shared



Largest archive

## Technology

- transparency and FOSS
- replicas all the way down

## Content (billions!)

- **intrinsic identifiers**
- facts and provenance

## Organization

- non-profit
- multi-stakeholder

## Sharing the vision



United Nations  
Educational, Scientific and  
Cultural Organization



And many more ...

[www.softwareheritage.org/support/testimonials](http://www.softwareheritage.org/support/testimonials)

## Donors, members, sponsors



### Diamond sponsors



### Platinum sponsors



### Gold sponsors



### Silver sponsors



### Bronze sponsors



- 1 Meeting Hassan
- 2 30+ years later: Software as Heritage
- 3 Meet Software Heritage
- 4 Preserving our software heritage
- 5 Preserving LIFE



Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

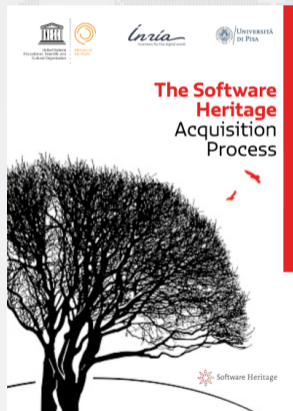
## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”



## Paris Call on Software Source Code

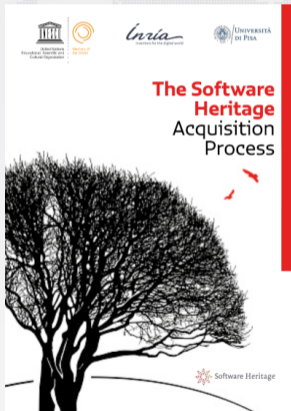
“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”



- **Rescue** Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported

## Paris Call on Software Source Code

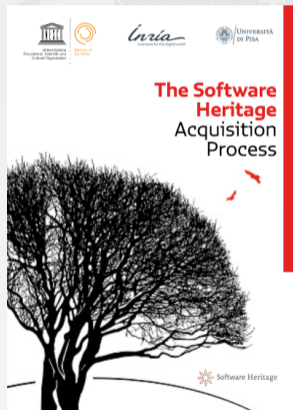
“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”



- **Rescue** Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported
- **Curate** the code
  - reconstructing the development history
  - collecting metadata

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”



- **Rescue** Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported
- **Curate** the code
  - reconstructing the development history
  - collecting metadata
- And **illustrate** with dedicated presentations

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”



The screenshot shows the Software Heritage website interface. At the top, it says "SOFTWARE STORIES" and "Software Heritage THE GREAT LIBRARY OF SOURCE CODE". Below that, the main heading is "The Pisa Collection" with the subtitle "Stories of landmark legacy code (Beta Version)" and "3 STORIES IN THIS COLLECTION". There is a search bar labeled "Search Collection...". Below the search bar is a photograph of a computer workstation from the 1970s, featuring a CRT monitor, a keyboard, and a large wooden cabinet. The caption below the photo reads "TAUmus".

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”



The screenshot shows the Software Heritage website interface. At the top, it says 'SOFTWARE STORIES' and 'Software Heritage THE GREAT LIBRARY OF SOURCE CODE'. Below that, the main heading is 'The Pisa Collection' with the subtitle 'Stories of landmark legacy code (Beta Version)' and '3 STORIES IN THIS COLLECTION'. There is a search bar labeled 'Search Collection...'. Below the search bar is a photograph of a computer workstation from the 1970s, including a CRT monitor, keyboard, and a large cabinet. The name 'TAUmus' is written below the photo.

- **Expand** the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”



- **Expand** the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- **Preserve and Present** all this material

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”



- **Expand** the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- **Preserve and Present** all this material
- **Share** process and tools (all open source!)
  - with museums, archives and all interested parties

## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”



- **Expand** the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- **Preserve and Present** all this material
- **Share** process and tools (all open source!)
  - with museums, archives and all interested parties

see this live on [the Software Stories website](#), and get *the guide*

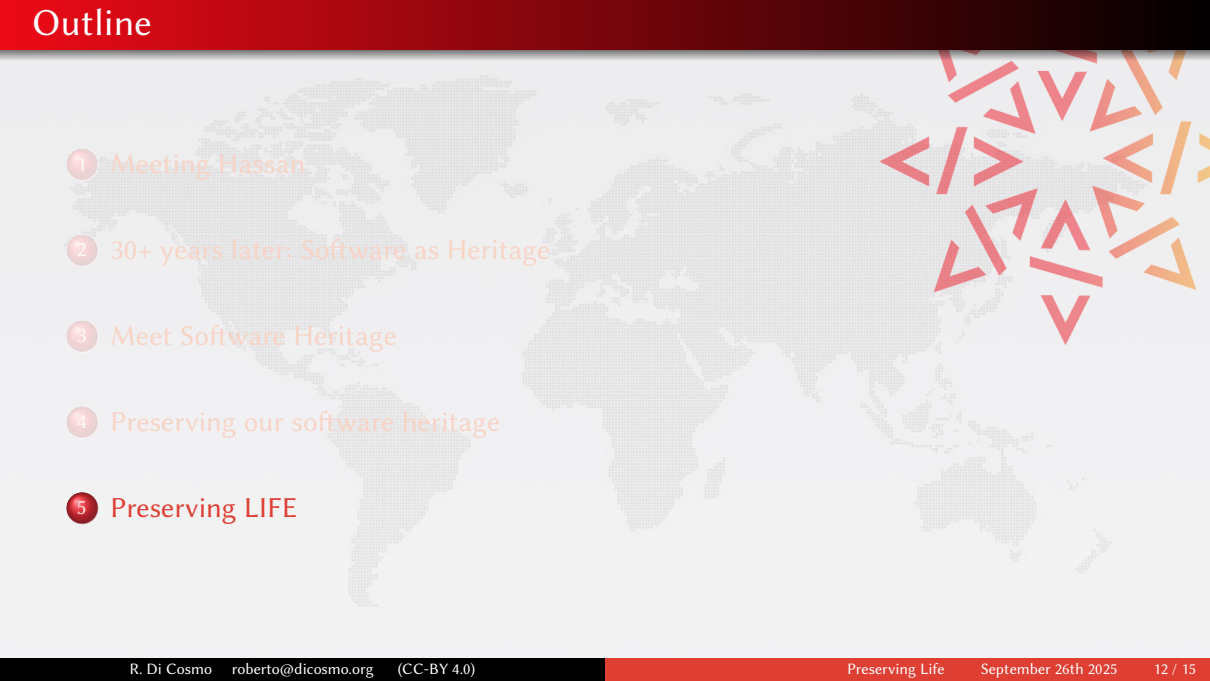
## Paris Call on Software Source Code

“[We call to] support efforts to gather and preserve the artifacts **and narratives** of the history of computing, while the earlier creators are still alive”



- **Expand** the SWHAP scope to
  - documents
  - media (videos, pictures, images, etc.)
  - oral history
- **Preserve and Present** all this material
- **Share** process and tools (all open source!)
  - with museums, archives and all interested parties

see this live on [the Software Stories website](#), and get [the guide](#) and [the SWHAP Days hybrid event](#), 19 and 20 october 2022

- 
- 1 Meeting Hassan
  - 2 30+ years later: Software as Heritage
  - 3 Meet Software Heritage
  - 4 Preserving our software heritage
  - 5 Preserving LIFE

Search engines are (more or less) your friends: [CMU has it!](#)

← → ↻ 🌐 cs.cmu.edu/afs/cs/project/ai-repository/ai/lang/prolog/impl/fp\_lp/life/

📖 | ★ Bookmarks 📁 Sel2Org 📄 Simple O... 📧 Debian P...

## Index of /afs/cs/project/ai-repository/ai/lang/prolog/impl/fp\_lp/life

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
📁 <a href="#">Parent Directory</a>		-	
📄 <a href="#">0.doc</a>	1994-12-07 14:44	5.5K	
📄 <a href="#">0.html</a>	1995-02-13 10:34	6.7K	
📄 <a href="#">0.lst</a>	1994-07-15 11:50	1.1K	
📄 <a href="#">announce.txt</a>	1994-03-29 23:05	4.9K	
📄 <a href="#">license.txt</a>	1993-01-28 07:55	2.9K	
📄 <a href="#">life_090.ltv</a>	1994-07-15 11:50	72K	
📄 <a href="#">life_090.tgz</a>	1993-08-09 22:16	537K	
📄 <a href="#">life_091.ltv</a>	1994-07-15 11:50	72K	
📄 <a href="#">life_091.tgz</a>	1993-08-09 22:17	571K	
📄 <a href="#">life_10.ltv</a>	1994-07-15 11:50	110K	
📄 <a href="#">life_10.tgz</a>	1994-03-24 13:39	1.3M	
📄 <a href="#">readme.txt</a>	1993-08-19 02:30	4.5K	

Apache/2.4.18 (Ubuntu) Server at www.cs.cmu.edu Port 443

📁 > ChatGPT 5 Thinking ▾

Great, now let's make an exercise on a real world case. Let's consider the code source de LIFE / Wild LIFE (le système de LIFE développé au PRL de Digital par l'équipe incluant Hassan Aït-Kaci) :

Dépôt CMU AI Repository (archives officielles) — contient les tarballs `life_090.tgz`, `life_091.tgz` et `life_10.tgz` avec le code C et LIFE de l'interpréteur `Wild_LIFE`, le README et la licence DEC (non-commerciale).

Indication rapide : [https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/lang/prolog/impl/fp\\_p/life/](https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/lang/prolog/impl/fp_p/life/) et la `life_10.tgz` est la dernière archive listée (1.3 MB, datée 1994-03-24). Propose a step by step series of commands to get the original material, and then process it according to the SWHAP official guide.

Thought for 1m 15s

Awesome choice. Here's a clean, reproducible SWHAP workflow for **LIFE / Wild\_LIFE** from the CMU AI Repository. I give you:

- a **one-shot Bash recipe** (download → verify → structure → metadata → Git reconstruction)
- plus the **alternative "archives" ingestion** path on Software Heritage (no Git reconstruction)

I've inlined exact URLs, dates, and license info from CMU's listing and files so your curation trail is airtight.

CMU School of C... +2



# LIFE (source code) saved!

SoftwareHeritage / Wild\_Life-swhap

Type to search

<> Code Issues Pull requests Security Insights Settings

Wild\_Life-swhap Public

Edit Pins Watch 0 Fork 0 Star 0

main 2 Branches 0 Tags

Go to file Add file Code

Roberto Di Cosmo Add badges with pointer to the archived versions 3b10d23 · 3 weeks ago 4 Commits

additional_materials	Add the WildLife_Handbook.ps	3 weeks ago
metadata	SWHAP depository: Wild_LIFE original tarballs + metadata	3 weeks ago
raw_materials	SWHAP depository: Wild_LIFE original tarballs + metadata	3 weeks ago
README.md	Add badges with pointer to the archived versions	3 weeks ago

README

## Wild\_LIFE — SWHAP Workbench

[archived repository](#) [swah:1-df40af2cf1134180d1e644b3cd2401e6f6da23988df](#)

This repository contains the curated acquisition of the **Wild\_LIFE / LIFE system** developed at the DEC Paris Research Lab, following the [Software Heritage Acquisition Process \(SWHAP\)](#).

The workbench provides both the *Depository* and the *Curated Source Code Deposit*:

- [raw\\_materials/](#)  
Contains the pristine release tarballs ( `life_090.tgz` , `life_091.tgz` , `life_10.tgz` ) and ancillary material (README, license).

About

Reconstructed version history of the Wild Life compiler, following the SWHAP process

Readme Activity Custom properties 0 stars 0 watching 0 forks Report repository

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

See the  
GitHub  
repository,  
flash:



# A call for action

## More memories from Hassan's LIFE?

Open a pull request in the repository at

[https://github.com/SoftwareHeritage/Wild\\_Life-swhap](https://github.com/SoftwareHeritage/Wild_Life-swhap) and add them to:

- raw-materials if it's code
- additional-materials otherwise (papers, pictures...)



Floor is yours!