1660-100
Comment Letter No. 149

SoFTEC

Software Finance & Tax Executives Council

www.softwarefinance.org

June 19, 2009

Via Email to: director@fasb.org

Mr. Russell G. Golden
Technical Director
Financial Accounting Standards Board
401 Merritt 7
PO Box 5116
Norwalk, CT 06856-5116

Re: Discussion Paper on Revenue Recognition
File Reference No. 1660-100

Dear Mr. Golden:

I write on behalf of the Software Finance and Tax Executives Council (SoFTEC) to comment on the Discussion Paper issued December 19, 2008 setting forth the Boards' Preliminary Views on Revenue Recognition in Contracts with Customers (the "Discussion Paper"). SoFTEC supports the Boards' goal of replacing the existing patchwork of industry specific guidance with a new, principles-based, revenue recognition standard. We believe the preliminary views set forth in the Discussion Paper further that goal.

SoFTEC is a trade association providing software industry specific public policy advocacy in the areas of tax, finance and accounting. Its members have long grappled with existing software industry-specific revenue recognition accounting guidance and naturally have an interest in commenting on the views set forth in the Discussion Paper.

As noted above, SoFTEC supports the Boards goal of crafting a new, principles-based revenue recognition standard having broad application. SoFTEC also supports the single, contract-based revenue recognition set out in the Discussion Paper. Adoption of such a standard will, we believe, lead to a more transparent display of software revenue in financial statements and provide more useful information to their users.

SoFTEC's members recognize that the Discussion Paper only sets forth preliminary views and therefore does not include all of the guidance one would expect to see in a final Standard. With this recognition in mind, and as more fully set forth below, SoFTEC suggests that any final standard more fully develop guidance applicable to services and similar contracts.

1. Software Industry Business Models:

Software is a unique product for a number of reasons. Software is predominantly a creature of intellectual property. The main intellectual property protection for software is copyright law which gives the author of software the exclusive rights to make copies, create derivatives or distribute copies to the public.[1] Additionally, these copyrights may be subdivided indefinitely.[2] These features of copyright law give software vendors tremendous flexibility in terms of the rights in the software that they might license or transfer to their customers.

Software is also somewhat unique because there is no manufacturing involved. Once the R & D on a software product is completed, the marginal cost of producing copies approaches zero. There is no need to build a factory to make the software once the first copy is made. The costs incurred after development are mainly selling, marketing and general and administrative expenses. Some packaging and shipping costs might be associated with software sold in shrink-wrapped boxes. Delivering software electronically is even less expensive than imprinting a copy of the software onto a physical medium such as a CD-ROM. Thus, concepts such as “margin” and “mark-up over cost” are somewhat foreign when applied to software copies.

a. Sales of Copies:

The simplest software business model is the sale of a single copy of the computer program. In the software industry’s early days, there were retail outlets that specialized in stocking shrink-wrapped boxes containing floppy disks or CD-ROMs on which was imbedded a copy of the vendor’s software products. These stores stocked an inventory of these boxes containing copies software from various vendors. Customers could come into the stores and browse the shelves, select the products they wanted and present them to the store clerk for purchase. In order to supply these retail outlets with product, the software vendors had to maintain a supply channel. Many times, the software boxes were provided to the retail distributors with a right to return unsold inventory, requiring the vendor to keep track of sell-through to individual consumers. As the members of the Boards are no doubt aware, there is very little of this retail marketing of software remaining.[3]

Today, software vendors directly supply their software to customers by making it available for download to customers. Many software vendors no longer market their products on physical media and only sell their software via download. These vendors do not need to supply any retail vendors with boxes containing copies of their product and thus they do not need to keep track of sell-through.

b. Online Software:

The advent of the Internet has made it possible for software vendors to provide customers with software functionality without ever delivering to the customer a copy of their software. A

[1] See 17 U.S.C. Sec. 106.
[2] See House Report No. 94-1476, p. 61, 94th Cong., 2d Sess., Sept. 3, 1976.
[3] An exception to this general statement might be for game software used on dedicated game consoles that require a copy of the game software imbedded on a CD-ROM.

growing trend in the market is for software vendors to provide customers with online access to their software's functionality. The vendor will maintain a facility with sufficient hardware that customers can access online and obtain the same software functionality available through purchase of a copy. A prime example of this business model is tax return preparation software. Most vendors of tax return preparation software sell copies of their software in retail outlets, make it available for download or allow customers to prepare their tax returns online. The online access business model requires that the vendor maintain a data center allowing customers 24-hour access to the software. Depending on the type of software, the customer may pay a one-time fee or be required to pay periodic fees for online access to the software's functionality.

c. Tiered Software Licenses:

Many software vendors closely control the type of machine that their software will be allowed to operate on. For instance, the software may be set to only run on a computer with a certain processing power. If the customer desires to migrate the software to a computer with greater processing power, the customer must pay a higher license fee and obtain the necessary access codes that will permit the software to run on the more powerful machine.

d. Enterprise Licenses:

Many software vendors provide a customer with a license to make a fixed number of copies (or as many copies as are necessary) for use in the customer's business. These licenses typically prohibit transfer of the copies of the software outside the business. Some of these arrangements require annual audits with true-up payments in the event a certain number of copies is exceeded.

e. Server Access Licenses:

Some computer programs are designed to operate on computer servers. The software may be set to permit a fixed number of users to access the software on the server at a given time. If the customer wants to allow more users to access the software than the license allows, it must pay a higher license fee and obtain from the vendor the necessary codes to allow more users.

f. Software Subscriptions:

A software subscription is a type of transaction under which a purchaser licenses software that requires continual updates in order to remain useful. The software vendor will sell the purchaser a copy of the underlying software along with a right to receive the updates for a specific period of time. When the time expires, while the purchaser remains entitled to use the copy of the software, it is no longer entitled to receive the updates. The vendor offers the purchaser the opportunity to renew the right to receive updates for a given period of time. An example of a type of software that is typically marketed using the subscription model is anti-virus software. In order to remain effective against newly developed virus threats, the software must be periodically updated with information about new threats. Many companies license a copy of the software along with a one-year subscription to the updates with the update subscriptions renewable annually.

g. Software Term Licenses:

A software term license is one that requires the purchaser to make recurring payments in order to retain the right to use the software. Many times, software is licensed for a fixed period of time under terms that require periodic payments. At the end of the term, the software will cease to function. Other times software may be licensed under terms that allow the purchaser to use the software for as long as it wants so long as it makes periodic payments.

If the purchaser fails to make a payment, the software ceases to function. Software vendors might enforce these sorts of licenses through the use what are known as software “keys.” When the purchaser makes its recurring payment, the vendor supplies the purchaser with a code that must be entered into the software in order to keep the software functioning. If the code is not entered prior to the expiration of the period, the software will shut down.

These types of licenses are to be distinguished from situations where a purchaser will license software on a perpetual basis but enter into an arrangement with the vendor to pay the license fee in installments. In these situations, the purchaser’s right to use the software is not contingent on the making of the future payments (although the vendor might be in a position to electronically repossess the software if the purchaser fails to make an installment payment).

h. Application Service Provider/Hosted Software:

The Application Service Provider (ASP) scenario involves three distinct business models. All three, however, have some facts in common. Generally, the ASPs depend on the desire of the end user of a computer program to outsource to a third-party the ownership and maintenance of the hardware on which the software runs. The “host” is the owner of the computer servers, networking and communications equipment on which the software runs. The host is responsible for maintaining the hardware and software and charges its customer a fee for this service. The three variations on the ASP theme are described below.

ASP-Separate License: Under the first scenario, a customer with a license to use a software product enters into a contract with a host entity, whereby the host entity loads a copy of the software on servers owned and operated by the host and provides technical support to protect against system failures. The host entity might either be an independent third-party or it could be the vendor of the software. The customer can access, execute and operate the software application remotely. The application is executed either onthe customer’s computer after it is downloaded, or remotely on the host’s server. This type of arrangement could apply, for example, to financial management, inventory control, human resource management or other enterprise resource management software applications. The customer has no control over the equipment used by the host entity.

ASP-Service Provider: Under the second scenario, the host entity has a license to use a software application in the course of its business. It hosts the software on a server that it owns, operates and maintains. The host entity enters into an agreement with a customer to manage a particular back-office function (e.g., the customer’s payment processing), and provides the customer with access to the software application, enabling the customer to perform specific tasks when required (e.g., data entry, addition of tombstone data for new suppliers and clients). However, the host entity is responsible for the major aspects of the payment processing, such as check issuance and bank verification, and uses the software application to automate these tasks.

The customer has no right to copy the software or use it other than for the specific functions assigned by the host entity; at no time does the customer have possession or control of the software (since it resides on the host entity's server). There is no provision of prewritten computer software to the end user and the end user acquires no rights to computer software, the fee is characterized as a service fee.

ASP-Bundled Contract: Under the third scenario, the host entity is also the software vendor. For a single fee, a user enters into a contract whereby a host entity, which is also the software vendor, allows access to one or more software applications, hosts the applications on a server owned and operated by the host, and provides technical support for the hardware and software. The user can access, execute and operate the software application remotely. The contract is renewable annually for an additional fee.

Since no software is ever delivered to the end user and the end user, under the terms of the contract obtains no rights to the software, there is no provision of prewritten computer software to the end user. Consequently, the fee paid by the end user to the host entity is a service fee.

2. Software Maintenance:

Typical software transactions involve multiple deliverables: a copy of the software and a "maintenance" contract or agreement. Generally, a computer software maintenance contract is an agreement between a software vendor and its customer whereby the vendor agrees, for a period of time, to provide its customer with additional software and/or technical support services. The additional software typically includes updates and bug fixes and upgrades or enhancements. Bug fixes typically do not provide additional features or functionality; they typically address incompatibilities between the vendor's software and software from other vendors that customers run concurrently on their systems. Upgrades and enhancements add features and functionality to the software and cause it to do things that the software was not initially programmed to do. Technical support services could include employee training, troubleshooting and merely helping the customer's personnel operate the software on a case-by-case basis.

Despite its widespread use in the industry, application of the term "maintenance" to computer software is somewhat misleading. Computer software is not tangible personal property as that term is normally understood in that software has no moving parts that might wear out and need replacement nor does it require any fluids such as coolants or lubricants which might need changing from time to time. Computer software does not require any sort of repair or replacement of its internal coded instructions due to wear and tear.

At the time computer software code initially is written, there is an understanding that the software will be running on customers' systems with various hardware configurations that might concurrently be running software from other vendors. An attempt is made to anticipate the combinations of other hardware and software that might concurrently be used and to develop the code so that the software will be compatible. As hard as they try, the computer programmers do not always anticipate every combination of hardware and software that customers might use; hence the requirement of periodic software updates that cure unanticipated incompatibilities. In addition the software might be used with other hardware and/or software that were not on the market at the time the software was released. As used in the computer software context, the term maintenance refers to this effort to “maintain” a given computer software product’s compatibility with other hardware and software products.

These software maintenance contracts are not a bundle of two separate products because many software vendors, as a matter of industry practice, typically will not sell rights to the additional software updates and upgrades separate and apart from the rights to the support services.

A few software companies will sell separately the rights to the upgrades and update separately from the rights to technical support services. The issue of separating the two elements does not arise in these cases. Also, in these limited cases, the existence of separate pricing for the elements provides a means of unbundling them when they are sold together by a company that does separately offer them.

Some software companies require that their customer purchase a maintenance contract as a condition to acquiring a license to the underlying software (known in the industry as “mandatory maintenance”). Mandatory maintenance comes in two flavors. Under one flavor, the software vendor will include with the price of the software a limited right to updates and technical support services. These rights typically expire within a short time after the customer installs the software. This form of mandatory maintenance is typically included with consumer oriented software. Under the other flavor, the vendor will separately price the maintenance and include it as a separate line item on the invoice. The vendor and the customer might negotiate the price of the mandatory maintenance and the customer may be given the right, but not undertake any obligation, to renew the maintenance on an annual basis. Many times, the price of the mandatory maintenance sets the price for the renewal. Other companies do not impose such a requirement (known as “optional maintenance”). Optional software maintenance would include renewal of a mandatory software maintenance contract.

Generally, when software maintenance is sold separately from the underlying software, it is priced based on a percentage of the license fee paid for the software. As noted above, many software maintenance agreements obligate the vendor to provide the customer with all of the technical support services it might consume during the term of the agreement. Thus, two customers purchasing identical maintenance contracts might consume vastly different abouts of support services, making it difficult to predict the ultimate profitability of any of these agreements.

Software Updates and Upgrades:

Most software maintenance agreements only require that the vendor supply an update or an upgrade only when the update or upgrade becomes available; there is no require that the vendor produce any updates or upgrades during the term of the agreement. This is know in the industry as an “when and if available” basis. The vendor obligation to supply any additional software arises only when the vendor creates the additional software; if the vendor creates no additional software during the term, no obligation arises.

Most software companies are constantly revising their software in order to take advantage of advances in technology and to maintain the competitiveness of their software. Thus, software updates and upgrades are created with new customers in mind, not existing ones. While the development of the updates and upgrades might be costly, delivering the additional software to the existing customer base can be done through inexpensive electronic delivery techniques.

Technical Support Services:

Technical support services provided under a software maintenance agreement generally are supplied through telephone help desks or are web-based. This type of service is provided on an as-needed basis such that the individual users of the software can access the services whenever the need arises. This generally requires that the software vendor have a staff of support technicians at the ready whenever a customer calls or seeks help through a web-based service.

3. Existing Software Industry Revenue Recognition guidance:

Today, the revenue recognition practices of software companies are governed by S.O.P. 97-2. This guidance generally requires that revenue be recognized when persuasive evidence of an agreement exists, delivery has occurred, the vendor’s price is fixed or determinable and collectability is probable. When all that is being delivered is a copy of the software, application of this standard is fairly straightforward. However, when the arrangement involves multiple deliverables, such as software and maintenance or hosting type services, the amount and timing of revenue recognition is complicated. The software provider often charges a single fee that must be allocated to the products delivered in the present and to be delivered in the future. SOP 97-2 requires that the vendor’s fee be allocated to the various elements based on vendor-specific objective evidence (VSOE) of fair value for each element. VSOE is limited to the price charged by the vendor for each element when it is sold separately. This requires the deferral of revenue until VSOE can be established for all elements in the arrangement or until all elements have been delivered. If technical support service is the only undelivered element in the arrangement, however, the entire fee can be recognized ratably over the term of the contract.

SOP 97-2 requires the allocation of revenue to all of the elements of a multiple-deliverable arrangement using the relative fair value method where objective and reliable evidence of fair value is present for all the elements in the arrangement. If objective and reliable evidence is available only for the undelivered elements, SOP 97-2 allows use of the residual method to value the elements that have been delivered. If objective and reliable evidence is available only for the delivered elements, no value may be assigned to any element until all of them are delivered. SOP 97-2 prohibits use of the residual method to value the undelivered elements where there is VSOE for the delivered elements.

As noted above, many times, software companies, for competitive reasons, do not separately price the various elements sold as a bundle to their customers. Application of the VSOE standard often requires that the vendor wait until the end of the term of the arrangement to recognize the bulk of the revenue from the sale. For instance, consider a sale that includes a copy of the software and a software maintenance agreement that obligates the vendor to provide both updates and upgrades to the software on a “when and if available” basis together with technical support services for a one year period. Because the vendor does not separate sell the software or either element of the maintenance agreement, it has no VSOE of a separate price for any elements. Under SOP 07-2, the vendor could not recognize any revenue associated with the sale of the software or the maintenance agreement until the maintenance agreement expired. This causes unnecessary distortion of the vendor’s revenue.

Under many of the software business models described above, the vendors many times sell term arrangements with multiple deliverables that include the provision of software functionality together with other elements that might include additional software and services of one sort or another for a single, undifferentiated price. Many of these agreements require the vendor to stand ready to deliver an element on a 24/7 basis. However, SOP 97-2 prohibits vendors from using management’s business judgment to assign reasonable values to the delivered and undelivered elements and recognizing revenue accordingly. SOP 97-2 also prohibits use of a ratable recognition scheme for multiple element arrangement that require the vendor to stand ready on a 27/7 basis to deliver one or more elements of the arrangement when use of such a scheme would provide the most relevant information to users of the vendor’s financial statements.

We are happy to see that the proposal set forth in the discussion draft would repeal industry specific revenue recognition guidance, including SOP 97-2.

4. The Discussion Draft Proposal:

As stated in the opening paragraph, SoFTEC supports the Boards’ preliminary decision to replace existing revenue recognition guidance with a contract based recognition principle with measurements based on the contract price. We believe adoption of such a principle would lead to increased reliability and comparability of financial statements issued by software firms. We recognize that the Boards’ proposal is preliminary and that any final standard would include enhanced detail on how the new principle applies to a variety of transactions. We urge the Boards to include in its final standard enhanced details on how the new standard would apply to transactions that require delivery of a variety of products and service, such as those that are prevalent in the software industry. We will point out specific areas where more detailed guidance is needed as we march our way through some of the specific questions posed in the discussion draft.

Question 1: Do you agree with the Boards’ proposal to base a single revenue recognition principle on changes in an entity’s contract asset or contract liability? Why of Why not?

We generally agree with the Boards’ proposal to base the revenue recognition model on changes in an entity’s contract assets or liabilities. However, has noted above, many software companies still sell packaged software which necessitates the shipment of inventory to downstream resellers and retailers. We are concerned that recognizing revenue based on changes in an entity's contract asset or contract liability could lead to a deceptive business practice of "channel stuffing" whereby a company would inflate its revenue by deliberately sending retailers or resellers along its distributions channels more inventory than they are able to sell to end users. By stuffing the channel, distributors temporarily increase their accounts receivables. However, when the downstream resellers or retailers are unable to sell the excess inventory, they will send the excess items instead of cash back to the distributor, who must readjust its accounts receivable and ultimately its bottom line. We suggest that your final revenue recognition standard include guidance that defers the recognition of revenue until the product is sold through to the end user.

Question 3: Do you agree with the Boards' definition of a contract?

Answer: No.

The Boards propose to define a contract as "an agreement between two or more parties that creates enforceable obligations." The discussion draft goes on to state that this definition is consistent with the FASB's current definition, cribbed from Black's Law Dictionary, which defines a contract as "an agreement between two or more parties creating obligations that are enforceable or otherwise recognizable at law." We see daylight between the two definitions. Black's definition specifically provides that in order for an agreement to rise to the level of a "contract" the agreement must impose obligations that are enforceable "at law."[4] The definition proposed in the discussion draft omits the "at law" requirement found in Black's definition.

We are concerned that omitting the "at law" concept could interject into revenue recognition determinations the possibility that an obligation not enforceable "at law" could be enforceable for moral, competitive or other reasons unrelated to the law. It would be impossible for any vendor to judge, at the time an agreement is executed, whether it imposes obligations not enforceable at law that are enforceable on some other basis and to assign prices to such obligations such that recognition of revenue associated with the obligation can be deferred.

We suggest that you revise your definition of "contract as follows:

> A contract is an agreement between two or more parties that creates legally enforceable obligations.

Question 4: Do you think the Boards' proposed definition of a performance obligation would help entities to identify consistently the deliverables in (or components of) a contract?

Answer: No.

The Boards propose to define "performance obligation" as "a promise in a contract with a customer to transfer an asset (such as a good or a service) to that customer." We agree with this. The discussion draft goes on to explain that the promise underpinning a performance obligation usually is stated in the contract or could arise by operation of law. We also agree with this. However, the discussion draft also further explains that an enforceable obligation could be

[4] The IASB's definition of contract also includes an "enforceable by law" concept.

created through an entity's customary business practice even if neither the contract nor the law explicitly imposes the obligation. We do not agree with this; further explanation is needed.

We are concerned that representations made with regard to future product plans that do not make their way into the contract could fall within this exception to what appears to be a general rule that performance obligations are to be discerned from the four corners of the contract and the applicable law. We fail to see how an entity's customary practice, which presumably it is free to change at any time, could result in an obligation that is legally enforceable. This gets into our suggestion, above, that you revise your definition of "contract" so that it incorporates the concept of "legal" enforceability.

We believe that the focus should be on the terms of the contract without reference to extraneous information about a vendor's plans for future products or services, unless those plans somehow find their way into the contract between the vendor and the customer such that a legally enforceable obligation arises.

We also ask that the Boards include guidance dispelling the notion found in some interpretations of SOP 97-2 that a promise to provide a software feature or functionality on a "when and if available" basis creates a legally enforceable obligation. In such cases, the entity undertakes no obligation to develop the functionality; the entity only promises to provide the functionality to the customer when and if it does develop the functionality. We do not believe an entity should ever have to defer the recognition of revenue associated with a promise to provide functionality on a "when and if available" basis because such a promise does not give rise to any legally enforceable obligation to develop the functionality.

Question 6: Do you think an entity's obligation to accept a returned good and refund the customer's consideration is a performance obligation? Why of Why not?

Answer: No.

We think it would be inappropriate to defer the recognition of revenue until a right of return and an obligation to issue a refund expire. Identifying performance obligations related to returned or refunded goods as elements of a contract can be complex and require judgment based on facts and circumstances. The value related to returns or refunds are not readily determinable at the origination of the contract. It is difficult to establish a valuable measure on a per contract basis particularly in a multiple element arrangement. We believe the better way is to allow entities to use their historical experience in estimating the portion of sales that will be returned. We suggest existing guidance under FAS 5, Accounting for Contingencies, and FAS 48, Revenue Recognition When Right of Return Exists, should be incorporated into the final standard. Deferring a portion of the revenue from each sale attributable to expected return and refund rates would produce more reliable results.

Question 8: Do you agree that an entity transfers an asset to a customer (and satisfies a performance obligation) when the customer controls the promised good or when the customer receives the promised service? Why or why not? If not, please suggest an alternative for determining when a promised good or service is transferred.

Question 9: The Boards propose that an entity should recognize revenue only when a performance obligation is satisfied. Are there contracts for which that proposal would not provide decision-useful information? If so, please provide examples.

The concept of an asset transfer goes in and out of focus in the context of services. The examples set forth in the discussion draft are cases where the entity obligates itself to provide a discrete service such as painting or construction. The sorts of services contracts of concern to SoFTEC members, described in the business models section of this letter, are contracts that require the vendor to make a service available to the customer on the customer's demand. These services contracts give rise to an asset transfer in that they give the customer the right to receive the services for the duration of the contract. That asset wastes as the term of the contract approaches its end.

Take the case of a contract that obligates a vendor to provide technical support services as a part of software maintenance. The vendor must hire and train a workforce and set up a telephone support center such that customers can call in whenever they need help with the software. Even though the customer has paid for the right to have the help center personnel at its beck and call, it may never use the service. Nevertheless, the vendor satisfies its obligation to provide the support by making the technical support services available. But when would the customer be said to have received the service? These same difficulties arise when the vendor is obligated to maintain a data center in which the customer's software is running or to which the customer has a right of access to software functionality. The obligation of the vendor is to maintain the help desk or data center and make them available to customers around the clock.

These types of services are not the sort of services which the customer receives all at once at a discrete point in time. We believe the customer receives the services over time throughout the duration of the contract and that revenue received from the delivery of the service be recognized ratably over the life of the contract, not all at the end. We urge the Boards to include in the final standard some examples providing guidance with respect to these sorts of services contracts.

Question 12: Do you agree that the transaction price should be allocated to the performance obligations on the basis of the entity's standalone selling prices of the goods or services underlying those performance obligations? Why or why not? If not, on what basis would you allocate the transaction price?

Question 13: Do you agree that if an entity does not sell a good or service separately, it should estimate the standalone selling price of that good or service for purposes of allocating the transaction price? Why or why not? When, if ever, should the use of estimates be constrained?

We agree that the transaction price should be allocated to the various elements of the transaction based on the entity's standalone selling prices of the individual elements. This approach brings the benefits of simplicity and objectivity which together further the objectives of reliable and comparable financial statements. We also agree that if the entity does not separately sell a product or a service it should estimate the selling price for purposes of allocation. Use of estimates, however imperfect, would produce more meaningful results than an alternative model that would defer all revenue until all performance obligations have been satisfied.

In some cases, the marginal benefits to be gained by attempting to use estimates to assign selling prices to products not sold separately are so minimal that it may not be worth the exercise. We believe allowing use of a simple ratable revenue recognition method in complex delivery situations (such as are prevalent in the software industry) would provide meaningful and comparable results in a cost/benefit analysis.

5. Conclusion:

The software industry generally support the revenue recognition approach laid out in the discussion draft and urge the Boards to include additional examples providing guidance with respect to transactions that apply the contract-based revenue recognition principle to transactions peculiar to the software industry. We thank the Boards and their staffs for the opportunity to presents these comments on the discussion draft and look forward to working with the Boards as them move towards exposing a more complete revenue recognition proposal. I can be reached at mnebergall@softwarefinance.org with any questions.

Respectfully submitted,

Mark E. Nebergall
President
Software Finance and Tax Executives Council