

**WHITE PAPER**  
**SUPERCHARGING INNOVATION**

**WHY THE STATE SHOULD RELEASE ITS SOFTWARE UNDER AN ACCESS REGIME**

**BRENDAN SCOTT**  
**OCTOBER 2003**

Contact: Brendan Scott

email: [brendanscott@optusnet.com.au](mailto:brendanscott@optusnet.com.au)

web: [www.members.optusnet.com.au/brendanscott/papers](http://www.members.optusnet.com.au/brendanscott/papers)

**1. SUMMARY**

- 1.1 Governments should institute an access regime to provide access to software components where the development of those components was funded by the State. An access regime can provide net benefits when applied to most software components. Components which ought to remain confidential, and those for which the State has a real intention to commercialise should not be the subject of such an access regime.
- 1.2 The historical absence of access regime structures has led to significant opportunity costs being suffered by the State and its taxpayers. An access regime will provide the opportunity to inject significant value into the State's economy going forward through the creation of a "Software Base". The Software Base can be seeded from existing software development projects and receive ongoing contributions of code from Government software development work. This value will emerge through direct contributions and through efficiency gains.
- 1.3 An access regime can be viewed as a form of public private partnership (PPP) structure in which the Government establishes the fundamentals of a market for access, with private enterprise driving that market going forward.
- 1.4 An access regime over software maximises taxpayer value from software development spend.
- 1.5 The benefits can be accessed at a modest reengineering cost, but without long term maintenance costs.
- 1.6 The Government currently has an informal practice for the sharing of code across Government. An access regime would formalise and extend that practice.

- 1.7 The arguments in this white paper are generally applicable to any Government which engages in the practice of developing software, whether in house or through contractors.

## 2. **OUTLINE OF THE ISSUE**

- 2.1 The design goals of copyright as a legislative system are to grant vendors a monopoly in order to give the market an incentive for the creation of works. Without this incentive free riders can sell works at their reproduction and marketing costs, while the developer must amortise their development costs into their sale price. Copyright also provides an incentive for the vendor to distribute that software. Copyright is specifically targeted not only at the sell side of transactions, but also towards ventures which are: speculative, “big bang”, and non-collaborative.

- 2.2 Not only are the incentives mentioned in section 2.1 usually irrelevant where the development is funded by the Government, they can be actively counter productive. This is because copyright promotes a focus on ownership and a bipolar – “I (ie. Government) own/you (ie Vendor) own” – approach rather than on how to maximise taxpayer benefits from the product going forward. This can lead to a suboptimal result even where the State owns the copyright – *copyright ownership without an access regime can be a loss for Government* (usually Vendor ownership is a greater loss, although sometimes Vendor ownership is a win). This is because, despite the best of intentions, Government rarely, if ever, then acts on the sell-side of any transactions in relation to the work. It owns the copyright, but makes no use of it. That ownership locks the rest of the community out of realising the true value of the product.

- 2.3 The last decade has seen the rise of inventive uses of the copyright monopoly to spur innovation. The most successful of these turn commonly understood paradigms of the copyright monopoly on their head, providing solutions for the customer side of a software development transaction which provide more benefits, and fewer detriments than apportioning copyright to either customer or vendor and leaving it at that. It is these new models that underlie the concept of an access regime for software.

## 3. **PRACTICAL CONSEQUENCES OF COPYRIGHT OWNERSHIP**

- 3.1 In this section we demonstrate (by reference to an analysis of copyright ownership in the absence of an access regime) that the traditional model leads to suboptimal results for the “buy side” of a copyright transaction.

3.2 The following consequences flow from ownership of the copyright in a piece of software by a vendor:

- (a) the vendor can amortise development costs across all users (but the total aggregate cost to the community for the product is increased as more profit is able to be extracted);
- (b) the vendor can make strategic decisions about and control product's direction – ie can move product forward;
- (c) the Government can take the benefit from involvement of broader user base;
- (d) the Government is exposed to vendor lock in – with attendant high switching costs in the event of unresponsive vendor;
- (e) the Government (ie the taxpayer) funds vendor's R&D from which vendor subsequently profits;
- (f) the Government is captured and held hostage to the vendor's vision for the product (related to (d) lock in above);
- (g) there is a lack of competition for ancillary services (maintenance, training etc) so ability to extract monopoly rents. This is because any ancillary services in relation to the software must ultimately be authorised by the vendor as a result of their copyright monopoly.
- (h) there is a tendency towards data formats to be specifically structured by the vendor to promote vendor lock in. These will provide additional anti-competitive effects/lock in;
- (i) the insolvency or corporate restructuring of the vendor can mean the end of product support (escrow of source code makes this survivable but unpleasant - see section [5.2(d)] below);
- (j) the vendor is able to sell the product to other members of the community.

3.3 The flip side to consider is that situation where the Government retains copyright in the software produced but does not make it available under an access regime. In this situation, the following are likely consequences:

- (a) both the Government and the vendor are locked out of subsequent development. Unless the Government makes a positive decision (and expends requisite effort) to move the product forward it will stagnate.
- (b) there is less ability for the vendor to provide value to other customers/leverage off the skills and knowledge they have acquired during the development process (ie sub optimal contribution to general economy);
- (c) the product is quarantined in practice, so (i) other users can't leverage off value of product; and (ii) other users can't contribute to improving the product;
- (d) the insolvency of the vendor is survivable, but unpleasant. It is unpleasant because there is no general community with an interest in acquiring support in relation to the product, Government needs to skill up from scratch and begin from a standing start;
- (e) there is a tendency towards the evolution of closed standards in practice because no visibility of standards is provided;
- (f) mitigating against all of these is the *theoretical* potential for Government to commercialise the product, offset development costs, and develop a community of participants which will address the other issues. In *practice*, Government never does this (for a variety of reasons, not least of which is that commercialisation is difficult, time consuming and requires skills that Government does not usually have). Often such developments are fundamentally unsuited (as a package) to productisation and commercialisation (being too specific to the needs of Government).

3.4 There is another option we have not considered, that of joint copyright holding. Unfortunately this provides the worse outcome for both parties. Joint holding means that each party effectively has a veto over what is done with the software. In practice the software is encumbered with all of the detriments of both scenarios set out above, but endowed with none of the benefits.

3.5 What the Government (and any customer) really wants is the benefits of community involvement in the product without the downsides of vendor lock in. In reality who owns copyright is irrelevant. What is important is a workable "access regime" for the product.

#### 4. **KEY AIMS OF AN ACCESS REGIME**

4.1 The main purposes of an access regime are:

- (a) to create a free market for all services related to a software product.
- (b) to reduce Government wastage by encouraging reuse of software (either on a whole of product or individual component level).
- (c) to minimise opportunity cost from software hoarding incurred by the State and its residents.
- (d) to maximise opportunities to multi-source.

## 5. ELEMENTS OF AN ACCESS REGIME

5.1 An access regime is fundamentally a grant of a licence on terms which include the following:

- (a) Rights of access to code;
- (b) Rights to distribute code but source code must be distributed concurrently or otherwise made available;
- (c) Rights to use code;
- (d) Rights to modify code;
- (e) Rights to distribute modifications of code;
- (f) Obligation to distribute on terms of the access regime (ie. if a person distribute modifications, then they must contribute them to the access regime's code base). This term is critically important to the proper operation of an access regime. In its absence an access regime can promote incompatible "forking", which in turn leads to higher administration and compliance costs, undermining the value of an access regime. At its worst access with out a contribution requirement can become just another vendor subsidy.

5.2 In each case, subject to 5.1(f) and the specific limitation in 5.1(b), with no limits on the rights granted – eg anyone may access, modify, distribute etc. Access, use, modification etc can be for any purpose. None of these rights are conditional on any other circumstances.

5.3 The terms of an access regime may (and often do) include provisions relating to a disclaimer of liability for the use or distribution of the code.

5.4 In practice these access rights need to be supported by a means of storing the code and providing visibility of the code base.

5.5 An access regime can be successful despite only minimal or modest marketing of the code and the regime. This arises because, given the minimal compliance and transaction costs, viable code effectively sells itself.

## 6. **COSTINGS**

6.1 The main cost components of an access regime are

(a) **Legal** – to review and adopt an appropriate licence for the access regime, ensure development agreements give sufficient rights to allow access regime contributions.

(b) **Procedural/Administrative** – creating methodologies to determine when a component has been closed off, determining whether it meets qualifying criteria, documenting and transferring a copy of the component into the Software Base.

(c) **Hosting/Access** – provision of a common archive location for contributed code. However, it is possible to seek private capital for this cost (see, for example, [www.sourceforge.net](http://www.sourceforge.net)).

6.2 None of these heads will involve significant up front or ongoing costs to Government.

## 7. **THE LONG TERM BENEFITS OF AN ACCESS REGIME**

7.1 **Greater value contributed to economy.** first the component of monopoly rents from copyright are stripped out of development pricing; and second lower barriers to acquisition mean greater use. These benefits arise because an access regime introduces competition in relation to the software, thus stripping out monopoly rents over time (see section 7.4). The State effectively multiplies the value of its investment by the number of users who adopt the application.

7.2 **Better Localisation of Expenditure:** An access regime effectively requires software services to be provided as services, rather than products. Ordinarily development and maintenance operations are geared to creating and servicing a “one size fits all” approach. Under an access regime a broader ecology is sustainable. Furthermore service providers who are physically located near to the end customer are more likely to understand that customer’s circumstances and be responsive to their needs. As such Government expenditure on software expenditure tends to be localised as local businesses will tend to be able to provide the best service.

- 7.3 **Reducing Digital Divide, Encouraging community and participation:** An access regime by its nature has no barriers to participation and no barriers to accessing the benefits of software. Where appropriate, components can be repackaged for use by disadvantaged in the community. It also provides a means to empower the disadvantaged and provide them with an opportunity for meaningful participation in society.
- 7.4 **Greater contestability and competition:** access regime increases contestability of a number of markets – eg software development, maintenance, training, design, installation, customisation. Increased contestability means greater competition. For example, a broader range of people will be in a position to tender for software services under an access regime.
- 7.5 **Will emphasise code quality:** Vendors need to compete on service and quality rather than monopoly control over a key input (or specific knowledge gained from previous work). Hence, the market will produce better quality outcomes. See also the next point (code auditing).
- 1.1 **Better code auditing:** As the code is visible, third parties will audit the code for their own purposes. This will lead to better quality code being generated (as a Vendor strategy to avoid criticism/advertise their services going forward) and better error identification in existing code.
- 1.2 **Standardisation:** The access regime provides code visibility. This, coupled with the fact that development costs will be lower for interoperable code provides a strong incentive for code standardisation, without mandating any standards to the market. Note: this standardisation occurs not only across Government, but will also tend to encourage standardisation across the State, so efficiency gains are multiplied. This point relates to coding methodology, the next point (open standards) relates to data storage and interchange formats.
- 7.8 **Open standards:** A further consequence of the access regime is that any standards which result are necessarily open standards. This (a) increases portability and therefore lowers cost going forward; and (b) increases community access to information. Again, this standardisation will occur naturally through the operation of the market, without Government mandating the standard. Further, open standards better preserve data resilience and accessibility over time and across applications.
- 7.9 **Modular design:** An access regime preferences modular software development. This software design method is recognised by software engineers to be superior to monolithic design.

- 7.10 **Creation of a software infrastructure/common libraries:** An access regime means visibility of code. Over time this will result in a common code base that can be used by programmers and businesses for their software projects. In addition to the efficiencies this provides, it also means that skilling up costs will not be at the expense of the Government (because the skills will not be Government specific).
- 7.11 **Better focus development effort:** An access regime will provide access to commonly used code components. This means that developers need not constantly reinvent the wheel (development costs will be replaced by search costs which are lower).
- 7.12 **Self supporting:** The access regime will operate without Government oversight. Products and components will compete for market share based on functionality and useability. Those which do not have sufficient of either will fall from prominence. However, they will never be lost, so if new uses arise old components can be revived through the action of the market.
- 7.13 **Real value provided to business:** If a piece of software is developed for \$1 million for internal use by a Department, but has a use value to individual businesses of \$10, if it is used by 100,000 businesses, the access regime contributes \$1 million of value to the economy without detracting from the Departmental use. Businesses can also take the benefit of the other benefits provided by the regime, such as standardisation and increased competition. We note that this example, and that is section 7.14 are necessarily simplistic but serve to illustrate the point.
- 7.14 **Value multiplying:** If 1 of those businesses improves the software and releases those improvements – say by increasing the use value of the software to \$11, that incremental improvement can be accessed by all of those businesses. If 10 of those 100,000 each make a \$1 improvement, the value to the State has doubled – this from an initial seed with no further Government involvement. If the Department which initially developed the product is able to use these improvements, the value to the State has effectively tripled. Note: the access regime permits businesses to undertake software development with internal resources so development occurs on a lower cost base than the original development. The regime also permits participation even for small, or minor incremental additions which would be excluded by the transaction costs involved in a traditional model.



- 7.15 **Acts as an investment in businesses for complementary products and services:** Where an access regime results (through community participation over time) in a market ready product this effectively acts as an investment in complementary products and services – a suite of business software available under an access regime would increase sales of (for example) hardware to run that software or of training services to use the software.
- 7.16 **Reducing Software Piracy:** An access regime creates a structural barrier to software piracy. Not only does it tend to create a piracy proof environment, it reduces incentives for piracy more generally.
- 7.17 **More efficient use of resources:** This is another way of restating the cumulative effect of the other benefits mentioned above. At a macro level an access regime means less duplication of effort, more open standards (and therefore better interchange of information), and reduction of vested interest in the status quo – in other words, a lessening of restrictions on innovation. An access regime grants benefits to the State in a similar way to those of a major infrastructure project (calculate all the money spent on software development by the State within a calendar year). However, (a) they will come with minimal ongoing maintenance costs; (b) it can be seeded incrementally and at the State’s own pace (although the earlier it is done, the quicker access to benefits); (c) it has no environmental impacts; (d) once the kernel is built, it will grow itself – organically responding to the needs of the residents of the State; (e) it is recycling and repurposing assets already paid for; (f) it involves no element of speculation – innovations are made by contributors who have a direct need for the innovation; (g) it has no marketing and distribution overheads (those which are incurred are incurred by the members of the community); (h) it provides visibility of what code is available; and (i) it will not need the State to act as guarantor in order to receive private sector buy in.

## 8. **EXAMPLE**

- 8.1 Government department D contracts vendor V to develop a software product P. P has features X, and Y. Assume P is subject to a software access regime.
- 8.2 Business B1 sees P and believes they can use a product similar to P in their business. B1 can contract V to add feature Z to P, creating P2. B1 could also, if they chose, contract V2 or V3 to do the same work. In order to secure the work V, V2 and V3 would need to compete on service and quality, **not** on monopoly control over an input (P). B1 is not required to go to the expense of creating P from scratch as it can be reused. Because P2 was developed under the access regime, if B1 makes feature Z publicly available, it must do so on the terms of the access regime. That is, feature Z

is contributed back into the pool for others, including D, to use but at no development cost to D.

- 8.3 Vendor V2 is developing a software product Q. It becomes aware of P and realises that while P and Q share little common high level functionality, the means that P uses to perform its functions can be used to help Q achieve its functions. V2 is able to reuse some parts of P in developing Q, reducing the overall cost to develop Q.
- 8.4 D is dissatisfied with V's service. Because of the access regime D can contract with vendors V2 or V3 to support P. Note, this would still be possible if D only owns copyright – but more expensive. Without an access regime, P is quarantined in a dark box somewhere. Hiring V2 or V3 requires time and money for them to skill up in order to service P. Under an access regime, there is an existing community of users of P so V2 and V3 have already skilled themselves in order to provide services to that community, leading both to faster reaction times and lower costs.
- 8.5 Trainer T hears about P and would like to investigate the opportunities for training users on P. T's ability to access P allows T to assess and become familiar with P and to set up a training lab at the cost of hardware alone. T is better able to set their prices at market, without the need to pay premiums for access to P or to licence P in a training environment.

## 9. **COMMON QUESTIONS**

- 9.1 Doesn't this mean giving up copyright ownership? No, an access regime requires copyright ownership to make sense/be enforceable.
- 9.2 Are you sure? Aren't you saying I should put the software into the public domain? Definitely not. Putting software into the public domain will not achieve any of the objectives of an access regime because there is no requirement for subsequent pooling of innovation.
- 9.3 Aren't I getting something for nothing? No. Price reductions occur through:  
(i) efficiency gains resulting from greater visibility, modularity and standardisation;  
(ii) increasing contestability in each ancillary market relating to the product; and  
(iii) removing above market rents enjoyed by vendors as a result of monopoly control over essential inputs (the product and its source code).
- 9.4 Will this kill software entrepreneurship? No, an access regime will not affect existing copyright incentives. By fostering modular code and creating a standardised code base it will provide a code infrastructure for entrepreneurs to leverage off (ie. they

need not continually reinvent the wheel and can focus on inventing the axle). It will also provide additional tools for SMEs – the traditional powerhouses of entrepreneurial spirit.

- 9.5 Should the Government be licensing for a fee rather than just providing access? No, practice has demonstrated that the Government is ill equipped to operate as a software vendor. In addition: search and transaction costs typically exceed the use value of the software in question to a potential user; the strength of an access regime results from leveraging off incidental use of components and the subsequent contribution of incremental improvements; licensing for fee would render access to the software uncommercial for users (who probably only want one or two components and therefore a price for the components bundled in the form of a product will be uncommercial); licensing for a fee will also prevent the subsequent use and development of secondary markets – which is where substantial benefits of an access regime arise.
- 9.6 If an access regime is so great why hasn't it been tried on other products or industries before? An access regime is useful for copyright works because, as a general rule, they are (a) durable – they don't wear out and (b) non rival – use by one does not inhibit use by another (ie. in economics speak they are “public goods”). In relation to software, it is also modular (so a user can derive value from using a small part, not necessarily through using the whole). This alignment of characteristics does not occur in other industries.
- 9.7 Won't people just pirate the code? The history of access regimes indicates remarkably low levels of unlicensed copies of software the subject of an access regime. Access regimes appear to create structural barriers to piracy and, at a broader level, reduce the incentives for piracy.
- 9.8 Won't this require a lot of effort on the part of Government to implement? No, most of the costs are born by those making use of the access regime. There are costs associated with housing software and with administering entry of the software into the regime. However, by and large the community administers itself.
- 9.9 Isn't the State giving up something? No, by adopting an access regime the State trades a theoretical and unused right (ie. the right of sale of the software) for a tangible, but indirect benefit (greater value in the economy in the first instance and, later, improved software). Further, the tangible benefits are self supporting and compound over time. It is better to unshackle software and let others work it rather than locking it away in the vain hope of one day selling it.

9.10 Who will be clear losers from an access regime? No one. Software companies which rely on the existence of a Government monopoly as part of their business model may appear to be *prima facie* disadvantaged. However, they will derive benefits through the access regime, so it is not clear whether it is a net loss in their case.

## 10. **WHERE IS AN ACCESS REGIME NOT APPROPRIATE?**

10.1 There are a number of circumstances in which an access regime would be inappropriate. However, in many of these cases they would not apply to the whole of a software development but, rather, only to specified components.

10.2 **Community involvement:** An access regime relies on broader community participation in the use of the product (ie broader than the Department that commissioned the work). Where a work is so tailored to idiosyncratic needs of a given Department, the same leverage will not be available. Query whether individual components could be reused even in the absence of a market for the bundled product.

10.3 **Confidentiality:** An access regime necessarily requires openness in relation to the product. Aspects which need to remain confidential should not be subject to an access regime. However, confidentiality requirements will not apply to all components in a software product even if the product as a whole is confidential.

10.4 **Sale:** Where the Government actually gears up to make a profit through selling the product. Mere inchoate intention to sell is not sufficient as it never translates into practice. It is also not appropriate to make a distinction between private and public enterprise (ie grant an access regime to public enterprises, but not to private ones). If this is done it undermines many of the benefits of such a regime (such as standardisation, open standards and incremental innovation).

## 11. **CONCLUSION**

11.1 The State currently incurs large opportunity costs through failure to reuse software developed for the Government. An access regime will provide a means of minimising these costs while providing tangible benefits to businesses within the State. In the long term an access regime will result in the establishment of a Software Base which can, itself drive growth in ICT industries within the State.