FEDERAL COURT OF AUSTRALIA

IPC Global Pty Ltd v Pavetest Pty Ltd (No 3) [2017] FCA 82

|  |  |
| --- | --- |
| File number: |  |
|  |  |
| Judge: | **MOSHINSKY J** |
|  |  |
| Date of judgment: | 10 February 2017 |
|  |  |
| Catchwords: | **COPYRIGHT** – infringement – literary work – computer program – software for materials testing equipment – where executives of applicant resigned and established rival company – where one of the executives retained copy of applicant’s source code – where executive copied that source code onto computer provided to rival company’s computer programmer – where programmer copied source code from system library of applicant’s software – where source code related to message format and data structures – whether respondent infringed copyright in applicant’s software – whether respondent’s software reproduced a substantial part of the applicant’s software  **INTELLECTUAL PROPERTY** – confidential information – computer program – software for materials testing equipment – where executive of applicant received copy of source code – where executive resigned and provided applicant’s source code to programmer of rival company – where programmer referred to and copied applicant’s source code in course of writing software – whether applicant’s source code was confidential – whether source code received by executive in circumstances imparting obligation of confidence – whether ‘springboard’ benefit gained |
|  |  |
| Legislation: | *Copyright Act 1968* (Cth), ss 10, 14, 22, 29, 31, 32, 36, 115 |
|  |  |
| Cases cited: | *Australian Medic-Care Company Ltd v Hamilton Pharmaceutical Pty Ltd* (2009) 261 ALR 501  *CA Inc v ISI Pty Ltd* (2012) 95 IPR 424; [2012] FCA 35  *Coco v AN Clark (Engineers) Ltd* [1969] RPC 41  *Del Casale v Artedomus (Aust) Pty Ltd* (2007) 165 IR 148; [2007] NSWCA 172  *IceTV Pty Ltd v Nine Network Australia Pty Ltd* (2009) 239 CLR 458  *IPC Global Pty Ltd v Pavetest Pty Ltd (No 2)* [2016] FCA 1332  *JR Consulting & Drafting Pty Ltd v Cummings* (2016) 329 ALR 625  *Leica Geosystems Pty Ltd v Koudstaal (No 3)* (2014) 245 IR 422; [2014] FCA 1129  *Nexus Adhesives Pty Ltd v RLA Polymers Pty Ltd* (2012) 97 IPR 160; [2012] FCAFC 135  *O’Brien v Komesaroff* (1982) 150 CLR 310  *Optus Networks Pty Ltd v Telstra Corporation Ltd* (2010) 265 ALR 281  *RLA Polymners Pty Ltd v Nexus Adhesives Pty Ltd* (2011) 280 ALR 125  *Smith Kline & French Laboratories (Auft) Ltd v Secretary, Dept of Community Services and Health* (1990) 22 FCR 73  *Wright v Gasweld Pty Ltd* (1991) 22 NSWLR 317  *Zomojo Pty Ltd v Hurd (No 2)* (2012) 299 ALR 261 |
|  |  |
| Date of hearing: | 14, 17, 19, 20 October and 7, 8 November 2016 |
|  |  |
| Registry: | New South Wales |
|  |  |
| Division: | General Division |
|  |  |
| National Practice Area: | Intellectual Property |
|  |  |
| Sub-area: | Copyright and Industrial Designs |
|  |  |
| Category: | Catchwords |
|  |  |
| Number of paragraphs: | 222 |
|  |  |
| Counsel for the Applicant: | Mr R Cobden SC with Ms E Bathurst |
|  |  |
| Solicitor for the Applicant: | Bird & Bird |
|  |  |
| Counsel for the Respondents: | Mr CP Thompson |
|  |  |
| Solicitor for the Respondents: | Stephens Lawyers & Consultants |

ORDERS

|  |  |  |
| --- | --- | --- |
|  | | NSD 1203 of 2015 |
|  | | |
| BETWEEN: | IPC GLOBAL PTY LTD (ACN 168 265 156)  Applicant | |
| AND: | PAVETEST PTY LTD (ACN 160 146 083)  First Respondent  CON SINADINOS  Second Respondent  ALAN JOHN FEELEY (and another named in the Schedule)  Third Respondent | |

|  |  |
| --- | --- |
| JUDGE: | MOSHINSKY J |
| DATE OF ORDER: | 10 FEBRUARY 2017 |

THE COURT ORDERS THAT:

1. By 4.00 pm on 24 February 2017, each party file and serve proposed minutes of orders to give effect to these reasons and a short outline of submissions in support of those orders.
2. By 4.00 pm on 3 March 2017, each party file and serve any short outline of submissions in response.
3. Liberty to apply.

Note: Entry of orders is dealt with in Rule 39.32 of the *Federal Court Rules 2011*.

REASONS FOR JUDGMENT

MOSHINSKY J:

## Introduction

1. The applicant (**IPC Global**) makes equipment for testing materials such as asphalt and other construction materials. The operator inputs commands to the testing equipment and results are reported back. Those functions are mediated by software, with which the operator interacts, and firmware, which mediates between the software and the equipment.
2. IPC Global’s software, which resides on the customer’s personal computer, is known as Universal Testing Software or UTS (the **UTS software**). IPC Global’s firmware resides in a controller known as Integrated Multi Access Controller or IMACS. The firmware has generally been known as the **IMACS firmware**.
3. The UTS software and the IMACS firmware exist in both object code and source code. Object code is machine-readable. Source code is human-readable (with appropriate skills).
4. On 31 March 2014, pursuant to an asset sale agreement (the **Asset Sale Agreement**), IPC Global acquired all of the assets of Industrial Process Controls Ltd (**Old IPC**). (Unless otherwise indicated, the expression “IPC Global” is intended to include Old IPC.)
5. Until mid-2012, the second respondent (**Mr Sinadinos**) was the Chief Executive Officer of Old IPC and the third respondent (**Mr Feeley**) was the Manager of Research & Development at Old IPC. Between 1999 and 2003, Mr Feeley’s engagement by Old IPC was pursuant to a contract of employment. From July 2003 to mid-2012, he worked for Old IPC pursuant to a consultancy agreement (the **Aleuta Consultancy Agreement**) between Old IPC and the fourth respondent (**Aleuta**), which was his family company. The documents constituting the Aleuta Consultancy Agreement have not been located.
6. In mid-2012, Mr Sinadinos and Mr Feeley resigned from IPC Global and, shortly afterward, they established the first respondent (**Pavetest**). Subsequently, Pavetest launched a range of testing products that compete with IPC Global’s products. Pavetest’s software and firmware in relation to the relevant Pavetest products are known as the **TestLab software** and the **CDAS firmware**. (The reference to CDAS is a reference to the Control and Data Acquisition System, which is the name of Pavetest’s controller.)
7. At the time Mr Feeley resigned from IPC Global he had copies of the UTS software on his computer. He retained those copies after his resignation and subsequently provided them to Mr Li, a computer programmer engaged by Pavetest to write the TestLab software, by copying them onto a computer provided to Mr Li. Mr Li referred to the UTS software in the course of writing (what may be referred to as) version 1 of the TestLab software. The evidence in this case shows that some parts of the source code of the TestLab software (version 1) are identical to parts of the UTS software, and that other parts are similar.
8. Subsequently, following the commencement of this proceeding, Pavetest instructed Mr Li to make changes to the TestLab software to remove the parts that were identical or similar. The version he produced is referred to as version 2 of the TestLab software.
9. In this proceeding, IPC Global has alleged that Pavetest infringed its copyright in the UTS software and the IMACS firmware and that Mr Sinadinos and Mr Feeley authorised the infringement; that Mr Sinadinos and Mr Feeley breached duties of confidence towards IPC Global in relation to the UTS software and IMACS firmware; and that Mr Feeley and Aleuta breached contractual duties of good faith and fidelity towards IPC Global.
10. On the third day of the trial of this proceeding (on issues of liability only), the respondents conceded liability in relation to the firmware part of the case, both as regards copyright and breach of confidence. The trial continued in relation to the software part of the case. However, IPC Global submitted that it was appropriate for final orders to be made in relation to the copyright aspect of the firmware part of the case in advance of publication of my reasons on the software issues. It was arranged that, in conjunction with the making of closing submissions on the software part of the case, the parties would address on whether final orders should be made, and if so what orders, in relation to the copyright aspect of the firmware case. Following those submissions, I published reasons and made orders in relation to the copyright aspect of the firmware part of the case: *IPC Global Pty Ltd v Pavetest Pty Ltd (No 2)* [2016] FCA 1332.
11. In relation to the software part of the case, there was a narrowing of the issues both in the lead up to, and during the course of, the trial. In particular:
    1. In the respondents’ opening written submissions, they stated that it was not disputed that source code of the UTS software was used and copied in developing the TestLab software.
    2. The respondents also stated in their opening written submissions that they did not contest that:  copyright subsists in all of the versions of the programs in issue; IPC Global owns the copyright in the relevant versions of the UTS software; and, if and to the extent that the primary liability of Pavetest for copyright infringement is established, authorisation liability of Mr Sinadinos and Mr Feeley will also be established.
    3. In IPC Global’s closing written submissions, it indicated that it no longer pressed its copyright claim in relation to version 2 of the TestLab software.
    4. In the respondents’ closing written submissions, they stated that it was not disputed that:  the UTS source code includes confidential information, which IPC Global is entitled to protect; substantial parts of the UTS software were copied in late 2012, by the copying of UTS source code and object code onto Mr Li’s computer; UTS source code confidential information was referred to in the process of creating version 1 of the TestLab software; and the acts were done on behalf of Pavetest, and authorised by Mr Sinadinos and Mr Feeley.
    5. In the course of closing oral submissions, IPC Global’s senior counsel, in response to a question, indicated that IPC Global did not press a claim based on a consultancy agreement between Old IPC and Mr Feeley (as distinct from the Aleuta Consultancy Agreement, between Old IPC and Aleuta).
12. As a consequence of the above, the main issues to be determined (and which form the subject of these reasons) can be summarised as follows:
    1. In relation to copyright:  did Pavetest infringe copyright in the UTS software by the creation of version 1 of the TestLab software? In particular, did version 1 of the TestLab software reproduce a substantial part of the UTS software?
    2. In relation to breach of confidence:  has IPC Global established the elements of the cause of action for breach of confidence?
    3. In relation to contract:
       1. Did Mr Feeley’s employment contract or the Aleuta Consultancy Agreement contain implied terms that Mr Feeley or Aleuta owed Old IPC a duty of good faith and fidelity not to engage in conduct which was destructive of the necessary confidence between IPC Global and Mr Feeley or Aleuta; and that Mr Feeley or Aleuta would not utilise the property and confidential information of IPC Global for any purposes other than those that benefitted IPC Global or its related corporate entities?
       2. If yes, did Mr Feeley or Aleuta breach these terms?
13. My conclusions and reasons can be summarised as follows:
    1. It is established that Pavetest infringed IPC Global’s copyright in the UTS software by Mr Feeley copying the source code for the system library and several test modules of the UTS software onto the computer provided to Mr Li in or about November 2012; and that Mr Sinadinos and Mr Feeley authorised the infringement. Further, it is established that Pavetest infringed IPC Global’s copyright in the UTS software by the creation of version 1 of the TestLab software; and that Mr Sinadinos and Mr Feeley authorised the infringement. In my view, for the following reasons, version 1 of the TestLab software reproduces a substantial part of the UTS software. For the purposes of this analysis I focus on the sequences of source code set out in the document referred to below as “Annex D”. First, there is originality in the *expression* of the sequences of UTS source code set out in Annex D. Secondly, the evidence establishes that these sequences of source code play a functionally significant role in the operation of the UTS software as whole. Thirdly, although the quantum of source code set out in Annex D is small relative to the total size of the UTS software, the cases make clear that the emphasis is on a qualitative rather than quantitative assessment of substantiality. The evidence establishes that there is significant duplication across UTS test modules. Having regard to this duplication, I consider the quantity of copied source code to be sufficient to constitute a substantial part.
    2. The breach of confidence cause of action is also established. First, the relevant information has been specifically identified, namely (a) the UTS source code, as a *corpus*; and (b) the documents referred to below as the UTS communications protocol documents. Secondly, the relevant information is confidential. Thirdly, Mr Feeley received the confidential information in circumstances imparting an obligation of confidence. Fourthly, the confidential information was misused. Further, the evidence establishes that Pavetest gained a substantial ‘springboard’ benefit by the misuse of the confidential information.
    3. IPC Global’s contract claims are not established. In relation to the Aleuta Consultancy Agreement, in circumstances where this agreement cannot be located and the evidence about its written terms is limited, I am not satisfied that it is necessary to imply the terms alleged by IPC Global.

## Procedural matters

1. IPC Global was given leave to amend its originating application and pleading during the course of the trial. Following amendment, these documents were:  the second further amended originating application and the second further amended statement of claim. The respondents’ defence was not amended following those amendments and remained the further amended defence.
2. The trial was on issues of liability only, with issues of pecuniary relief deferred to be dealt with later, if necessary. During the course of the hearing, the parties requested that they have the opportunity to make submissions on the form of any relief (including injunctive relief) to be granted following the publication of my reasons.
3. An issue arises with IPC Global’s claim for additional damages pursuant to s 115(4) of the *Copyright Act 1968* (Cth). IPC Global submitted in its closing written submissions that the better approach would be to make findings to support the enlivening of the provision, but defer setting any quantum; that should be done when other monetary remedies are being assessed. The respondents submitted that the Court should defer to the quantum stage any determination of whether, and if so what, additional damages to award for the admitted or proved copyright infringements, and any necessary findings of fact in respect of such a determination. I do not propose to address directly in these reasons whether additional damages should be awarded, preferring to defer this question to the stage of the proceeding concerned with pecuniary relief. However, it may be that, in addressing other issues, some findings are made which will ultimately bear on whether additional damages should be awarded and, if so, what amount.

## The hearing

1. IPC Global called evidence from the following witnesses:
   1. Kieran McGrane (the Chief Executive Officer of IPC Global);
   2. Stephen King (a senior research and development engineer at IPC Global);
   3. Bart Fernando (the Research and Development Manager of IPC Global);
   4. Geoffrey Sizer (an independent expert in relation to computer software).
2. The respondents called evidence from the following witnesses:
   1. Bin Li (a computer programmer engaged by Pavetest, who wrote the TestLab software);
   2. Mr Sinadinos;
   3. Mr Feeley.

The respondents did not call an independent expert witness.

1. Each of the witnesses was cross-examined. Mr Li gave evidence during cross-examination with the assistance of an interpreter.
2. I make the following observations about the evidence of the witnesses:
   1. Mr McGrane’s evidence was of a confined nature and the cross-examination of him was brief. He presented as a reliable witness and I accept his evidence.
   2. The cross-examination of Mr King was also quite limited. Mr King also presented as a reliable witness and I accept his evidence.
   3. Mr Fernando was, in my view, a very good witness. He answered questions in a straightforward way. He displayed a good knowledge of the subject-matter about which he gave evidence. He accepted propositions put to him during cross-examination where it was appropriate to do so, enhancing his credibility. I consider him to be a reliable witness and I accept his evidence.
   4. Mr Sizer has university qualifications in electronics engineering and in computer science. His industry and career experience has been predominantly in the development of electronic and microcontroller-based, software-driven systems, including test systems and electromechanical systems which have computers controlling apparatus of one sort or another. Mr Sizer presented as someone with a strong knowledge of the field about which he gave evidence. He gave evidence in clear terms. He accepted points that were put to him during cross-examination where it was appropriate to do so. He was a very good witness. I consider his evidence to be reliable and I accept his evidence.
   5. Mr Li generally gave careful consideration to questions before answering them and expressed his answers in precise terms, indicating a clear recollection of the relevant events. His evidence included significant concessions. There was no attack on his credit as a witness. In these circumstances, I generally accept his evidence.
   6. Mr Sinadinos gave evidence in a straight-forward way. There was no attack on his credit as a witness. In these circumstances, I generally accept his evidence.
   7. There was a substantial attack on Mr Feeley’s credit as a witness in IPC Global’s closing written submissions. It was submitted that he was not a reliable witness and that the Court should not rely on his evidence except where it is against his interests or independently corroborated. I do not accept that Mr Feeley’s evidence was generally unreliable. He made some frank and sensible concessions during the course of cross-examination, which suggest that other parts of his evidence should be treated as credible. Rather than forming any general view about his evidence, I will consider it in the context of specific factual issues as they arise.

## Factual findings

1. The following factual findings are based on the affidavit evidence and oral evidence during the hearing. Although some of the documentary evidence is confidential, it is not necessary to refer to those aspects of the evidence in detail in these reasons.

### IPC Global

1. IPC Global is in the business of developing, manufacturing and supplying materials testing equipment for asphalt, rock, soil, unbound and other construction materials. This is its core business and it has no other undertaking. IPC Global’s materials testing equipment is typically used for the purposes of research, quality assurance and control for road pavement design, civil engineering and construction. IPC Global supplies its products in Australia and internationally. It has about 400 customers in about 80 territories.
2. In 1981, Old IPC was incorporated.
3. In or about 2005, Old IPC entered into a distribution agreement with Controls S.r.l. (**Controls**) for a term of six years. Among other things, the distribution agreement provided that Controls would act as the exclusive distributor of products developed and manufactured by Old IPC in most territories throughout the world.
4. In or about 2010, Old IPC notified Controls that it did not wish to renew the distribution agreement and Controls ceased distributing the products of Old IPC shortly thereafter. Once the distribution agreement with Controls ended, Old IPC took steps to set up its own distribution network.
5. On or about 14 February 2011, Old IPC appointed Matest S.r.l. (**Matest**) as its distributor for the territory of Italy. Matest is a competitor of Controls.
6. On or about 7 August 2012, Matest notified Old IPC that the distribution arrangement for the territory of Italy would cease with effect from 7 November 2012.
7. In or about September 2013, the then owners of Old IPC (primarily, Paul Harpur) began talks with Controls in relation to a possible deal under which Controls (or an entity associated with it) would purchase the business assets of Old IPC.
8. On 31 March 2014, the business assets of Old IPC were transferred to IPC Global (which is 100% related to Controls).

### Pavetest

1. Pavetest was incorporated by Mr Sinadinos and Mr Feeley on 30 August 2012. Subsequently, in November 2012, an agreement was reached for Matest to acquire half of the issued shares in Pavetest.
2. Pavetest manufactures and sells products in direct competition with IPC Global.

### Mr Sinadinos

1. From January 1991 to June 2012, Mr Sinadinos was employed by Old IPC. In or about July 1994, he was appointed General Manager of Old IPC. In or about November 2009, his title was changed to Chief Executive Officer.
2. In April 2012, Mr Sinadinos gave notice of his resignation from Old IPC (the letter of resignation is dated 4 April 2012). His final day of work there was 22 June 2012. Despite searches, a copy of Mr Sinadinos’s employment contract with Old IPC has not been located.
3. Mr Sinadinos is the Managing Director of Pavetest.

### Mr Feeley

1. Mr Feeley joined Old IPC in about 1991. He took on a full-time role of designing and developing new products. In about 1993, the title of Research and Development Manager was created for him. Mr Feeley remained an employee of Old IPC until mid-2003. In July 2003, Mr Feeley resigned as an employee and was immediately re-engaged as a consultant, with the same title. Mr Feeley provided consultancy services pursuant to a consultancy agreement between Old IPC and Aleuta, which invoiced Old IPC for Mr Feeley’s services. That agreement is referred to in these reasons as the Aleuta Consultancy Agreement. It appears that it was initially constituted by an exchange of letters and later by a memorandum of understanding. Despite searches, the documents constituting the Aleuta Consultancy Agreement have not been located.
2. From July 2003, Mr Feeley generally worked about four days a week for Old IPC, usually for about 35 hours or a little more per week. He generally also worked about two days a week for another company, Celltek. Mr Feeley’s role as Manager, Research and Development extended to recruiting employees in the R&D Department, as well as overseeing all development activities in that department.
3. Mr Feeley was provided with a laptop computer by Old IPC. However, he also had a personal computer at home, purchased by Aleuta, on which he carried out source code development work.
4. On 29 June 2012, Mr Feeley gave notice to Old IPC that the consultancy would cease on or about 27 July 2012.
5. Mr Feeley is the Technical Director of Pavetest. Mr Feeley is also a director of Aleuta, which is his family company and under which he provides services to Pavetest.

### IPC Global’s materials testing equipment

1. Materials testing is the testing of materials such as asphalt, soils, unbound granular and pavement materials (eg, for fatigue, strength and creep).
2. Old IPC released its first materials testing apparatus in the early 1990s. The first product was called “MATTA”. Later apparatus were referred to as “universal testing machines” (**UTM**).
3. The apparatus manufactured by IPC Global has three main components:
   1. a loading frame with or without an environment chamber where the material being tested is subjected to stress or strain controlled tests to enable the data to be generated;
   2. a controller known before 2002 as a “CDAS” and from 2002 onwards as the “Integrated Multi Axis Controller” or “IMACS”; and
   3. a personal computer (**PC**) on which software with test modules, namely the UTS software, is installed. (This is usually the customer’s own PC.)
4. In general terms, the operator using the PC, with the assistance of the UTS software, creates a set of instructions to pass to the firmware for the firmware both to operate the testing machinery and to receive the appropriate responses from the testing machinery to send back to the UTS software. When the firmware sends back the relevant data to the UTS software, it is utilised by the software to generate screen reports and reports that can be printed.
5. The UTS software installed on the PC has a series of “test modules”. “Test modules” are the specific modules relevant to a particular Australian or international test standard or method. There are more than 40 test modules in the UTS software.

### Development of the UTS software

1. The first IPC Global materials testing apparatus, “MATTA”, operated with a PC on which software that was referred to as “UMAT/UTM1” was installed. This was released in 1991. Later versions were referred to as UTM2, UTM3 and UTM4.
2. Mr King said in his affidavit that, since 1992, approximately 80% of his working time at Old IPC has been devoted to programming and developing the UTM software and, later, the UTS software; from about 1992, he worked alongside Ken de Vos in programming and developing the UTM software and, later, the UTS software; from about 1995, both Mr de Vos and Mr King worked under the managerial supervision of Mr Feeley. I accept Mr King’s evidence as set out in this paragraph.
3. The concept work on the UTS software began in or around 1999 in conjunction with the development of a new controller. The programming work on the UTS software commenced in late 2001/early 2002. The programming work was carried out by Mr King and Mr de Vos. The ‘look and feel’ of the UTS software was very similar to the UTM4 software. Both Mr King and Mr de Vos were employees of Old IPC (as distinct from independent contractors).
4. Mr Feeley stated in his affidavit dated 13 September 2016, his third affidavit filed in the proceeding although the second tendered at trial (**Mr Feeley’s third affidavit**), that the first version of the UTS software involved about three person-years of development work, most of which was Mr de Vos’s time, the balance Mr King’s. Mr Feeley gave evidence to similar effect during cross-examination. I accept three person-years as a rough estimate of the time involved in developing the initial version of the UTS software, but note that the programmers, Mr King and Mr de Vos, had considerable experience already in the ‘problem domain’ through their work on the previous software.
5. Mr King said during cross-examination that he created about 19 of the test modules for UTS and did so over a period of about a decade. Mr King accepted during cross-examination that some of the test modules took a considerable time to develop; and that one of the challenges in programming a test module is the need to implement and code all of the mathematical equations of the testing standards. I accept Mr King’s evidence set out in this paragraph.
6. The UTS software was written using the “Delphi” environment. Delphi is an application development tool for programming Microsoft Windows applications. The Delphi development environment allows a Microsoft Windows application to be designed visually by ‘dragging’ and ‘dropping’ standard and custom components from a palette onto a form. This form design then represents the appearance of the Windows application. The behaviour of the visual components is then programmed in parallel. For example, source code is written to describe what should happen when a particular button is clicked or how to interpret and graph sensor data that has arrived from the IMACS controller. As well as the functionality described above, the Delphi environment allows the user to extend the standard visual components provided, as well as design custom visual components (that is, custom visual components powered by purpose-written source code). These components can then be used in the same way as a standard component (that is, to provide functionality common to a field of applications without rewriting/recreating a particular aspect every time). Windows applications use (what is referred to as) a graphical user interface (**GUI**).
7. The UTS software was initially written using Delphi 4 or 5 (the evidence is not clear on which version it was). From late 2004, it was written using Delphi 7.
8. One of Mr King’s first projects was to develop what is known as the “Virtual Pendant”. The Virtual Pendant is a software application made to mimic a hand held physical controller pendant, so that a physical controller is not necessary and all controls can be controlled from the personal computer. All buttons in the GUI were designed and created by Mr King.
9. Mr de Vos left Old IPC in about May 2011. Mr King took over the further development work on most of the tests developed by him.
10. In or about October 2011, Mr Fernando, at the request of Mr Feeley, prepared a first version of parts of the UTS software in Delphi 2009. The purpose of using Delphi 2009 was to enable non-English characters, which were needed for certain foreign language versions, to be used. Although work continued for some time on the Delphi 2009 UTS software, it has not been released. This is because, in or about late 2012, Old IPC made a decision to develop a new generation of the UTS software which, among other enhancements, would be multi-lingual. That project was ongoing as at 15 March 2016, when Mr Fernando affirmed his third affidavit.

### The UTS software

1. The UTS software comprises a series of test modules and the Dynamically Linked Libraries (the **DLL** or **system library**).
2. The test modules are the parts of the software that the end-user interacts with on his or her PC. These modules are written to implement a particular test standard. Each of the modules has a name and a number using the format UTSXXX; for example, some of the modules referred to in Mr King’s affidavit are:
   1. Cyclic Triaxial (soils) (UTS012);
   2. Overlay Test (UTS036);
   3. Tuning Test (UTS001);
   4. SPT Flow Test (UTS005).
3. In general terms, system library code is code that is common between programs; the programmer puts the code in a library so all the different programs can use it. In this way, the programmer does not have to write exactly the same code again every time he or she is writing a new test. When a UTS test module is running, effectively the code in the libraries is merged with the code in the test module as a single process.
4. There was a difference in emphasis, but perhaps not of substance, between Mr Fernando’s and Mr Feeley’s evidence regarding the role and significance of the DLL.
5. In Mr Feeley’s third affidavit he stated that the UTS system library provided infrastructure support, or a building block, which could be used or called up to implement UTS application test modules; in general, it provided a software interface to allow the high-level application test modules to communicate with the IMACS firmware, using a shared message format. He stated: “It thus worked in a similar way to a ‘device driver’ such as a printer driver, which is software installed on a PC that allows a word processor on that PC to communicate (via a specified communication protocol) with the firmware of the printer.”
6. Mr Fernando gave the following evidence in relation to the role and significance of the DLL:
   1. In his fifth affidavit, Mr Fernando stated that the DLL implements the functionality to set up transducer allocations, calibration of transducers, leading control setup, sensor data acquisition and timing setup, test synchronisation and control system parameters set up. He stated: “Put simply, the source code for the DLL exploits the functionality of the IMACS to provide all the functionality to execute a materials [test] (i.e. the particular UTS application modules), while the source code for each of these test module[s] passes the specific parameters to the DLL to implement the particulars of a test standard, and apply post processing and calculations to acquired data as described in the test standard.” Mr Fernando also stated that, while he agrees with Mr Feeley’s affidavit evidence that the DLL is akin to a device or printer driver because it provides the UTS test modules with the infrastructure to drive the IMACS, this infrastructure is the core intelligence of the UTS functionality; specifically, the DLL exploits the full functionality of the IMACS (eg, transducer data acquisition, timing, loading and control).
   2. Mr Fernando accepted during cross-examination that the test modules pass down the parameters that will be needed by the firmware; once those parameters have been passed down, the libraries will send those parameters; as the test is running or when it is complete, data will be sent back to the software; the common code has a role in decoding and passing on that data to the test module; and the test module does whatever calculations need to be done. Mr Fernando accepted during cross-examination that the passing of the specific parameters to the DLL is the populating of the various data structures with the parameters required by the test design. (I refer below to “data structures”.)
   3. Mr Fernando said during cross-examination that the DLL sets up all the default settings in the IMACS; displays the calibration data for each transducer as interpreted by the IMACS; sets up default machine limits; and “has all the communications with the IMACS”. He said that, with any piece of hardware of this nature, you need a piece of software to explore every single function of the hardware.
   4. Mr Fernando accepted that, essentially, the way that the DLL exploits the functionality of the IMACS is by sending message packets to the firmware which contain particular information to do particular things.
7. Although there was a difference in emphasis as between Mr Feeley and Mr Fernando as to the relative significance of the system library and the test modules, there does not appear to be any substantive dispute about what the system library does. I accept the descriptions of Mr Feeley and Mr Fernando, set out above, as to the functions performed by the system library.
8. Mr Feeley stated in his third affidavit that the UTS software (in its Delphi 7 version) has about 250,000 lines of source code, including application code (ie, the code for test modules) and the system library code. I accept this evidence, which was not challenged. Mr Fernando stated in his fifth affidavit that the UTS system library has about 6,000 lines of source code and that, on average, each test module has a similar number of lines. I also accept this evidence, which was not challenged.

### The message format and data structures

1. In order to understand some of the issues and evidence discussed below, it is necessary to refer to two key concepts or expressions: the “message format” and “data structures”.
2. In his third affidavit, Mr Feeley explained the IMACS message format as follows:
   1. Messages had to be exchanged between the UTS and the IMACS firmware in the materials testing machine, so that a user could use the software both to control the machine (so as to ensure the test was performed according to the software design) and to obtain the test results for analysis. A shared message format was necessary to allow exchange of that information.
   2. The IMACS message format was structured as two layers. Generally, the lower (or network) layer specified the control information to be carried by each message packet (that is, the data identifying, addressing and validating the message) and the higher (or application) layer specified the message ‘payload’ (that is, the substantive content of the message). A layered structure of this kind is typical for a message format for a network communication.
   3. The lower layer of the IMACS message format was an implementation of a third party communication protocol known as Scalable Network Application Protocol or **SNAP**. SNAP defined a generic message format for sending messages across a communication medium. (Mr Feeley said in oral evidence that some minor amendments, which he described as “trivial”, were made to the open-source SNAP protocol.)
   4. The higher layer of the IMACS message format was an application protocol that defined the structure of the payload of each of the messages that needed to be sent to and from UTS. (Mr Feeley said in oral evidence that the payload is the application-specific information that is sent backwards and forwards between the UTS and the IMACS in a defined format.) The higher layer of the format was ‘bespoke’ to IMACS.
   5. In all, there are about 100 messages that are described by the IMACS message format, most of which are control messages.
   6. Mr Feeley designed the message format used for IMACS. That format was recorded in a protocol document. (Mr Feeley said during cross-examination that he did this work at Old IPC before 2003).
3. In oral evidence, Mr Feeley referred to the protocol document. In response to a question from the Court, he said that this was a document that sets out the messages that are sent between the PC and the controller; and that it was essentially a document that explains in English what the message format is. During cross-examination, he agreed with the proposition that the message format is embedded in the firmware and software.
4. It was put to Mr Sizer during cross-examination that the protocol is not itself part of the computer program, and that it was a protocol for communication between computer programs. He responded: “The protocol in the sense of the definition of the fields within a packet and what their meaning is, would normally be captured in a text document. And I would not regard that protocol definition as part of the computer program. However, the software at either end of the link that does the compilation and de-compilation of the message packets is part of the software which implements what the specification of the protocol tells it in needs to do.” It was then put to him that “the software, when it’s sending a message to the firmware, that software instruction, for example, that’s encoding a message, that’s part of the program”. Mr Sizer said this was correct.
5. I do not think there is any substantive dispute about the evidence of Mr Feeley and Mr Sizer set out in [64]-[66] above. There is, however, some ambiguity in the expressions “message format” and “communications protocol”. They can be used, first, to refer to an ordinary text document that defines or describes a system for sending messages between two points. Such a document will not set out the actual lines of code required to implement the system it defines or describes. The lines of code will need to be written by a programmer, writing within the confines of a protocol document. I will use the expression “communications protocol document” to make clear when this sense is intended.
6. The second sense in which the expressions may be used is to refer to the actual lines of code (forming part of the system library) that implements a system for sending messages between two points. I will use the expression “source code relating to the message format” to make clear when this sense is intended.
7. The purpose of a “data structure” in programming terms is to encapsulate data that is going to be useful to the programmer to be dealt with as a group. Data structures can be groups of variables and constants. A “variable” is a value that can change as the program is running. For example, it may change in response to user input. A “constant” is a value that does not change. In the software, constants that are used in various places are defined. The reason for this is as follows. The constant may be used throughout the software in hundreds of different places. If there is a need to change that amount (for example, in the next generation of controller) it can be redefined in one place. Names are given to the constants to make it easier for the programmer to remember what he or she is manipulating.
8. In the context of executable code, the variables and constants have a meaning and are used within the lines of the executable code.
9. The data structures that are on the ‘firmware side’ and the data structures that are on the ‘software side’ in effect mirror each other, because both of them have the same types of parameters. They may not look identical in code form, but they hold the same kind of parameters. These are the parameters which, when the data structure is filled with actual values, are sent in the message packet.

### Confidentiality of the UTS software and communications protocol document

1. It is common ground that the UTS source code is confidential. But there was dispute about whether the communications protocol document was confidential. I refer to the evidence and then make findings.
2. Mr Fernando gave evidence in his third affidavit as follows:
   1. Since he commenced his employment in January 2011, Old IPC and now IPC Global have kept the source code for the UTS software in a specific location on the server.
   2. Throughout that time, access to the UTS software source code has been restricted to the Chief Executive Officer and certain personnel. (These included Mr Feeley prior to his resignation.) The restricted access is achieved by network access credentials.
   3. To the best of Mr Fernando’s knowledge, the UTS software source code has never been provided to customers, distributors or any person external to IPC Global (other than for the purposes of this proceeding).
3. Mr Feeley stated in his third affidavit that he was not aware of any specific restrictions on accessing the source code on Old IPC systems.
4. Mr Feeley accepted during cross-examination that he held the view, in July 2012, that source code was confidential. He accepted that during the interlocutory stages of this proceeding there had been a couple of disputes in which Pavetest had very firmly enforced the confidentiality of its source code.
5. Mr Feeley stated in his third affidavit that he had never considered or treated the IMACS message format as proprietary, or a trade secret, and had never heard anyone at Old IPC refer to it in this way. In oral evidence during the hearing, Mr Feeley referred to a communications protocol document and said that he did not consider this to be a confidential document as IPC had considered sending it to a number of companies over the years and had done so in respect of some machines. No details were provided.
6. Mr Fernando, in his fifth affidavit, stated that the restricted access applicable to the source code for the UTS software was applied to the source code and design documents relating to the IMACS messaging format.
7. To the extent that there is a difference between the evidence of Mr Fernando and Mr Feeley in relation to confidentiality, I prefer the evidence of Mr Fernando. It seems to me that Mr Fernando’s evidence is more likely to be correct, when considered in the context of the evidence generally. It seems to me to be likely that IPC Global treated, and treats, the communications protocol document (or documents) and the source code relating to the message format as confidential. The communications protocol document is intimately connected with the development of the UTS source code, which itself is treated as confidential.

### Backups

1. Mr Feeley stated in his third affidavit that, when he worked at Old IPC, the company did not have any formal system of making off-site back-up copies of the code on its computer system; accordingly, from time to time, especially before holiday periods such as Christmas and Easter, he made copies of the complete source code of the UTS software and the IMACS firmware onto compact disks or DVDs and took them home by way of back-up. I accept this evidence.

### March to July 2012

1. In March 2012, Mr Sinadinos visited Matest in Italy. He was, at this time, the CEO of Old IPC. The primary purpose of Mr Sinadinos’s visit was to install an Asphalt Mixture Performance Tester (or AMPT) at the University of Torino, but before returning to Australia he visited Matest to discuss general sales and marketing issues. Mr Sinadinos met with Robert Maestroni, the founder and owner of the Matest business, and his daughter, Paula Maestroni, the Manager of Matest. During the meeting, Mr Maestroni asked Mr Sinadinos if he was happy at Old IPC. He said that he was frustrated as he was concerned that the owner was taking too much cash out of the business to finance other business ventures. Mr Maestroni asked if the owners would be interested in selling the business to Matest. Mr Sinadinos said that he thought the owners would be interested in selling part of the company, but he did not believe they would offer to sell part of the business for a realistic market price. Mr Maestroni asked Mr Sinadinos if he would be interested in a joint venture with Matest. Mr Sinadinos said he would need to consult his wife and that, if they were going to start a new business, Mr Feeley and Mr McGrane should also be part of the business. Mr Sinadinos spoke with his wife that night and informed Mr Maestroni the next morning that he was interested. Mr Sinadinos said during cross-examination that the proposal was to establish a company in competition with Old IPC, which would manufacture or distribute, or design for manufacture and distribution, materials testing equipment driven by software.
2. Following Mr Sinadinos’s return to Australia, in late March 2012, he asked Mr Feeley and Mr McGrane to meet with him at the Knox Club in Wantirna, Victoria. At the club, Mr Sinadinos informed Mr Feeley and Mr McGrane in confidence that he would be resigning from Old IPC and was planning to create a new company with Matest, which would develop products to compete against Old IPC. Mr Sinadinos said that he believed the then owners were not investing sufficiently in the development of the company. Mr Sinadinos then invited Mr Feeley and Mr McGrane to join him in the venture.
3. Subsequent to the meeting at the Knox Club, Mr Sinadinos, Mr Feeley and Mr McGrane had a number of discussions about how a new business could be created. Mr McGrane said in his affidavit that, during one such meeting, Mr Sinadinos and Mr Feeley informed him that they intended to seek legal advice about their respective employment/contractor agreements with Old IPC and the restrictions on their ability to compete against Old IPC. Mr McGrane was not questioned about this during cross-examination and I accept the evidence that this was said.
4. There is a difference between Mr Sinadinos’s evidence and Mr Feeley’s evidence regarding the timing of the events that come next, but I do not think much, if anything, turns on this.
5. Mr Sinadinos stated in his second affidavit that, on 4 April 2012, he handed Mr Harpur his letter of resignation (dated 4 April 2012); after he handed in his resignation, he told Mr McGrane that he had resigned; shortly afterwards, in or around April 2012, Mr Sinadinos, Mr McGrane and Mr Feeley met with Mr Harpur to see if Mr Harpur would be interested in selling Old IPC to its managers, but this was rejected; Mr Sinadinos ceased employment with Old IPC on 22 June 2012. Mr Sinadinos said during cross-examination that, between late March and 4 April 2012, he made up his mind that he was going to enter into some kind of joint venture with Matest.
6. Mr Feeley stated in his third affidavit that the three of them (Mr Sinadinos, Mr McGrane and Mr Feeley) had a second meeting to discuss the new business proposal on Easter Friday (which was 6 April 2014); the following day, Mr McGrane called him to say he was considering staying at Old IPC; at this time, he (Mr Feeley) was yet to decide whether to go into business with Mr Sinadinos and Matest; he decided that, before making a decision, he should visit Matest; he flew to Italy after Easter 2012 (that is, during April 2012) and met with Ms Maestroni, Mr Maestroni and their engineering team; on his return from Italy, which was in about late April 2012, Mr Sinadinos came into Mr Feeley’s office and told him he had just resigned; after Mr Sinadinos resigned, he, Mr McGrane and Mr Feeley met with Mr Harpur to see if he would allow them to invest in Old IPC or would sell the company to them, but this was rejected; in June 2012, Mr Sinadinos finished at Old IPC; it was shortly after he left that Mr Feeley finally decided to leave Old IPC and to join Mr Sinadinos.
7. On the basis of this evidence, I find that:
   1. A number of discussions took place between Mr Sinadinos, Mr Feeley and Mr McGrane in late March and early April 2012 regarding the establishment of a new business in conjunction with Matest, which business would manufacture or distribute, or design for manufacture and distribution, materials testing equipment in competition with Old IPC.
   2. Mr Sinadinos resigned from Old IPC on or about 4 April 2012. He finished working there on 22 June 2012.
   3. Further discussions took place with Mr Feeley and Mr McGrane after Mr Sinadinos’s resignation.
   4. Mr Feeley decided to leave Old IPC (end the consultancy) and join Mr Sinadinos shortly after Mr Sinadinos finished working at Old IPC, that is, shortly after 22 June 2012.
8. Although Mr McGrane was initially interested in joining Mr Sinadinos, he ultimately decided not to join Mr Sinadinos and Mr Feeley in the new business venture. Mr McGrane informed Mr Sinadinos and Mr Feeley of this in late April 2012.
9. As at 22 June 2012, Mr Sinadinos’s plan was that the hardware to be offered by the new company would align itself very closely with the range of hardware being offered by Old IPC. Mr Sinadinos accepted during cross-examination that, in practical terms, having the software and firmware available within a reasonable timeframe depended on Mr Feeley coming with him to the new company.
10. On or about 29 June 2012, Mr Feeley handed Mr McGrane (who had been appointed the Chief Executive Officer of Old IPC following Mr Sinadinos’s resignation) a letter of that date by which Aleuta gave notice that it was ceasing to provide services as from 27 July 2012. Mr Feeley’s role ceased on or about 27 July 2012.
11. It is clear that, at the time Mr Feeley left Old IPC, he retained copies of the UTS software. An issue arose as to whether some of these copies were made deliberately with the intention of using them in the new business (as distinct from merely being backup copies). I refer to the evidence on this issue in the following paragraphs and then make findings.
12. Mr Feeley said during cross-examination that when he finished consulting with Old IPC (on 27 July 2012) he had retained copies of the source code in the circumstances of the informal back-up process described in his affidavit.
13. In paragraph 50(a) of the respondents’ defence, the respondents stated that, after July 2012, Mr Feeley retained backups of source code for the UTS software. On 23 December 2015, an order was made in this proceeding to the effect that (a) the respondents deliver up to IPC Global all copies of the retained UTS source code referred to in paragraph 50(a) of the defence; and (b) Mr Feeley file and serve an affidavit attesting to the fact that he no longer retains any further copies of the retained UTS source code referred to in that paragraph of the defence. In his first affidavit, Mr Feeley stated that, pursuant to the orders made in the proceeding, he had copied the retained UTS source code onto DVD disks and then deleted that code; those disks were exhibits to his affidavit.
14. During cross-examination, Mr Feeley was taken to printouts of certain ‘pages’ from one of those disks (Ex A18). These pages showed (as Mr Feeley accepted during cross-examination) that, among the software that had been in his possession, was some Delphi 7 and some Delphi 2009 UTS software. One of these pages, with a heading “SystemLib-combined\_latest”, listed three zip folders (page 4 of Ex A18). The “Date modified” column indicated that they were modified on 25 and 26 June 2012. It was put to Mr Feeley that this was an indication that those zip folders were created on those dates. He accepted that it seemed that way. Those dates were after the date on which Mr Sinadinos left Old IPC (22 June 2012) and shortly before Mr Feeley’s resignation letter on 29 June 2012. It was put to Mr Feeley that he made those copies into zip files for the purpose of taking them with him when he left. Mr Feeley said he did not recall that. Later in cross-examination, Mr Feeley said that, from April 2012 until the time he left, Mr Fernando and he were working on a Delphi 2009 system library and there were a number of issues with ‘lockups’. Noting that Ex A18 showed three zip files of similar versions separated by a day or two, Mr Feeley said that this indicated to him that he was investigating or helping Mr Fernando investigate a problem in the system library; he said that he may well have emailed those three very focused small parts of Delphi 2009 UTS to assist in understanding the issues that they were having in the communications lockup.
15. In re-examination, Mr Feeley was taken to a larger copy of page 4 of Ex A18 which included the full file or folder names and an additional column, headed “Date created” (Ex R10). This showed that the full name of one of the files modified on 26 June 2012 was “System – 2012June26Tue0234pm – Comms Code Cleanup 64 reintroducing debug pipe”. Mr Feeley said that this indicated that they were investigating issues with the system library in UTS at the time, that is, between March and June 2012. He said that Mr Fernando was investigating a process of implementing pipes as a solution to “communications lockups” they were having.
16. Having regard to Mr Feeley’s explanation, as set out above, I am not satisfied that Mr Feeley deliberately copied the UTS software with a view to using it in the new business. Nevertheless, as is accepted by the respondents, he retained copies of the UTS software when he left Old IPC.

### August to November 2012

1. On 30 August 2012, Mr Sinadinos and Mr Feeley incorporated Pavetest. Initially, they were its only directors. From this time, they began negotiating with Mr Maestroni and (as to the details) Ms Maestroni for Matest to acquire half of the issued shares of Pavetest.
2. In August or September 2012, Mr Feeley first spoke to Mr Li about Pavetest. Mr Li holds degrees in computer software and computational mathematics and has more than 16 years’ experience in information technology systems development across multiple platforms in a variety of industries, including 10 years’ experience in Delphi. Mr Feeley and Mr Li knew each other as they had both done work at or for Celltek. Mr Li gave evidence (which I accept) that, as at August/September 2012, Celltek, where Mr Li was working as a senior software developer, was closing and his “boss” told him he might be able to work for Mr Feeley who had another project.
3. In late November 2012, an agreement was reached for Matest to acquire half of the issued shares in Pavetest. Under the agreement, Matest was allocated 50% of the issued and allotted shares in Pavetest, and two additional directors from Matest were appointed to Pavetest.
4. Under the shareholders agreement, Matest was to fund the activities of Pavetest, at least for a significant period, and up to profitability.
5. Annex A to the shareholders agreement set out equipment intended to be developed by Pavetest. Each of those pieces of equipment was a piece of equipment sold by Old IPC. In relation to the Asphalt Mixture Performance Tester (or AMPT), the key milestones included: the first prototype hardware of main unit to be completed by March 2013; control and data acquisition system hardware and firmware to be completed by September 2013; the application software to be completed by December 2013; and the product to be ready for sale by February 2014. These timeframes were not in fact met.
6. Mr Sinadinos did not accept during cross-examination that the timelines agreed with Matest depended upon a view that Mr Feeley would be able to use code that he had been using at Old IPC. In light of this evidence, I do not think IPC Global has established that Pavetest, in formulating these timeframes, proceeded on the basis that it would need to copy the UTS software.

### Development of the TestLab software (version 1)

1. Mr Feeley stated in his third affidavit that he is the architect of the TestLab software and that the source code for the software was written, to his design, by Mr Li, who was engaged informally for this purpose by Pavetest. I accept this evidence.
2. All, or virtually all, of the source code for the TestLab software (both version 1 and version 2) was written by Mr Li.
3. Mr Li commenced working for Pavetest in September/October 2012. He worked effectively full-time on writing version 1 of the TestLab software from that time until it was completed in early 2015.
4. Based on Mr Feeley’s evidence during the hearing, it seems that, for a period of about one month in September/October 2012, Pavetest hired Mr Li through Celltek and paid Celltek for his services. In November 2012, Pavetest engaged Mr Li as a consultant. Mr Li has provided services to Pavetest as a consultant since then. He has a continuing relationship with Pavetest; he continues to work full-time for the company. Since November 2012, Mr Li has issued Pavetest with weekly invoices for his work. Although the early invoices were brief, from February 2013 the invoices contained a more detailed description of the work carried out during the relevant week.
5. In about November 2012, Mr Feeley gave Mr Li a computer to use for the project. At or about that time, Mr Feeley copied parts of the UTS software onto this computer, including the whole of the system library and a number of test modules.
6. I will set out the evidence of Mr Li, then the evidence of Mr Feeley, and then make some further findings in relation to the development of the TestLab software.
7. Mr Li stated in his affidavit that in about November 2012, Mr Feeley showed him some source code, and said it was from his previous company; Mr Feeley and Mr Li referred to this as the “old code”; Mr Feeley provided Mr Li with a computer for use on the project, on which Mr Feeley installed the old code; specifically, Mr Li received the system library code and test modules numbered 001, 002, 003, 005, 006, 018 and 019. Mr Li said during cross-examination that he still had the computer but the old code was no longer on it. He said that he was asked to remove it by Mr Feeley at some time during 2015.
8. Mr Li also stated in his affidavit that Mr Feeley told him that he wanted to create a new Pavetest product, to be called “TestLab”, which had a similar purpose to the old code, but would be better; Mr Feeley explained that he wanted to create a ‘universal’ piece of software, which would handle all types of existing tests, and all potential tests in the future.
9. Mr Li stated in his affidavit that he had previously developed applications in Delphi for companies in the building intelligence and home automation, industrial automation and telecommunications industries; those applications had a significant feature in common, which was also significant for the TestLab application, namely real time data acquisition and transmission; as a result of his earlier experience, Mr Li believed he could understand and implement Mr Feeley’s design. Mr Li also stated that, as he was new to the industry of materials testing, he did not have any knowledge about the material tests carried out in that field; he considered that he needed to understand what a materials test was, and how software could be used to create and manage a test. He stated: “I found it useful to read the old code to learn something about this general issue”.
10. Mr Li stated at [22] in his affidavit that, in his view, the old code was not written efficiently; every piece of the old code was able to perform only one type of test; as a result, a number of projects were created to cater for all types of tests; this seemed to Mr Li to have caused problems with code management; first, there was code for many different projects and tests, that all had to be separately maintained; secondly, customisation and variation to some tests, seemingly as a response to requests from clients, resulted in more copies of the same code for projects. Mr Li accepted during cross-examination that he studied the old code closely and, in particular, closely enough to be able to make the observations he made in [22] of his affidavit.
11. Mr Li gave the following evidence in his affidavit:
    1. The part of the old code that he copied was the “communications and protocols part” which was within the system library.
    2. There are two layers of protocol for communicating between the software and the interface of the materials testing machine’s firmware – the lower layer, called SNAP, and the higher layer, which is the application protocol.
    3. The application protocol layer specifies the precise format in which data must be exchanged between the software and the interface of the materials testing machine.
    4. Mr Feeley specified the application protocol data format for Mr Li; that format was “based on and very similar to” the application protocol data format for the old code.
    5. Based on the format that Mr Feeley specified, Mr Li designed the internal data structures for the system library of the Pavetest system. (It is convenient to note at this point that the TestLab software also has a system library.) Mr Li “used data structures of the old system library as the base for writing the internal data structures of the TestLab system library”. Mr Li added: “However, those data structures followed from the application protocol data format”.
    6. In version 1 of the TestLab software, “many functions for converting data from the data structures used by the system library to the application protocol data format (*i.e.* the functions for packaging data), and many functions that accessed those structures, were also based on corresponding functions in the old system library code”.
    7. He made significant improvements to the design and implementation of old system library code.
    8. By reusing data structures used by the old code, and system library functions, instead of developing these from scratch based on the application protocol, Mr Li estimated that he saved himself “about a couple of months of development time”.
    9. Other than code in the system libraries, he did not copy any of the old code.
12. Mr Li said during cross-examination that, although he did use some content from the data structures of the old code, the communication aspects were “totally redone”.
13. Mr Li stated in his affidavit (as corrected during the hearing) that Mr Feeley gave him a document titled “Notes on TestLab Test States, Acquisition and Loading”; and that Mr Li read this in about March/April 2013 and used it as a guide when he was writing the TestLab software. Mr Li said during cross-examination that Mr Feeley gave him other documents as well, but this was the most important.
14. Mr Li stated in his affidavit that, once he fully understood how the new system worked according to Mr Feeley’s design, which was in early 2013, he only rarely referred back to the old code. Mr Li said during cross-examination that, from about April 2013, he only rarely referred to the old code. He said:  “I wasn’t referring back to it every day. So only when it comes to a part where I’m not quite sure, I would refer back, and after a quick look then I will not be continuing looking at it”. Mr Li accepted that the purpose of reviewing the old code was to see if there was something it could tell him that would be useful in creating new source code.
15. Mr Li accepted that his invoices dated 7 June 2013, 17 June 2013, 4 October 2013, 18 October 2013 and 23 January 2015 contained references to his consulting the old code as part of the description of the work covered by those invoices.
16. Mr Feeley gave the following evidence in his third affidavit about his conceptual approach in developing the TestLab software. He stated:
    1. Although the TestLab software would necessarily have the same general purpose as the UTS software, he conceived a different core architecture and design, which he believed would be superior to that of UTS.
    2. UTS had grown to include code for about 40 different tests, as well as code from supporting libraries. Although code for the various tests was often shared, and the tests used the same common libraries, each test was a separate module and, as such, the code for that test was separately developed and maintained.
    3. The result was that every time a materials testing standard was revised, or a new materials testing standard was specified, a skilled developer had to be engaged to write or modify the relevant test code to implement that standard.
    4. Mr Feeley wanted to take a different, and more efficient, approach to the design of TestLab, which he considered would reduce costs over time. His vision was to have a truly “universal” piece of software, which could be used by a customer to implement any materials testing standard, whether new or modified, without the need to engage a skilled developer to write a new application test module each time.
17. Mr Feeley stated in his third affidavit that before Mr Li started any coding, he specified to Mr Li the required message format for information exchange between the TestLab software and the Pavetest CDAS; that format was based on the IMACS message format that he had earlier developed; Mr Feeley considered it to be preferable to make the IMACS and Pavetest message formats incompatible (among other things, there would be features of TestLab that could not be supported by the IMACS firmware); accordingly, the TestLab software’s application layer is designed to be different from the UTS software application layer, in a way that prevents the TestLab software from communicating at all with IMACS firmware (or UTS with Pavetest firmware); to achieve this design, Mr Feeley specified a different way of processing the cyclic redundancy check 16 checksum on the network protocol.
18. Mr Feeley stated in his third affidavit that he had retained copies of the UTS source code (in circumstances where he had taken them home as a back-up before leaving Old IPC); in or around December 2012, he gave Mr Li access to parts of the UTS source code; he specifically recalls supplying him with copies of Delphi 7 versions of the source code for the application test modules UTS001, UTS006, UTS018 and UTS019 and the UTS system library. Mr Feeley stated that he supplied Mr Li with UTS code as he did not have prior experience in the problem domain, that is, the real world context in which the TestLab software would need to operate. Mr Feeley also stated that he wanted Mr Li to understand what some of the requisite tests were actually seeking to do, and how users needed to conduct them, so that he could understand the fundamental technical requirements of software in this field. Mr Feeley stated: “I did not consider this to be an improper or illegitimate use of the UTS source code. The problem domain of materials testing is not confidential, but consists of published international materials testing standards that stipulate the required tests and largely determine and detail the necessary calculations, and data acquisition and control requirements of those tests, along with the loading control (*i.e.* material stimulus) requirements.” Mr Feeley also stated that, while he appreciated that Mr Li would inevitably gain some insight into the particular software solution of UTS, he intended to implement a different core architecture and design.
19. Mr Feeley said in his third affidavit that he could have communicated his domain knowledge to Mr Li in a series of face-to-face meetings, but Mr Li did not have strong verbal English. Mr Feeley said during cross-examination that Mr Li’s written English was very good and that he (Mr Feeley) could have prepared documents for him to read that ‘downloaded’ his knowledge and experience of materials testing and software for materials testing. Mr Feeley said in his affidavit that he thought an efficient way to bring Mr Li up to speed in the problem domain would be for him to read UTS source code.
20. Mr Feeley stated in his third affidavit that when he reviewed Confidential Schedule 4, filed as part of IPC Global’s claim, he “was surprised to see the number of the similarities that existed between the source code of functions in the UTS system library and of corresponding functions in the TestLab system library”. Mr Feeley also stated: “It was clear to me that Mr Li had used some UTS data structures, and had copied functions in the UTS system library that used those structures”.
21. Mr Feeley also stated in his third affidavit that, while he expected to see some similar data structures because of the similarities between message formats, he had not expected to see similarities between the source codes. In evidence admitted on a limited basis (in connection with issues under s 115(4) of the *Copyright Act*), Mr Feeley stated that he believes Mr Li copied some source code in the system library because of a miscommunication on Mr Feeley’s part, perhaps due to a language barrier; and that he (Mr Feeley) had not made his intentions sufficiently clear to Mr Li.
22. Mr Feeley wrote and provided to Mr Li a number of documents relating to the design of the TestLab software. The documents were listed in chronological order, and described, in Mr Feeley’s affidavit dated 11 October 2016 (**Mr Feeley’s fourth affidavit**).
23. Mr Feeley gave the following evidence during cross-examination in relation to his giving the source code of the UTS software to Mr Li.
    1. It was put to Mr Feeley that it was not right to give Mr Li access, for the purposes of working on TestLab, to source code that had been written for a rival business, namely the UTS source code. Mr Feeley responded: “I considered it actually fair use. I considered that as architect of the software … I had some right to at least look at it. And the reason I gave it to Mr Li was to research an approach that I had taken in the past or that I had been involved with in the past. … And that we were looking to do something completely different.” He said that his view that it was “fair use” was a view he reached about an aspect of copyright law; that “fair use” was terminology that explained his thinking at the time; and that he did not pay close attention to the requirements of copyright law at the time. He accepted that he held the view that the source code was confidential and that he understood that the confidentiality of the UTS source code belonged to Old IPC. The gist of Mr Feeley’s responses to further questions as to the propriety of providing the UTS source code to Mr Li was that, as he was involved as the architect of the whole of that software, that gave him certain rights, and the provision of the source code to Mr Li, for research purposes in the context of developing completely different software, was fair use.
    2. Mr Feeley said that he believes he said to Mr Li, “We should not use any of this code. Any code we should write should be our own.”
    3. Mr Feeley said he had no reason to disagree with Mr Li’s evidence as to the files of UTS he was given. Mr Feeley said that the application modules from the UTS software he gave to Mr Li were a sample representing the different application modules that you might find out of the set of 40. Mr Feeley accepted during cross-examination that giving Mr Li the UTS source code gave Mr Li insight into the types of tests that Pavetest would need to address.
    4. Mr Feeley said that, when he received a letter from IPC Global’s solicitors during 2014, he advised Mr Li that they should not be looking at UTS source code.
    5. Mr Feeley said that Mr Li’s invoices were provided to him by email and he forwarded them on for payment. Mr Feeley was taken to a number of invoices showing Mr Li reviewing the old code. Mr Feeley said during cross-examination that in 2013 and 2014 he told Mr Li on a couple of occasions that he should not be looking at old source code. It was put to Mr Feeley that the invoices show that from early 2013 until an occasion in early 2015, when Mr Li encountered issues of some sort with the task he had of writing software for Pavetest, Mr Li went back and consulted the UTS software for some reason or other. Mr Feeley said that, if Mr Li’s references to “old code” mean the UTS software, then it appears that way.
    6. Mr Feeley said that the data structures and the message formats were specified by him to Mr Li. These were very similar to those present in the UTS software.
    7. Mr Feeley said that there would be probably only two to three weeks’ work in coming up with a message protocol. He accepted that, rather than do that work, he used the work he had done before, at Old IPC (and before 2003, when he was an employee).
    8. Mr Feeley said that he did not consider the communications protocol to be confidential. (As set out in [78] above, I consider that the communications protocol document was confidential to Old IPC.)
    9. Although Mr Feeley said that the communications protocol was set out in a protocol document, he accepted that the format was embedded in the firmware and software. (In re-examination, Mr Feeley explained that by “embedded”, there are two parts to consider. The first part is formulating the message to send and the second part is in decoding the message to be received. “So if we take the decoding part first, when the PC software received a message or a packet, which contains a message, it needs to … interpret that message into a data structure for use in the rest of the program. In the transmission side of things, there is a similar process but, typically, the information comes from the application that defines or sets parameters in a data structure, and then a function is called that takes that data structure, produces a packet or a message, and sends it to the control and data acquisition system.”)
    10. Mr Feeley said that he also loaded onto a laptop being used by Mr Li the executable code for parts of the UTS software. He said that he did this to educate Mr Li about the different types of tests. He said that he assumes that the parts of the UTS executable code that he gave Mr Li were the same parts as the parts of the UTS source code that he gave Mr Li. Mr Feeley accepted that, as at the end of 2012 and the beginning of 2013, he understood that the copyright in the executable code of the UTS software was owned by Old IPC; and that he did not think he had Old IPC’s permission to reproduce that software.
24. To the extent that there were differences between the evidence of Mr Li and that of Mr Feeley, I prefer the evidence of Mr Li. His evidence was given precisely and he is the person who actually did the programming and, thus, the copying. However, as noted below, I do not accept all aspects of Mr Li’s evidence. Having regard to the evidence of Mr Li and Mr Feeley set out in [108]-[124] above, I make the following findings:
    1. In about November 2012, Mr Feeley provided Mr Li with a computer onto which he had copied UTS source code and object code. Mr Feeley told Mr Li it was from his previous company. Mr Feeley and Mr Li referred to this as the “old code”. The parts of the UTS source code provided by Mr Feeley to Mr Li were the source code for the system library and test modules numbered UTS001, 002, 003, 005, 006, 018 and 019.
    2. Mr Feeley told Mr Li that he wanted to create software with the same general purpose as the UTS software, but with improvements; in particular, Mr Feeley wanted to create a ‘universal’ piece of software, which would handle all types of existing tests, and all potential tests in the future. Thus the overall architecture of the TestLab software is different to that of the UTS software.
    3. Although Mr Li was an experienced programmer, he was new to the industry of materials testing. In these circumstances, he found it useful to read the old code to learn something about materials testing.
    4. Mr Feeley specified the message format for information exchange between the TestLab Software and the Pavetest CDAS. The format was based on and very similar to the IMACS message format.
    5. Given that similarity, Mr Li used data structures of the UTS system library as the base for writing data structures of the TestLab system library. He also based many functions for converting data from the data structures used by the system library to the message format (ie, the functions for packaging data), and many functions that accessed those structures, on corresponding functions in the UTS system library code.
    6. Mr Li referred to the UTS source code frequently during the period November 2012 to April 2013. This includes, not only the UTS system library code, but also the source code for the test modules he was provided with. This helped him generally with the development of the TestLab software.
    7. After April 2013, Mr Li consulted the UTS source code from time to time, as needed. Although Mr Li said that he “rarely” referred to it from about April 2013, he also said that he “wasn’t referring back to it every day”. This implies that he may still have referred to it several times each week. Mr Li’s invoices dated 7 June 2013, 17 June 2013, 4 October 2013, 18 October 2013 and 23 January 2015 show that he consulted the UTS code in the weeks covered by those invoices.
    8. I do not accept Mr Feeley’s evidence that he told Mr Li, “We should not use any of this code. Any code we should write should be our own.” The fact that Mr Li used the UTS source code suggests that he was not given any such instruction. Mr Li did not give evidence of any such instruction. There is no objective (eg, documentary) evidence to suggest such an instruction was given.
    9. I do not accept Mr Feeley’s evidence that, when he received a letter from IPC Global’s solicitors during 2014, he advised Mr Li that they should not be looking at UTS source code. The fact that Mr Li continued to look at the code after 2014 suggests that this instruction was not given. Mr Li did not give evidence of any such instruction. In any event, Mr Feeley did not take steps to delete the copy of the UTS source code on Mr Li’s computer until 2015.
    10. Although Mr Li said that, other than code in the UTS system libraries, he did not copy any of the UTS code, there is evidence (discussed below) that version 2 of the TestLab software contains a number of unusual names that may suggest copying beyond the system library. Whether or not Mr Li copied parts of the UTS source code beyond the system library, I think it is likely that he consulted and derived assistance from the UTS test modules that were provided to him.
    11. I accept Mr Feeley’s evidence that there would be probably only two to three weeks’ work in coming up with a new message protocol rather than using the work he had done before at Old IPC.
    12. Mr Li gained considerable assistance in the development of the TestLab software from consulting the UTS source code. Mr Li said that, by reusing data structures used by the UTS code, and system library functions, instead of developing these from scratch, he estimated that he saved himself “about a couple of months of development time”. This does not take into account the benefit he gained generally from consulting the UTS software. I think it likely that the amount of time saved was considerably greater than two months.
25. There is a factual issue (although nothing seems to turn on it) whether the version of UTS copied by Mr Li was a Delphi 7 version or a Delphi 2009 version. The expert evidence filed by IPC Global (discussed below) shows correspondence between a Delphi 2009 version of UTS and the TestLab software. Mr Fernando gave evidence in his fifth affidavit that certain features of the TestLab software appeared only in the Delphi 2009 version of UTS. On the other hand, Mr Feeley stated in his third affidavit (prepared after much of the expert evidence was filed) that, to the best of his knowledge, all of the copied code was from a Delphi 7 version of UTS. He stated that the similarities between Delphi 2009 UTS and the TestLab software code (discussed below), in his view, merely reflect that much, although not all, of the system library code in Delphi 7 UTS can also be found in the corresponding libraries of Delphi 2009 UTS. Mr Li said during the hearing that the old code was written using Delphi 7 and, when he was writing the new code he first used Delphi 2009 and, later, XE3 (a version of Delphi). He said that in most circumstances it is possible to cut and paste source code from Delphi 7 code to Delphi 2009 code. An exception relates to Unicode conversion: in simple terms, English characters (supported by both Delphi 7 and 2009) use one byte, whereas non-English (or Unicode) characters (supported by Delphi 2009 but not Delphi 7) use two bytes; more room is therefore needed to save the extra string. In light of Mr Feeley’s and Mr Li’s evidence, it seems that, at least in the main, the version of UTS that Mr Li copied was a Delphi 7 version.
26. Mr Sinadinos stated in his second affidavit that, although he was not involved in the software or firmware development, it was his understanding that Mr Feeley had a copy of the firmware source code. In relation to the software source code, Mr Sinadinos said that he knew Mr Feeley had made copies from time to time and taken them home by way of back-up while he was doing work for Old IPC. I accept this evidence.

### March 2014: the Asset Sale Agreement

1. On 31 March 2014, the business assets of Old IPC were transferred to IPC Global under the terms of the Asset Sale Agreement, which is dated 7 March 2014. The agreement included (in clause 3.1) a sale and purchase of the “Assets”. That term was defined as meaning (among other things) “Vendor IP” which was defined as meaning all “Intellectual Property Rights” (a defined term) owned by, used by or licensed to Old IPC in the conduct of the “Business” (also a defined term). The expression “Intellectual Property Rights” was defined as meaning all intellectual property rights including in respect of (among other things) confidential information; and computer programs and copyright. The definition of “Vendor IP” also specified certain matters as being included in the definition. One of these was “any right to have information relating to the Business or the Assets (including confidential information) kept confidential”. Another was “all rights and copyright in … the UTS software”.
2. The Asset Sale Agreement also included, in clause 13.1, an assignment of all of Old IPC’s rights in “Potential Claim Contracts” including any rights to issue proceedings against any party to the Potential Claim Contracts. The expression “Potential Claim Contracts” was defined as meaning (among other things) “the employment agreement between [Old IPC] and Sinadinos” and “the consultancy agreement between [Old IPC] and Feeley”. An issue arises whether the reference to the consultancy agreement between Old IPC and Mr Feeley should be read as referring to the Aleuta Consultancy Agreement. There was no consultancy agreement between Old IPC and Mr Feeley. Although it is not necessary to decide the point (for reasons set out below), I am inclined to think that, in the factual circumstances, the reference to “the consultancy agreement between [Old IPC] and Feeley” should be read as referring to the Aleuta Consultancy Agreement.

### April/May 2015

1. Pavetest’s materials testing machines, including the TestLab software and the CDAS firmware, were marketed at the Asphaltica conference in about April or May 2015.

### July to October 2015

1. In July 2015, IPC Global obtained a copy on a USB drive of the executable version of Pavetest’s TestLab software. Mr Fernando conducted an analysis of the TestLab software. After obtaining legal advice, IPC Global engaged Mr Sizer to conduct an analysis and comparison of the UTS software and the TestLab software.
2. On 1 October 2015, Mr Sizer provided his first report (**First Sizer Report**). This was based on the object code (rather than source code) of the TestLab software, as the source code was not available at that stage.
3. On 7 October 2015, IPC Global commenced this proceeding. It sought and obtained *ex parte* interim orders on that day.
4. On 23 October 2015, interlocutory orders were made by a judge of this Court. Among other things, those orders provided for the respondents to produce to IPC Global’s solicitors the following materials (subject to a detailed confidentiality regime):
   1. a copy of the TestLab software as it existed on 7 October 2015, including its source code;
   2. a copy of each existing historical version of the TestLab software (preceding the 7 October 2015 version), including its source code;
   3. any design documentation or specification of the TestLab software.

### December 2015 to February 2016

1. Following the commencement of the proceeding, in December 2015 parts of the source code of the TestLab software were rewritten to remove or reduce the similarities with the UTS software. The new version is referred to as version 2 of the TestLab software. Version 2 of the TestLab software was released to the market in February 2016 and was rolled out to customers as a compulsory upgrade.
2. Mr Li gave evidence in his affidavit that, after these proceedings were commenced, at Mr Feeley’s instruction, he rewrote parts of the TestLab system library code; wherever practical, this involved rewriting the code inside the function, so that the function would still have the same behaviour (ie, it would still achieve the same result) but would achieve that result in a different way. Mr Li also stated that the similarities that still exist are where the internal data structures dictate the contents of the functions. Mr Li stated that he did not rewrite the internal data structures used by the system library, because in his view the form of those data structures followed from the application protocol data format; he could not see the utility in writing new data structures that would be the same as the previous data structures.
3. Mr Li said during cross-examination that, in December 2015, Mr Feeley said to him words to the effect, “We need to start changing parts of the TestLab software so it is less like the old code”. Mr Li said that he had done all of the re-writing of the TestLab software code to change it from version 1 to version 2; in creating version 2, he started with TestLab version 1 and made changes to it; and Mr Feeley indicated the names of functions and variables that needed to be changed.
4. Mr Li’s invoice dated 11 December 2015 referred to a visit to lawyers with Mr Sinadinos (for 2 hours) and then work carried out to rename, clean up and revise the TestLab software. Mr Feeley said during cross-examination that the renaming referred to in the invoice was a reaction to this proceeding. He said: “We did everything we could to make the software different; and when you’ve got a record or a structure all you can [do] is essentially rename the element of it.” In response to a further question, he said: “If your structure is dictated by the communication protocol – in this case it is – all we can do without significantly impacting on the design and our solution for the problem domain is to rename the element – the variable name, essentially, of the structure.”
5. On the basis of the evidence set out in [136]-[138] above, I make the following findings:
   1. After these proceedings were commenced, Mr Li, at Mr Feeley’s instruction, rewrote parts of the TestLab system library code. Wherever practical, he rewrote the code inside the relevant function, so that the function would still have the same behaviour (ie, it would still achieve the same result) but would achieve that result in a different way.
   2. Mr Li did not rewrite the internal data structures used by the system library, because in his view the form of those data structures followed from the application protocol data format. This was because he could not see the utility in writing new data structures that would be the same as the previous data structures.
   3. In creating version 2 of the TestLab software, Mr Li started with TestLab version 1 and made changes to it.
6. I note for completeness that IPC Global has not granted to the respondents or any of them a licence to reproduce in a material form, publish, communicate to the public or make an adaptation of the UTS software.

### Textual similarity between the UTS software and the TestLab software (version 1)

1. I will now address the degree of textual similarity (referred to in the expert evidence as “correspondence”) between the UTS software and version 1 of the TestLab software, and the significance of the source code that is the same or similar. I will set out the evidence in relation to these matters and then make findings.
2. As noted above, at the time Mr Sizer prepared the First Sizer Report, only executable files of the TestLab software were available. Mr Sizer set out at [5.3.2] of that report a table of common symbols appearing in the system libraries of the UTS software and the TestLab software. He stated that, when comparing certain system library files, “it can be seen that there are regular obvious matches between the two implementations in terms of the symbol (mostly code ‘functions’ or ‘procedures’) names. It can be stated that the name of the files is very similar, and that this is complemented by the large degree of similarity of internal symbol names”. For the purposes of preparing the First Sizer Report, Mr Sizer was provided with a summary, prepared by IPC Global (annexure “GDS-5”) which listed function names which were common to the UTS software and TestLab software system libraries.
3. Subsequently, following the commencement of the proceeding, and the order referred to above for the provision of source code for the TestLab software (subject to a strict confidentiality regime), Mr Sizer prepared a second report, dated 4 March 2016 (**Second Sizer Report**). This is the main report addressing textual similarities between the UTS software and the TestLab software (version 1). For the purpose of this report, Mr Sizer used the version of the UTS source code that he was provided with by IPC Global’s solicitors (which was a Delphi 2009 version of UTS).
4. Mr Sizer stated in [2.1] of the Second Sizer Report that the analysis process compared TestLab software source code with UTS source code with the objective of determining the degree of similarity. He explained at [4.2] that the UTS source code and the TestLab software source code folder structures and file naming conventions are somewhat different, making the simplest approach (a direct text-level comparison using a file compare utility such as “Beyond Compare”) impractical, as such utilities rely on close alignment of the text being compared. Mr Sizer also said during cross-examination that one of the tools he used was Beyond Compare, but Beyond Compare is limited in its ability to determine correspondence if there is not effectively physical alignment of rows of text in the file; relocation or reorganisation of the contents of a file will render Beyond Compare not particularly useful.
5. Annex D to the Second Sizer Report (**Annex D**) sets out, side by side, extracts from the source code from the UTS software and the TestLab software where the text is identical or similar. Mr Sizer explained during cross-examination that Annex D contains a presentation of the source code for the functions in respect of which the function names correlated as noted in the First Sizer Report; the list of these function names had been prepared by IPC Global and provided to Mr Sizer in connection with the First Sizer Report; he used these function names as an indicator as to where to search for correspondence in the source code.
6. Annex D includes (in the column headed “Comparison”) Mr Sizer’s assessment of similarity between the source code extracts, using words such as “Identical”, “Nearly identical”, “Cosmetic differences”, “Similar” and “Different”. During cross-examination, it was put to Mr Sizer that he did not comment on the significance of what he had found other than to identify whether something was identical or similar or the like. He said this was correct. It was put to him that this was something Mr Fernando did. He accepted this and said that, at this point in time, he did not attempt to determine the significance of the source code correspondence; the activity was purely to find what correspondence existed without placing any value judgment on its importance or otherwise.
7. Mr Sizer noted in [4.8] of the Second Sizer Report (as corrected during the hearing) that further analysis to uncover similarities between the UTS software and the TestLab software could be undertaken, but would incur substantial expense. Mr Sizer set out in Annex F to the Second Sizer Report a summary of possible additional analysis work.
8. Mr Sizer expressed the following opinions in [6.2]-[6.5] of the Second Sizer Report:

6.2 Analysis of UTS Source Code and TestLab Source Code confirms and strengthens the conclusions presented in section 6.3 of [the First Sizer Report], namely:

*The analysis performed by Genesys [Mr Sizer’s company] of technical similarities and differences between the deployed Pavetest software and the IPC Global UTS software leads to the conclusion that it is highly likely that Pavetest’s Testlab system design and software have been copied from, derived from, or built on, the IPC Global UTS system design and software.*

6.3 The source code analysis reveals that the software code structure of the TestLab Source Code follows closely that of the UTS Source Code. Specifically, the presence of a large number of identical procedure, function, variable and constant names puts beyond doubt the conclusion that the TestLab Source Code has evolved from the UTS Source Code.

6.4 Large and functionally significant sections of the TestLab Source Code are identical to the UTS Source Code. Other large and functionally significant sections of the TestLab Source Code are similar to the UTS Source Code. These equivalences and similarities are a sure sign that these sections of the TestLab Source Code have been created from a copy of the corresponding UTS Source Code, and subsequently modified.

6.5 It is highly likely that at some point in time which pre-dates the earliest available copy of the UTS source code, a complete copy of the UTS Source Code was taken and used as the starting point for the TestLab source code. This was subsequently modified, including file restructuring, cosmetic changes and functional changes. Regardless of these changes, it is obvious that the TestLab Source Code remains a close copy of the UTS Source Code.

1. In relation to [6.3] in the above extract, Mr Sizer said in oral evidence, in response to a question from the Court, that the “software code structure” referred to is the fine level structure, not the overall system level structure. In relation to [6.4], Mr Sizer said that, by “large and functionally significant sections”, he was referring to significant correspondence between about 20 to 50 lines of code; and that this was based on the quantum of code and making a judgment on what the code was doing. Mr Sizer said in further cross-examination that he did not express a view as to what the sections of code actually did, as, at this point in time, this would have been largely speculation on his part. In light of this evidence, and the evidence referred to in [146] above, I take Mr Sizer to be saying that the sections of TestLab source code which are identical or similar to sections of UTS source code set out in Annex D are functionally significant in the sense that they perform a role in the operation of the software, but he did not investigate what their particular function was.
2. Mr Sizer said, in response to a question from the Court, that, in considering version 1 of the TestLab software, he would say that in the order of 1,000 lines of source code corresponded with the UTS software. He said that, for the purposes of considering the proportion that this represented of the UTS software, given the duplication in UTS (where code is replicated across some test modules), he would treat this as about 1,000 lines out of perhaps 10,000 or 20,000 lines. In further cross-examination, he said that he had not done an exact count of the corresponding lines of code; he had just done a rough estimate. He accepted that he did not do a rough estimate in any of his reports.
3. Although Mr Fernando works for IPC Global, a competitor of Pavetest, he was given access to limited parts of the TestLab software (being parts identified by Mr Sizer as similar) for the purpose of commenting on the significance of those parts to the UTS software as a whole.
4. In his second affidavit, Mr Fernando referred to a comparison document created by Mr Sizer comparing extracts of source code from “UTSDefine.pas” (being part of the UTS software) and “SystemDefine.pas” (being the equivalent TestLab source code) (Exhibit “BJRF8”). Mr Fernando gave the following evidence in relation to this comparison document:

(a) The primary function of each of the common parts of the extracts is “to interface the software with the firmware in the controller”.

(b) In each case, the method by which the software interfaces with the firmware is by way of a common definition of data structures, commands and responses.

(c) The data structure design in the TestLab software is substantially identical to the corresponding parts of the UTS software; the modifications are minor.

(d) The data structure in the TestLab extract contains an identical “CommandID” field, which is part of the command and addressing scheme used by the UTS software and IMACS firmware.

(e) Those data structures are utilised within the IMACS firmware to implement the following functionality:

(i) sensor reading correction within the IMACS;

(ii) the setup and timing of transducer signal acquisition;

(iii) the description of material stimulus patterns;

(iv) the setup and control of the application of material stimulus;

(v) the synchronisation of transducer acquisition with the application of material stimulus;

(f) The functionality described in (e) above forms a significant part of the source code for the IMACS firmware.

1. For the purposes of preparing his third affidavit, Mr Fernando was provided with a confidential schedule, being a document including a copy of Annex D as the first 23 pages, and asked to annotate it. The confidential schedule with Mr Fernando’s annotations is exhibit “BJRF13” to Mr Fernando’s third affidavit (**Exhibit “BJRF13”**). In relation to this document, Mr Fernando gave evidence in his third affidavit as follows:

(a) He had only annotated the first section (pages 1 to 23) to indicate confidentiality because “the first section sets out the methods that form the core of the Delphi 2009 UTS Software”.

(b) He stated that the confidential schedule “provides a convenient way to identify most of the key parts of IPC Global’s confidential information”.

(c) He stated that, based on his experience and intimate knowledge of the UTS software, he considers the key parts of the Delphi 2009 UTS software are:

(i) the data structures common to the IMACS controller;

(ii) command and response definitions common to the IMACS controller;

(iii) sensor calibration and interpretation schemes;

(iv) functionality to acquire sensor data from the IMACS;

(v) functionality to apply material stimulus via the IMACS; and

(vi) timing and synchronisation functionality.

Mr Fernando stated that he had used a colour coding scheme to indicate each of these parts on Exhibit “BJRF13”.

(d) Mr Fernando also stated that the source code reproduced in the confidential schedule is primarily related to the UTSSystemDLL component (which includes the transducer levels screen) of the Delphi 2009 UTS software and the TestLab software equivalent. He stated that the UTSSystemDLL is the part of the Delphi 2009 UTS Software that implements the functionality described in sub-paragraph (c) above, and presents that functionality as a library of software methods that can be used to build materials testing applications powered by the IMACS controller. He stated: “It is the key content that IPC Global’s UTS software is built upon.”

1. Mr Fernando was taken through Exhibit “BJRF13” in some detail during cross-examination. He gave the following evidence (both in his affidavit and during cross-examination) in relation to that document.
   1. One of the items in Exhibit “BJRF13” is called “MakeDefaultAcqTask Descriptor”. Mr Fernando annotated this item (in the exhibit) with the comment that the source code generates a data structure used both in the IMACS firmware and the UTS software to set up the acquisition of sensor data. Mr Fernando said during cross-examination that the source code “makes a variable which is an instance of the data structure. And it sets the field in that variable to the default values”. He accepted that this constituted initialisation of a data structure. He also accepted that “what you see happening is the assignment of default values to each of [the] parameters” set out in the source code; and that this is what the “colon equals” is doing. Mr Fernando accepted that it was standard in Delphi programing to initialise all of the data structures that you have defined, and that this was just one example of doing this. It was put to him that he was not suggesting that the actual code to initialise a data structure, as such, is difficult or unconventional. He accepted that he was not suggesting this. In relation to this data structure, it was put to Mr Fernando during cross-examination that it contains all of the various parameters that are needed by the firmware, so that the firmware can acquire sensor data in accordance with the text specifications. Mr Fernando responded: “That’s correct. But it also has some other elements which tie in the IMACS firmware organisation. For example, it has event mask triggers, and they’re also used in the event generation system and they’re also used in the loading and stimulus application. And so it enables synchronisation between all the various sub-blocks in the IMACS firmware.”
   2. Another item in Exhibit “BJRF13” is called “MakeDefaultAxisControlBlock Descriptor”. Mr Fernando annotated this item with the comment that the source code generates a data structure with default values, which is mirrored in the IMACS firmware; and that it is used to set up the application of material stimulus. He accepted during cross-examination that the purpose of this source code is to initialise the main data structure that defines the operation of loading control.
   3. In relation to the item in Exhibit “BJRF13” called “DownloadAxilControl Block”, Mr Fernando annotated this item with the comment that the source code downloads a data structure which is described in both the UTS software and the IMACS firmware; this function relays this data structure to the IMACS to set up material stimulus. Mr Fernando accepted during cross-examination that this is a function that sends the control block data parameters to the firmware; and that it was not sending the default values that have been initialised – it is sending the values that have been determined elsewhere in the software.
   4. The first item in Exhibit “BJRF13” is called “AcqCalcOptimumClusterSize”. In his annotations, Mr Fernando commented that the source code set out alongside this item is an algorithm used to interface/communicate with and configure the IMACS in relation to acquiring sensor data from the IMACS controller; specifically, the number of sensor samples to transport in one data packet from the IMACS. He accepted during cross-examination that the source code works out how many sensor samples it wants the firmware to send back in each packet; that he had described this as an algorithm because there is a computation being made; and that the computation is not complex.
   5. In relation to the item “LevelsDisplay1Setup” in Exhibit “BJRF13”, Mr Fernando accepted during cross-examination that this function displays transducer reading levels to the user. He said:  “So at any point when the IMACS is connected and you have UTS tests loaded, you can bring up the level screen, which will tell you what every transducer is reading at that time. … [I]t’s sometimes very useful for setting up test jigs and just moving the actuator and making sure everything is set up right before starting a test.” It was put to Mr Fernando during cross-examination that, although this is a relatively long piece of code, it is just setting up a Delphi bar chart. He answered that it is setting up a Delphi bar chart using the information in the transducer allocations table.
   6. In relation to the item “LevelsSetClearChannelZeroSuppression” in Exhibit “BJRF13”, Mr Fernando accepted during cross-examination that this is a function that you can use to set a transducer level to zero.
   7. It was put to Mr Fernando during cross-examination that essentially all the information you need to put into the parameters of your data structure are set out in the test standard. He said:  “For the most part, yes.” Mr Fernando also said:  “Typically we get those parameters from the test standard, but … there is a level of customisation that we offer as well”.
2. Mr Feeley commented in his third affidavit on Annex D and Exhibit “BJRF13”. He stated as follows:
   1. Annex D includes a total of about 500 lines of UTS source code from the system library. The code consists of function calls, which are the interfaces between system library code and the main program.
   2. Save that, in Mr Feeley’s view, it is clear that the copied code was from Delphi 7 UTS rather than Delphi 2009 UTS, he agreed with Mr Fernando’s affidavit evidence that the UTS code in Annex D is primarily related to the UTS system library component, including the transducer levels screen.
   3. UTS001.pas, UTS002.pas, UTS006.pas, UTS018.pas and UTS032.pas are the main application modules for the Delphi 2009 UTS tests, which consist of about 26,000 lines of code. (It appears that this is a total figure. There was other evidence – in Mr Fernando’s fifth affidavit – that on average each UTS test module has about 6,000 lines of source code.) The main application program of the TestLab software is TestManager.pas. Mr Feeley performed a “Beyond Compare” comparison between each of those UTS files and TestManager.pas. The results show that no lines are duplicated, other than for a function which is a third party code.
   4. Mr Feeley denied that there are similarities between UTS and TestLab software source code, other than what Mr Sizer reported.
3. Mr Feeley responded in his third affidavit to Mr Fernando’s evidence regarding the significance of the code in Annex D and Exhibit “BJRF13”. Mr Feeley stated as follows:
   1. He fundamentally disagrees with Mr Fernando that the UTS system library code that has been copied constitutes the key parts of the UTS software.
   2. The application test modules of the UTS software use the UTS library functions to implement a working materials testing application software. It is these application modules that define the parameters, necessary settings and calculations to perform the materials test and present the result.
   3. The high level application modules are, in Mr Feeley’s opinion, based on his experience, the “core” intelligence defining the UTS functionality and the valuable intellectual property in UTS.
   4. The UTS system library presents a layer of communication and setup functions to the UTS application, and provides some services. In Mr Feeley’s opinion, this is not the core intellectual property of UTS; it is, rather, in the nature of routine “infrastructure” code.
   5. In Mr Feeley’s view, much of what was copied consists of extremely rudimentary functions that any competent programmer should in any case be expected to know or easily work out.
4. Mr Feeley responded in his third affidavit to comments made by Mr Sizer in the Second Sizer Report regarding the absence of any pre-August 2014 versions of the TestLab software. Mr Feeley stated that all versions of the code that still existed in October 2015 were discovered; it has been his practice to routinely delete old versions of source code, going back more than a month, in order to save hard disk space; the approach of only keeping recent code versions is consistent with “extreme programming” practice.
5. In relation to similarities in data structures, Mr Feeley stated in his third affidavit that the similarities do not imply similarity between the program algorithmic code implementation; and the similarities are (at least in some instances) dictated by the problem domain.
6. Mr Sizer’s fourth report is dated 4 October 2016 (**Fourth Sizer Report**). In this report, Mr Sizer expressed a number of opinions regarding the potential for reverse engineering the communications protocol (which had been discussed in Mr Feeley’s affidavit evidence), and the significance of the communications protocol. Mr Sizer gave the following evidence:

(a) He generally agreed with Mr Feeley in relation to the non-proprietary nature of the lower layers of the SNAP protocol.

(b) While the lower “message transport” layers of the UTS to IMACS protocol are based on public-domain SNAP foundations, the digitally encoded messages conveyed by the transport mechanism are highly application-specific. Unless copied, it is most likely that the structure and content of these messages would be unique to UTS and IMACS.

(c) Although the IPC Global and Pavetest Systems implement functions for the testing of asphalt specimens, in accordance with the requirements of a range of regulatory standards, and thus the functions performed by each system are closely aligned with the requirements of the standard and hence with each other, this does not mean that the systems are necessarily closely aligned in aspects such as hardware or software architecture, user interfaces or communications protocols between system elements; these can be freely selected by the system designer and can be quite different for two systems which implement similar tests.

(d) Mr Sizer expressed the opinion that, even with his prior knowledge of the UTS to IMACS protocol, it is unlikely that Mr Feeley would be able to reverse engineer and reproduce the exact protocol in full detail, without access to UTS or IMACS source code, or a detailed specification documentation; in any case, the effort involved would be likely to exceed the value gained as an aid to developing an independent materials test system.

(e) In Mr Sizer’s opinion, by copying and re-using the UTS to IMACS protocol for TestLab to CDAS communications, Pavetest has gained significant benefit.

(f) In Mr Sizer’s opinion, the communications protocol and its underlying data structures form the heart of the IPC Global and Pavetest systems, and the underlying software is arguably the single most important part of the source code, as it is intimately linked with the functions of the system and the way in which they are implemented.

1. Responding to Mr Feeley’s affidavit evidence regarding the relative significance of the test modules vis-à-vis the DLL, Mr Fernando said in his fifth affidavit: “Further, no UTS test module operates without the DLL and there is a significant level of similarity and source code reproduction between each test module. For these reasons, in my view and based upon my experience, it is the DLL that is the core of the UTS software, not each individual test module.” Mr Fernando also stated in his fifth affidavit: “In my view and based on my experience as a software developer who has worked with the DLL, the task of recreating the DLL from scratch would be an immense task. Once such a library has been developed and is functional it, in my experience, then becomes relatively straightforward to code for a particular materials test.”
2. Mr Feeley in oral evidence referred to a statement made by Mr Sizer to the effect that you would be better off spending the time developing your own message format (rather than reverse engineering it) as you could do it in a similar time. Mr Feeley said that he agreed with that.
3. On the basis of the evidence set out in [142]-[161] above, I make the following findings.
   1. Annex D sets out extracts from the source code of the UTS software and the TestLab software. In most of the extracts, the source code is identical or similar. It is clear from the evidence of Mr Li that this is the result of copying from the UTS software to the TestLab software, Mr Li having been provided with a copy of the system library and several of the test modules of the UTS software by Mr Feeley.
   2. Annex D comprises 23 pages. The estimates varied of the number of lines of source code set out on these pages. Mr Feeley, in affidavit evidence, said about 500 lines. Mr Sizer, in oral evidence, said about 1,000 lines. Based on my review of the document, my estimate is that it comprises about 800 lines of UTS source code in respect of which the TestLab source code is identical or similar (that is, not counting source code labelled as different).
   3. The UTS source code in Annex D is primarily related to the UTS system library component, including the transducer levels screen.
   4. The parts of the UTS software that have been copied (ie, those set out in Annex D), in general terms, contain code which initialises a data structure or code which sets out a function relating to the interface or communication between the software and the firmware. In general terms, in each case, the code reflects the message format specified by Mr Feeley to Mr Li, which was based on and very similar to the IMACS message format he had developed at Old IPC.
   5. I do not accept that the test modules constitute the “core intelligence” and the system library is merely “infrastructure”. Both the test modules and the system library play essential and significant roles in the operation of the software.
   6. The parts of the UTS software that have been copied constitute a functionally significant part of the UTS software. I accept Mr Fernando’s evidence, referred to in [153]-[154] above, as to the functionality of the source code set out in Exhibit “BJRF13”. This evidence was supported by the independent expert evidence of Mr Sizer, as set out in [148]-[149] above. There was no contrary independent expert evidence.
   7. A number of the copied sections contain code which would not be described as difficult or unconventional.

(h) Although the UTS software comprises about 250,000 lines of source code, this includes about 40 different test modules, and there is a significant amount of duplication of code across the test modules. This needs to be borne in mind when comparing the quantum of copied code with the size of the UTS software as a whole.

### Textual similarity between the UTS software and the TestLab software (version 2)

1. I have dealt with the development of version 2 of the TestLab software earlier in [135]-[139] above. Among other things, I found, in [139] above, that in creating version 2 of the TestLab software, Mr Li started with TestLab version 1 and made changes to it. I now consider the degree of textual similarity between the UTS software and the TestLab software (version 2).
2. Mr Sizer’s third report is dated 21 July 2016 (**Third Sizer Report**). It addresses (among other things) correspondence between the UTS software and the TestLab software (version 2). Mr Sizer expressed his conclusions at [8.1.2]-[8.1.3] of the report as follows:

8.1.2 TestLab V2.0 has evolved significantly from [TestLab version 1] which was the subject of the comparison in [the Second Sizer Report]. This evolutionary process has progressed to the extent that only relatively small sections of the TestLab V2.0 source code are an obvious exact or close match to sections of the UTS Source Code, based on a simple side-by-side text comparison.

8.1.3 I expressed the following opinion in [the Second Sizer Report] at paragraph 6.5

*It is highly likely that at some point in time which pre-dates the earliest available copy of the UTS source code, a complete copy of the UTS Source Code was taken and used as the starting point for the TestLab source code. This was subsequently modified, including file restructuring, cosmetic changes and functional changes. Regardless of these changes, it is obvious that the TestLab Source Code remains a close copy of the UTS Source Code.*

This remains my opinion, except that the process of modifying of the UTS Source Code to produce the TestLab V2.0 Source Code has progressed to the point where much of the similarity evident between UTS and earlier versions of TestLab has been removed or obscured. Accordingly, TestLab V2.0 is a less close copy of UTS than was TestLab [version 1]. However, significant parts of the UTS Source Code remain evident in the TestLab V2.0 Source Code, particularly between UTS *System/UTSSystem.pas* and TestLab *Ssystem/Comm/TaskManager.pas*. The fundamental structural similarity between TestLab V2.0 and UTS source code remains.

1. In relation to the Third Sizer Report, Mr Sizer said during cross-examination that the equivalence or similarity of names and functions as they were organised was still evident in the TestLab software version 2. (This related to the functions that had been the subject of his comparison in relation to version 1.) Mr Sizer said during cross-examination that, to the extent that he commented (in the Third Sizer Report) on the significance of the source code, he was relying on Mr Fernando’s affidavit evidence.
2. In section 2.8 of the Fourth Sizer Report, Mr Sizer gave some further evidence regarding correspondence between UTS software and the TestLab software (version 2), in light of Mr Feeley’s explanation of certain structural aspects in his affidavit evidence. Mr Sizer noted at [2.8.1] that Mr Feeley drew attention to UTS001.pas, UTS002.pas, UTS006.pas, UTS018.pas and UTS032.pas being the main application modules of UTS, with TestManager.pas being the main application program of the TestLab software. At [2.8.5], Mr Sizer wrote that, based on information provided by Mr Feeley about the relationship between TestLab’s TestManager.pas and the UTS files identified above, Mr Sizer had undertaken further analysis of the files in question to determine if there were similarities; specifically, he undertook analysis to determine the number of words and text strings appearing in these particular TestLab software files in comparison with the UTS files referred to above. The results of this analysis were presented in Annex C to the report. Having excluded “key words” associated with the Delphi programming language and common words, Mr Sizer found that there were a large number of unusual words and text strings in both the UTS and the TestLab software files that he analysed. Mr Sizer expressed the following opinion in [2.8.7] of the Fourth Sizer Report:

In my opinion, the occurrence of a large number of unusual words and text strings in UTS and TestLab V2.0 source code files comprises evidence of initial copying, and of residual artefacts of the copying process. The likelihood that two independent developers would choose so many identical but unusual names for procedures, functions, variables and constants is negligible.

1. Mr Sizer accepted during cross-examination that it was possible that some of the words he had identified in this part of his evidence could be words used in Delphi; it was possible that some of the similarities could relate directly to the system library similarities he had identified; and it was possible that some could relate to third party functionality. As to whether some similarities could relate to the problem domain, he said that, while this was possible, in his view it is less likely when one considers the exact nature of the words and the correspondence of the concatenated words and their capitalisation.
2. In response to that evidence, Mr Feeley, in his fourth affidavit, denied that there was any copying of the UTS outside the DLL (and the communications protocol and data structures that it uses). Mr Feeley was taken in cross-examination to a document (Ex A19) which conveniently set out in table form Mr Feeley’s comments on Mr Sizer’s evidence to the effect that there were a large number of unusual words and text strings in both UTS and the TestLab software files. (I note that the document also contains some columns which were put forward by IPC Global merely as submissions rather than evidence.) Mr Feeley was taken to a number of the similarities in names. He said that, when he was at Old IPC, he had adopted a naming convention of using a capital letter for the first letter of each word and that Mr Li had followed the same convention. He said that Mr Li had obtained some names from “the protocol document”. In relation to one name (which appeared in UTS and also in TestLab version 2), Mr Feeley said that, “Obviously … that’s a name that we decided … if we changed it, it would detract too much from the problem domain that we’re trying to solve”. In relation to a significant number of names, Mr Feeley had commented (as reflected in Ex A19) that this was a “coincidence”. However, he accepted during cross-examination that in these cases, the name may have come from Mr Li’s checking and revisiting the old code (that is, the UTS source code).
3. On the basis of the evidence set out in [164]-[168] above, I make the following findings.
   1. I accept Mr Sizer’s evidence that version 2 has evolved significantly from version 1 and that this evolutionary process has progressed to the extent that only relatively small sections of the TestLab version 2 source code are an exact or close match to sections of the UTS source code. I also accept his evidence that much of the similarity evident between UTS and version 1 of TestLab has been removed or obscured.
   2. I accept Mr Sizer’s evidence, based on his review of UTS001.pas, UTS002.pas, UTS006.pas, UTS018.pas and UTS032.pas (being the modules identified by Mr Feeley as the main application modules of UTS) and TestManager.pas (being the program identified by Mr Feeley as the main application program of the TestLab software), that there are a large number of unusual (or particular) words and text strings in both UTS and TestLab version 2. I think this unlikely to be a coincidence, at least in most cases. I think it more likely that these similarities flowed from the copying of data structures and other functions from the UTS system library.

## Copyright

### Applicable principles

1. Section 10(1) of the *Copyright Act* provides that a “literary work” includes a computer program or compilation of computer programs. The expression “computer program” is defined in the same section as meaning “a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result”. Copyright subsists in an unpublished original literary work of which the author was a qualified person at the time when the work was made or, if the making of the work extended over a period, was a qualified person for a substantial part of that period: s 32(1). Copyright subsists in an original literary work which has been published or, if copyright subsisted in the work immediately before its first publication, copyright continues to subsist in the work, subject to the factors set out in s 32(2)(c), (d) and (e).
2. A literary work is *made*, relevantly, when the work is first reduced to writing or to some other material form: s 22(1). Material form, relevantly, in relation to a work, includes any form (whether visible or not) of storage of the work or a substantial part of the work: s 10(1). The combined effect of s 32(1), s 22 and the definition of “material form” is to provide for subsistence of copyright in an unpublished literary work at the time or period of first fixation in a material form by a qualified person, whether then “visible or not”: see *IceTV Pty Ltd v Nine Network Australia Pty Ltd* (2009) 239 CLR 458 at [146] per Gummow, Hayne and Heydon JJ. The first publication occurs upon supplying a reproduction of the work to the public (that is, the whole work not just a substantial part of it): s 29(1), s 29(2) (cf, s 14(1)): see *JR Consulting & Drafting Pty Ltd v Cummings* (2016) 329 ALR 625 at [253] per Bennett, Greenwood and Besanko JJ.
3. Section 36(1) of the *Copyright Act* relevantly provides that copyright in a literary work is infringed by a person who, not being the owner of the copyright, and without the licence of the owner of the copyright, does in Australia, or authorises the doing in Australia of, any act comprised in the copyright. Two such acts comprised in the copyright in the case of a literary work include reproducing the work in a material form and communicating the work to the public: s 31(1)(a)(i) and (iv). It is not necessary that those acts be done in respect of the whole of the literary work. That is because a reference to the doing of an act in relation to a work includes a reference to the doing of that act in relation to a substantial part of the work: s 14(1)(a). In the context of infringement, the test directs attention to the degree of originality in the *expression* of the part of the work reproduced: *IceTV Pty Ltd v Nine Network Australia Pty Ltd* (2009) 239 CLR 458 at [40] per French CJ, Crennan and Kiefel JJ. In the same case, Gummow, Hayne and Heydon JJ said (at [157]) that the statutory requirement that the part of the work taken must be substantial assumes that there may be some measure of legitimate appropriation. Their Honours also said (at [158]):

The case law does disclose that special difficulty has been encountered in considering the relationship between the phrase “a substantial part” in s 14(1) of the Act and the definition in s 10(1) of that species of “literary work” which is a “computer program”, being: “a set of statements or instructions to be used directly or indirectly in a computer *in order to bring about a certain result*.” (Emphasis added.) The phrase emphasised suggests the importance of function, although this is usually encountered in patent and designs law, rather than in the “traditional” law of copyright respecting original literary works.

1. The relevant principles were recently considered by a Full Court of this Court in *JR Consulting & Drafting Pty Ltd v Cummings* (2016) 329 ALR 625 at [246]-[279]. That case was also concerned with the provisions relating to computer programs. In that case, Bennett, Greenwood and Besanko JJ said in relation to infringement at [267]-[272]:

267 Questions of originality, however, arise not only in relation to subsistence of copyright in a work but also *infringement* of the work in suit. This is especially so when the question is whether a *substantial part* of the work in suit has been reproduced, as such an analysis necessarily engages a consideration of, importantly, the *quality* of the part of the work alleged to have been reproduced. In this context, the authorities direct attention to the “critical question” of the “degree of originality” in the “*expression* of the part of the work reproduced”: *IceTV* at [40] and [52]. The fact that a part reproduced “originates” from the author of that part does not, of itself, mean that it is necessarily a substantial part of the whole work because, when giving attention to the *quality* of the part taken, it may be there has been reproduction of “something that is itself ***largely*** unoriginal” [emphasis added]: *Autodesk Inc v Dyason [No 2]* (1993) 176 CLR 300 at 306, Mason CJ (in dissent), a view affirmed and adopted, however, by Gleeson CJ, McHugh, Gummow and Hayne JJ in *Data Access v Powerflex* at [83] to [87], and particularly at [84]. Consideration of the skill and labour of the author “may assist” in answering the critical question of the degree of originality in the expression of the part of the work taken because the circumstance that “the creation of a work required skill and labour may indicate that the particular form of expression adopted was ***highly*** original” [emphasis added]: *IceTV*, French CJ, Crennan and Kiefel JJ at [52].

268 It follows that in determining whether the *quality* of what is taken *makes* it a substantial part of the copyright work in suit, as the source of the rights, it is important to enquire into the importance which the taken portion bears in relation to the work as a whole. Is it an essential or material part of the work in suit?: *Autodesk Inc v Dyason [No 2]*, Mason CJ at 305, as approved in *Data Access v Powerflex* at [83] to [87].

269 In *Data Access v Powerflex* at [84], Gleeson CJ, McHugh, Gummow and Hayne JJ observe that “in determining whether something is a reproduction of a substantial part of a computer program, the ‘essential or material features of [the computer program] should be ascertained by considering the originality of the part allegedly taken’”: quoting Mason CJ in *Autodesk Inc v Dyason [No 2]* at 305. Since the statutory definition of a computer program is a “set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result”, it follows that in determining whether there has been a reproduction of a substantial part of the computer program, it would be necessary to examine what was allegedly taken, the set of statements or instructions comprehended by what was taken and compare it with the original, as part of the enquiry of originality for the purposes of infringement.

270 Although the statutory definition of computer program in *Data Access v Powerflex* was in different terms, the nature of the enquiry, as a matter of principle, remains the same adapted however to the language of the definition: see *Data Access v Powerflex* at [84] and [85].

271 Moreover, if a person does no more than reproduce parts of a program which are “data” or “related information” and which are not relevant to the statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result, the person will be unlikely to have reproduced a substantial part of the computer program: *Data Access v Powerflex* at [86].

272 As to the notion of reproduction (leaving aside for the moment any question of the element of causal connection), the copyright owner must show a sufficient degree of “objective similarity” between the work in suit and the contended infringing work.

1. The test of infringement in the context of a computer program was also discussed by Bennett J in *CA Inc v ISI Pty Ltd* (2012) 95 IPR 424; [2012] FCA 35 at [173]-[179]. Her Honour said (at [178]-[179]):

178 It is apparent from *IceTV* and *Data Access* that special considerations may arise when considering whether or not a substantial part of a “computer program” has been taken. In *IceTV*, Gummow, Hayne and Heydon JJ referred to the importance of function in the determination of substantial part for “computer programs”. Although the current definition of “computer program” introduced by the *Copyright Amendment (Digital Agenda) Act 2000* (Cth) does not expressly refer to “function”, unlike the former definition introduced by the *Copyright Amendment Act 1984* (Cth), their Honours observed at [158] that the current definition nevertheless has a functional aspect.

179 In order to determine whether a substantial part of a computer program has been reproduced, the question is not whether one necessary integer of a computer program is present (*IceTV* at [159] per Gummow, Hayne and Heydon JJ; *Data Access* at [85]-[87] per Gleeson CJ, McHugh, Gummow and Hayne JJ). In *IceTV*, their Honours said at [159] that what is required is ‘*the need for some process of qualitative abstraction of the material features of the computer program in question*’. This was in the context of a reaffirmation that what is entitled to protection is not the idea of the author but the fixed expression of that idea.

### Application of principles to the facts

1. Relevantly for the copyright case, the respondents concede that:
   1. copyright subsists in the relevant versions of the UTS software, and is owned by IPC Global;
   2. substantial parts of the UTS software were copied in late 2012, by copying UTS source code and executable code onto Mr Li’s computer; and
   3. the acts were done on behalf of Pavetest, and authorised by Mr Sinadinos and Mr Feeley.
2. In light of the findings set out above and these concessions, it is established that Pavetest infringed IPC Global’s copyright in the UTS software by Mr Feeley copying the source code for the system library (or DLL) and several test modules (namely test modules UTS001, 002, 003, 005, 006, 018 and 019) of the UTS software onto the computer provided to Mr Li in or about November 2012; and that Mr Sinadinos and Mr Feeley authorised the infringement.
3. The next issue to be considered (which was the main issue agitated at trial) is whether version 1 of the TestLab software reproduced a substantial part of the UTS software.
4. The respondents submit that version 1 of the TestLab software does not constitute reproduction of a substantial part of the UTS software as a whole, for the following reasons, in summary:
   1. Copying of the UTS software was confined to its system library, which is common code used by the various UTS high level application modules, and objective similarity between the respective programs is limited accordingly. The TestLab program was not otherwise copied from, and is not otherwise objectively similar to, the UTS program.
   2. Objective similarities are thus confined to parts of UTS which are supporting infrastructure to the main program, and not its core intelligence.
   3. The similarities to the UTS system library amount to about 500 code lines from a library of about 6,000 code lines, which is itself quite a small minority of the whole code base.
   4. Further, the copied code is largely if not entirely unoriginal and/or relates to inessential features of UTS.
   5. The copying was not otherwise of original expression of the statements and instructions of a computer program (ie, it was not of functional code) but was instead of other aspects of a software interface, specifically some data structures used by UTS for information exchange with the firmware, which were derived from a shared message format.
   6. Further, the relevant data structures, and the message format they were derived from, were largely if not entirely unoriginal, as they resulted from choices about how to combine into manageable units the various types of information that had to be exchanged with the controller firmware, and the choices of those structures and their parameters were largely or substantially dictated by the requirements of the problem domain.
5. In my view, for the reasons that follow, version 1 of the TestLab software reproduces a substantial part of the UTS software. (It follows that it is established that Pavetest infringed IPC Global’s copyright in the UTS software by the creation of version 1 of the TestLab software and that Mr Sinadinos and Mr Feeley authorised the infringement.) For the purposes of this analysis I will focus on the sequences of source code set out in Annex D (and in the first 23 pages of Exhibit “BJRF13”). I note at the outset that there is objective correspondence between most of the sequences of UTS source code and TestLab source code set out, side by side, in that document. As identified by Mr Sizer in Annex D, in most cases, the sequences of source code are identical or similar. Further, there is no question that the TestLab source code was copied from the UTS source code: Mr Li was provided with UTS source code by Mr Feeley and copied the sequences of code when writing the TestLab software.
6. First, there is originality in the *expression* of the sequences of UTS source code set out in Annex D. The fact that Mr Li was able to rewrite most of these passages of source code as between version 1 and version 2 of the TestLab software demonstrates that there is room for choices to be made (and judgment applied) as to how the function is expressed. In relation to the sequences of source code that initialise data structures, it may be accepted that, once the communications protocol document has been developed, the content or composition of the data structures flows from that document. But the evidence generally indicates that choice and judgment is involved in developing the communications protocol document and thus, by extension, the content or composition of the data structures. In these circumstances, there is originality in the form of expression of the sequences of source code that initialise the data structures, taking into account the choice and judgment involved in developing the communications protocol document. The point may be illustrated in this way. In the present case, different people were involved in developing the communications protocol document and writing the source code for the UTS software: Mr Feeley developed the communications protocol document and Mr King and Mr de Vos wrote the source code. However, both these tasks could have been undertaken by the one person. In such a case, it may more readily be seen that there is originality in the form of expression of the sequences of source code that initialise a data structure. I do not think there is a substantive difference for present purposes between the two situations. The position may be different if the composition of the data structures was dictated by the ‘problem domain’ of materials testing, including, for example, the requirements of particular test standards. But the evidence does not establish this. Rather, the evidence is to the effect that, while the communications protocol document and data structures are heavily influenced by the problem domain, there are still choices to be made and judgment to be applied. For these reasons, in my view, the sequences of source code that initialise data structures possess sufficient originality of expression to be capable of constituting a substantial part of the UTS software.
7. The respondents submit that the copied code is largely, if not entirely, unoriginal and/or relates to inessential features of UTS. I do not accept this submission. For the reasons given in the preceding paragraph, the form of expression of the sequences of source code set out in Annex D involves choice and judgment. The sequences which comprise functions other than generating a data structure can be expressed in different ways, as demonstrated by the changes made as between version 1 and version 2. The sequences that initialise data structures involve choice and judgment as to the variables and constants comprising the data structures. I do not accept that the composition of the data structures is dictated by the problem domain. As for the proposition that the sequences of source code relate to inessential features of UTS, I will discuss this below.
8. The respondents submit that the copying was not otherwise of original expression of statements and instructions of a computer program (ie, it was not functional code) but was instead of other aspects of a software interface, specifically some data structures used by UTS for information exchange with the firmware, which were derived from a shared message format. I do not accept that the sequences of source code set out in Annex D are not “functional code”. While the communications protocol document (see [67] above) would not constitute a “computer program”, in my view the source code relating to the message format (see [68] above) forms part of a computer program, namely a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result (see [170] above). Further, even in the cases where the sequence initialises a data structure, it is still performing a function, namely the generation of that data structure. I also do not accept that the source code for such sequences lacks originality because the data structure is derived from the communications protocol document. As discussed above, the formulation of the communications protocol document involved the application of judgment; it was not dictated by the problem domain (although it was heavily influenced by it). This may be taken into account in considering whether the source code for sequences generating data structures possesses the necessary degree of originality. It may be different if the problem domain (an objective, external matter) dictated the composition of the data structures; but that is not the case.
9. The respondents also submit that the relevant data structures, and the message format they were derived from, were largely if not entirely unoriginal, as they resulted from choices about how to combine into manageable units the various types of information that had to be exchanged with the controller firmware, and the choices of those structures and their parameters were largely or substantially dictated by the requirements of the problem domain. It may be accepted that the choices of variables and constants to include in the data structures was heavily influenced by the problem domain. But I think it is nevertheless clear on the evidence that choices are involved and judgment applied in the process of determining which variables and constants to include in particular data structures. In these circumstances, the resultant source code is capable of possessing, and does possess, a sufficient degree of originality.
10. Secondly, the parts of the UTS software set out in Annex D, in general terms, contain code which initialises a data structure or code which sets out a function relating to the interface or communication between the software and the firmware: see [125](e) and [162](d) above. The evidence of Mr Fernando and Mr Sizer establishes, in my view, that these sequences of source code play a functionally significant role in the operation of the UTS software as whole: see [162] above. To the extent that these passages of source code initialise data structures, the evidence generally indicates that these data structures play an important role in the operation of the UTS software. To the extent that these passages of source code set out other functions relating to the interface or communication between the software and the firmware, these functions appear to play an important role (see [153]-[154] above and also Mr Fernando’s annotations on Exhibit “BJRF13”, including the colour coding applied by Mr Fernando.) I have found that, in a number of cases, the source code is not difficult or unconventional (see [162](g) above). But the point being addressed at this stage of the analysis is whether the sequences of source code set out in Annex D (and Exhibit “BJRF13”) are functionally significant to the UTS software as a whole. In my view, these sequences of source code are functionally significant. They are not merely, for example, collections of data or related information (cf [173] above); rather, they set out functions which have an important role to play in the operation of the software.
11. The respondents submit that copying of the UTS software was confined to its system library, which is “common code” used by the various UTS high level application modules, and that objective similarities are thus confined to parts of UTS which are “supporting infrastructure” to the main program and not its “core intelligence”. It needs to be borne in mind that the issue concerns the functionality and significance of the copied parts of the UTS software rather than the functionality and significance of the system library as a whole. To the extent that it bears on this issue, I do not accept that the test modules are the “core intelligence” and the system library is merely “infrastructure” (see [162](e) above). I have also concluded that both the test modules and the system library play essential and significant roles in the operation of the software (see [162](e) above).
12. As noted above, the respondents submit that the copied code is largely, if not entirely, unoriginal and/or relates to inessential features of UTS. I have dealt with the first part of this submission already. As for the proposition that the copied code relates to inessential features of the UTS software, I do not accept this submission. The evidence of Mr Fernando, supported by the evidence of Mr Sizer, establishes that the parts of the UTS software set out in Annex D and Exhibit “BJRF13” constitute a functionally significant part of the UTS software: see [162] above. It follows from that finding that I do not accept that the copied code relates to inessential features of UTS.
13. Thirdly, although the quantum of source code set out in Annex D is small relative to the total size of the UTS software, the cases referred to above make clear that the emphasis is on a qualitative rather than quantitative assessment of substantiality. Annex D contains approximately 800 lines of UTS source code in respect of which the TestLab source code is identical or similar (see [162] above). On the basis that I have found that the copying was (at least in the main) from a Delphi 7 version of UTS (rather than a Delphi 2009 version), it is appropriate to have regard to the size of the Delphi 7 version of UTS. The Delphi 7 version of UTS had about 250,000 lines of source code (see [62] above). However, as noted above, it needs to be borne in mind that this includes about 40 test modules and there is a significant amount of duplication of code across the test modules. The evidence did not go into detail about the amount of duplication. Mr Sizer considered that, for the purposes of making this type of comparison, he would treat the UTS software as about 10,000 to 20,000 lines (see [150] above). However, it is not clear how he came up with these figures, which were provided in response to a question from the Court. I note that Mr Feeley identified five test modules in UTS as being “the main application modules” and said that they consist of about 26,000 lines of code (see [155] above). It may be that, taking a practical approach, one should focus on the main application modules. In any event, the evidence establishes that there is significant duplication across test modules. Having regard to this duplication, I consider the quantity of copied source code to be sufficient to constitute a substantial part.
14. No point was taken in the respondents’ written closing submissions that IPC Global did not have the right to pursue this claim. In any event, I consider that IPC Global acquired the right to bring this claim even though it related to a past infringement as at the time of the Asset Sale Agreement. In my view, the provisions of that agreement described in [128]-[129] above were sufficiently broad to transfer such rights from Old IPC to IPC Global.

## Breach of confidence

### Applicable principles

1. As Finn, Sundberg and Jacobson JJ stated in *Optus Networks Pty Ltd v Telstra Corporation Ltd* (2010) 265 ALR 281 at [39], the cause of action for breach of confidence has four elements:
   1. the information in question must be identified with specificity;
   2. it must have the necessary quality of confidence;
   3. it must have been received by the respondent in circumstances importing an obligation of confidence; and
   4. there must be an actual or threatened misuse of the information without the applicant’s consent.
2. The Full Court referred to *Smith Kline & French Laboratories (Auft) Ltd v Secretary, Dept of Community Services and Health* (1990) 22 FCR 73 at 87 per Gummow J. See also *Coco v AN Clark (Engineers) Ltd* [1969] RPC 41 at 47-48; *Australian Medic-Care Company Ltd v Hamilton Pharmaceutical Pty Ltd* (2009) 261 ALR 501 at [632]-[634] per Finn J; *Leica Geosystems Pty Ltd v Koudstaal (No 3)* (2014) 245 IR 422; [2014] FCA 1129 at [47]-[48] per Collier J.
3. In relation to the first of the above elements, in *O’Brien v Komesaroff* (1982) 150 CLR 310, Mason J (with whom Murphy, Aickin, Wilson and Brennan JJ agreed) said (at 326) that “the respondent has consistently failed to identify the particular contents of the documents which he asserts constitute information the confidentiality of which he is entitled to protect. The consequence is that he has failed to formulate a basis on which the court could grant him relief on the assumption that some part or parts of the documents constitute confidential information”.
4. In *Leica Geosystems*, Collier J referred to the observations of Kirby P in *Wright v Gasweld Pty Ltd* (1991) 22 NSWLR 317, where his Honour identified certain factors relevant to a determination as to whether information is of a confidential nature. Kirby P said (at 334):

Determining what is confidential involves a decision on a question of fact in each case where that quality is asserted. Considerations which courts have found to be relevant, in particular cases, in determining this question include:

(a) The fact that skill and effort was expended to acquire the information …

(b) The fact that the information is jealously guarded by the employer, is not readily made available to employees and could not, without considerable effort and/or risk, be acquired by others …

(c) The fact that it was plainly made known to the employee that the material was regarded by the employer as confidential …

(d) The fact that the usages and practices of the industry support the assertion of confidentiality … and

(e) The fact that the employee in question has been permitted to share the information only by reason of his or her seniority or high responsibility within the employer’s organisation …

(Case references omitted.)

1. In relation to the third element set out above (the information must have been received by the respondent in circumstances importing an obligation of confidence), in *Del Casale v Artedomus (Aust) Pty Ltd* (2007) 165 IR 148; [2007] NSWCA 172, Campbell JA (with whom McColl JA agreed), said (at [133]):

The circumstances have to be such that the court can conclude it would have been clear to a reasonable person, on reasonable grounds, that he or she was not free to deal with the information as his or her own, or could only deal with it subject to certain limitations.

1. The authorities recognise that confidential information can be of different grades of secrecy. In some cases, the respondent contends that, even without access to the confidential information, he or she would have developed the relevant product. In this context, the “springboard doctrine” has been invoked. The doctrine was discussed by Gordon J in *Zomojo Pty Ltd v Hurd (No 2)* (2012) 299 ALR 261. Her Honour said (at [201]-[202]):

201 Equity will restrain a former employee who seeks to use an employer’s information as a ‘springboard’ to gain a head start, even where that information is capable of being independently ascertained: *Terrapin Ltd v Builders Supply Co (Hayes) Ltd* [1967] RPC 375 at 391; *Seager v Copydex Ltd* [1967] 1 WLR 923 at 931-2; *Mense & Ampere Electrical Manufacturing Co Pty Ltd v Milenkovic* [1973] VR 784 at 791 and *Deta Nominees Pty Ltd v Viscount Plastic Products Pty Ltd* [1979] VR 167 at 194-5.

202 The springboard doctrine seeks to prevent the misuse by one party of another’s confidential information in order to bring out its own product in a manner or time that it would not otherwise have been able to achieve: *Terrapin* at 391; *Aquaculture Corporation v New Zealand Green Mussel Co Ltd* (1985) 5 IPR 353 at 383; *Dart Industries Inc v David Bryar & Associates Pty Ltd* (1997) 38 IPR 389 at 408-9; *RLA Polymers* at [70]-[75]. The doctrine is founded on a concept of fairness. Parties are free to use information that becomes public so long as they do not take advantage of the ‘head start’ of having the knowledge ahead of the public: *Aquaculture Corporation* at 383. As Goldberg J said in *Dart Industries* at 408-9:

In short, if a person wishes to design a product without it being alleged the person has used confidential information he must proceed through an independent design sequence and not use confidential information as a springboard to jump through the sequence.

1. In *RLA Polymners Pty Ltd v Nexus Adhesives Pty Ltd* (2011) 280 ALR 125 (appeal dismissed: *Nexus Adhesives Pty Ltd v RLA Polymers Pty Ltd* (2012) 97 IPR 160; [2012] FCAFC 135), Ryan J said (at [70]):

The “springboard doctrine” is inextricably linked to the concepts of misuse of confidential information and trade secrets outlined above … and, at its simplest, reflects the misuse by one party of another’s confidential information in order to bring out its own product in a manner or time which it would not otherwise have been able to achieve.

1. After quoting from relevant authorities, Ryan J said (at [73]):

The fundamental principle discernible from the authorities is that a recipient of confidential information or trade secrets should not be allowed to use it as a springboard into a better position than would have been achieved from the use of publicly available information and the recipient’s own independent skill and ingenuity.

### Application of principles to the facts

1. The main issue to be addressed is whether IPC Global has established each of the elements of the cause of action for breach of confidence.
2. IPC Global submits that the information here is succinctly identified: it is the source code, and the collocation of ideas, structures and solutions within it. IPC Global also refers to, and appears to identify, Annexures “BJRF5” and “BJRF6” to Mr Fernando’s first affidavit, which are copies of the change log comments for the messaging format specification between the PC and the IMACS. I will refer to these as the **UTS communications protocol documents**. IPC Global notes that Mr Fernando gave evidence during cross-examination that these were confidential and was not challenged on that evidence.
3. IPC Global submits that the source code for the UTS software is of a confidential nature, relying on a series of factors.
4. In relation to unauthorised use, IPC submits that: Mr Feeley took copies of the complete source code of the UTS software to “have back-up copies for an emergency”; taking copies home for work or for offsite security is one thing – keeping the software and deploying it in a rival business is quite another; that does not negate the confidentiality of this material; nor does it of itself answer IPC Global’s claims; there is no evidence that IPC Global authorised Mr Feeley to copy and permanently remove this material; the sheer volume and complexity of the material taken by Mr Feeley point to a breach of Mr Feeley’s duty of confidentiality. IPC Global also submits that Mr Sinadinos breached his duty of confidentiality owed to IPC Global on the basis that he admitted that he knew that Mr Feeley had a copy of the IMACS firmware, that Mr Feeley had previously made copies of the UTS software when they were both engaged by Old IPC and that Pavetest’s software would use the same or similar message format to the UTS software. IPC Global also submits that Mr Feeley and Mr Sinadinos are the moving minds of Pavetest and accordingly their knowledge and awareness of confidentiality is to be imputed to Pavetest.
5. In relation to “springboarding”, IPC Global submits that: with respect to TestLab (version 1), the evidence of springboarding is clear; Mr Li has conceded that, by having the source code for the UTS software, instead of developing it from scratch, he saved “about a couple of months of development time”; his oral evidence showed that to be a very significant underestimate; not only did Mr Li consult the source code for the UTS software to develop TestLab, but he admittedly copied it; in those circumstances, Mr Feeley’s evidence that he only gave the UTS software to Mr Li to “get an understanding of the application but more so to understand the communications interface and software design that had been done in the past and how we might do it differently in the future” ought not be accepted.
6. In relation to version 2, IPC Global submits that: version 2 has been used effectively as a “ruse” to mask unlawful use of IPC Global’s confidential information in the UTS software; Mr Li admitted that he used TestLab version 1 as a base to develop TestLab version 2; thus, although the source code may have been rewritten for copyright purposes, so as to distance it textually from UTS, it retained conceptual links to UTS; what the respondents have done in developing TestLab version 2 is used the UTS software to springboard into a better position than would have been achieved from the use of publicly available information and Mr Feeley’s own independent skill and ingenuity; IPC Global should be entitled to relief to compensate it in respect of the unlawful and unfair advantage obtained by the respondents in the development of the TestLab software including version 2.
7. The respondents’ submissions in relation to the confidential information case can be summarised as follows:
   1. Insofar as the allegation is that the UTS source code and the UTS communications protocol documents, each considered as a *corpus* or body, constitute confidential information, it is accepted that IPC Global has adequately identified the confidential information in issue.
   2. It is not disputed that the UTS source code as a *corpus* includes information that is by its nature confidential.
   3. It is disputed that the UTS communications protocol documents are truly confidential information, and on no view is it in the nature of a secret formula or process. The evidence shows that it is information that is specific to the problem domain of materials testing, and that accordingly a largely similar format could be independently developed by Mr Feeley or any other person with relevant knowledge and experience in the field. Further, even as to the precise form of IPC Global’s message format, the evidence is that it could have been ascertained by Mr Feeley, if he was determined to do so, by a process of enquiry or experiment, without access to source code, given that he had been the original designer of the protocol and had deep knowledge of the software and firmware gained over many years; as such, the use of the protocol should not be enjoined.
   4. IPC Global has not adequately identified, or established the confidentiality of, any *specific* information in the UTS source code that was used in the creation of TestLab, or forms part of the released product. In particular, the respondents submit that none of the information relating to the copied source code of the UTS system library was shown to be truly confidential in nature. First, the evidence does not establish that the UTS system library functions were unconventional or difficult in programming terms. Secondly, the totality of the evidence about the data structures indicated that the choice of data structures, and their parameters, were essentially dictated (or at least heavily influenced by) requirements of the problem domain of materials testing.
   5. It is not disputed that the UTS source code, as a *corpus*, is confidential information that was used by Mr Li as a reference, or research tool, from in around November 2012 until around February 2013, and also from time to time thereafter as indicated by his time sheets (ie, the invoices) (Ex A16). The exhibit indicates that the use had largely ceased by November 2013, with a couple of exceptions in late 2014 and early 2015.
   6. It is not disputed that Mr Li gained a benefit from learning about the operation of the UTS software, and a particular benefit in developing the TestLab system library source code.
   7. It is also not disputed that some development time was saved by reason of the combination of Mr Feeley’s specific decision to re-use Old IPC’s communications protocol, rather than to write a new one or even (if he was so determined) to reverse engineer Old IPC’s; and Mr Li’s resulting decisions to copy some data structures and functions used by the system library.
   8. However, it is submitted that the evidence falls short of establishing that the communications protocol, data structures and functions were truly confidential information, and does not in any case show there was more than a modest saving of time through its use. It was not, and could not be, suggested that Mr Li lacked the experience or expertise to efficiently implement data structures and functions for information exchange between the software and firmware, given the starting point of an application protocol. His frank evidence was that he saved about a couple of months of development time.
8. In my view, for the reasons that follow, the breach of confidence cause of action is established.
9. First, the relevant information has been specifically identified, namely (a) the UTS source code, as a *corpus*; and (b) the UTS communications protocol documents. There is no issue that these are identified with sufficient specificity.
10. Secondly, the relevant information is confidential. In relation to the UTS source code as a *corpus*, I rely on the following matters:
    1. Skill and effort were expended to develop the source code for the UTS software.
    2. IPC Global led evidence, which I accept, that the source code for the UTS software was protected by IPC Global and not made readily available to all IPC personnel. In particular, Mr Fernando gave evidence that since he commenced work at IPC Global, access to the UTS software source code has been restricted to certain personnel (see [73](b) above).
    3. Mr Feeley accepted during cross-examination that he held the view, in July 2012, that source code was confidential (see [75] above).
    4. There is no dispute that the UTS source code, as a *corpus*, is confidential.
11. In relation to the UTS communications protocol documents, I rely on the following matters:
    1. Mr Fernando gave evidence, which I accept, that the restricted access applicable to the source code for the UTS software was applied to the source code and design documents relating to the IMACS messaging format (see [77] above).
    2. Mr Fernando gave evidence during cross-examination that Annexures “BJRF5” and “BJRF6” to Mr Fernando’s first affidavit were confidential.
12. I do not accept the respondents’ submission that none of the information relating to the copied source code of the UTS system library was shown to be truly confidential in nature. The evidence establishes that the UTS source code set out in Annex D (comprising, in general terms, source code to initialise data structures and functions relating to the interface or communication between the software and the firmware) is confidential. Mr Fernando’s evidence, set out in [153](b) above, was that confidential schedule (ie, Annex D) “provides a convenient way to identify most of the key parts of IPC Global’s confidential information”. I accept this evidence. It is consistent with the confidentiality of the source code generally. The fact that Mr Li copied the relevant source code indicates that it has value. It may be accepted that a number of the sequences of code in Annex D are not difficult or unconventional. But this does not negate the proposition that they may be confidential. I do not accept that the evidence establishes that the sequences of source code were dictated by the problem domain; at its highest, the evidence is that the source code was heavily influenced by the problem domain. It does not follow from this that the source code lacks confidentiality.
13. Further, I do not accept the respondents’ submission that the evidence falls short of establishing that the communications protocols, data structures and functions were truly confidential information. For the reasons given in [207] above, I consider the UTS communications protocol documents to be confidential. For the reasons given in the preceding paragraph, I do not accept the submission that the source code initialising data structures (in Annex D) and the functions set out in Annex D were not confidential. In my view, they were.
14. The third element set out in [189] above is that the information must have been received by the respondent in circumstances importing an obligation of confidence. I accept that Mr Feeley initially received the UTS source code as an offsite backup or to do work on it at home. Nevertheless, he received the UTS source code in circumstances imparting an obligation of confidence. Mr Feeley accepted during cross-examination that he held the view, in July 2012, that source code was confidential (see [75] above). The fact that, from mid-2003, he was a consultant rather than an employee does not affect the position. The critical point is the nature of the source code and the way in which it was protected by Old IPC. These were unaffected by the change in his arrangements from employment to consultancy. The position is more complex in relation to the UTS communications protocol documents. These documents, or at least the information contained in them, were probably developed by Mr Feeley himself. It is a little artificial, in these circumstances, to speak of Mr Feeley receiving the documents or information contained in them in circumstances importing an obligation of confidence. Perhaps the better way to approach this element, in the circumstances of the present case, is to ask whether a reasonable person in Mr Feeley’s position would have considered that he or she was not free to deal with the documents as his or her own, or could only deal with them subject to certain limitations. On balance, I consider then that a reasonable person in Mr Feeley’s position would have considered that he or she was not free to deal with the documents. The documents were developed by Mr Feeley while an employee of Old IPC (that is, before mid-2003). The development of the documents involved skill and effort. The documents were treated as confidential by Old IPC. I note that the UTS communications protocol documents contain detailed information which is unlikely to have been remembered by Mr Feeley without reference to the documents or other documents (including the UTS source code itself). It is a detailed piece of work unlikely to be remembered. For these reasons, I accept that Mr Feeley received the UTS source code and the UTS communications protocol documents in circumstances importing an obligation of confidence. I note for completeness that, had Mr Feeley developed a new communications protocol document ‘from scratch’, relying on his experience and memory, and without reference to any UTS confidential documents, then the product of this work would have fallen into a different category.
15. I do not think the third element is established with respect to Mr Sinadinos. The evidence does not establish that he received the UTS source code or the UTS communications protocol documents.
16. The fourth element referred to in [189] above is that there must be an actual or threatened misuse of the information without the applicant’s consent. Mr Feeley provided the confidential UTS source code to Mr Li to assist him in connection with the development of the TestLab software for Pavetest, a rival company. In these circumstances, the UTS source code was misused by Mr Feeley. In relation to the UTS communications protocol documents, Mr Feeley’s evidence is that he specified to Mr Li the required message format for information exchange (see [118] above). Mr Li said that the application protocol data format specified by Mr Feeley was “based on and very similar to” the application protocol data format for the UTS source code (see [112](d) above). In these circumstances, Mr Feeley misused the confidential information contained in the UTS communications protocol documents. On this basis, each of the elements of the cause of action against Mr Feeley are established.
17. It is convenient to refer next to the submissions regarding “springboarding”. The evidence indicates that, while the relevant information is confidential, it is not of such a kind that could not have been be developed by Pavetest independently of the confidential information. In particular, Mr Feeley’s evidence, which I accept, is to the effect that he could have developed a communications protocol document ‘from scratch’, based on memory and experience, and without reference to UTS confidential information, in two to three weeks (see [124](g) above). Further, it is not suggested that Mr Li would not have been able to develop the TestLab software without reference to the UTS software; he would have required additional instruction and information about the problem domain from other sources, and the process would have taken longer, but it is not suggested that he could not have done this. Ultimately, there did not seem to be any real dispute between the parties about these general propositions. The respondents said in oral closing submissions that no issue was taken that a general benefit was obtained and that “no issue is taken that there was some specific benefit obtained in speeding up the development of the system library, … by effectively not having to rewrite the protocol documents and then the functions based on that” (T73). In the course of closing submissions, I sought clarification from counsel for the respondents as to whether their position was (a) that they conceded that there had been a breach of confidence, but took issue with further findings being made that there was confidentiality and breach of confidence in the reproduction of specific sections; or (b) that there was no breach of confidence. Counsel for the respondents responded that their position was the former (T73).
18. In my view, the evidence establishes that Pavetest gained a substantial springboard benefit by the misuse of the confidential information discussed above. Apart from the two to three weeks that were saved by Mr Feeley not having to develop a new communications protocol document, Pavetest saved a substantial amount of time by Mr Li having access to the UTS source code. The benefit gained is not limited to using the UTS source code to educate Mr Li about the problem domain, and the copying of source code as set out in Annex D. I would infer that Mr Li utilised ideas from, and otherwise gained a benefit from examining, other aspects of the UTS source code in writing the TestLab software. Given that he was writing software for the same type of equipment, it would be surprising if he did not draw on other aspects of the UTS source code, notwithstanding the new and more efficient approach to the architecture of the software. Thus, while I accept that he did not copy (in the sense of reproducing the text of) functions in the UTS test modules, I do not accept that he did not draw on other aspects of the UTS source code, including in the test modules, in the course of writing the TestLab software.
19. I note that, although IPC Global’s primary breach of confidence case focused on Mr Feeley and Mr Sinadinos, IPC Global’s springboarding submissions were directed to the position of Pavetest. I also note that, in its pleading, IPC Global alleges that Pavetest has made profits by reason of the breach of confidence. No point was taken by the respondents that IPC Global could not seek relief against Pavetest in connection with the breach of confidence case. (They did, however, flag a dispute as to the form of any such relief.) In my view, Pavetest is to be attributed with the knowledge of Mr Feeley and therefore may be taken to have known that the information was confidential, had been received in circumstances imparting an obligation of confidence, and was being misused. Thus I think it is open to IPC Global to seek relief as against Pavetest in relation to the springboard benefit it gained as a result of the misuse of the confidential information.
20. In the course of closing oral submissions, both parties indicated that they would like the opportunity to make submissions on the form of any such relief following the publication of my reasons. I will defer until I receive those further submissions making a finding, should this be necessary, as to the precise amount of time saved by Pavetest. However, I note that, as set out in [125](l) above, I consider that considerably more time was saved than the two months referred to by Mr Li in his evidence.
21. As with the copyright claim, no point was taken in the respondents’ written closing submissions that IPC Global did not have the right to pursue the breach of confidence claim. In any event, I consider that IPC Global acquired the right to bring this claim even though it related to a past infringement as at the time of the Asset Sale Agreement. In my view, the provisions of that agreement described in [128]-[129] above were sufficiently broad to transfer such rights from Old IPC to IPC Global.

## Breach of contract

1. The two contracts in issue are Mr Feeley’s employment contract with Old IPC (for the period up to mid-2003) and the Aleuta Consultancy Agreement (which was in place from mid-2003 until about 27 July 2012). In relation to the Aleuta Consultancy Agreement, IPC Global seeks to imply terms to the following effect:
   1. that Aleuta owed Old IPC a duty of good faith and fidelity not to engage in conduct which impeded the faithful performance of its obligations, or was destructive of the necessary confidence between Old IPC and Aleuta; and
   2. that Aleuta would not utilise, and would ensure that Mr Feeley would not utilise, the confidential information of Old IPC for any purposes other than those that benefitted Old IPC or its related corporate entities, either during or after the termination of the Aleuta Consultancy Agreement.

Terms to like effect are sought to be implied in the employment contract.

1. In relation to the Aleuta Consultancy Agreement, in circumstances where this agreement cannot be located and the evidence about its written terms is limited, I am not satisfied that it is necessary to imply the terms alleged by IPC Global. Further, it is unclear whether such terms would be inconsistent with the express terms, which are not known.
2. In relation to Mr Feeley’s employment contract, IPC Global’s contract claim does not add anything of substance to its breach of confidence claim. It is preferable to approach the matter on the basis of confidential information principles in the circumstances of the present case, which circumstances include that no copy of the employment contract can be located and, for the period mid-2003 to about 27 July 2012, Mr Feeley worked as a consultant pursuant to the Aleuta Consultancy Agreement rather than as an employee. Indeed, in its closing written submissions, IPC Global did not develop any submissions in relation to this claim.
3. For these reasons, IPC Global’s contract claims are not established.

## Conclusion

1. I will make orders to the effect that each party file and serve proposed minutes of orders to give effect to these reasons and a short outline of submissions in support of those orders; and for the filing and service of submissions in response.

|  |
| --- |
| I certify that the preceding two hundred and twenty-two (222) numbered paragraphs are a true copy of the Reasons for Judgment herein of the Honourable Justice Moshinsky. |

Associate:

Dated: 10 February 2017

SCHEDULE OF PARTIES

|  |  |
| --- | --- |
|  |  |
| Respondents |  |
| Fourth Respondent: | ALEUTA TECHNOLOGY PTY LTD (ACN 104 583 708) |