

---

**The  $\aleph$  (Aleph) project**

Giuseppe Bilotta

**Abstract**

A brief introduction to the  $\aleph$  project, a  $\TeX$  extension providing most  $\Omega$  and  $\varepsilon$ - $\TeX$  features.

**The path**

$\TeX$  was created by Donald E. Knuth more than 20 years ago. Initially, it was supposed to serve one main purpose (providing a high-quality typesetting workbench for Knuth's books), but was general-purpose and powerful enough to be quickly adopted as a more or less standard environment in the scientific community, thanks to its capability to easily typeset complex formulas.

Usage of  $\TeX$  outside the scientific/technical domain has always been confined to niche applications, partly due to the absence of high-level formats like  $\LaTeX$  that are geared more towards nontechnical writing, and partly because  $\TeX$ , in its original design, had very limited support for languages other than English.

Efforts to push the limits of  $\TeX$  have been made, in at least three different directions, by different teams. This led to the creation of multiple, sometimes incompatible extensions of the original engine; we have for example

- pdf $\TeX$ , which gives  $\TeX$  the capability to produce output directly in PDF form, and introduces micro-typesetting capabilities;

- $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$ , based on prior extensions of  $\text{T}_{\text{E}}\text{X}$  which added right-to-left typesetting capabilities, which strives to remove some of the structural limitations of  $\text{T}_{\text{E}}\text{X}$  while maintaining maximum compatibility;
- $\Omega$  (Omega), an effort to bring the  $\text{T}_{\text{E}}\text{X}$  world to up-to-date standards and push it towards a multicultural world.

As mentioned before, not all these extensions are compatible with each other; specifically, while  $\text{pdfT}_{\text{E}}\text{X}$  and  $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$  can be merged in a single program, the changes in  $\Omega$  are so extensive that they put the program in a rather isolated position.

While hopes and desires for a unified  $\text{T}_{\text{E}}\text{X}$  extension have always been present, they have not been pressing because the most common format ( $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ) did not make use of  $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$  extensions, and other formats that did take advantage of those extensions (like  $\text{ConT}_{\text{E}}\text{Xt}$ ) didn't have enough market to be of interest to  $\Omega$  users. Things started to change recently, as the  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  team found the original  $\text{T}_{\text{E}}\text{X}$  more and more restrictive, and  $\text{ConT}_{\text{E}}\text{Xt}$  started spreading, its power and flexibility appealing to users outside the domain of Latin scripts.

### Birth of a new branch

The  $\Omega$  project started with the best of intentions and reached some outstanding results; for example:

- 16-bit registers allow  $\Omega$  to typeset documents too complex to be handled by  $\text{T}_{\text{E}}\text{X}$  (or even  $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$ );
- $\Omega$  introduces the concept of  $\Omega\text{TP}$  ( $\Omega$  Translation Process), a way to transcode texts, therefore allowing the input of text in any script in any encoding, in a font independent way and without the use of active characters; for example, one can write in Greek or Arabic using a plain English keyboard:  $\Omega\text{TPs}$  take care of translating Latin characters (or sequences thereof) into the appropriate Arabic or Greek characters;
- $\Omega$  can typeset text in many directions;  $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$  provided some support for right-to-left typesetting, but  $\Omega$  brings this capability to any combination of direction (left to right, right to left, top to bottom, bottom to top), easily combining these layouts in the same page.

But  $\Omega$  presents some characteristics that can make its adoption a difficult choice with an uncertain future. These are the characteristics of an experimental project with a very broad (maybe too broad) final destination, a moving target where stability is only a secondary if not even tertiary target. While this experimental nature of the project is not intrinsi-

cally negative (on the contrary, it guarantees that the project has enough dynamism to project itself into the future; in fact, it has been what brought  $\Omega$  to its current status) it does hinder a widespread adoption of the tool for production use.

We therefore have a program,  $\Omega$ , that has a lot of potential, and that needs to continue to develop; but this potential has to be somewhat harnessed to make the results available in production use contexts. Therefore, we chose to take the  $\Omega$  code base and use it to start a new project, called  $\aleph$  (Aleph),<sup>1</sup> which could provide the power of  $\Omega$  in an “affordable” manner.

Four main goals were set:

- it had to be stable;
- it had to be fast;
- it had to be powerful;
- it had to be readily available.

### Meeting goals

**Stability and speed** When the  $\aleph$  project was started in late 2002, there were two publicly available versions of  $\Omega$  which could be called “current”: version 1.15 and version 1.23 (which I would call the “old” and “new” version respectively).

The reason why the old one was not taken off the distributions when the new one was published is that the new one suffered from “excessive bloat” which rendered it essentially unusable: processing even a simple document with the new version could take from five to ten times longer than processing it in the old version, memory consumption during the processing was at least twice as much, and the resulting DVI was enormous. These shortcomings of the new version were a result of the introduction of a very powerful enhancement, with interesting potential but that needed to be greatly refined before it could become of common usage.

Because of this, most  $\Omega$  people kept using the old version, which was almost as fast as  $\text{T}_{\text{E}}\text{X}$  (although obviously not as slim). This version, on the other hand, had some extremely serious bugs which caused it to crash whenever overfull boxes were present.

Finding which version to choose to base  $\aleph$  on was not easy. Indeed, the first release of  $\aleph$  (at the time called  $\varepsilon$ - $\Omega$ , Release Candidate 0), which was just a proof of concept that  $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$  could be merged with  $\Omega$ , was available both in a 1.15-based version and in a 1.23-based version, although the officially supported one was the former, therefore with an

<sup>1</sup> The project was originally named  $\varepsilon$ - $\Omega$ , since it provided both  $\varepsilon$ - $\text{T}_{\text{E}}\text{X}$  and  $\Omega$  features.

implied preference for speed over stability, on the assumption that fixing bugs would have been easier than solving the speed/bloat problem. This has later proved to be indeed the best choice.

**Power** The immediate outcome of the third goal was that  $\aleph$  should have provided both  $\varepsilon$ - $\text{\TeX}$  and  $\Omega$  features; since of course  $\Omega$  already provides some of the extensions provided by  $\varepsilon$ - $\text{\TeX}$ , we could limit ourselves to the programming enhancement (extra marks, protected macros, `\scantokens`, etc.).

**Availability** This was probably the most important goal, since it would have been the one that “made the difference” with, e.g.,  $\Omega$  2:  $\aleph$  had to be available in a usable status as quickly as possible; this led, among other things, to the choice of stripping from the  $\Omega$  base the code that dealt with SGML and XML, since it conflicted with the code that implemented the `\middle` primitive in  $\varepsilon$ - $\text{\TeX}$ . Priorities led to this decision.

### History of releases

The first version of  $\varepsilon$ - $\Omega$  was released in December 2002; while the “official” version merged  $\varepsilon$ - $\text{\TeX}$  on  $\Omega$  1.15, a parallel release based on  $\Omega$  1.23 was also made available. That version was no more stable than any of its components, and had an extra few bugs that crept in during the adaptation of the  $\varepsilon$ - $\text{\TeX}$  change files to the  $\Omega$  structure. In particular, it had all the bugs present in  $\Omega$  1.15, which made it scarcely usable due to the major problem with overfull boxes.

The second step was trying to fix the most outstanding bugs coming from the  $\Omega$  1.15 codebase. This led to the first version of  $\varepsilon$ - $\Omega$  testable in production-use environments, Release Candidate 1, around June 2003. This was the version presented at TUG 2004.

Subsequent versions finally officially switched to the  $\aleph$  name; the last public release (Release Candidate 2) also fixed some other significant bugs and started introducing some minor new features (the most important being the `\boxdir` primitive to retrieve/change the direction of a box, a feature backported from  $\Omega$  1.23).

### Status

$\aleph$  is actively developed on the  $\text{\TeX}$  Live repository. A mailing list for discussions concerning the present and future of the project, including both the core program itself and any ancillary tool, is available (`aleph@ntg.nl`).

Development, as always, is focused mainly on

the discovering (and fixing) of bugs, but discussions on possible future features are welcome. Currently, issues and bugs in  $\Omega$ TPs and their interaction with certain  $\varepsilon$ - $\text{\TeX}$  features (namely protected macros and the `\scantokens` primitive) are the most prominent targets.

### Progress

The main focus of  $\aleph$  will always be stability. This means, among other things, that each new release is supposed to be at least as stable as the previous one. A test suite analogous to the TRIP test for  $\text{\TeX}$  is being discussed. Indeed, TRIP proved itself a trusted friend in the discovery and resolution of the most notable bugs coming from the  $\Omega$  1.15 code base, which dealt with overfull boxes and leaders.

Given the stability of  $\aleph$ , it is important to remark that this does not imply a static, frozen behavior (à la  $\text{\TeX}$ ); on the contrary,  $\aleph$  should be considered a foundation on which to build: experimental projects to test the implementation of new features and ideas are welcome, provided they are developed separately; once they reach enough stability to be available for production uses, they might be candidates for introduction in future versions of  $\aleph$ .

### Acknowledgements

I wish to thank

- Donald Ervin Knuth, for providing us all with  $\text{\TeX}$
- John Plaice and Yannis Haralambous, for giving us  $\Omega$
- Peter Breitenlohner and the  $\mathcal{N}\mathcal{T}\mathcal{S}$  team, for giving us  $\varepsilon$ - $\text{\TeX}$
- Idris S Hamid, Alan Hoenig and Hans Hagen for pushing me into attempting the merge and supporting me for all this long time
- all the distribution maintainers for their constant feedback, help and support, with particular thanks to Christian Schenk and Fabrice Popineau for their essential help in getting me started with the coding
- everybody in the  $\text{\TeX}$  world for making it the great community it is

◇ Giuseppe Bilotta  
Dipartimento di Matematica e  
Informatica  
Università di Catania  
viale A. Doria, 6  
95125 Catania, Italy  
`gip.bilotta@iol.it`