



Overview

RISC OS Developments is launching a new initiative to:

- Bring new features to the OS at a time when focus is on 64bit
- Convert modules to C or other high level languages in the process
- Field test new C code alongside these new features
- Built on top of existing RISC OS 5
- Delivered in partnership with other developers
- Using modern development tools to deliver new features to you sooner
- Open Source for the benefit of the whole community
- No additional funding required to get started
- First deliverables in September 2026

Announcement

RISC OS Developments is pleased to announce the RISC OS Renaissance project, beginning September 2026. RISC OS Renaissance represents our response to a number of key challenges facing RISC OS right now, and is intended to be a joined-up approach to handling them:

- 1) The need to move to 64bit
- 2) The limited number of programmers available for working on RISC OS
- 3) The resulting possibility of complete stagnation whilst 64bit is tackled
- 4) The slow pace of OS feature developments in the last few years
- 5) An aging userbase with established hardware who deserve new functionality
- 6) Outstanding disparities between other Operating Systems (including Select/6) and OS 5

The Renaissance project addresses these by introducing more active development focussing on OS features whilst simultaneously converting code to portable, high level languages (primarily C) away from difficult-to-maintain assembly. Users will gain access to new functionality, and as they test and enjoy these new elements - they will be actively helping to test the new C components of the OS.

The idea of the project is to allow you access to new functionality on top of your existing 32bit RISC OS 5, so that your existing software all continues to run, but with the added benefits of new features - much like how Pinboard 2.0 and the RISC OS Developments TCP/IP integrated with the existing RISC OS 5 OS.

The new code produced as part of the Renaissance project will be published via the RISC OS Developments git repositories, and will be licensed wherever possible to be compatible with the existing RISC OS 5 codebase - at worst a few extra lines may need to be included in the Licence text file. This means that the code will be usable by everyone, regardless of existing projects. RISC OS Developments intends to keep a small financial fund available to assist developers wishing to integrate Renaissance project code with the classic ROOL codebase.

The Renaissance project is something we'd like the community to engage in - it need not be limited to RISC OS Developments. We hope that other developers will join this "golden age" of RISC OS progress with new software releases to work alongside the new OS components - the idea is to make RISC OS exciting again. If others wish to join the RISC OS Developments git community, they will be welcome, but github/lab are perfectly good alternatives. Targeted funding is available from ROD for various projects and developers to assist with hardware purchases, development time, debug and development tools etc.

We should stress that this project is not about a "one true way" to 64bit - it welcomes the work of all developers because we believe that only through shared development will RISC OS make the leap to 64bit and retain the necessary backwards compatibility expected by users of existing RISC OS software. It is merely about providing both development of features, and the underlying progress needed towards 64bit in a way that can involve the community in both enjoyment of, and testing of, new code.

Renaissance is about agile development, making things available to the community sooner than has been traditionally possible. Advances in development tools, testing and release-automation should mean that elements of the project can release as soon as September 2026.

Frequently asked questions....

1) Are you forking the OS?

The Renaissance project builds on top of the existing RISC OS 5 codebase, much as Pinboard 2.0 does. You'll need to be booting a traditional RISC OS 5 ROM on your machine before loading the new content. In this way, and by provision of code, we expect to keep forking to a minimum - the code will be available to everyone. However, some users will be more or less comfortable trying new modules and features, and we absolutely understand that some users will prefer not to run the Renaissance modules, just as some users choose not to run Pinboard 2 or the ROD TCP/IP stack. This is about choice, not forking.

It is, however, likely that at some point whole new ROMs will be offered, as there are modules which cannot easily be softloaded. Some will classify this as a fork, but with easily available source code, and common code/modules, the new versions should easily trickle back into the traditional RISC OS 5 tree.

2) Is ROD going to be actively pushing code back into the RISC OS 5 tree?

No - this is not a priority for us, as previous attempts in this area have been both exceedingly slow, and in many cases not possible or unwelcome. Instead our focus is on delivering code as soon as possible to our own public servers and to make prompt releases. However, we're conscious of the need for the code to make its way back, and have set aside funds to facilitate developers able/willing to assist us in this area.

Ultimately we do not wish the Renaissance to be held back by the time-availability of others, and understand the many good reasons behind the existing code acceptance policies. However, the need for increased agility in developing towards 64bit before the end of the decade means we need to focus our internal energies on producing new/updated code.

3) How much will it cost?

The RISC OS Renaissance deliverables are all intended to be provided free of charge to the community, both in terms of executable code, and source.

At present, RISC OS Developments is not asking the community for additional funds (although we always welcome donations). Instead, we would prefer to highlight the on-going business costs that both ROD and ROOL incur year-on-year such as accountancy, taxes, domain/server hosting, electricity etc. ROOL list this under their "general" bounty, and we encourage community members to support that. Similarly, RISC OS Developments accept general donations which may be used for things like this. We would urge the community to support both ROD and ROOL to ensure the healthy survival of RISC OS.

We're not seeking funds directly for Renaissance at the moment because until specific needs arise that we cannot meet, we would prefer the community saved their money to spend within the RISC OS marketplace. It is essential that people buy software, hardware and services from RISC OS vendors to keep the whole scene healthy. Helping RISC OS suppliers stay in business is just as important to the ongoing success of RISC OS - so we would urge readers to support the RISC OS scene as a whole. But yes, please help cover our accountancy bills too.

4) What about timescales?

RISC OS Developments is well aware that RISC OS use skews towards an older demographic than other operating systems. That's why we think it is so important to deliver new content ASAP. We don't want to be waiting 5+ years for things if they can be done (at least in part) in a faster time frame. The plan is to start

delivering new features almost immediately, and build and expand on that month-on-month, with content from as many developers as possible. As a community we need to ensure RISC OS is 64bit ready by the end of the decade (due to hardware availability) but we intend to try and do much better.

5) Where does this leave Moonshot and Bounties?

The Renaissance project should be a huge boon for both Moonshot and Bounties by virtue of providing repositories of visible code that has been tested through the release of feature-led updates for existing users. It may be that not all features are desirable for those schemes, but the availability of this new Open Source code should be a great asset.

There may be crossover between Renaissance and some bounties where we need to update modules and/or applications to achieve new functionality. We apologise in advance for any toes we may inadvertently tread on in our desire to progress forwards. However, as explained previously, we hope community members will be willing to take Renaissance enhancements and feed them back via the bounties (etc) into the main codebase, and ROD will support this.

It is important to understand that both the Moonshot and Renaissance projects have aligned goals with respect to 64bit code - the need to transition RISC OS from assembly to high level languages like C. Where the two differ will most likely be in flavour, management style and feature-focus.

We should be clear - the goal of Renaissance is to lay the foundations of a golden era of RISC OS progress, and we know that this will come from ALL sides - RISC OS Developments, ROOL, and the community of RISC OS developers and companies. The platform has nothing to gain from politics and/or duplication of effort - those things have held us back in the past, but are of no use whatsoever in a 64bit future.

6) Will you be using AI?

We are acutely aware that AI represents a future that few relish. However, both inside and outside of the RISC OS world, it cannot be avoided - much of the new development in 2026 has already been AI assisted.

RISC OS has a real problem - there are mere handfuls of programmers capable/willing to work on the complex challenges that lie before us. It is therefore absolutely essential that we utilise any aides available that demonstrably cut development time and avoid duplication of effort. So yes, that means AI will play a significant role.

HOWEVER (yes, in capitals) AI is only as good as the human hands that guide it, check it and test it. And it is those PEOPLE (again in capitals) that will be key to the success of this project. We have already built up tool chains that comprehend RISC OS, its modules, APIs, Programmers Reference Manuals and even schematics and technical references of motherboards - that understand Toolbox, Templates, the Wimp, sprites, drawfiles, RISC OS fonts and many more “uniquely british” quirks of our beloved OS. This needs to grow, but the progress so far has been exceptional. Work that would take weeks can now be completed in hours or days.

But one thing we have also seen is that people are needed to “stick the landing”. AI is useless without a strong hand on the tiller, and finishing things to release quality is still something that requires the human touch. And that’s where we will be applying it - AI to crunch the grunt-work, and people to make it shine.

We hope you’ll be a part of it - your hands through testing, checking and using the fruits of the project, are exactly what we need to make this a success.

Thanks for reading - we look forward to bringing you a smorgasbord of RISC OS goodness as soon as possible.