

ASSOCIATION OF CORPORATE COUNSEL

TITLE: Attorney Best Practices for Minimizing Risk for Technology Licensees & Licensors

DATE: May 27th, 2008

PRESENTED BY: ACC's Law Department Management Committee

SPONSORED BY: The International Association for Contract & Commercial Management (IACCM) and Iron Mountain.

FACULTY: John Boruvka, Vice President, IPM, Iron Mountain Digital
Tim Cummins, President & Executive Director, IACCM

MODERATOR: Jason Anderman, Corporate Counsel, Becton, Dickinson & Company & Vice Chair of ACC's Law Department Management Committee

Operator: Welcome to this ACC Webcast.

Jason, please go ahead.

Jason Anderman: All right. Thank you for everybody joining today. Today's presentation is Attorney Best Practices for Minimizing Risk for Technology, Licensees and Licensors. This is being presented today by the ACC's Law Department Management Committee and kindly sponsored by our sponsor for this, Iron Mountain.

Next slide, please.

The presenters today include myself, Jason Anderman. I'm Vice Chair of the ACC's Law Department Management Committee and In-house Counsel at Becton, Dickinson and Company. We also have Tim Cummins, who is President and Executive Director of the International Association for Contract and Commercial Management; and John Boruvka, who is Vice President for IPM at Iron Mountain Digital.

Next slide, please.

Today's presentation will be in two parts. The first part is going to go over technology escrow basics, and the second part will be on advanced technology escrow. In between the two parts we're going to ask for questions. We'll also ask for questions at the end.

If you look to the lower left-hand corner of your window next to the presentation, you should see a chat box there. Just type in any question that you might have and we will, of course, try and get to as many questions as we possibly can.

We will remind you at the end, but just to highlight it for you now as well, if you could also please, at the end of the presentation, click the web cast evaluation link and submit your thoughts on today's presentation, that would be tremendously useful, as well.

Next slide, please.

So what are the questions that really come up when it comes down to technology escrow basics? Well, the first thing is, why is today's information society exposed to risk and how do we get a better understanding of those risks? What is technology escrow and how does it ameliorate that risk? Why do licensees, namely, people who are in the buyer-user customer-type role, need technology escrow? And why does the licensor, usually people in the vendor or developer role, need it as well?

And Tim Cummins from the International Association of Contract and Commercial Management will speak now to this.

Tim Cummins: Thank you, Jason, and it's great to be on the program today for ((inaudible)).

We're under onto the right chart, I believe.

So why is it that we're facing this environment of greater risk and we will want, of course, to talk about how escrow plays a role in managing of it.

The network world, I guess, lies at the core of the conversation today. We are, all of us I think, to varying degrees, dealing with new markets, whether as buyers or sellers. The drive obviously is to expand these new markets, either to bring new business opportunity or alternatively to acquire new innovative products and services or new services sources of supplies to drive down costs. These are, of course, dramatic shifts for virtually all significant corporations in the world.

And what that takes us to is, of course, a world where we are frequently encountering different standards of practice and of different values, and of course, very different modes and abilities around enforcement.

We're also gaining in a world where, as we move into our information age, we are, I think, are becoming increasingly uncertain of exactly what is an asset. If we think about the growth of things like confidentiality and non-disclosure agreements over recent years, that's perhaps a great illustration of this point.

We've moved from a world where most business was really about the management of fixed assets to a world where, in fact, increasingly our assets are the relationships we build and the information and knowledge we exchange. Disruption in the way the business and the legal community particularly needs to look at the asset base and its ability to protect it.

We're also seeing a world where, of course, we're seeing the growth of dramatic innovation from small companies. The sources of innovation are changing significantly. Perhaps they have always tended to come from small companies and we just haven't had such visibility. But the truth is that now many of the innovative ideas and practices that are coming forward, and in particular in areas like software, but also in the more innovative services, are often coming out of small 20, 30, 40-person companies with no substantial asset base. And big questions are obvious of sustainable development, our ability for recourse, and of course the potential for them to be acquired by other bigger players, perhaps challenging the security of supply that we have otherwise achieved.

And on the line of all of this, as the final comment says, trust is becoming something that's at a premium. As we venture beyond our traditional relationships, beyond those that have been tried and trusted, beyond those that are governed by our domestic legal systems, we really do need to reconsider how we deal with trust and also how we deal in an environment where perhaps there is not that same imbedded level of trust.

So ((inaudible)) of all of this, as Eric Smith at Google recently pointed out, there are no global laws. And so the law itself is really not adequate to fix a lot of these risk issues. We need to

utilize a growing range of mechanisms to protect our assets and to assure that our business operations are protected. That's where, certainly, our ((inaudible)) plays in creating and assisting in the ((inaudible)) work for trust uncertainty and is very much where escrow plays as part of an asset and capability protection strategy.

So, John, with that introduction, let's pass over to you and get into the detail and use escrow as part of that strategy.

John Boruvka: Tim and Jason, thank you very much, and thank you to everybody out there that's joined the webcast.

So Tim makes an excellent point about the fact that trust is at a premium. And the basic concept of escrow has been around for many, many years to the legal community and financial securities. The constant of technology escrow is a little younger. It's only been around since the early '80s. And it's a constant whereby a neutral third party, typically a trusted third party, retains proprietary information of another party in some sort of a licensing transaction or technology transfer agreement.

And the idea is that at some point in the future if certain conditions are met, that the information that the escrow agent is holding can then be transferred and made available to that buyer or licensee.

The most common place where technology escrow is used today is in software transactions. And in software transactions that describe, if you will, software that is being internally used by a corporation, like a payroll system, a human resources system, or some financial or analytical soft, or technology that's being imbedded in a company's product.

And later today we're going to talk about a couple different scenarios when we get into the advance section on escrow.

But escrow, essentially, is a safety net. It's a form of insurance, if you will, where the payout is not financial or money but the payout is access and restricted use to the know-how and information that's being placed in escrow.

Often, we ask a lot because we refer to as a prenuptial agreement between a software licensee and the software buyer for mission-critical technology investments. And in any escrow transaction there's normally going to be, at a minimum, three parties. There's the developer or licensor or the owner of the information. There is the licensee or the buyer or the beneficiary. And then there's the escrow agent.

I'll comment on something that Tim said is that markets are starting to diversify and the assets are becoming much more diverse. And as an escrow agent, about 80 percent of what we see is software driven type escrows, meaning that it's software being held in escrow. But we have also seen escrows set up to manage and protect ideas, formulas, firmware, ship designs, robotic designs, and things like that.

So it's not unique to just the software market, but really the proprietary technology that your company may be licensing out or licensing in to your organization.

Now, when you come back to the basic issue of the technology escrow service and the realization that the core of that 80 percent is in the software market – why is this so critical?

Well, software as it's delivered to us today on the desktops and on the servers and mainframes that are out there that run our businesses are typically delivered in something called object code. So it's an executed compiled version. For a layman like myself it's ones and zeros that your computer understands what to do in order render the program in its functionality.

Source code is something that's on the back end, which is the English version of what the programmer uses to code and create the product. And that essentially is the secret sauce, if you will, or the know-how on how the product works, how it's designed, how it's put together.

So in order for somebody to be able to maintain, support, modify or change or continue to use a product – a software product – you have to have access to this source code. And this source code, when compiled, turns into the shrink wrap disks that we normally see when we go to the corner computer store to buy some software.

And so access to that development information and that underlying source code which is where the intellectual property and the asset sits is crucial because without that the user or the beneficiary would not be able to understand how the product works and make use of it.

And that's a big distinction that separates technology escrows from financial escrows or real estate escrows or things like that, is that the technology is constantly evolving and constantly improving as the product changes and, therefore, what's held in escrow needs to move in that course as well.

Now let's talk a little bit about how do we identify technology risks?

So we've defined escrow. We understand why source code and maintenance material is important to this process. The question that we always get by first-time escrow users is when do I use escrow? How do I use it, when do I know what application to place into escrow if I am either, again, a software vendor licensing that product out or if I'm a corporation that's bringing that software in-house?

And we like to think about it in terms of four buckets. Your operational dependencies, the cost associated with the technology, the investment of time and your vendor assessment, so your provider assessment if you're the beneficiary.

Twenty years ago the decisions around escrow were primarily made around the second bullet under the license fee, which was – under cost – excuse me, which was the license fee. And by that I mean that companies simply said, "If I'm paying a lot of money for this software application, say \$50,000, \$100,000, \$200,000, then maybe we should consider placing it into escrow."

Today what we find is that risk management departments inside corporations work with their legal departments, the users and their IT folks to better look at things like how many users are using this product. Is the product customer-facing? Does it have an impact to my customers? Is there revenue tied to this? What are the costs to bring the product in, customize it, hardware, software training, all of that? And then how long is it going to take me to identify a replacement product? Are there available substitute products in the marketplace? What's it going to take me to renegotiate this application or this technology if I have to go somewhere else?

And then what's the stability of my vendor. Are they a start-up, as Tim pointed out, that are coming to market with a lot of innovation? Or are they a large, trusted company? Or are they somewhere in the middle and may be a possible target of an acquisition?

And it's a combination of these areas that will allow you as a beneficiary if you're bringing the technology in to say, "Hey, I have a risk factor that requires me to consider using escrow as a way to mitigate the risk that I may perceive I have with that vendor."

If you're a software provider or a vendor on the call today, this will help you think about are customers going to ask me about escrow and should I be proactive in trying to establish that escrow so that when we come to the negotiation table we don't spend time negotiating small

details like the escrow but we can spend time negotiating the bigger pieces around the license and the use of the technology.

Next I want to switch gears here is talk a little bit about the beneficiary's pain overview, or if you think of it this way – this is the concern that the buyer or the beneficiary to the escrow agent is going to have. And it's really an expansion on Tim's information society risk exposure slide because what we see is corporations saying, "Am I bringing in new technology that may not be around for a few years and do I have risk in that?"

"Am I looking at small vendors where I'm concerned that they may not be in business or they may get acquired and the direction of the product may be different than the one that we want to take the product as a corporation several years from now?"

"Are we acquiring intellectual property or buying and then acquiring where we may want to have this in place? Or am I working with a large vendor that is customizing the product and tailoring it substantially where the meaningful changes are all in that customization and I want to make sure I have access to that plus the core product."

So these things are typically the scenarios where we see the licensee or the large beneficiary saying we need to consider escrow as a way to mitigate the risk, along with proficient contracting and good vendor selection and evaluation.

Finally, if I try to bring it together and speak to the issue between why escrow is necessary from a developer's standpoint or a beneficiary's standpoint, I'll think of it in the two columns.

One is that developers, especially if they're smaller, they will use technology escrow to establish credibility in the marketplace. And by this I mean they're trying to get out there to level the playing field against the bigger guys. So they have better technology, a better response, a better design, a better commitment to the corporation but the corporation may be concerned about contracting with somebody who's small who may not be around.

And a way to appease that concern or mitigate that risk is for the software vendor to say, "Look, we believe so much in the product and in the technology and in our ability to support you that we're going to establish an escrow and make that available to you under certain conditions like us going bankrupt or us not supporting the product or perhaps even being acquired to appease and alleviate those concerns."

Ultimately developers want to do an escrow because the alternative is either to walk away from the deal or to provide intellectual property rights to the customer. And ultimately they don't want to do that either because that is the source of their recurring revenue and maintenance revenues.

So at the end of the day there's not only an IP protection strategy but a practical strategy that essentially helps the developer reduce their workload.

From a beneficiary standpoint it's whoever ((inaudible)) the license has been signed. And by that I mean that once you negotiate the agreement you put it into the place, the technology is delivered, the software developer moves on to the next transaction and your dependence, month over month, year over year, is probably going to increase in that software and that technology. So a well-constructed escrow agreement is going to allow you to keep your issues and your customer in front of that software developer to make sure that they get what they need.

And ultimately, if all else fails, then of course, it's access to that current set of source code and maintenance materials. And that's really all about having an option to control the future that avoid having to go through the court system to litigate against what potentially may be a defunct company with no assets.

Tim, I know you had some comments that you'd like to share on this slide.

Tim Cummins: Well, I do. I think the major things, really as far as I'm concerned, John, or as you were saying is that I feel that many of the developers particularly who have been very reluctant to really understand the benefit that escrow can bring in creating this environment of trust. Rather than actually anticipate the real concerns that many buyers are going to have they sort of force this into a case-by-case discussion, which obviously is not only inefficient but really does undermine that ability to demonstrate that they are a trustworthy provider.

So I think taking a proactive position as a provider in addressing the understandable concerns of your supply community is a very important proactive step. And it's also one that the lawyers can certainly lead in terms of really trying to eliminate and manage workload more effectively.

John Boruvka: Great. Thank you ((inaudible)). I'm going to turn it over to Jason now.

Jason Anderman: Great. So that kind of wraps up part one on the technology escrow basics. I think at this time what we would probably like to touch base on a bit is give everybody an opportunity to ask some questions.

I am looking at the chat box and we have a couple of questions up right now. The first one is "Which party should pay for the escrow agent?"

That question is kind of a nice transition from Tim's point just now about many developers probably don't realize that escrow really is something that can help with the sale. I've been on both sides. Before I was in-house counsel I represented Benders where we negotiated escrow issues. Now that I'm in-house counsel I'm usually on the user side where we're asking for escrow.

And I think pretty much universally, Tim's point is well taken because most of the time the escrow is not offered from the developer's side. If it is offered, generally the demand is that the customer pays for it. And I've negotiated that back to splitting the fee or asking the developer to pay for it. And on a case-by-case basis, depending on who has the most leverage and how badly someone wants the software versus how badly someone wants to make the sale that gets worked out.

But that's pretty much the experience that I've had with that.

Tim, John, how have you seen things go with paying for the escrow?

John Boruvka: Jason, this is John. Our experience is obviously we're in a lucky place because we get to see both sides of the coin, much like you have. And this is something that moves a little bit in one direction and a little bit in the other direction every year when we review – when we review this statistically. But what we find is that often the fees are pretty closely paid half and half between the software developer or the licensee.

Now if you look closely I'll argue that more often than not the licensee is ultimately paying for the escrow. And sometimes it's for the right reasons; other times it's for the wrong reasons. And let me explain myself.

I think that if you come back to my first statement when we defined technology escrow I referred to it as an insurance policy. We would always tell the corporation a large user who is bringing technology in and making the demand for escrow that if this is insurance then you should negotiate the contract. You should – and the terms of that contract. You should select your insurance carrier, if you will. And you should negotiate and pay for the premium because all those things are going to put you in the driver's seat in terms of getting the terms and conditions that you want.

But we have a fair amount of ...

Jason Anderman: Well, we've got a lot of questions here. So maybe we could wrap this up and move to the next question that would be great. Sorry about that.

John Boruvka: No problem. So the point is that at the end of the day, I think it's 50/50 between who pays for it.

Jason Anderman: Great. So the next question is, "What should a source code escrow clause look like?"

And I guess the short answer to that is the way they tend to look like is all over the map. Sadly, like most legal provisions, there's not a lot of unanimity as to what would be best-in-class for this.

I have very strong opinions on this and I really feel like it should be set up to be as easy as possible and the least likely that there's going to be a dispute when you need this. Because when you need this normally you're in a fairly disputed situation because you are not getting the maintenance that you need or the company – the developer has gone bankrupt or any of the other normal release conditions where you're really going to need the source code.

Also, because, hopefully you've done this for your mission-critical applications that keep you in business as well as any other application that would cause you tremendous pain to go out and buy a substitute and implement it. You're going to be under a lot of pressure to get this done quickly. So you want something very short and very clear.

And the best way of going about that, in my opinion, is pick a vendor. For instance, if you're going to pick Iron Mountain and make John Boruvka very happy – pick them and put them in the clause and say that this is the vendor that we're going to use.

Every major reputable vendor already has an agreement written. They have several agreements depending on the particular type of escrow service you're buying. And the agreement spells out crystal clear what the developer's responsibilities are, what the user's responsibilities are and what the escrow vendor's responsibilities are, and what the release conditions would be to get it.

Usually those release conditions involve the developer going bankrupt, materially defaulting on their maintenance responsibilities, breaching the agreement, things of that nature.

And also a good escrow agreement from your escrow vendor will spell out what's the evidence you have to present as a user in order to demonstrate the release condition has been met and if there's a dispute between the developer and the user what's the timing on how that dispute gets handled.

And all those things are spelled out in the escrow agreement. So reference the vendor, reference the escrow agreement, sign the escrow agreement with your vendor and with your user or your developer, depending on who you are, at the same time you get the agreement done or right thereafter. And then everything should be extremely smooth and easy.

Tim, John, do you have any thoughts on that?

Tim Cummins: I would only add that I know that certainly we've made extensive use of a lot of the guidance materials actually provided by (R. Manten). So certainly from the association's point of view I think there are some tremendous insights clearly from the range of experience that they have.

Jason Anderman: Great.

John Boruvka: You hit it on the head.

Jason Anderman: Great. Wonderful.

The next question is, "While many include escrows in practice for key license agreements it seems rare that an escrow is activated successfully when needed. Please comment."

Well, I would say that's kind of a telling point about everything you do with a contract. Right? It's rare that you have a dispute. It's rare that the dispute is something that you can't settle with a phone call and you actually have to take action. It's rare that you actually get to the point where you're mediating it or arbitrating it. It's rare that you get where you get to the point where someone sues each other. And it's extremely rare that you even go to trial.

So what you're essentially doing with contracts like this is you're not just writing for the rare situation, for the rainy day. I believe that you write these contracts and write all your contract provisions, including escrow, to head off disputes before they can happen.

And as John and Tim keep saying, create a basis for trust.

If the agreement spells out exactly what happens if things go wrong and effectively protects both companies, what it does, in my experience, is really motivates everybody to be above-board before you can get into a really heavily disputed situation.

If the question is this is never going to happen ...

Operator: Due to normal maintenance or technical difficulties, the conference will be terminated immediately. Please dial back after 15 minutes to continue your conference. We apologize for this.

Jason Anderman: I'm sorry. Is everybody still on?

Tim Cummins: Well, yes. I'm still here.

Operator: Yes, go right ahead. Go right ahead.

Jason Anderman: That was a weird message.

What I generally have found, though, is I've had this happen. And I've been able – and we had a clear clause, we had a clear agreement. We had it signed, pulled it out, read the release conditions, read the evidence we presented and we got ourselves out of escrow. And it was a life-saver.

Tim, John, any thoughts on this?

Tim Cummins: I think, Jason, the key point here is actually the point of why you're using a trusted intermediary? It's because you don't actually want to have a hassle of trying to keep, maintain and monitor all of these relationships yourself. There are a lot of them and no legal department is going to do it. And to be honest, no sourcing department is going to do it.

Jason Anderman: Right.

Tim Cummins: So you bring in a trusted intermediary who is going to provide that purpose for you and give you reliability. And they're going to give you early warning and alerts. So what you're really doing is looking more at what's the quality of the trusted intermediary you bring in so you only have to make one judgment and not multiple judgments.

Jason Anderman: Right.

((inaudible))

Jason Anderman: I'm sorry, John. Go ahead.

John Boruvka: Jason, the only thing I would add to that is the point about escrow providing leverage after the license is signed. And you made some excellent points. And the best escrow is the one you never need to use or the one that you threaten to use and the software developer resolves your problem before it gets to the point of releasing it.

Jason Anderman: Right. Absolutely.

The next question is, "To what extent does the resource capacity to use the source code, a.k.a. the buyer's technical capabilities, bear on the benefit to the buyer of having an escrow?"

This is a really outstanding question. And given the time we're at I apologize to the people that asked the additional questions. We're going to have to move on. We'll try and get back to you at the end after this.

But usually most companies that are going to be getting software, even if you have a really big IT department – we have a massive IT department. We have over 900 employees in our IT department worldwide. There's a lot of software that we license that none of those 900 people would be able to do a thing with if they got the source code.

So what's the solution? Well, when you are negotiating your software license or license development agreement you should always try – and I know you can't always get this concession – but you should always try and get a concession and a reasonable vendor should really give this where if a release condition happens you are allowed to hire a third party company to be your subcontractor and use the source code under strict confidentiality for your business purposes only.

And that's a clause that I kind of fight tooth and nail for in all my agreements.

It is very rare I don't get that. Just about every vendor that I'll deal with will give that to me, except for the really, really huge ones that have absolute dominant market share. And I think we know who those are, without naming names.

John, Tim, any thoughts before we go to part two?

John Boruvka: I think I'm good.

Jason Anderman: Good. All right. We'll try and get back to the additional questions after part two if we have time. Again, my apologies.

So if we can go to the first slide after the part two title page, John?

John Boruvka: Yes.

Jason Anderman: OK. So in part two we're going to get into more advanced issues for technology escrow that you may find yourself struggling with or dealing with.

What are these key issues that you can address to minimize SAAS applications becoming very popular – software as a service where the software commonly is residing on a web site that's on a server owned by the vendor and it's not sitting on your internet, on your servers? It's outside your firewall.

What are the challenges or risks surrounding off-shore development and licensing of software?

What are the challenges or risks surrounding exclusive supply agreements? I think this is one of the more ignored areas for escrow and a tremendous opportunity that I think can really help.

And as John pointed out earlier, 80 percent of his stuff is software. So Iron Mountain and all over escrow vendors would love to see growth in this area and it's something really to think about.

We'll get more into it later.

And finally, "Why and when should I recommend verification of the content of an escrow deposit?" Verification is an extremely important issue.

So I think we can move to you here, John, on changing license models.

John Boruvka: Great. Thank you, Jason.

So, Jason actually gave us a great quick description of SAAS, or software as a service, and we're seeing a lot of changing licensing models there from what has traditionally been called the straight licensing to on-premise now which is most of the licensing that we're used to, to SAAS, and now also in the last couple of years to open-source license.

But today we're going to specifically talk about the challenge around SAAS. And it's like Jason pointed out. In a SAAS model the application and often the data is sitting with the SAAS provider and the corporation using this technology is only accessing it thorough the Internet and through a web browser.

So in this case you don't have – a couple things are going on. One, you don't have access to the source code, so you don't have – and you don't have that under the on-premise situation. You don't have access to the object code because that object code or the run time version of the product is sitting on a server that you're reaching through the Internet. And then your data is likely not being stored in-house on your premises. And that data might be with a ((inaudible)) location provider or it might be with SAAS provider themselves.

And so the challenge there is really now two-fold. It is if this SAAS provider disappears you as a subscriber have the same issues we had in escrow basics, which is, you don't have access to the source code and therefore you can't recreate that. But you have two other problems. You don't have the data and you don't have that object code or that run time version.

So even if you wanted to continue to run the product as it stood before the software developer closed its doors, you wouldn't be able to do that.

So there's an implication for use rights and for disaster recovery and business continuity.

Iron Mountain, we have actually set up a couple of (weather arms) with Tim's group that spend a whole hour talking about that. But there's a lot of actionable guidance both for the provider and the subscriber on our web site in content that we would be happy to make available to folks.

But the key thing to remember here is I don't have source code, I don't have object code, and I don't have data. And the big thing I'll tell you is if the software provider, the developer, tells you that they back up their data even with somebody like Iron Mountain who does data back-up of both tape and electronic you have a concern that you may not, as an individual subscriber, have access to that data because that data has co-mingled all the data of all the other subscribers and customers of that sized vendor.

So there are some issues there that need to be resolved and watched for when you enter into these transactions.

A perfect example of one of our customers that used this to help them mitigate this risk is a company – a group called the Common Application. And they offer – or they have an on-line non-profit site that lets students apply to colleges around the world. And they were using a third party technology partner to deliver that application.

They didn't have access to the data. They didn't have access to the source code or the object code. So they ended up establishing an escrow agreement with Iron Mountain where we captured the source code into one account.

We physically captured the object code into a second account with different triggers and release. And then we have a mirroring of the data into a data center where now the common app can go and get access to that data in the event that the primary copy for the provider were to be destroyed or disappear.

And it's only their data and not every customer's data that uses this application.

Now in this case, it's probably a custom and tailored solution. And so it's easier to separate the data than when you have the co-mingling of data in applications that are more widely used across a variety of subscribers.

And so that's some thought around the risks and challenges around SAAS. And I'll now bring it over to Tim to speak about the off-shore out-source piece.

Tim Cummins: Well, thank you, John.

And of course in many ways an extension of this whole software as service argument – again it's back to this issue of the dramatic transitions being created by a networked economy. And that is forcing many organizations, obviously, to look at off-shoring and out-sourcing, interestingly not just as economic logic anymore but – for example, I had a very interesting conversation with one of the major defense agencies just last week who was talking about how they in fact started moving more and more stuff into software as a service, but also into other out-sourced environments because by utilizing specialists they have recognized that not only are they often less vulnerable to attacks of various sorts but they are also a last likely target.

So in fact it's probably their risk strategy as well as their economic strategy they have decided these are directions they go in.

So I think we can pretty safely say that – to the extent that anybody after the things like the recent finance industry about to ((inaudible)) can safely predict anything.

I think we can fairly safely say that out-sourcing and off-shoring are really becoming and will remain for the foreseeable future norms for any modern business.

But obviously in the event that we go down that route then we really need to think very carefully, again, about the risk strategies we put around those relationships.

So if we're looking at new organizational models continuing to drive us down this path, obviously the key here is how do we get prepared? As long as we've obviously got to make sure that we are not warriors to getting deals put in place and that we're not causing delays in the business.

So we've really got to be thinking about the challenges and risks that are surrounding the development and licensing.

We've got to consider, obviously, the broad question of ownership rights, often with perhaps joint developments of various sorts where we've got the complexity, of course, of wanting either complete ownership or – the (Spoke) software for example, or other developments.

Of course this goes beyond your software as we've touched on once or twice. This goes right into the field of things like business process, of designs and of methods. Therefore, it's that whole base of intellectual capital that we're increasingly looking at in our off-shore and out-sourced world where most companies are, of course, now undertaking much of their development work. If you look at something like pharmaceuticals where I think some 80 percent of the development work now for new products is being undertaken through off-shore out-sourced facilities.

Now clearly we need then to protect those designs. We need to make sure that we are building methods to recapture those.

But we also have to think about very actively around things like, of course, broader IP ((inaudible)). How do we limit the access to overall methods we – getting back to the trust issue we raised at the beginning. We're dealing with domains where probably trying to enforce things through the traditional legal methods are not going to work

So many more groups are coming up again with creative solutions like, of course, the splitting of work between logical dividers to make sure no one of them has complete access to all of the secrets. You alone are the people who can put the great recipe together to make whatever it is you want to make.

So we're looking again here in this section at many of these more advanced strategies and thoughts, which are about how, do you ensure that you have security? How do you ensure that you've got continued access? How do you protect business continuity?

And those sorts of things also get us into, in the world of outsourcing, things like transition. Many organizations, as you know, have faced one or more failed outsourcing relationships where, perhaps, there's been a need either to bring work back in-house or to transition to another outsourcer.

If you transition, that other outsourcer is of course a competitor to your existing provider.

So again, what are the mechanisms that you may put into place to make sure that that is a seamless transition where you have some level of control and can insure that your business is not disrupted?

These are, I think, the key issues that we get into with off-shoring and outsourcing, John. So I'm sure you'll tell us what you would suggest we should be doing about it.

John Boruvka: Thank you, Tim.

I'm actually going to transition over to Jason who is going to talk to us about the supply arrangement.

Jason Anderman: Sure. So this is a really underused area in escrow and I think really ripe for expansion.

It's very common for a lot of companies to be in an exclusive supply arrangement. Usually it's exclusive for one of two reasons. Either one you're getting a big financial concession to make a particular vendor the exclusive supplier. And usually it's a long-term contract. And they're the only ones supplying this particular raw material.

Two, the vendor has a particular patented or trade secret process for making the critical raw material for you and you've got to do a lot of validation internally to make sure their particular method of making it an ultimate raw material outcome of that process can fit into your product. And it would be tremendous time waste to switch to an alternative.

So let me give you an example. We have a particular product that we sell to our customers. It's one of our flag ship products. In order for that product to work there's an absolutely critical raw material that we buy from a particular vendor. That raw material is chemically derived and also has to be in a certain chemical structure. The vendor has internal know-how that only they have access to on how to make that. We don't know how to do it. And if we didn't have this particular product line would just go down.

And it's not only one of our products used by our customers, it's absolutely critical to what they do day-to-day.

So it would be a real nightmare if this happened.

We are negotiating with this vendor to put a clause in the contract that would say that they are going to put into escrow all the specifications, the manufacturing process, all their know-how, that we would then be able to take and give to a third party manufacturer under strict confidentiality who can then make this critical raw material for us as long as our initial vendor is unable to or is not coming through according to specifications.

So that's kind of the supply agreement situation in a nutshell. It's something rarely thought of and I think it's something that really needs to be expanded and emphasized more in the future.

So that really raised the question, then – and it's the same thing with the source code is how do you know what you know? How do you have in escrow is what you actually need and matches it?

And that really is an issue of verification.

John, you want to talk about that?

John Boruvka: Thanks, Jason.

And absolutely you said it right. You can write the best legal contract and legal escrow agreement with all the right terms and conditions but if the day you get access to the information it's not all there, you're going to have an unpleasant surprise, to say the least.

And so something I think the general community using escrow, escrow agreements, has awakened to – and back to some comments that Tim made at the very, very start of today's session is that technology is becoming complex. We're becoming very dependent on that technology and so if we establish an escrow and we really want not have the assurance that it can be used, we should do some level of validation or verification on the materials.

And you can do that either yourself if you have the expertise as the company. We have seen corporations go on-site to the software developer site to watch the code and the information being captured and built and produced into that physical medium that ends up with the escrow agent. And we've also seen corporations hire third parties or used the escrow agents themselves.

So the verification is simply the process of having somebody go through the materials and make sure that what was delivered is going to be sufficient to do the job if it ever comes out

And the reason that this service gets a lot of attention, or that this point gets a lot of negotiation in the contract, is that from Iron Mountain's perspective we see an alarming amount of these

transactions, these validations or verifications coming up as not being complete. As a matter of fact, (97.4) percent of those deposits that go to analysis we find don't contain the necessary build information.

So what does that mean?

We got the source code. We got the information. But there isn't a clear recipe that's evident in the deposit that would help one of those 900 people that Jason mentioned at his company be able to make some sense of this. And the smaller the corporation, the smaller the developer gets, remember that probably the discipline around source code control and source code management isn't there like with the big guys.

So having some process to validate that is going to be important. And most escrow agents out there in the marketplace as well as third parties will speak to verification in terms of different levels. They'll do a basic level that says, "Hey, read this stuff. Is it not encrypted? Is it virus-free? Can you print out a directory listing? Can I see that there's files that correspond to make files or executable files? Then can I take it and can I compile it? Can I create a run time version of the product and then ultimately can I do some sort of binary comparison or some usability test?"

And that would be the ultimate test where you really have a neutral party build it, create it and then you run some test data through it to make sure it works.

In Jason's example on the exclusive supply arrangements it would be having a third party produce the raw material.

Now it's probably not going to be cost-effective or really possible for somebody independent to do that. And this is where, again coming back to the identification of the risks in the first section, it's so critical to say, "How comfortable do I feel with this vendor?" And "What kind of results am I getting?" If you're doing low-level verifications and everything's messed up then you want to go higher. If you get some good data back and it builds your confidence then maybe you say, "I'm good where I am," and don't take it to the next stop.

((inaudible))

Jason Anderman: John, this is Jason. Sorry to interrupt you. Two things – one just a time check. I have 1:57.

John Boruvka: Yes.

Jason Anderman: And number two, I think this would be a good time to bring up one of the questions that I saw that came up on the screen which I found to be a rough situation. It says, "The escrow agent is refusing to allow us to hire a third party to verify the source code. Instead, the escrow agent says that it will perform its own verification services using policy best practices.

"What are these practices and is it a good or bad thing for me as a software licensee?"

John Boruvka: Well, I would tell you this, Jason – hopefully it's not Iron Mountain in our legal department refusing that. But I would be surprised because all that's in our standard agreement is we want to make sure that the right is there to verify and then if we're hired after the materials come in our only concern is going to be if somebody else is hired to do the work that there's a clear and good chain of custody.

So I think you don't have Iron Mountain in that scenario. And if you do, whoever submitted that question should call me after this session and I'll go look into it. Because for us it's more important that somebody verify it than we verify it.

Now we have experience, we do this frequently, we do it a lot, so it's easier for us to understand what to look for. But we also have good guidance in terms of what makes a good deposit that we make available to anybody that uses an agreement.

Jason Anderman: Great.

And I guess to be a little more concrete with the question, I'm shocked and I really recommend that you double-check what the escrow agreement says. I've never heard of that. And echoing John, every reputable escrow vendor I've ever dealt with actually encourages the customer to verify and hire anybody they want to verify because it prevents a later unhappiness with the escrow vendor that the escrow vendor considers unfair.

John Boruvka: Ultimately when this stuff comes out of escrow we want you to be happy. So it would be unfortunate to take that position. And I think you got somebody who is probably trying to drive the revenue in their direction and not really looking at what's the best interest for the customer.

So, Jason, given the time check – and I was aware of that – you know – (TWE) is just an example that we wanted to share where the customer has a custom code, placed it in escrow. We verified it and were able to use it that much faster and better when it did come out of escrow and it did get released.

And then last the regulatory compliance was simply to say that we talked about risk management but for public companies escrow is also part of a way of helping with compliance regulation around ((inaudible)) (SSIEC) if you're in a financial services sector, to make sure that the escrow applications are really protecting the technology that's doing all that reporting and, as you said, the mission-critical technology that runs the business.

So with that, I'm sort of going to bring it back to you, Jason, and the Q and A. And pretend to close the meeting.

Jason Anderman: OK. I think we're probably going to run a little bit over. Please stay on with us. But for those of you who need to go, I'm going to jump a bit ahead and just mention that you might want to put up the slide on the white papers, John. If you e-mail IPM-Info at IronMountain.com you can get whitepapers from Iron Mountain on technology escrow and verification services.

In addition, as I mentioned at the start, if you could click on the evaluation link and give us your thoughts on today's presentation that would be a tremendous help as well.

Just to go back up to the question and answer, I'm going to try and get through as many of these questions as we can while we have time just because I know if I was asking a question I would very much like it to be answered.

So ...

John Boruvka: Jason, just one quick note.

Jason Anderman: Sure.

John Boruvka: The other thing that we would be happy to do – I'm sure Tim, we could do this is if we don't get to all of them we could put them back on the slide and sort of answer them as an FAQ and publish them back out to everybody as well.

Jason Anderman: Great. Very nice of you.

So I think the next question was, "So using a source code, escrows are analogous to using a letter of credit, in essence?" I think that's a great analogy. Just as you set up a letter of credit to be able to draw down funds under certain conditions when you need them, you're setting up an escrow agreement to be able to draw down knowledge, whether it's in the form of source code or whether it's in the form of a manufacturing capacity, to give two examples we talked about today.

Another question that is on here is – I'm sorry. I'm having a little trouble scrolling down here.

Oh. "What percentage of escrow accounts are ultimately triggered?" I think it's a pretty low number. But, John, you tell me.

John Boruvka: We probably release somewhere in the range of 10 to 15 percent. I'm thinking now the last time we had this stat the most prevalent release condition, believe it or not, is failure to support. Then the second one is ceasing to do business in an ordinary course, which is not a formal bankruptcy filing. And third is the actual Chapter 11 or Chapter 7 bankruptcy file.

Jason Anderman: But do you know off the top of your head how many out of the total of escrow agreements what percentage you actually need to trigger a release condition? About?

John Boruvka: About 40 percent of – 30 percent of the releases are for support.

Jason Anderman: No, not the reason for it. Of the total – say you do 1000 escrow agreements. Out of 1000 how many actually what was escrowed get released? I think that's the question.

John Boruvka: Between 10 to 15 percent.

Jason Anderman: Ten to 15 percent. That's much higher than I would have thought. So that kind of emphasizes the need, right?

John Boruvka: And the interesting thing is that we probably get – which we don't measure – we should but we don't – is we get more requests to release material where just the escalation of the conversation between the parties resolves the problem so we're told to stand down and not deliver anything.

Jason Anderman: Right. So that's a good point. That circles us back to Tim's point which is again creating a basis of trust.

That's interesting.

A lot of people have asked questions about getting the slides. You can access a copy of the slides in the links box on the left side of your screen. Just click on number six labeled "Web slides." That should open a new window and get you the document that you can print.

Another question that I have is what are you seeing in the marketplace with regard to escrow vendor's willingness to accept liability for consequential damages for the escrow vendors gross negligence? That is in the case where the vendor loses control of the escrowed code.

I've never seen that happen. Not even once. And I think it's probably a waste of time to try and go for that.

The amount of the fee is so low in relation to the potential liability when the escrow vendor does the calculus that I think an escrow vendor would walk if that was made a deal-breaker issue every time.

The next question says, "Most software license agreements restrict licensees from maintaining, modifying or creating derivative works of the software. Given the bankruptcy code, section 365 in

the provision, that does not allow the license to be expanded upon bankruptcy to include such rights which would be needed if the license ((inaudible)) actual source code, and given recent case law affirming spring licenses ((inaudible)) upon bankruptcy, how can a licensee convince a licensor that it needs to grow its current license to maintain, to modify, to ((inaudible)) the escrow arrangement?

“Do you have alternative advice for overcoming the (springing) license issue?”

Sadly, I don't. I think as the question assumes you're just in a really bad situation if you haven't done that up front. And really you're kind of left to begging in that kind of situation the vendor to work with you on it. The best thing you can do is do it from a business negotiating standpoint where you say, “If you don't help me out on this then I'm just not going to think about you for future business because I'm just not going to trust you next time.”

In terms of convincing the vendor, I think candor – I generally find candor is the most important thing in these kinds of negotiations. I'd walk them through the exact parade of horrors that we've talked about today and the release conditions and say, “Look, we're just trying to create the basis of trust.”

That's what I do every time. And it is very, very rare, except for absolutely dominant companies with massive market share for someone not to give on this. I almost always get it. So that's what I would recommend for that.

Tim Cummins: Jason, if I can just sort of ...

Jason Anderman: Sure, please.

Tim Cummins: I mean I think this is another one where of course it's very important for you to understand also the jurisdiction in which you're actually looking for satisfaction . . .

Jason Anderman: Yes.

Tim Cummins: ... because clearly you would not have quite those same rules as we just read out. I mean a great question but of course if you were, for example, dealing in Europe you would not have those provisions and you would not have the same restrictive rights for the supplier.

So it may well be – and this is where obviously as a community we need to get better at understanding the relative basics of different jurisdictions because it could well be – in fact we sometimes force it back into a U.S. jurisdiction when it's to our disadvantage.

Jason Anderman: Right. And I think that what you just said is also a perfect point for the next question, which is, “How does the contractual right to a deposit release affect the beneficiary's right to traditional damages for breach of contract? Are these contractual rights mutually exclusive?”

That jurisdictional issue, again, is absolutely critical in answering that question. And, frankly, I don't have a good answer to this question. I think it would very much turn on the forum and governing law involved as to what would happen. And whether a court would deem that you've covered or not and, of course, most importantly, did you actually sustain damages or did access to the escrow limit your damages just to the loss productivity in implementing what you got out of escrow. I think that would be the question that a judge would consider in that situation.

The next question – again I'm sorry for going so fast. I just want to get to everybody.

“Doesn't verification risk large.”

I'm sorry. I'm having a little trouble understanding this. But I think what he's asking is engaging in verification creates the risk of disclosing things that are confidential that you don't want disclosed or a breach of confidentiality.

And certainly I think verification first and foremost you should ask yourself, "Can I verify this internally? Do I have the expertise to be able to verify this myself?" If you're a big huge company like mine where we have 28,000 employees the answer to that is a lot more likely to be yes than if you're a smaller company.

If you are a smaller company and the answer is no, we would have to hire somebody else. Yes. It does pose a confidentiality risk if there's stuff in there about your own data within the escrowed source code, for instance, that you don't want released. Also if you're the vendor if you take it from that perspective, and you're scared, well, I don't think that you're at any more risk than you are every time you give your stuff out.

And I tend to think – and I'm speaking of someone who used to represent vendors all the time – people are a bit too paranoid and afraid of the risk of confidentiality breach or access to their source code. I think that's the risk of being in business. And you do have very strong rights. You have very strong rights under every reputable escrow vendor's escrow agreement protecting you. And if somebody's going to take your source code and engage in that kind of nefarious behavior, you have rights under the law to sue them for it.

I think in many ways things – if the source code gets released to the customer, if your customer is going to be viewed as the enemy and highly likely to be a bad actor I think then you want to make sure you ramp up really looking at them and what they're doing with things. And I think that's the best you can do.

John, Tim, any thoughts?

John Boruvka: I think I'm good. I think we're getting asked to wrap this up so we'll pass any other comments through the questions and the ACC group.

Jason Anderman: OK.

Tim, did you have any quick closing thoughts you wanted to conclude with?

Tim Cummins: Jason, I think really after what we've managed to outline to everybody today is the critical information of our information society the fact that we need to start thinking in new and different ways.

We need to be open to understanding these different mechanisms that could be available to us in the management and handling of risk. Those risks are going to continue to increase but there are great opportunities out there to find good solutions.

And I know that John has given us some great outlines today and certainly through ACCM and ACC resources but also, importantly, through the sorts of materials Iron Mountain is able to offer there is some great and easy insights available to people to be able to get maybe a better understanding of how to use this as part of that overall risk strategy.

Jason Anderman: Great. And if – I'm sorry. I thought I heard somebody. Sorry about that.

If I were to leave everybody with just a couple of things. If you got nothing out of this presentation today, I would say make sure number one, you always consider escrow for a mission-critical application and mission-critical supplies as well as anything else that would be tremendously costly and painful to implement an alternative.

Number two, you always make sure in your agreement that you get the right to hire a third party under confidentiality to use what you take out of escrow and implement it for you because you're probably not going to have that expertise.

Number three, make sure you really think about this for supply agreements. It's something that really gets ignored.

And I guess that's about it for today. So thank you, everybody, for your time. I really appreciate everybody participating.

John Boruvka: Thank you, Jason, and thank you, Tim, and everybody out there.

Tim Cummins: Thank you and bye-bye.

Operator: Speakers, please stand by to let our participants disconnect and to let us end the recording. Thank you.

One moment please, speakers. Thank you.

OK. Speakers, I think we should be good now. I'm sorry for the technical problems that we had today but I think as far as the participants go it went off flawlessly. Everybody good?

John Boruvka: No complaints. I thought it was good. I was scrolling through the questions and there's a ton of them.

Operator: There's so many. Yes there are. And what I'll do is I download all of those questions and I'll send them on to Cherise at ACC. And then she can pass those along to you for any that you might not have been able to get to and then I believe they are posted on the ACC web site.

John Boruvka: Perfect. OK.

Female: Do you have any of the ((inaudible)) stats in terms of the total registered?

Operator: Yes. I can tell you. There were about 140 attendees today, give or take a few.

Female: OK.

Operator: Give or take a few. But it was about 140, maybe even up to 150.

Female: OK. Yes. I think at one point I saw 145. How many was the total registered? I had 249 earlier today, Sherrese. Did that stay the same?

Female: Earlier today it was – let me check. It'll take me a second to pull up that report how many registered. One moment.

John Boruvka: I'm sorry. I was switching ears. 140 ended up at it?

Female: Yes. Well, I know at one point I saw 145 actually.

Female: Yes.

John Boruvka: That's good.

Female: At the very end, I noticed a lot stayed through. We had 94 still on the line even though we were running terribly over.

Female: Right.

John Boruvka: That's pretty good.

Female: Yes. I think it's a great result.

Operator: Let's see here. You had a total of 286 registrants and a total 165 attendees today.

Female: Great.

Operator: Oh, that's a great number. Very good. Very good.

Excellent. Well, it was a pleasure working with you all. Again, I apologize for the technical glitches with sounds. We're working on that issue. But as I said, I think as far as the participants go it went smoothly.

Female: Yes. It certainly did.

Thanks very much, Tim.

Thank you, Jason, John.

Good job as always, everyone. I really appreciate it.

John Boruvka: Great. Thanks, everybody. Awesome job.

And, Tim, thanks a lot. Hopefully, our hotel bill isn't too big.

Tim Cummins: That's right. We'll hope so.

Operator: All right. Take care. You all have a good day.

John Boruvka: Goodbye.

Operator: Bye-bye.

END