

Webcast: Implementing a Best Practice Technology Escrow Program for Your Organization's IP

Date and Time: Wednesday, March 1, 2006 at 1:00 PM ET

Sponsored by [Iron Mountain](#) Intellectual Property Management Division

Presenters: Frank Bruno, Senior Business Strategist, Iron Mountain Intellectual Property Management and Max E. Iori, Vice President and Assistant General Counsel, Legal Department, JPMorgan Chase Bank, N.A

Moderator: Christopher W. Ekren, Vice President, Law Department, Sony Electronics Inc.

ASSOCIATION OF CORPORATE COUNSEL

Moderator: Christopher W. Ekren

March 1, 2006

Operator: Go ahead, (Chris).

(Chris): Welcome, everyone.

My name is (Chris Ekron). I'm a vice president for the (Semi Electronics) law department and I'm also, this year, the chair of the ACC IT committee.

I'd like to welcome our speakers: Frank Bruno, Senior Business Strategist with Iron Mountain Intellectual Property Management; and (Max Fiori), who is Vice President and Assistant General Counsel ((inaudible)) ...

Operator: Just a reminder, today's conference is being recorded.

(Chris Ekron): ... legal department at JP Morgan Chase Bank.

Two administrative matters – if you have questions, we have a chat function. Please enter your question at any time by entering text into the lower right hand corner box, which you can enlarge using the enlargement function, if you want to have more space. And then hit the send button. However, what we'll be doing is taking all the questions at the end of the Webcast and then, at that point, we'll read them out and respond to those.

Secondly, I'd like to remind everyone that your opinions are important and there will be a survey and evaluation available at the end and we'd like to encourage everybody to take the two or three minutes it takes to fill out that survey before you log off.

So, with that brief introduction, perhaps our speakers can begin with today's Webcast.

Frank Bruno: Thanks, (Chris).

This is Frank Bruno and thank you for joining us on implementing a best practice technology escrow program.

Joined by (Max Fiori). (Max), how we doing today?

(Max Fiori): Doing good.

Frank Bruno: Terrific. Terrific.

Let's move on to our learning objectives if we can. We want to cover three things today, obviously, to understand the dynamics of escrow protection and when to use it; secondly, we want to focus on the contract language to maximize that protection; and then, finally, we'll move into how you optimize the vendor relationship and use an escrow to your advantage. It's also something that's going to help, hopefully, your company IT folks to expand on their software asset management best practices.

So, the grim reality – we may or may not have been there in the past; but if this should happen to you, you'll get that cold sweat building on your brow quite ((inaudible)). And so, you know, you may have a problem with your software application. You call your software vendor and this is what their offices look like. Now, this may be a case (of) bankruptcy. The developer may be out of business completely. But, one other thing to consider, and something that's become a growing trend, is mergers and acquisitions. And so, you know, if you look at that figure, 952 mergers and acquisitions took place last year; which would result in certain products either being acquired and subsequently sunsetted.

So, you know, (Max), tell me, does this issue get discussed frequently when negotiating escrows?

(Max Fiori): Yes, it does, Frank. It is a – it is a high number, but I'm not surprised. Just because for every instance where you read in the Wall Street Journal or the New York Times regarding acquisitions such as JD Edwards or Oracle or PeopleSoft, there's quite a number of smaller software vendors that are being acquired and who are acquiring their competitors. So, in those cases, we, right at the beginning stages of picking a vendor, we keep that into account because if they – if an acquiring company has the intent to sunset the acquired

company's product, we need to make sure that we have the ability to support that software in-house if the maintenance is no longer offered.

Frank Bruno: Thanks for that.

Case in point, we encountered a major food service company, a few years back, who had to replace 10 mission-critical applications after their vendors disappeared. This cost the company 15 million in unbudgeted funds; which, you know, in today's standards, could have represented a most painful Sarbanes-Oxley experience.

On the flip side, we've also had experiences where, for instance, a film production company was able to utilize and quickly redeploy their special effects technology after their vendor quit. And fortunately, they had terms and conditions written into their agreement to have their code verified. And who knows? Without it, we might not be enjoying some of the special effect that we see in the movies today.

So, on with our first learning objective; understanding the dynamics of escrow protection. Let's take a look at the definition of escrow. And you know, it's a practice of securing access to property through a trusted, neutral third party under certain limited circumstances; a triggering event, a release condition. Parties include the depositor – obviously, this would be the seller in the underlying agreement; also referred to as the developer, licensor, grantor. The beneficiary is the acquiring company, the buyer; also referred to as the licensee, the user, grantee, et cetera. And then, finally, there's the escrow agent, which is the trusted, neutral third party; also referred to in certain cases as the custodian, holder and agent.

Why is the source code so critical?

I'm not sure if everyone's having a problem looking at this slide on their screen. If so, just please, follow along as best you can.

But, source code is critical. The source code is the human, readable version of the software. Compilers are used to translate this computer-readable language into what we refer to as object code. We will also interchangeably use the term executable along with object code; they mean the exact same thing so please, bear with me. And then, finally, your computer system reads the ones and zeros that run the software application. So, the code is critical because most developers don't provide that to you and since the source code – or rather, the object code – cannot be reverse-engineered back into source code, the licensee would be unable to support the program.

So the access to the key development data is essential to maintaining the technology and the source code is the secret recipe – if you will – to a vendor maintaining a sustainable revenue stream. And that will come in the form of the maintenance invoices that most companies pay for each year.

So (Max), in your opinion, when you negotiate escrow for JP Morgan Chase, how well do you think that the business understands how critical the source code is to them?

(Max Fiori): I would say the business understands it very well; just because, generally, the business is made up of project managers and technologists who not only need to operate the software on their own, but they also have a very integral understanding of the software and what it

needs to do. However, I should mention that I think that when you're negotiating a software license deal, the – because the client or the business is oftentimes managing the entire project and perhaps considering sun setting a prior version or a prior product that the new product is going to be taking over for, it's also the responsibility of the procurement or the sourcing professional, in conjunction with legal, to keep the requirements of the software escrow in mind while you're going through the transaction.

Frank Bruno: Great.

So, here we have the developer/licensee dilemma. On the one side, you know, developer has certain desires. They want credibility because, of course, they want to sell some product. They also want to shorten their sales cycle and ultimately satisfy their client. On the other side of the coin, the licensee – obviously, they want an improved ((inaudible)); that's the third bullet to the right there. But in the process, they want to mitigate risk and avoid litigation at some point.

So the developers have increased competition. The licensees have increased business pressures to get things done. And of course the outcome, while the ultimate goal is common between the two, there are some desired (actions) which may differ; you know, closing deals, market traction, IP protection differs from what a licensee might want, which is a controllable budget, leverage or support after the license agreement has been signed and then, of course, you know, they both want a long term relationship. And the escrow kind of provides an ideal solution for both parties.

If we look at the target audience, the people that we deal with, this slide is pretty significant because what we often find is that there is a breakdown in communication somewhere between the time that the license and the escrow is signed to the time that the release actually takes place. And if a release does actually take place will the code be usable? So, let's look at this target audience closely if we can.

You've got the IT folks who provide the service delivery. They're the ones who are responsible for providing 99.99 percent up time. You've got the business owners on the upper right hand side who are ultimately responsible for the application and they depend on the IT folks for that support. And then, below those two folks are the procurement and contract folks, part of the supply chain or acquisitions process and then the legal folks who are – who are consulted to review the agreement and to make sure that the appropriate risk management issues are addressed. So, you know, this is very much the common scenario; the people that we deal with on a day-to-day basis.

(Max), am I missing anyone here?

(Max Fiori): I wouldn't say so, Frank. As I mentioned before, I think it's a conjunction of the procurement sourcing professionals with the client, and also legal, when the agreement's first being negotiated. And when I say agreement, I mean the software license agreement which should also be negotiated in conjunction with the escrow agreement.

Just one point I'd like to mention is that, going forward, you should also have information risk managers who, on a regular basis, audit the use of the software at the licensee location

to make sure that all updates and anything else that's needed to support the software is being added to the escrow on an ongoing basis.

(Chris Ekron): Hey, Frank and (Max), could you guys speak up a little bit more? We're getting a few comments from people on that.

Male: Will do.

Male: Sure, will do.

(Chris Ekron): Try ((inaudible)) your speaker – your microphones. Thank you.

Frank Bruno: And that's an excellent point – discussing the info risk managers – and we will address that a little later in the presentation as well; in terms of how you manage the software asset.

Let me ask you one more question, (Max), on this slide. Who typically drives the escrow requirement for JP Morgan Chase?

(Max Fiori): Who drives it while the agreement is being negotiated, I would say, is the sourcing and procurement department. They're the ones who make sure that we get the software into escrow. This is generally in conjunction with legal, who assists with the negotiation of the underlying agreement.

Frank Bruno: Great, great.

OK. Well, let's take a look at some other uses for escrow if we can. At the top of the list, I have the application service providers and that is a significant and, actually, newer trend in our business; where in addition to traditional deposit materials, you actually want to look for a copy of the object code – again, we interchangeably refer to that as the executable – the hardware environment and then, of course, having access to current data in the event that there was a failure for the application service provider. And then, there is a matter of managing or protecting your own intellectual property. So if your company has patents, copyrights or trade secrets documented for ownership, this may be something that you'd like to put in escrow just to document and track the product genealogy if you will; but ultimately to prove ownership in a court of law if it comes to that.

We have many, many organizations – a lot of venture capital firms – that look to preserve their capital when they fund a startup. And so, collateral preservation is another decent application for escrow. And then, joint ventures; (reseller) arrangements. Many developers today don't service their own applications. What they do is through channel sales, they'll find a valued (reseller) to maintain the code for the end user. And so, you have a lot of joint venture or (reseller) arrangements which will require an escrow and that extends beyond the traditional two or three-party deal and now you've got additional parties involved which we can address at some later point.

And then finally, anything else that you might need access to, under certain conditions, that the other party doesn't want you to have right now.

Let's just take a quick look at source code versus object code if we will. And I mentioned in the earlier slide, application service providers, this is a pretty significant development in

our business in that the application service providers are becoming more and more prevalent.

I'm sure many of you have written – or rather, read – some news about software as a service.

And this is really an application that is served up through the web, through a provider who is hosting the application. So, the source code is not resident on your computers. It's resident on your vendor's computers and being served up through the web, accessible through your Internet Explorer or one of the other applications to surf the internet.

And the significance here is that, you know, (ASPs) sell software as a service. The source code and the object code is proprietary. The event of a failure – and let's say, for instance, the application service were to stop providing a service or to have a failure of some sort that was going to extend beyond days or beyond a point of disaster recovery – you would need to get access to that application to run without the vendor. So, access to the object code would enable the subscriber to keep using the application without the application service provider.

Let's look at some types of escrows that we – that we typically support. The traditional arrangement is the three-party arrangement; that's where all three parties negotiate and sign the escrow contract. There is typically one developer, one user and one set of deposit materials. And then, on the right hand side, I've got two and three-party masters and this is an upcoming trend that we're seeing more and more of; where either the licensee or, in certain cases, the beneficiary will execute a master agreement and then link third parties to that agreement so that they could standardize the terms and allow for multiple products to be secured using one agreement.

So, on this slide, you see the two/three-party master with the developer as the contract party and they're adding licensees to their escrow agreement and then they could all (share) multiple products. And then, of course, on the next slide, you'll see that we have a master three-party beneficiary agreement where the beneficiaries are now standardizing the terms and conditions of their escrow agreement, presenting it to their vendor before the license agreement is signed and then, as the escrows are executed, they're simply adding the depositor as a contract party to their escrow agreement through deposit or acceptance form.

So (Max), let me ask you this question. Has JP Morgan Chase standardized their escrow agreement and do you have any opinions on a master?

(Max Fiori): Yes, we put together a form master escrow agreement that has been agreed to by both Iron Mountain and JP Morgan Chase. So, when we enter into a software licensing deal with a vendor, we're able to present the vendor with that form agreement that's already been agreed to by two of the three parties. So, it really makes the whole transaction and the negotiation proceed smoother. And I think it is the way to go.

Frank Bruno: Thank you.

OK, on to objective number two; leveraging the contract language. In this section, we're going to cover off on deposit materials; how you verify those deposit materials. We'll talk a little bit about release conditions in the process. And then, finally, the rights to use.

First off, leverage after the license is signed. As you look at the slide, it'll kind of take you through the process and let's say that there's a problem. You know, obviously, you call your

developer; hopefully, they answer the phone. And if they rectify the problem, you know, the desired outcome is that you're a happy customer. However, if the situation is not rectified, then you would likely refer to the remedies of your license; which would include the escrow agreement and, potentially, you may seek a release. If the developer is still operating, again, we hope that they take care of business; but if the problem goes unresolved or if the developer's no longer operating, then we would ((inaudible)) and simply release the source code.

Knowing your terms and conditions is very, very important. And you know, what we typically look at or what we see negotiated many times is, you know, the deposit contents; the update frequency and the update process. We often see people negotiating verification rights. Certainly, release conditions is something that changes with every agreement that we see. And then, of course, the objection period; how long does a developer have to respond to a release request and what kind of contrary instructions can they give us. And then, you know, the rights to use the software source code after release has been made; who's going to pay for all this; and finally, you know, what happens if there's a problem or a dispute that needs to be resolved.

(Max), in your opinion, what's the most frequently negotiated term here?

(Max Fiori): Actually, Frank, you covered most of them. I would certainly say the release conditions is one of the most negotiated points. And I would say the failure to support or the failure to continue to provide maintenance for the software is one of the most important of those release conditions.

I would also say, regarding dispute resolution, if a licensee feels that a release condition has occurred, what's the next step? That process there is also heavily negotiated; what kind of notice goes out to the licensor, what kind of time frame is the licensee going to be waiting through until Iron Mountain releases the software source code or until the licensor responds or provides some sort of remedy for the – for the failure to maintain or whatever the release condition was. I would also say the payment of the fees is also heavily negotiated; not only the annual – the annual fee for maintaining the escrow, but also the initial setup fee. Also the fees for the verification services are also negotiated. So I would say those are – those are mainly the major points that are negotiated.

Frank Bruno: Thank you. And the good news is by the end of this presentation, you'll have some strategies to address with the other party as it relates to paying some of these costs. So we'll cover that in another – in the next few minutes.

OK, let's talk a little bit about deposit contents and practices. Obviously, you want to make sure that you will be able to create a similar environment that your application currently resides on. And these are some of the things that we typically look for in a deposit; you know, the maintenance tools, proprietary third-party system utilities, instructions on where to get third-party utilities and how to deploy them, the description of the system and the program generation. Essentially, we're talking about build instructions. We can refer to those things as the build instruction.

And one of the most important things that I find missing from a lot of deposits is the names and home addresses of key technical people who can assist with recreating the application development environment, absent of the developer itself. And so, if you're thinking about

whether or not this is something that a developer would agree to, you know, ask yourself this question, “If my developer was out of business, might this person be looking for work anyway?”

So, considering that, you may want to ((inaudible)) on that and it’s not like this information is going to be posted on the internet somewhere. It’s going to be stored on some magnetic media or electronically in a vault and never to be seen until a release request is made.

And then, finally, the list of encryption keys or passwords used in the escrow deposit. I did slip in one last ((inaudible)) and that was relative to the application service provider arrangements that we see happening a lot. And so, ((inaudible)) make sure that you’ve got a copy of that executable – again, we also refer to it as ((inaudible)) code – as long as the hardware environment and the data that goes on that environment.

So (Max), you know, tell me – how important is it ((inaudible)) this stuff up front and would you recommend anything else here?

(Max Fiori): Frank, I think this is a very thorough list and I think it is very important, up front, that the – that legal, along with the procurement folks and the client and the business, all get together to really understand what would be needed to operate and continue to support the software on an ongoing basis if the – if the vendor was no longer around. It’s also important to understand the use that you’re going to be using the software. Meaning, if you have any sort of sublicense ability, if you are turning around and using the software on a customer installation, let’s say, and each customer installation has specific customizations or specific

development for that customer's operating environment, you would want to get the source code for all those customizations; each one of those; continually updated to the escrow that Iron Mountain is holding.

In addition to that, I would also say all enhancement; any updates or upgrades need to be escrowed. And, as I mentioned, the customizations. And as you said, Frank, I'd like to echo that it's very important to get the names and the contact information of the actual developers for the source code. Because, as you mentioned, if the vendor's no longer in business, most likely they're going to be looking for a job.

Frank Bruno: Great.

OK. Well, now, when you talk about deposit contents, you know, the only way that you know that you're going to get those things is if you verify it. And you know, typically, what we look to do in a verification is to ask three questions. You know, the first is, you know, are the items that we've referenced, do they appear to be present. And then, the next question – which requires us to kind of, you know, roll up our sleeves and get into the code a little bit – is can the executable be recompiled. So, we're taking the piles that are contained and actually putting it together to see if we could get the object code. And then finally, you know, once you have been able to compile the code, does it work.

Frank Bruno: The trends that we've seen as – and again, this is – this is pursuant to Iron Mountain information only and not industry general information – I'll say that 66 percent of all deposits that were sent in for analysis determined – were determined to be incomplete. And so, something was missing. Ninety-two percent of the time, we wouldn't – we were not able

to compile the code without additional input from the developers. So, if you look at the value proposition of verification, essentially what it does is it ensures that the build instructions are going to be complete. And if there's any questions relative to the build instructions, they're going to come out during the verification testing process.

(So is it) necessary? You know, our experience indicates that 38 percent of all verified deposits didn't contain any build instructions. And – or there may have been corrupt media or missing files 28 percent of the time. And then, of course, 50 percent of the time we required major, major input from the depositor. And not for anything else. I mean a depositor, in all good faith, is looking to satisfy their obligations under the escrow and provide a usable deposit but they may be going through it for the first time and they may have left something out. So, innocently, you know, these percentages really represent more human error than anything else. And you won't know it until you – until you dig into the code and figure it out for yourself.

At the bottom of the slide, you'll see a timeline and this is typically how verification testing unfolds. ((inaudible)) the depositor completes a questionnaire which the escrow agent ((inaudible)) just to provide some background information about the software application. The escrow agent develops a statement of work, which would ((inaudible)) any resources, time and cost associated with recreating the application development environment. The licensee would review and approve the statement of work and then, of course, execute the statement of work. The test is conducted and then, of course, once the test is completed, the report is issued and any corrective action is taken at that time.

Let's normalize this to some of the other industries that are out there. I mean, when you look at banks, they're always going to examine collateral. If you look at a real estate deal – I mean, personally, I can't get a mortgage without having an inspection or an appraisal done on my home. And then finally, if you look at a life insurance company – and those deals are not that large – they actually perform a physical examination on every policy that they sell. The question is you know, why would you spend hundreds of thousands of dollars – potentially, millions of dollars – on mission-critical technology without verifying the code?

OK. Let's take a look at the standard release process. If you follow the arrows, starting from the upper left, the licensee believes that a release condition has been met so they send a release request to the escrow agent in writing. The agent then sends a copy of it to the developer and then, of course, the developer has a certain period of time to respond or to provide contrary instructions. At the top, if the developer agrees that a release condition has occurred, the escrow agent would deliver a copy of the deposit ((inaudible)). If the developer does not respond by the end of the contrary instructions period, then we would go ahead and release a copy of the code to the licensee. But if they do not agree that a release condition has occurred, they would issue contrary instructions to us and then, of course, we would submit that to a dispute process and we would retain the deposit while a determination is made.

(Release trends). If you look at this slide, you know, it tells a story. You know, there have been a significant increase in the number of releases, much to do with the number of software (licenses) that have been written. But mostly, beneficiaries are realizing that it's not really the responsibility of the depositor to get the escrow in place. It is their responsibility if they are to represent their own best interests. And so, what we ((inaudible))

is that the beneficiary community – the user community – is becoming more savvy. And so, we see a 61 percent increase in the number of release requests from last decade to this decade and we're only through 2005.

Hear are some of the release conditions, as ranked by the percentage of total release requests that we get. And as you see, bankruptcy is the number three item on there and it falls behind ((inaudible)) business operations at 22 percent. And we see this because it does cost money to file for bankruptcy. And so, some companies – if they're a small developer – will simply shut their doors and disappear. So it's very, very important to be specific, cover all possible circumstances and think beyond bankruptcy.

And (Max), you know, in your opinion, if you had to choose any particular release condition to be the most critical, you know, which one would that be?

(Max Fiori): Frank, I would say it's the first one, which is loss of support or if the vendor is no longer able to provide maintenance for the product. That's the most important one.

Frank Bruno: OK. And we're going to cover a little bit more of that. On this slide here, you'll ((inaudible)) release conditions. We won't focus too hard on it but suffice it to say if anyone needs a copy of this presentation, we will make that available to you.

And you know, in terms of the rights to use the code after a release has taken place – (Max), I'll just ask – have we left anything out here?

(Max Fiori): I don't think so. I just think it's important that if a licensee needs to maintain and support the software on its own, it has everything that it needs to do so. It has the source code and all the related materials, the commentary, any customizations and the source codes to such customizations. I think it's also important to make sure that if the licensee needs to allow its third-party vendor or some sort of outsourcing company to use the source code as well to support the licensee's business; so only on behalf of the licensee. And also the requirement transfer the software offshore – near shore if needed to be used again totally on the licensee's behalf.

Frank Bruno: Very important. Very important to consider that.

OK. So, let's move on to learning objective number three. This is actually my favorite part; optimizing the vendor relationship and expanding software asset management best practices. In a nutshell, check the escrow agreement; check the escrow deposit; and verify the deposit.

(Gardner) has supported a lot of what we – what we counsel our clients to do. And you know, if you read – if you read the quote, you know, it's very important. If you don't plan on doing regular audits or verifying the version of the software you're using, that is in escrow, the agreement may be worthless. If the vendor falls behind on their deposits or if the deposits are incomplete or unusable, the escrow agreement is basically useless. You know? And it's a tough thing for us to say, but if we're going to do the escrow, we want to make sure that we're pointing out to our clients that the right way to do it is to make sure that you go the whole step and budget for appropriate verification testing in addition to securing the contract language.

So, the pre-licensee protection strategy – we always counsel clients to determine whether or not they need escrow and verification. And if we're talking about a software package that's shrink-wrap, off the shelf, it may not be necessary to escrow because you can easily replace it. You do want to start early. We would counsel our clients to do this during the vendor selections process; perhaps at a time when they're sending out requests for information – we also refer to those as RFIs – and include, you know, a copy of your escrow agreement. Include a copy of the deposit questionnaire. Ask them the ((inaudible)) fill that stuff out. And this way, you can determine what the total cost of the escrow is going to be and then budget for those anticipated costs to protect those investments.

Very, very important that you have a well-balanced agreement. Obviously, you want to get through the negotiations process as quickly as you can and so, you know, having something that the vendor will readily agree to is very important to speed up the deal. Also, you want to specify the deposit materials. You want to make sure that you're getting everything that you're going to need to recreate the application development environment and then verify it. Finally, you know, you want to use somebody that you can trust to guide you through the process. Obviously, many times we find clients who are going through the escrow process; they may be going through it for the first time. And so, if you have somebody that you can trust, you know that you're going to be able to get it done consistently every time. And then, finally, we do advise to properly manage the escrow agreements, (stay involved).

And then, you know, in terms of best practice planning for licensees, here is the timeline on how we see things unfolding. Obviously, starting early; getting the deposit questionnaire filled out at the RFI stage; the completed questionnaires provided to the escrow agent; and a

statement of work is created. That statement of work includes the cost to get that done.

And so now you can include that in your program budget and not have to go back to the CFO at the eleventh hour to say, “Yes, we have to spend an extra couple thousand dollars to get this code – this code verified.” Of course, you do also – if this is early enough in the vendor selections process, you can consider these costs as you decide which vendor to work with and then, once the winning vendor is selected and the agreements have been executed, you can get the agreements (marked “Enforced”) and commence the verification testing.

(Max), does JP Morgan Chase have a documented process for safely acquiring mission-critical technology?

(Max Fiori): We place all software used in the production environment here. The source code for such software must always be placed into escrow. So it’s not just for mission-critical technology but it’s across all software. So any software that’s used in production needs to be – the source code for such needs to be placed into escrow. And as you mentioned before, on an ongoing basis we do have reviews and audits by our information risk management team who makes sure that that’s being done not only for the base code but also any updates, upgrades, customizations, so on and so forth.

Frank Bruno: Terrific. And when we talk about identifying your risk, you know, we’ve got four quadrants here inclusive of operational dependencies; the costs associated with it; of course, there’s the investment of time; and then, finally, the vendor. I’m not going to read off the criteria but, you know, all of these things contribute to your acceptable threshold of pain. And I say that tongue-in-cheek.

But (Max), how would you rank these quadrants by order of importance?

(Max Fiori): I would say that most of all the vendor assessment is the most important. It's important to understand the financial viability of the vendor; its future plans to the best that you can; what they are – what they're planning to do with the software on a going-forward basis. You do not want to go down the road with a particular vendor and place all your eggs in one basket, so to say, with that vendor, only to find out that that vendor is acquired or is planning on sunsetting the product. So I would say, above all of them, the vendor assessment is the most important.

Frank Bruno: Thank you.

And so, post-license strategy – obviously the agreement, once it's put in place, the application is in production, it's important for the people who are going to be required to maintain the application, to be the owner of the vendor relationship to gain understanding of the rights under the escrow agreement and the budgetary costs to maintain it. You may be talking about your information risk managers. This may be a part of your software asset management program. But it's very, very important that you consider these criteria when – once the license has been signed. And so, leveraging the escrow to optimize the relationship with your vendor is very important. I mean, obviously, if they're not supporting the way they should be and you know, as an information risk manager, that this is the release condition written into your escrow, you can use this or leverage this provision as teeth to support the service level agreements that you have with your vendors.

And so, you know, let's take this timeline and say, for instance, you know, the software asset manager's in receipt of an annual invoice. And so, as part of their software asset management program, they're going to go through and do all their normal checks. But then they're going to check with the escrow agent to make sure that the version of the software that's in escrow is current with what they're supporting in production. And then, of course, you know, if there's a major release of that code, you may consider budgeting to verify that deposit to make sure that it works. And you know, finally, don't process or approve any invoices until everything has been completed. And if you don't – you know, if you don't have an escrow or if the terms are weak, you really want to consider the risks associated with doing nothing.

So, the software asset management best practices compliance and – compliance audit and contract remediation – if a company is wondering whether or not they have a software escrow in place, it may be a good idea to review the underlying license agreement for any agreed upon escrow provisions. And then, of course, if there is an escrow provision in the license agreement but an escrow has not been executed, then obviously you want to make the business stakeholders aware of this and you want to just point out the inherent risks of not having an escrow in place.

And then, you know, finally, you want to do a cost benefit analysis to justify the cost of getting an escrow in place versus perhaps finding a new solution if you didn't have access to the code and your vendor failed you. This is something that, you know, you need to consider. If this is the case and you need to get the escrow done, the next step in the process, obviously, is to pursue a dialogue with your vendor. And then, if they're not willing to cooperate with you, you need to consider alternatives and contingencies.

And of course, you know, the cost benefit analysis is pretty simple. You know, to get an escrow and to have the code verified might run you anywhere from seven to \$15,000; whereas the cost to replace a mission-critical piece of software, you know, may run you – you know, the sky's the limit. You know, we just need to consider that as a big part of the budget if it should come to that.

And now, here's the timeline. Obviously, the – for, you know, the vendor to receive payment on his invoice, you kind of want to go through these steps just to ensure your own protection. And so the vendor's invoice is received. The software asset manager checks the license agreement to make sure that they are in compliance; they have the number of seats that they've got the software residing on the specified machines et cetera, et cetera so they're in compliance with the license agreement. But then you also want to check the escrow deposit to make sure that it is current with what you're supporting in production. And then, if there's any deficiencies discovered along those lines, you need to communicate that to the vendor and the business ((inaudible)).

Finally, you need to conduct verification on any new major version release and then, once the audit is completed, you can process ((inaudible)) and get it paid.

The next slide; selecting a reputable agent. You know, there are qualitative conditions and quantitative conditions which should help you decide which escrow agent to work with. I won't go through all of these – all of these criteria. But I will ask (Max), you know, what would you think – what would you add here as it relates to the qualities that you would look for?

(Max Fiori): I think it's important to choose an established escrow agent that you know is going to be in it for the long haul. While you can be worried that your software vendor might disappear, stop supporting the software, get acquired, you want to make sure that the escrow agent also has the same – the same viability. I also think it's important that you choose an escrow agent that's able to not only offer the simple service of holding on to the source code but also the full breadth of services regarding the verification. Because if it is mission-critical software, most likely you are going to want to have it verified that what's actually being deposited is enough for you to support the software in your own environment.

Frank Bruno: And I'll say one last thing about this slide and that is if you did a Google search on software escrow, you would probably get about 10 hits. And it's very, very important to consider the company that you're working with for all the reasons that we stated here and the ones that (Max) added. But most importantly, many of these companies do operate out of their homes; out of a shared office suite. Many of them don't have the infrastructure to support the escrow. Some of them may outsource the vaulting of the deposit materials, which is a very, very risky proposition. Our guidance is and will always be that if you are considering an escrow agent, to go and check them out. Go to their facility, meet their people, see their systems, check (their) disaster recovery plan before you made a decision on that.

The next slide basically determines what makes an ideal storage environment. And if you look through – obviously security systems want the video, motion sensors, biometric access, physical onsite presence. You want to make sure that your code is limited to only those bondage (framed) vault administrators who know the barcoding inventory and can locate

your deposit materials. But you want to make sure that the vault only contains technology-related materials only – typically, this is a media vault so we're only storing media like CDs, DVDs, tapes, things of that nature – and that the environment has a constant room temperature and humidity; four-hour fire rated fire walls; and of course, a non-water based fire extinguishing system; and most importantly, a documented process for handling deposits, for handling problem resolution and a disaster recovery plan; very, very important to consider.

And finally, a recap. (Max), you know, obviously the slide makes it look so easy; read the contract, verify the escrow deposit. But we all know that it isn't easy. Can you share any pointers as it relates to executing an on – and a holistic strategy to get those done right?

(Max Fiori): I think, to drive the point home, I have four major points just to mention here. One is you should negotiate your escrow agreement at the same time you're negotiating your software license agreement. If you negotiate your software license agreement then try to follow up with the escrow agreement, as a licensee, I think you've lost a bit of leverage. If you do both at the same time, they're both going to be pushed along equally at the same speed and in one shot you should be able to get both requirements done.

The second point is that you need to make sure that you get into escrow everything you need to support your client. You need to have close communications with your client. You really need to understand how they're using the software. And make sure that all customizations and upgrades and enhancements are not only placed into the software but – this is my third point – you have to make that it's continually being updated.

And just the fourth point I wanted to mention is the release condition of failure to support or failure to continue to receive maintenance on the software is, in my opinion, one of the most important release conditions.

So, those are just the four points that I wanted to mention here, Frank.

Frank Bruno: Hey (Max), that's good advice. Thank you very much for that.

Finally folks, a quick slide on Iron Mountain. And just know that technology escrow with verification is only one piece of the total compliance solution. Iron Mountain reduces the cost, complexity and risk associated with maintaining compliance. And I want to thank you guys for joining us today.

We do have an offer and many of the tips and strategies that we discussed today can be – can be downloaded off of our web site. We refer to it as the customer resource kit. And if you look to the right of your screen, just below where it says “F. Bruno Bio,” is number four – “Iron Mountain.” If you click on that link, it will take you to the – to the web page that'll enable you to download that information.

(Chris Ekron): So thank you very much, Frank and (Max). I think we have about seven-and-a-half minutes to deal with a couple of questions.

First a couple of comments on my part; being also someone who's very involved in the process of negotiating these agreements. What I always tell my clients internally is it's absolutely worthless to have an escrow agreement if you don't have people that actually can

work on the code once you have it released to you. So that obviously means you have to make sure that there's someone who can actually evaluate things. And then, you have the right to hire contractors or ex-employees of the company that has gone out of business or is no longer able to support your software.

And I guess a question for either Frank or (Max) is where do you often see the clarification that there is no restriction on the ability to hire ex-employees or even employees if there is a release condition? Do you see that in an escrow agreement or do you see that in some other ancillary agreement? Or is that very common?

Frank Bruno: Well, from the escrow agent's perspective, we don't see a lot of documentation to that effect in the escrow agreement itself. So I'll defer that question to (Max).

(Max Fiori): I would say that we – in our – in our software license form agreement and also in our (ASP) form agreement that we use with vendors, we do have a software source code escrow section where we do mention that. And that contains in it a reference to a separate – to the master agreement that we have with the escrow agent. So, if we do not address it in the escrow agreement, we do address it in the underlying agreement for the technology; whether it's the (ASP) agreement or the software license.

(Chris Ekron): But the point is there is some affirmative language that specifically says you will have the right to hire ex-employees or even employees if necessary.

(Max Fiori): We do have that, yes.

(Chris Ekron): OK.

Some of the questions we had from the field – and I’m afraid we probably won’t be able to get to all of them. One question was in terms of the license grant that exists in the documentation, should that be couched in the present or the future?

Frank Bruno: The license documentation ((inaudible)) ...

(Chris Ekron): Right. In other words, you want to have access to the source code. Technically, you probably don’t – you don’t need a license or don’t have a license to it immediately because you just need – you need the object code. But at the point you need the source code, the question is should you have a springing license or should it just be in the license ((inaudible)) (those in existence) it just has release conditions?

Frank Bruno: We do have a license to the source code and it’s written – I don’t have it in front of me but I think it’s something to the effect, “In the event a release condition occurs, then licensee is granted a so on and so forth type license to the source code.”

(Chris Ekron): Right. Have you seen any complexities associated with bankruptcy law related to that? That when you, for example, have bankruptcy triggering something therefore it becomes something that is not as enforceable, for example.

(Max Fiori): I have not. No.

(Chris Ekron): OK.

(Frank Bruno): We've never – we've never been stopped in a bankruptcy case before from a release ((inaudible)) ...

(Chris Ekron): Yes.

We have a question on – several questions on costs of verification; maybe a sense of what people can expect to pay there. Maybe there may be two levels of that; maybe the internal cost – which I suppose would be subjective; the second issue is what you can sort of expect from a vendor participation standpoint.

Frank Bruno: I can take that question, (Max).

You know, essentially, the cost associated with verifying code – if you're looking for ((inaudible)) figures to work with – in any – in any three ((inaudible)) arrangement, if you don't already have a master agreement in place, you may budget somewhere in the neighborhood of \$5,000. And then, depending on the complexity of the application, the verification may run you anywhere from an additional five to \$15,000 on average. So, if you're – if you're working a program budget and you need to get approval, I would – I would certainly benchmark somewhere in the neighborhood of 15 to \$20,000 to get the verification done. And that's a pretty safe figure to come in at.

(Chris Ekron): Question about expedited dispute resolution. And I understand that in your forms kit you probably will have some various alternatives that you've seen people use. But is

there what you would call a common approach that people can refer to when they're negotiating?

Frank Bruno: In terms of our standard escrow agreements, we do have, you know, language to the effect that, you know, the matter would be submitted to arbitration. But many contract parties do elect to avoid going to arbitration and just take it straight to the courts.

(Max), what does JP Morgan Chase do in this case?

(Max Fiori): With regard to dispute resolution?

(Chris Ekron): Yes.

(Max Fiori): Generally, it really goes down to the fact that we will reach out to Iron Mountain, letting them know that a release condition has occurred and for Iron Mountain to release the software to us. I know in the agreement there is an escalation provision for dispute; that it will – that it will escalate throughout the levels of the organization.

(Chris Ekron): OK, two more questions. One is regarding ala carte verification. Assuming you – do you see it – do you see the case where the parties agree that if there is going to be a verification, that this is the process and this is the cost but there's absolutely no requirement; it just gets triggered by the requester? Is that something you often see?

Frank Bruno: Yes, actually that is the more common occurrence; is the parties will agree that verification testing can take place at any time during the term of the agreement. And then,

of course, typically what's specified is who's paying for the verification. So, if you are a licensee and you want to try and share in the cost, one justification for that would be this, if I'm a developer and I want to increase credibility amongst my prospect base and I have my code in escrow with a reputable agent and then I've taken the steps to have my code verified, you know, might that put me on a higher playing level than some of my competitors? And the answer is yes. So there is justification or benefit to the developer to have their code verified.

(Chris Ekron): OK. Really, really quick question in terms of (ASP) arrangements. We've had several questions there. The first question is – I guess for some people is – is it really possible to get (ASPs) through (greeter) escrows? Then, how many of them generally do include escrows percentage wise? And then, I guess a final question is – I think that – I think that can end it. You know, how often do you actually see (ASP) agreements having escrows associated with them?

Frank Bruno: Well, it's very common. It's hard to measure, you know, the percentage rate of (ASPs) going into escrow. But I will tell you that the trend has been growing and what we've been advising our client to do in that – in that case is to, you know, not only get the source code to put in escrow but to have access to the object code as well as a description of the hardware environment and how you would get access to your data. And then, of course, if you're talking in terms of disaster recovery, you know, what is the recovery point objective for the data?

So, you know, if you're doing – if you're doing an (ASP) escrow, it's very, very important to consider those things as something you will need as soon as possible. And if you have a

disaster on your hands, you know, you need to be able to submit a demand release request to get access to that – to that stuff as soon as possible so that you can recreate your production environment and to continue to serve up the application to your end users. Without it, of course, you've got a disaster on your hands.

Now, as it relates to the source code itself, you know, we typically see those going into a traditional escrow agreement. Will it help the user in the event that the (ASP) failed? Well, not really; because it's going to take some time to recreate the application development environment.

(Max Fiori): Frank, if I could just add 30 seconds of commentary to that. We do require (ASPs) to put their source code into escrow for us. I think a few years ago it was – over the past few years, it's been more of a growing trend. But it certainly is a requirement from our perspective.

(Chris Ekron): So, on behalf of ACC and all the attendees, I'd like to thank Frank and (Max); in particular, Iron Mountain for their support of this programming.

Just again, a couple of reminders. There is a survey evaluation. It's available in the links box. Please let us know. I understand there were some audio quality problems; I'm sure we'll be working on improving those. But any other suggestions ...

Also, I think questions can be directed to the people with the contact information provided. And I know we did not get to the majority of the questions but I'm sure there's availability to answer those questions.

So, again, with – again, the reminder that we have the evaluation, I would like to thank everybody for their participation. This concludes the call.

Jacqueline Windley: This is Jacqueline Windley with ACCA. For those persons who requested slides for the presentation, the slides will be available at the Webcast link within an hour after the end of the presentation. Or you can email a request for the slides to me at (Windley@ACCA.com). That's W-i-n-d-l-e-y@ACCA.com.

(Chris Ekron): Thank you, everybody.

Male: Thank you.

END