



# Porting: The Business Imperative

Mindfire Solutions  
[www.mindfiresolutions.com](http://www.mindfiresolutions.com)

December 5, 2001

<b>PORTING: THE BUSINESS IMPERATIVE.....</b>	<b>1</b>
INTRODUCTION .....	1
BENEFITS OF PORTING.....	1
• <i>ROIP (Return on Intellectual Property)</i> .....	2
• <i>Revenue stability</i> .....	2
• <i>Positive fall-out on primary platform</i> .....	2
• <i>Competitive advantage</i> .....	3
• <i>Buzz</i> .....	3
• <i>Cleaning up code base</i> .....	3
IS IT THAT SIMPLE?.....	3
• <i>Porting costs</i> .....	3
• <i>Code maintenance</i> .....	3
• <i>Support costs</i> .....	3
• <i>Marketing?</i> .....	3
OPTIONS IN MOVING TO A NEW PLATFORM .....	3
• <i>Platform emulator</i> .....	3
• <i>Porting: The In-house Option</i> .....	3
• <i>Porting: The Out-source Option</i> .....	3
SELECTING AN OUT-SOURCE PARTNER .....	3
CONCLUSION .....	3

## Introduction

Porting is a significant, if oft-neglected, strategy for software businesses. Multi-platform products have multiple benefits, while bringing with them possible headaches. The porting process itself carries a few risks. In this white paper, we discuss the pros and cons of porting. We also consider some pitfalls, and finally outline options available for porting.

## Benefits of Porting

The thought of porting may have crossed your mind often. I will try to present benefits that arise from porting.



- **ROIP (Return on Intellectual Property)**

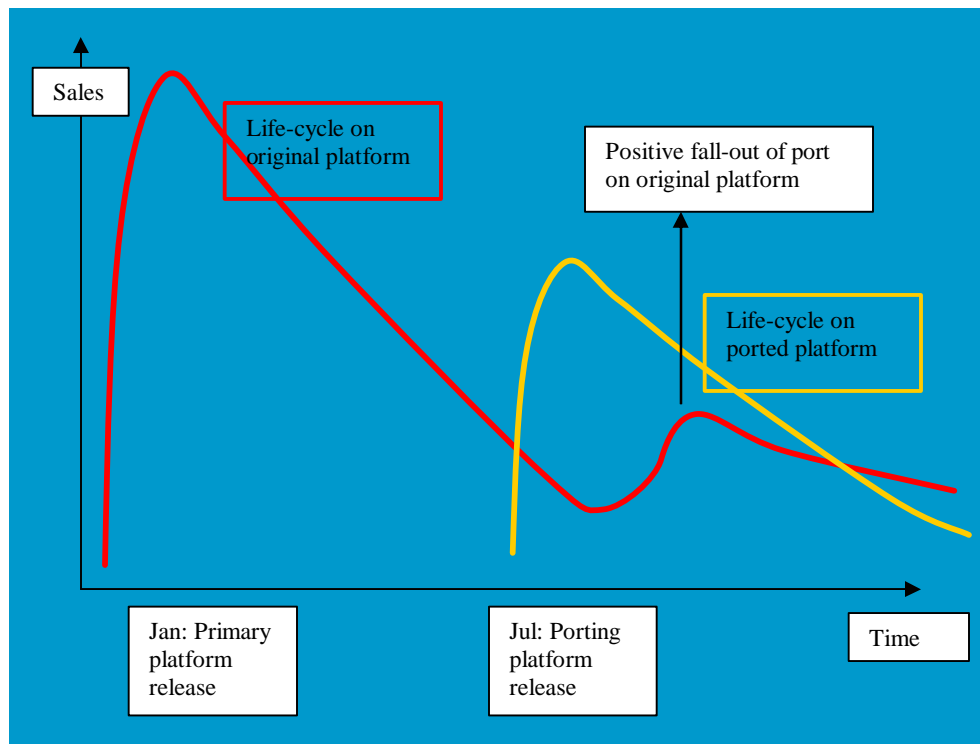
The fundamental imperative behind porting is simple: how can you multiply the effect of your ideas, your intellectual property, your primary-platform development effort, and your investment?

Product development cycles involve majority effort spent in ideas, design, coding, testing and marketing. Defining a product could itself take weeks, if not months. Much of this effort is re-used in porting. You create an entirely new product using incremental investment.

You may define success as sales revenue, profitability, user base, or any other business parameter. In each of these cases, porting your product usually achieves growth objectives.

- **Revenue stability**

Porting a product opens up opportunities for improved revenue stability. Staggered platform releases could drive revenue growth when the revenues of your primary platform reach steady state.



- **Positive fall-out on primary platform**



Every new platform you release your product on, usually has a positive impact on the first platform. This may be due to renewed interest in the product, a greater word-of-mouth experience, or simply side-effects of marketing for the new platform.

Very often, product selection decisions are based on the availability on multiple platforms. Corporate IT departments want alternative platform availability as safety for their investment. Home users typically have to deal with different platforms at home and at work, and would want their applications to work seamlessly at both places. Porting would, thus, swing decisions in your favor – both on original and ported platforms.

Whatever the cause, this is a welcome boost to your product sales on the original platform.

- ***Competitive advantage***

Porting would generate a competitive advantage for you, and allow you to establish a beachhead into the new platform. It establishes your clear commitment to the product, and puts in a significant plus in side-by-side comparisons with the competition.

- ***Buzz***

Porting a product offers opportunity to create buzz around your product. Existing customers share in the excitement, and you may make press releases, web site announcements etc. to increase awareness about your product. It also means a time when you can keep in touch with existing customers through a newsletter.

- ***Cleaning up code base***

Surprisingly, porting a product could actually improve its code quality. When a product is coded against a specific platform, one generally tends to forget good coding practices such as abstraction, layering, modularity (class boundaries and encapsulation) and parameterization. Architecture and design may also lack in finesse – the effect of coding to release against a deadline. Algorithmic optimizations may also be possible. Solutions to thorny known bugs on the original platform may be accidentally discovered during porting!

As a preparatory step to porting, existing code will always be studied. This is the first step where the porting developer is trying to build a model of your existing code. Further, porting requires a study of code in detail. Ongoing work also means that code receives a clinical inspection from the porting developer.

Thus, porting may actually result in strengthening your code base and cause improvements in the original code too.



## **Is it that simple?**

So, with everything going for it, what stops you from porting? What are the negatives, what are the pitfalls to watch out for?

- ***Porting costs***

The porting effort may be prohibitively expensive. Several factors (state of current code, development tools and languages used, libraries and components used etc.) may drive the costs of porting. The amount of time required for the port may also be too high, while availability of skills may be an issue in other cases.

- ***Code maintenance***

After porting, you are likely to end up with multiple code bases. You should try to achieve your port with as much abstraction and cross-platform code files as possible. Some files can be platform-specific and provide core services, whereas functionality and processing code is cross-platform. Project files (make-files) and conditional compilation may be used to compile alternative code streams on different platforms.

Such techniques have one objective: minimize costs of maintaining multiple code bases. These are especially relevant when developing the next version of your product (likely to be developed on all your platforms now).

- ***Support costs***

You may face support issues, in case you don't have in-house expertise in the platform ported to. If your newly-acquired Linux customer asks you about problems during installation, or normal operation, you need to be able to respond.

You also have to maintain consistent support policies with your earlier platform. This could impose a burden to support your new customers, whereas user base may be relatively smaller. Your commitment may also include ability to respond to bugs fast.

Porting should take into consideration support costs and issues when deciding on development strategy. For example, separate file formats would mean users cannot carry work across platforms, even though your product runs on both platforms. This will lead to dissatisfaction as well as numerous support calls. A common file format would be an appropriate choice, though it may imply increased development effort.

- ***Marketing?***

You have been successful on your earlier platform, so you know what it takes. However, every platform has its own channels of distribution, and you may need to establish yourself on your target platform.



## **Options in moving to a new platform**

- ***Platform emulator***

The simplest option in selling to users on another platform is to ask them to take the burden. Simply ask them to buy/get a platform emulator which will simulate one platform on another. This approach is good as an immediate fix. There are a bunch of emulators available, and they do a pretty good job.

Needless to say, this approach can only be treated as a quick fix, and for temporary gains. It might also have downside in your market reputation, and the emulator approach could generate unknown bugs.

- ***Porting: The In-house Option***

Native software is the only sustainable approach to having happy customers and leadership position on any platform.

You may prefer to do in-house development on the target platform, when you have requisite expertise with you. Even when you don't have the expertise, you may consider getting expert manpower on a contract basis, or train your own team in the required platform. However, such skills may not be as deep as you would want, and may take longer than you would desire. It may also detract from your mainline development.

If you have idle manpower with expert development skills in other platforms, you wouldn't be reading this article. You would be well under way. If you are not, please start now!

- ***Porting: The Out-source Option***

Porting is usually a clean, encapsulated operation and may be out-sourced to an expert party.

Out-sourcing would have a few problems: no development of in-house team, which may raise product support and future development issues. It also means releasing source code to an external party, which needs careful screening of out-sourcing partners.

However, some companies provide mature porting services. Such companies would be ideal partners for out-sourcing your porting projects.

## **Selecting an out-source partner**

Selecting an out-source partner is like selecting any other out-sourcing vendor and follows the same process. However, there are a few things you should specifically look out for:



1. Expertise in *both* platforms (original and target)
2. Expertise in porting, specifically
3. Provides good prices
4. In case you are not developing in-house expertise, you need the partner to provide:
  - a. Code base integrity and maintenance assurance
  - b. Commitment to work on future versions
  - c. Ongoing debugging support
  - d. Ongoing user support

## **Conclusion**

Porting may not be as infeasible as one may believe, without investigation. Proper time-and-cost equations make ports not only possible, but highly desirable. The many benefits of porting range from tangible to intangible, and software companies should seriously consider porting as part of their product strategies.

---

*Mindfire Solutions is an off-shore software services company in India. Mindfire possesses expertise in multiple platforms, and has built a strong track record of delivery. Mindfire passionately believes in the power of porting and its many advantages for software product companies.*

*We have developed specialized techniques to make porting efficient and smooth, and to solve the issues specific to porting. We offer core development and QA/testing services for your porting requirements, as well as complete life-cycle support for porting.*

*If you want to explore the potential of porting, please drop us an email at [info@mindfiresolutions.com](mailto:info@mindfiresolutions.com). We will be glad to help you.*

*To know more about Mindfire Solutions, please visit us on [www.mindfiresolutions.com](http://www.mindfiresolutions.com)*

---