UIC John Marshall Journal of Information Technology & Privacy Law

Volume 6
Issue 1 *Computer/Law Journal - Summer 1985*

Article 4

Summer 1985

Emerging Issues in Computer Procurements, 6 Computer L.J. 119 (1985)

Mark L. Gordon

Steven B. Starr

EMERGING ISSUES IN COMPUTER PROCUREMENTS†

by MARK L. GORDON*
STEVEN B. STARR*

 Adapted from a speech delivered by Mr. Mark L. Gordon on May 18, 1984 at the Fifth Annual University of Southern California Computer Law Institute.

* Gordon & Glickson, P.C., Chicago, Illinois.

In the beginning, there was the Old Testament of computer contracting: The Almighty and All-knowing (but not necessarily benevolent) Vendor would come down and not only tell the poor mass of users what data processing products and services were needed, but in addition would deliver the Vendor version of the stone tablets: the Form Contract. Naturally, bewildered and in awe, the user dared not question the Almighty, and accepted the Vendor's recommendations and signed on the dotted line.

This was the status quo for quite some time, until the user was enticed (by whom it is not known, although lawyers and consultants head the list of suspects) into eating of the tree of knowledge and was no longer afraid of the Vendor. Suddenly the Vendor found itself unprepared to deal with an informed, and worse, aggressive, user. The Vendor found that its stone tablets were crumbling and reforming under the weight of such blasphemous phrases as "Acceptability Criteria," "Liquidated Damages," "Meantime Between Failures," and the dreaded "Warranties." The Vendor, while down, was not out, and has risen again to match wits with the new user.

Thus a renewed and respectful, but no longer almighty, vendor has reasserted some of its former power and confidence to meet the challenge of an equally confident and self-assured user. This is the New Testament of computer contracting: Vendors can no longer expect to bully knowledgeable users into signing vendor-oriented contracts, yet are unwilling to merely back down at more audacious contractual requests or demands of users; and users no longer have the element of surprise on their side in contract negotiations. Both sides are now on more equal footing; the vendors, however, have rebounded with more aggressive methods of protecting their interests against experienced users.

This Article explores several issues recently raised as a result of users being forced to confront the increasing willingness of vendors to aggressively protect their rights by sophisticated methods, both contractual and noncontractual. Although it should have been easy to predict, the world of computer contracting continues to evolve with the industry. Only a few of the many complexities now emerging will be addressed here. A guiding principal, however, can be applied not only to these areas but to many others. Creative and thoughtful approaches to contracting are available and will yield opportunity and advantage.

I. ACQUIRING SOFTWARE WITH A VIEW TOWARDS DISTRIBUTION: JOINT VENTURE OR REMARKETING AGREEMENTS

A. Users' Growing Proprietary and Pecuniary Interests

A relatively recent trend in the data processing industry is the tendency of users to have a more proprietary and pecuniary interest in certain software products that were formerly desired for internal purposes only.

For the purposes of this Article, there are three basic types of software acquisitions made by users: standard package software with no modifications; package software with modifications for special user requirements; and pure customized software programmed solely or primarily to user specifications. In addition, there are alternatives with respect to modified packaged software and straight developed software: either the vendor can perform the modifications or development without substantial user input, the vendor and the user can combine programming and/or specification development work, or the user alone can perform the modifications to package software.

B. Alternatives in Dealing With Vendor Modifications

Until recently, vendors typically maintained that any original programming work done by the vendor or with the assistance of the vendor would be owned by the vendor. In addition, vendors maintained that any original programming development work done by the user in connection with a vendor owned package was owned by the vendor as well. There have also been variations on these themes, depending on the relative negotiating strengths of the parties as well as surrounding circumstances.

1. *User License*

While the vendor retains ownership, it will almost always grant a license to the user for use of the modifications or developed software, either on a perpetual basis or at least coterminous with the underlying vendor-owned software, and with the same restrictions on use and disclosure.

2. *User Marketing Rights*

In addition to a license to use the developed or modified software, the user may be able to obtain marketing rights, including royalties or commissions on each sale or license generated by the vendor or the user, depending on the arrangement.

3. *User Ownership—Vendor Marketing Rights*

Alternatively, if the user's input into the project has been substantial the user may be able to acquire ownership of the modifications or developed software. The vendor, however, may request marketing rights, with similar royalties or commissions payable.

4. *Joint Ownership*

In some instances, the vendor and user may negotiate joint ownership of the developed software or modifications to the vendor's package software, with reciprocal marketing rights and obligations. In some instances, the transaction may include joint ownership of the underlying vendor software.

5. *Other Complexities*

The foregoing possibilities are certainly not comprehensive: there are a wide variety of variables that can be added. For example, any of the alternatives referred to above may substantially impact the cost or pricing to the user for the original work that is the subject of the marketing or royalty negotiations. The user should be aware of the possibility and flexibility of these types of arrangements, and the resulting costs and benefits associated with structuring them.

C. Considerations of Vendor and User in Contracting for Marketing Arrangements

The negotiating leverage of both vendor and user will play a significant role in determining whether and to what extent a joint ownership/marketing agreement is possible. There are, however, additional considerations that may influence both parties' desire to propose or reject such an arrangement.

1. *Competition*

Both the vendor and the user will be sensitive to concerns about competition. A user that contracts for software to be customized or created for the user's industry or method of operations will probably not want the vendor marketing the software to the user's competitors, since the software presumably provides a competitive advantage. Moreover, the user will not want to market the software itself. Naturally, if there were a market outside the user's industry, some restrictive marketing arrangement might be acceptable. If the user has contracted for specialized software that could be used by other users, the vendor is not likely to forego this additional market without substantial consideration from

the user. Such consideration, however, will typically be in excess of what is commercially reasonable for the user to pay.

On the other hand, where the user is not concerned about acquisition of the software by its competition, and where the user can negotiate pecuniary or proprietary rights to the software, the user may allow the vendor to retain ownership of the software with royalties on future licenses payable to the user. This would allow the user to take advantage of the vendor's more experienced and comprehensive marketing abilities.

Conversely, in some situations the user may be in the better position to market the software, using its contacts or market position. Thus the vendor may find it more advantageous to grant ownership rights to the user while retaining royalty rights for licenses granted by the user.

2. *Joint Ownership*

In some cases, joint ownership, with joint or complimentary marketing responsibilities and obligations, may be the appropriate means to maximize vendor and user interests. Often a vendor in a modified software package situation will insist on retaining ownership in the original package but will allow joint ownership of modifications made by or paid for by the user. This complicates the rights of the parties in several respects.

3. *User Entering Into Software Sales*

Occasionally a user who requests customized software may either concurrently with its planning for such software or subsequent to productive use, decide that its ideas for marketing the software are potentially more profitable than the business in which the user had originally intended to use the software. In such a case the user may terminate its business and market the software as a replacement business. This is especially likely where the software is capable of substantially enhancing operations and significantly increasing competition if widely distributed. If the competitive effect will not significantly affect the market, the user may continue its primary business but sell software to generate additional revenue. It is crucial to plan for these possibilities as early and as thoroughly as possible by negotiating an agreement with the software developer.

D. THE REMARKETING AGREEMENT—FORM AND ISSUES

Depending on the nature of the software and its customization or development, as well as the user's desires, a remarketing or joint venture arrangement regarding the user's ownership and marketing rights can be as simple as a paragraph contained in the vendor license or de-

velopment agreement, or as comprehensive as a fully negotiated document specifically dealing with the numerous issues involved. The user must evaluate as early as possible the potential market for and value of the software, and must confront the vendor with the appropriate level of negotiation. The following is a discussion of some of the issues that should be addressed when a user is involved in negotiating an agreement for continuing value and marketing rights in software developed solely by a software vendor or in conjunction with the user.

1. *Ownership*

The parties must clearly specify the ownership rights in the software. These rights may be divided between the parties to correspond to different elements of the software. Since different rights will attach to each party's ownership, it is important to define the various elements so that ownership of particular portions of the software is clearly delineated. Where one party is the owner of an original software product and the other party has developed (with permission) a marketable enhancement, revision, or derivative work, it must be made clear who owns what part of the "new" work. Joint ownership, however, may not alleviate this problem if the parties wish to have differing rights with respect to the work, such as the right to prepare derivative works and the right to market the work to different users or in different geographical areas.

2. *Licenses*

If either or both parties will have the right to market software in which they have no corresponding ownership rights, the agreement should grant reciprocal license rights to the party marketing software it does not own. The agreement should specify the nature and extent of the license granted in terms of limitations on marketing, such as exclusivity, geographical limitations, and internal use or modification restrictions. In addition, the license should set forth the term granted, whether perpetual or of limited duration. Normally the term will be for the duration of the agreement; this, however, will not always be the case, such as where a party has rights or obligations to end users with respect to the software after termination of the agreement. In such a case the license granted would change character after termination of the agreement to the extent of that party's post-agreement rights or obligations.

3. *Derivative Works*

A crucial point to be considered by the parties is the respective rights of the parties to create derivative works based on the software. A

well drafted agreement should set forth a detailed set of definitions and each party's rights and obligations. It may be helpful to define the following terms: derivative work; enhancement; modification; revision; new version; update; maintenance fixes; translation; and adaptation. Many, if not most, of these terms surprisingly have not attained a specific accepted industry meaning, and thus it is important that the parties agree on their own definitions.

The agreement should set forth each party's rights and obligations regarding creating of derivative works, enhancements, updates and so on. For example, the agreement should delineate whether one party can make enhancements to a derivative work made by the other party, where the other party had exclusive rights to make derivative works, and where the definition of a derivative work excludes enhancements. In addition, the agreement should indicate the extent to which these "additional items" can or must be provided to existing or future licensees of the software, and whether a royalty is due to either party as a result.

4. *End-User License Agreements*

The remarketing or joint venture agreement should include a sample end-user license agreement that will be the basis on which either party grants licenses to the software to end users. This sample license agreement should contain sufficient confidentiality and non-disclosure requirements to prevent the loss of proprietary rights to the parties, such as copyright or trade secret.

Naturally there will be occasions when the end user will request changes, and provision should be made for prior consent of the party not involved in contracting with that end user. In the event one of the parties does modify the end user license agreement without obtaining consent, the agreement should provide that the modifying party will indemnify and hold harmless the non-consenting party for any resulting damages.

5. *Royalties*

It is neither possible nor necessary to review the variations of royalty payment formulations. While the formulation of royalty payments is primarily a business decision, some issues should not be overlooked in making this decision. The parties should review the marketing opportunities not only for the initial software available, but also for derivative works, enhancements, translations, and so on, which may provide opportunities for additional revenue from end users and thus be appropriate for additional royalty rights. These royalty rights may or may not correspond to the party with the ownership rights to the derivative

work or enhancements. If appropriate, the parties should agree in advance with respect to revenue and royalties generated as a result of software used as a service bureau operation. In addition, there may be occasions for exemptions from royalty payments where a party licenses the software to an affiliated entity or merely for demonstrative use.

6. *Ownership Warranties and Indemnification*

Any party that claims ownership rights in all or any portion of the software should warrant that it is the owner and author thereof. Such ownership rights should be warranted to include the right to license the software to the other party and to perform all obligations of the agreement. Additionally, the parties should warrant that the software and related documentation has not been published or disclosed under circumstances that may have caused a loss of intellectual property right protection.

The agreement should also contain appropriate warranties that the software and documentation do not infringe on any copyright, trade secret, patent, or other intellectual property right, and indemnifications should be made covering a breach of such warranty. This issue becomes complicated if the original software was not entirely created and owned by the vendor but is composed of a patchwork of third party software contributions. The rights of these third parties must be dealt with, not only in connection with the original software and the rights to modify it, but also as their rights relate to the marketing agreement and to any subsequent modifications.

7. *Copyright Notices*

The agreement should establish requirements regarding the placement of appropriate copyright notice on the software and related works. The forms of notice, as they relate to media, documentation, listings, and so on, should be agreed on by the parties and set forth or referred to in the agreement. The name associated with the copyright notice will generally be that of the owner and/or author. Where the work is jointly owned or if otherwise agreed upon, the copyright notice may be in both names. Furthermore, the parties should agree in advance on the copyright notice to be provided in the case of permitted derivative works; usually the name of the author will be appropriate. Provisions relating to the registration of a copyright by either party should also be included, specifying the cooperation of the other party.

8. *Other Issues*

a. *Termination*: The parties should agree on the duration of the agreement. In some cases the agreement may remain in force until the

last-to-expire copyright has expired. The agreement may also call for earlier termination upon certain events or default.

Provision should be made for the survival of certain sections of the agreement, such as ownership rights, right to royalties for pre-termination licenses, warranties of ownership and non-infringement, protection of copyrights, indemnifications, and other relevant terms as agreed on by the parties. All existing licenses made to end users should survive termination as should either or both parties' obligations with respect to such end-user licenses. Finally, conditions relating to return of documentation and code should, if appropriate, be detailed.

b. *Advertising*: The rights and obligations of the parties with respect to marketing the software may be specified. This is especially important with regard to either party's use of the other's name or trademarks in marketing the software.

c. *Independent development and competition*: The right of any party to independently develop materials or programs that are competitive with or similar to the software being marketed under the agreement should be considered.

d. *Marketing obligations*: It is primarily a business decision whether either or both parties wish to impose minimum marketing obligations on the other party or themselves. Marketing obligations may include a "best efforts" clause or provisions for minimum royalties and whether royalties should be guaranteed or conditioned upon continuation of the agreement.

e. *International distribution*: If either party contemplates distribution or marketing in foreign countries, the agreement should provide that any party marketing the software internationally must follow certain procedures to insure that such marketing or distribution does not cause a loss of proprietary rights under foreign laws, customs laws or export restrictions, or otherwise endanger the value of the software. Unless detailed procedures are set forth in advance, the consent of both parties prior to international distribution is appropriate.

f. *Employee or independent contractor rights*: The agreement should consider and protect against a loss of proprietary or pecuniary rights in the software as a result of poorly documented arrangements with employees or independent contractors connected with the software. If appropriate guidelines on contracting with such individuals are not formulated in advance, third parties may acquire rights to portions of the software and create unnecessary and unwanted burdens on the agreement and on the parties' economic benefit associated with it.

g. *Agreement administration*: In an agreement of this type, especially if the deal is large, it will aid the progress and success of the venture if the agreement names coordinators for each party, with each coordinator given certain authority and responsibility in connection with carrying out the terms of the agreement.

9. *Additional Terms*

The above list of considerations is not meant to be comprehensive or exhaustive. Other terms commonly found in license or distributor agreements obviously need to be included, such as confidentiality, force majeure, and solicitation of employees. Additional or substitute provisions will be appropriate where the parties agree or where the nature of the transaction so dictates.

II. MEETING SOFTWARE VENDORS' INCREASING SENSITIVITY TO THE FUNCTIONALITY OF SOFTWARE IN SETTING LICENSE FEES

Software vendors are increasingly able to measure or gauge the functionality of the software they place into the market, by measuring the variations of uses to which the software is put and assigning values to those varying uses. Eventually this will mean an abandonment of the simplistic one-copy, one charge structure and of the related assumption that the program will be used only on one central processing unit with one visual terminal.

A. CURRENT VENDOR INITIATIVES

In attempting to maximize the revenue from software product licenses, vendors are attempting to more accurately measure the use of the software, or stated in the alternative, to restrict use of the software in a way that creates an easily identifiable and logical measure for changing license fees associated with fluctuations in such measure.

1. *Location*

One method commonly used by vendors to restrict software use is to restrict use of the software to one specified location of the user. Where the vendor has not narrowed the scope of such a restriction by clearly defining its parameters and fully articulating the issue with the user, the user may be able to successfully argue that the term "single location" or "specified address" encompasses the use of multiple copies of the software at different buildings in a single office complex or on different floors of a single building. The user may also argue that use of the software on one or more processing units at a single location, even

though remote access can be achieved from terminals in distant locations, does not exceed the single location restriction.

2. *CPU Serial Number Restrictions*

Restricting the software to a particular processing unit, using the unit's serial number, is also becoming a widespread vendor practice. A user should ensure that it at least has the right to transfer the software to a back-up processing unit when the originally specified unit is not operable. The user should also have the right to permanently transfer the software to a different processing unit if advance notice is given. Again, however, the user may achieve a wider scope of use than the vendor intends if the software is used in a distributive processing environment or through telecommunications networks.

3. *Multiprocessor Configurations and Distributive Network Processing*

A major problem currently facing software vendors is the use of software in a distributive processing environment where several interconnected central processing units are used through telecommunication networks with each other and with intelligent and nonintelligent terminals or front-line processing units, including microcomputers. Software vendors faced with this situation are becoming as creative in pricing software as the user has become in deriving maximum use of a program.[1]

B. User Concerns

The user must be sensitive to these increasingly restrictive and encompassing pricing structures. Users may have swung the pendulum too far their way by expanding the use of new concepts in hardware equipment to enhance the utility of software beyond that intended by the vendor's license. Consequently, users must guard against any tendency of vendors to swing the pendulum too far the other way by requiring users to pay excessive license fees, which are out of proportion to the actual productive use of the software, simply because of a particular machine configuration or capacity.

1. *User Negotiating Tactics*

In the midst of this scenario, the user will have certain value judgments to make when negotiating with software vendors. Where a vendor has not been careful in drafting a license agreement and has failed

1. Appendix A contains an example of a license fee structure which is flexible in its realization of networking possibilities.

to include a well defined measure of use, and where the user intends to use the software in an expansive way, the user may decide not to disclose his intentions and let the ambiguity of the vendor agreement work to his favor. This tactic may, however, produce harmful disputes which may not only sour a relationship but may also result in greater cost to the user (should the vendor prevail in a dispute) than might have been the case had the user openly discussed the intended use of the software with the vendor and agreed in advance on an appropriate license fee arrangement.

On the other hand, where a vendor has carefully drafted well-defined license agreements that attempt to capture all the possibilities of the user's use of software, the user must negotiate an agreement with the vendor that will more accurately reflect the nature of the use of the software than will the vendor's pre-determined form. In some cases the user may have to convince the vendor that correlating license fees with configuration capacity will not be the best or the fairest arrangement. For example, it would clearly be unfair for license fees to be based on the number of terminals or microprocessors accessible to a central host in which the software resides, where the user knows and can demonstrate to the vendor's satisfaction that only a few of the available terminals will ever call up that program.

2. *User Alternatives*

Where a user is faced with a major software installation and an aggressive and sophisticated vendor, the user should be prepared to offer pricing alternatives. Given the current state of software technology, and more importantly hardware technology, it is doubtful that any vendor form contract can adequately or fairly present the parties with a real measure of functionality. The user must decide whether the vendor's form structure operates adversely to the user, and must be prepared to offer alternative structures to the vendor and convince the vendor of their fairness in reflecting reality.

III. DEALING WITH TECHNICAL METHODS OF SOFTWARE PROTECTION BY VENDORS

Many software vendors rely on preprogrammed routines embedded within the software not only to protect confidentiality of the software and their proprietary interests, but also to measure or otherwise control usage.

A. EXAMPLES

The following are some, but certainly not all, of the methods used by vendors to protect their software and monitor its use.

1) Encryption: Encryption of software source code may protect the confidential nature of the licensed software by preventing or making more difficult reverse engineering.

2) Date bugs: A vendor may encode a date bug into a program routine which will cause the software to become inoperative or to function incorrectly after a particular date unless the vendor periodically modifies the program, usually corresponding with periodic license or support payments.

3) Data flow: Technical means allow measurement of data flowing through a program and can be the basis for charges to a user.

4) Time elapsed: Vendors may also include routines that measure elapsed processing time.

5) Data storage amounts: The amount of data stored in a system, if measurable, may serve as the basis for charges.

6) Processing unit serialization: The serial number of a particular processing unit is hard-coded into the program and that program will operate only on that unit.

B. User Concerns

These technical "limiting devices" can seriously affect the user's processing ability, with effects ranging from a decrease in the usefulness of the software, such as the inability to transfer the software to a back-up computer or to process or store data in an unexpectedly heavy processing environment, to potential disaster in the case of incorrect processing results.

1. *Discussion with Vendor*

Users who are aware of the possible presence of preprogrammed routines in the software they license must discuss this with the vendor. Vendors who are not forced to confront the issue with a knowledgeable or at least cautious user potentially face self-inflicted injury.

2. *Administrative Controls by User*

When faced with the issue the vendor may either agree to remove the limiting device or insist on maintaining it within the software. If the limiting device remains in the software the user must protect itself by placing an administrative control within its organization to ensure that the preprogrammed terminating or limiting event is not allowed to occur. Naturally it would be difficult to plan a fail-safe procedure, and sole reliance on the vendor, even if the user complies with its obligations regarding use, is not advisable.

3. *Potential Damage and Liability*

Date bugs, time bombs and other preprogrammed termination routines can cause serious damage to users if not implemented and maintained properly. While it is conceivable that the vendor may have some liability for such damages, such potential vendor liability is not adequate to protect a user. Programs that cease to function after a particular date may seriously interrupt a business operation. It is possible that a vendor may mistakenly fail to periodically correct the program when required to prevent such termination. Moreover, over-zealous vendors have placed programs which, after the preprogrammed event occurs, seem to continue operating but in fact output false data.

C. RECOMMENDATIONS

In the face of vendors' increasing use of technical means to control software use, the user has available alternatives which, while not certain to alleviate the potential problems, provide the user some comfort.

1. *Warranty*

Technical devices that merely measure data flow, data storage, and so on may be effective in monitoring a user's compliance with license restrictions, but unless these devices are also programmed to directly affect the user's ability to process data in the manner intended, the user may not have much to complain about. Where the software is capable of shutting off access by the user, or of creating false data or otherwise limiting or affecting the use of the software, the user must take affirmative action in advance, particularly in the context of contracting with the vendor. One way to expose the existence of such technical limiting devices in vendor software is to require the vendor to warrant in the license agreement that no such limiting design is contained in the software.[2]

2. *Limitations of Liability: Consequential Damages*

While warranty language will usually cause a vendor to admit the existence of a limiting design, the vendor may not be willing or able to eliminate this portion of the software.

If the vendor is willing to warrant the nonexistence of a limiting routine in the software, the user may ensure the vendor's sincerity, and add to the user's security, by including language in the contract that eliminates any limitations on liability in the event of a vendor breach of that warranty. In addition, any limitation on consequential damages

2. An example of such a warranty is contained in Appendix B.

should be specifically removed to the extent that such damages arise as a result of a breach of the warranty.

On the other hand, if the vendor admits the existence of a limiting routine and will not agree to such a warranty, and if the user decides to acquire the software despite the risks associated with inclusion of a limiting routine, the user must demand that the vendor detail the events that trigger the limiting routine. The user must then ensure that its staff puts administrative controls in place to prevent user responsibility for a triggering event. In addition the user should require the vendor to agree to language that eliminates limitations on liability and consequential damages in the event that the vendor has not disclosed one or more triggering events and one occurs causing user damages, or the limiting routine is activated through no fault of the user.

IV. CONTRACTING FOR SOFTWARE COMPATIBILITY

The issue of software compatibility is becoming increasingly important to users. In the past, hardware was the primary factor in a user's acquisition, and the software usually came from either the hardware manufacturer or a single software vendor that developed software specifically for a certain manufacturer's hardware. Today the situation is substantially different. Hardware has become more reliable and available, and software developers and dealers have proliferated at an incredible rate. As a result, today a user's data processing operations may typically include software packages from several vendors. In addition, brokers, OEM's and other distributors are putting together more and more "systems" that are made up of multiple vendor hardware and software components. Users must be cautious and sensitive to the issue of compatibility between components.

A. Compatibility Between Software and Hardware

There are basically two situations related to software acquisition in which the issue of compatibility must be addressed.

1. *Software Additions*

The first situation requiring compatibility considerations is acquisition of a new software component, either package or custom, for use on an existing system. The user should describe the existing hardware and operating systems software in the contract and have the software vendor warrant that its software is compatible with such hardware and operating systems software and will be fully capable of performing all the intended functions. In addition to providing legal protection, the warranty provides practical protection in that the vendor is forced to consider the compatibility issue. In addition to the warranty, the user

should insist on an acceptance test procedure to be performed on the user's existing hardware.

2. *System Acquisition*

The second situation in which compatibility must be considered is acquisition of both new hardware and new software from the same or different vendors. In general, the user should require the same warranty protection discussed in the preceeding section.[3] If the same vendor is supplying hardware and software, a warranty of compatibility is appropriate. If different vendors are supplying the two components, the user should specify the hardware that will be acquired and ask the vendor to warrant software compatibility with that hardware. Acceptance testing that demonstrates the compatibility of the software and hardware is always a necessary precaution. In the event the hardware has not been installed on the user's site, the user might ask the vendor to locate a similarly configured hardware system and demonstrate the software on it.

B. INSURING COMPATIBILITY OF MULTIPLE SOFTWARE PACKAGES

Where a user is acquiring custom software or several software packages for the user's hardware system, the user must take the appropriate steps to ensure that all components work properly together. This is especially true where a broker or dealer has put together a system of software originating from multiple vendors.

1. *Warranty*

The vendor should be required to warrant that all portions and modules of the software are compatible with the hardware configuration, including operating systems software, and with each other. The vendor should also be required to warrant that when implemented as a system the software will perform all of the functions for which it was intended and licensed from the vendor.

2. *Acceptance Testing*

Where multiple software packages will be installed on a system over time in a phased installation program, the acceptance testing should also be phased using a pyramid type of testing. The first program or group of programs should be loaded onto the system and tested to ensure compatibility with the hardware, and to ensure that the programs perform according to the appropriate specifications. After the first program has been successfully tested for compatibility and per-

3. *See supra* § 1.

formance, the next program should be loaded onto the system and tested in conjunction with all previously tested programs. This procedure will ensure that the program currently being tested not only performs according to its specifications, but also is compatible with all other software in the system. This pyramided testing should continue until the hardware and software is demonstrated to work together as contemplated by the user at the contracting stage.

As with any acceptance testing procedure, sufficient incentives and remedies should be available to the user in the event acceptance testing is not successful. Normally a percentage of each software license fee should be held back, as should a percentage of the hardware costs if the user is dealing with a hardware and software package acquisition, until all acceptance testing for all software modules is completed and the total system performs as intended. In addition, the user and vendor should agree in the contract on remedies available to the user in the event one or more of the software packages fails to perform to specifications during testing. If a minor software module fails to perform near the end of the total system implementation, it is doubtful that the user should be entitled to rescind the contract and force the vendor to take back all hardware and software. On the other hand, if a major software module fails, all remedial provisions should be activated.

C. The Problem of Defining Compatibility

"Compatibility" must be adequately defined for purposes of warranties and acceptance testing. Obviously compatibility is not black and white, and there are varying standards that may affect the ultimate issues of acceptance or of warranties of compatibility.

1. *Examples*

The following examples illustrate the difficulties that can confront the user and the vendor when contracting for data processing compatibility:

(i) the extent and facility of existing data base transfer and access to an application program being acquired;

(ii) the extent of compatibility between different models of a manufacturer, including compatibility upwards and downwards;

(iii) whether the new software addresses data the same way previously installed software does; and

(iv) whether file and record structures are compatible with the various software programs.

2. *Technical Guidelines*

There is no set of guidelines that can provide a general rule to en-

sure compatibility. Technical guidelines should be structured by technical personnel and will differ for different types of software and hardware configurations. The only rule is that it is insufficient protection to rely on a simplistic definition of compatibility, or worse, on no definition at all.

V. CONTRACTING FOR DISASTER RECOVERY SERVICES

For a user whose operations significantly depend upon data processing, an increasingly important consideration is the availability of disaster recovery services to enable the user to continue with vital processing requirements. Many forms of this type of service are available, such as off-site back-up archiving of programs and data, reciprocal support agreements between companies with compatible equipment, and third-party "shell" or full system availability.

A. FULL SERVICE OPERATION

In the most ambitious and expensive form of disaster recovery service, a vendor who provides such service maintains one or more sites with a fully configured computer system that can be made available to the user in the event the user suffers a computer operations disaster. The charges for such protection usually consist of:

(i) a monthly or yearly subscription fee corresponding to the user's needs as determined by the user's hardware configuration;

(ii) a disaster notification fee (usually a substantial amount) which must be paid if the user experiences a disaster and requires use of the recovery center equipment; and

(iii) a daily usage fee payable for each day that the center is used as a result of a user disaster.

Typically a vendor offering this type of service will require the user to contract for the recovery center availability over a period of time, rather than on a "disaster occurrence" basis. Presumably the profitability of the vendor requires this approach, but in addition the user has some assurance, depending on the contract language, that recovery services will be available.

B. CONTRACTUAL CONCERNS OF USER

Since the availability of disaster recovery services is critical to users who decide to contract for such an obligation on the part of the vendor, it is important that the user and the user's counsel ensure a maximum amount of protection in the written agreement. The user must understand the nature of the services contracted for, the variables that may impact the obligations of both parties, and the specific concerns that

must be included in any document that may affect the survival of the user's business.

1. *Definitions*

As with any contract, but particularly with these types of issues, definitions must be as precise as possible. The definition of "disaster" the event triggering the user's need for and the vendor's obligation to provide recovery facilities—is critical. Typically this is defined as any unplanned interruption of or inaccessability to the user's computer system expected to last over 24 hours. The user may, however, find it necessary to expand this definition.

The definition of "multiple disaster" is also important. Usually a multiple disaster is defined as one or more disasters experienced by two or more users who contract for recovery services with the vendor, thus entitling each user to use the recovery facilities for the same or overlapping periods.

2. *Compatibility*

The user will need to be assured that the recovery center facilities are compatible with the user's software and processing needs. While recovery service vendors may not readily be willing to do so, unless the user can be assured of compatibility through a review of the vendor's machine configurations, the user may want to insist on a "test run" to be assured of compatibility prior to entering into a long term obligation. As a less desirable alternative, and since most vendors allow contracting users to periodically test the recovery facilities during the term of the agreement, the user may want to specify that if the first testing indicates that the facilities are not compatible with the user's processing, the user may cancel the agreement and receive a full refund.

3. *Access to Recovery Facilities*

Most importantly, the user should make sure that the procedures through which the user may obtain access to the recovery facilities are clear and specific. The method of notifying the vendor that the user requests access should be considered. While the vendor may prefer and require written notification, the user may negotiate a provision for some form of oral notification with a written follow up, which will expedite the user's access. The notification should also specify the time and duration of the user's desired access. The agreement should also specify how soon after the vendor is notified of the user's request the vendor is obligated to provide access to the recovery facilities.

The minimum and maximum amounts of time the user can use the facilities as the result of a single disaster should be specified. The

agreement should provide that the user may continue to use the system beyond the maximum time allowed if it has made a good faith effort to repair its own facilities and no other contracting user has experienced a disaster.

The user may also desire to contract for access to the recovery facilities for nondisaster events such as minor disruptions, software development, or other systems work. Usually the vendor will agree, subject to usage charges to the user and subject to the rights of other contracting users to disaster-related use of the facilities.

4. *Multiple Disasters*

The occurrence of a multiple disaster, where one contracting user requests access to the recovery facilities at the same time another user is using the recovery facilities as the result of a disaster, must be considered by the parties in the agreement. One way to minimize such a possibility is to require the vendor to agree to a maximum number of users to which the vendor will be obligated at any one time. Even then the user must be sensitive to two possible situations.

Where the user experiences a disaster and requires access to the recovery facilities while another user is using the facilities, the user should have the right to use any alternate facilities of the vendor not being used for disaster purposes. A priority arrangement should be established by the vendor indicating the user's priority with respect to other users experiencing a disaster as well as the remaining access time to which the current user is entitled. The vendor should also be required to use its best efforts to work out a substitute arrangement with the users involved, such as alternating shifts during 24 hour periods.

Where the user has experienced a disaster and obtained access to the recovery facilities, and another contracting user requires access because of a disaster, the initial user should be assured of continued access for the maximum time promised under the agreement, subject to the first user's agreement to permit a subsequent user access at times when the first user does not need the facilities, such as off-hours. A previously-accessed user should be assured that it will not be bumped off the recovery facilities prior to its maximum time allowed without its consent.

5. *Remedies for Vendor Failure to Permit Access*

In any agreement of this type, the user should expect the vendor to be aggressive in limiting its liability for failure to provide the user access to the recovery facilities in the event of a disaster. Typically the vendor will disclaim any liability in the event of multiple disasters, acts

of God, power or utility outages, and other force majeure circumstances, and will severely limit its liability for other defaults.

While the vendor will usually not take any responsibility for failure of a user to gain access to the recovery facilities in the event of a multiple disaster, the user should have the option to terminate the agreement and seek aid elsewhere. Whether the user can also negotiate a return of all amounts paid to the vendor in such circumstances is doubtful but desirable.

Where the user is unable to access the recovery facilities for reasons other than a multiple disaster, force majeure, or other reasons beyond the control of the vendor, the user should consider requesting the following remedies, alternatively or cumulatively:

(i) direct damages;

(ii) vendor responsibility for locating and paying for alternate facilities sufficient for the user's processing needs;

(iii) return of all sums paid to the vendor from the inception of the agreement;

(iv) suspension of any payments due the vendor until access is obtained; and

(v) termination of the agreement.

6. *Equipment Changes*

During the term of the agreement, especially if it is a long term agreement, it is likely that either the vendor or the user will undergo some equipment configuration changes. The user should request advance notification of any changes in the vendor's equipment configuration and, if the user reasonably determines that such change may adversely affect compatibility, require free test time to test compatibility. Should the equipment adversely affect the user's ability to meet its processing needs after a disaster, the user should have termination rights.

If the user changes its equipment configuration such that compatibility is affected, the user should have the right to terminate the agreement provided the vendor is given notice and an opportunity to change its configuration to accommodate the user. The vendor may insist on some limiting language to prevent a user from attempting to get out of the agreement by making small alterations in its system.

7. *Confidentiality*

As with any service bureau contract, the user should insist on strong, well drafted language requiring the vendor to maintain strict safeguards to protect the confidentiality of the user's data and programs while the recovery facilities are being used.

8. *Vendor's Other Agreements*

If the user is successful in negotiating provisions of the disaster recovery agreement, it can be assumed that other users have been successful in the same regard. Consequently the user should require some assurance from the vendor that the vendor has the power and authority to enter into the agreement, and that the obligations it has undertaken with respect to the user do not conflict with obligations to any other user. In addition, the vendor should covenant that it will not in the future enter into any agreement that would conflict with the user's rights.

CONCLUSION

In this current state of rapid technological advances in computer hardware and software technology, and the corresponding rush of vendors to sell and users to acquire such technology, the user must remember that advances are also being made by vendors in contracting for the distribution of such technology. The most successful users will realize this and respond accordingly.

APPENDIX A

SOFTWARE LICENSE SAMPLE PRICING[4]

License Fees For Software:

a. The standard Initial License Fee for the Software for each Central Processing Unit (CPU or "Host") are:

No. of CPUs on which Software is installed	License Fee per CPU
1	$4,500
2 - 6	3,600
7 - 25	2,025
26 - 50	1,275
Over 50	875

b. In addition to the above fees, the Standard Initial License Fee for the software when it resides on a Host and can be transferred or accessed electronically from the Host to CRT's or other nonintelligent and intelligent terminals ("peripherals") in network fashion will subject the user to additional licensing requirements as follows:

4. Further refinement of definitions and standards might be possible (for example, whether or not intelligent peripherals and microcomputers should be considered Hosts).

Number of Peripherals accessible per Host	Additional License fees required per Host
1 - 2	0
3 - 5	1
6 - 9	2
10 - 16	3
17 - 25	4
26 - 36	5
37 - 49	6
50 - 64	7
65 - 81	8
82 - 100	9

*Further refinement of definitions and standards might be possible (for example, whether or not intelligent peripherals and microcomputers should be considered Hosts).

APPENDIX B

PRE-PROGRAMMED TERMINATION WARRANTY

Vendor represents and warrants that the Software System (and any portion thereof) does not contain any timer, clock, counter or other limiting design or routine which causes the Software System (or any portion thereof) to become erased, inoperable, or otherwise incapable of being used in the full manner for which it is designed and licensed pursuant to this Agreement after being used or copied a certain number of times, or after the lapse of a certain period of time, or after the occurrence or lapse of any similar triggering factor or event. Furthermore, Vendor represents and warrants that the Software System (or any portion thereof) does not contain any limiting design or routine which causes such software to be erased, become inoperable, or otherwise incapable of being used in the full manner for which it was designed and licensed pursuant to this Agreement solely because such Software System has been installed on or moved to a central processing unit or system which has a different serial number, model number, or other identification different from that on which the Software System was originally installed.