Event ID: 323209
What's New in CPLEX Optimization Studio 12.3

Jeremy Bloom: As many of you know, ILOG has been part of IBM since 2009. IBM is a company with a broad portfolio of products and services related to analytics. And we provide through CPLEX Optimization Studio a complete solution for analytical decision support based on optimization, utilizing high-performance, reliable and robust servers backed by the strength and commitment of the company, IBM, with a premier staff of mathematicians and operations research professionals.

IBM has CPLEX completely integrated development environment, which gives you rapid, reliable deployment and development of optimization applications. We have many case studies with documented return on investment for optimization, which provides you with dependable analytical decision support. We have powerful solvers with a history of continuous improvement that give you consistent performance in solving optimization problems.

IBM has a broad vision of advanced analytics as a critical business competency. And you can use the power of advanced analytics and optimization to transform your business. IBM has the world's leading analytics practice and a premier staff of mathematics and operations research professionals. And IBM ILOG Optimization products and services build on IBM's technical strength. And we have a reputation for dependability. IBM is one of the most trusted brands in the world.

Within the ILOG Optimization product line, there are two basic products, CPLEX Optimization Studio, which I am going to talk to -- talk about today, which is our tool for developing and deploying optimization-based decision support applications; and ODM Enterprise, which is our tool for developing and deploying enterprise decision support solutions based on optimization. CPLEX Optimization Studio is part of ODM Enterprise in addition to the development features of CPLEX Optimization Studio, which include our model development tools, our ILOG Concert Technology APIs and our optimization engines for mathematical programming and constraint programming.

In addition to that, ODM Enterprise offers a graphical environment for displaying the results of optimization applications, enabling business managers and business decision-makers to utilize models without having to understand the underlying technology. It provides the ability to run applications at the enterprise level using servers for data and for optimization. And it has ODM Studio, which is the ability to provide a custom interface – [business using our] interface.

I'm going to focus our discussion today on ODM -- I'm sorry, on CPLEX Optimization Studio. I'm particularly going to highlight some of the new features in version 12.3, which became available in June. And so those of you who are interested, you can download a trial version from our website on ibm.com.

Let me briefly summarize what the new features of version 12.3 are, and then I will talk about them in more detail as we walk through the structure of Optimization Studio. First is we now have a connector in Optimization Studio through SPSS Modeler. This is an integrated modeling perspective for both prescriptive and predictive analytics. And you can use SPSS Modeler streams as data sources for optimization problems. And Ferenc will show you a demo of that capability later in this webinars. Secondly, as you all know, we now have an interactive development environment based on Eclipse, and we've now made that available on Linux as well as Windows for Intel-compatible processors.

We have extended our quadratic programming capability to include indefinite quadratic programs, which are non-convex. We have increased the ability to handle very large problems. So you have essentially an unlimited number of non-zero coefficients in your model. We have made performance improvements for both of our optimizers, the CPLEX Optimizer for mathematical program and the CPLEX CP Optimizer for constraint programming. And we have now made the OPL runtime available on the IBM POWER7 architecture. And on many of our problems, we've discovered that POWER7 architecture offers faster performance than Windows.

Let me start out by talking about the integrated development environment. This is the ability of a model developer to interact with the model, to test it and to debug it and to create it using our OPL language, and I'll talk about OPL a bit separately.

So CPLEX Optimization Studio IDE permits you to rapidly develop and deploy optimization models. The CPLEX Optimization Studio IDE permits you to develop, debug, test and tune analytical decision support applications based on mathematical programming and constraint programming. It enables you to access major relational database products and spreadsheets for data sources. It enables you to track progress during the solve process. It has a profiler to help you understand the use of memory and time in different phases of execution. It has the ability to visualize data and solutions in tabular and graphical displays. And it supports a complete process for model development from prototyping through deployment.

This screenshot shows the basic view that a developer has for an optimization project in the CPLEX Optimization Studio integrated development environment. And I'm not going to spend a lot of time on this slide, but you should see the major pieces of it. In the middle -- at the top, we have the problem editor, and this is the place where you actually enter your optimization problem using the OPL language. And I'll discuss OPL in a minute.

On the left, we have the project navigator. So CPLEX Optimization Studio permits you to have multiple projects opened. A project is a set of related models and data. And within the project, you have one configuration. So you select which model and which data to run each time. There are standard debugging tools such as problem browsers, and on the right side the problem outline, which show you all the components of the model that are available. And in the bottom, there is inspection windows that allow you to see the solution and see the process of the solve.

CPLEX Optimization Studio has -- in addition to the developer feature, it has the ability to do graphical displays for reviewing and debugging. And this is especially relevant when you're talking about scheduling solutions and models using the CP Optimizer, the constraint programming optimizer. You can view, for example, the schedule on a Gantt chart or you can use resource utilization, the cumulative charts. So there is very powerful ways to visualize the solutions that you're getting out of the optimization model.

Now since version 12.2, we've bundled all of the CPLEX Optimization components into a single product called CPLEX Optimization Studio. They provide a complete set of services from model development for operations research experts and practitioners. All the development interfaces and all the optimization algorithms are available through the CPLEX Optimization Studio, and we have a single deployment license that covers both mathematical programming and constraint programming.

Some of you who have been long-time ILOG users recall that we used to sell CPLEX Optimizer separately from the OPL Studio. Now, all the components are bundled together into CPLEX Optimization Studio and individual components are no longer available except the CPLEX Optimizer is available for deployment.

And we've removed license key enforcement. You still need a license, but there is no longer an ILM license key. So we now rely on other means to make sure that people are acting within their -- means that they have a license. But this greatly simplifies the process of developing and deployment. So you don't have to worry about whether you have the key installed on a particular machine where you're running the application.

I'd like to talk a little bit more about the modeling capabilities of CPLEX Optimization Studio. These modeling capabilities enable you to represent business problems in a straightforward mathematical language.

I have a question here. Ferenc, I'm going to ask you to address this. Does CPLEX Optimization Studio have an interface – an interactive interface on the MAC OS, or they only would be able to use the CPLEX from the terminal?

Ferenc Katai: No, we don't MAC OS on the Studio side.

Jeremy Bloom: Okay. Let's talk a little bit about OPL. OPL is a language, a modeling language, mathematical programming that provides a natural mathematical description of optimization models. It has high-level syntax for mathematical models that produces substantially simpler and shorter code than general-purpose programming languages such as Java or C++. This reduces the effort and improves reliability of development, of upgrades and of maintenance. And OPL supports all the expressions that are needed to model and solve problems using both mathematical programming and constraint programming.

So in the diagram on the right, you see some of the features that are available in the OPL language, again displayed in the CPLEX Optimization Studio IDE. So you can define parameters. You can define decision variables. You can create decision expressions that are used in the objective function. You can create constraints, again in a natural mathematical language. And you can actually use OPLScript for pre-processing, post-processing and flow control in order to develop more sophisticated data analysis and modeling algorithms.

OPL is a compact language to represent optimization problems. It has advanced data types for data organization. It allows you to define ranges, which are numerical ranges, either in integers or in floating point values. It has the ability to define sets and tuples, which are structures that have multiple components in them, and you can define sets of tuples. And you can use sets, tuples and tuple sets as indices in your mathematical programming model.

You have the ability to support both mathematical and constraint programming models, and I'll show some of the syntax for scheduling in constraint programming in a couple of minutes.

You can represent linear and quadratic objective functions in constraints. You can have real or integer variables. In OPL, reals are called floats as is standard in most programming language.

We have a special syntax that enables you to represent detailed scheduling problems. You can define decision variables which are intervals of time. You can set presence constraints among activities represented by these intervals. You can define intensity and cumulative functions that represent the use of resources during intervals. So you have a live syntax that enables you to develop scheduling problems within OPL. You can connect a relational database into Excel spreadsheets. So you can bring data in from your enterprise sources and make ideas as simple or as complex as needed.

We have -- in addition to the OPL language for describing mathematical programs and constraint programs, we have OPLScript, which is a scripting language for data processing and iterative solving. You can do data pre-processing. So if the data doesn't come in the format you wanted, you need to manipulate it. You can use script for data manipulation. You can do solution post-processing. So you can take the solution and put it in a format that's more suitable for further analysis. And you can provide execution controls for iterative solutions, decomposition and things like that.

A new feature that we've added in version 12.3 is interactive scripting. When you have a long script that's running and you need to make a change, you can save time and avoid rewriting the script by inserting a new code into the script while it's running. This is a great tool for debugging when you have a long script that's running.

This slide demonstrates some of the features that are available in OPL for specifying detailed scheduling problems. So you can define -- as I said, you can define intervals as decision variables. So in this statement that's in the box at the top, we are defining an interval. In fact, we're running a set of intervals over a bunch of jobs. It has a begin and an end, which can be decision variables in the model, and it has a duration.

We can also define cumulative functions to represent the resource use. We can define an objective function, which in this case is a minimum mix function, and we can define scheduling constraints based on the decision -- interval decision variables that we have.

So this specialized syntax enables you to -- very compact. It represents a wide variety of scheduling problems. I should point out that as OPL supports both mathematical programming and constraint programming, you can also do complex hard combinatorial optimization problems using constraint programming.

The next slide demonstrates some of the variety of ways that you can use OPL to access various kinds of data. So on the upper left is the description of the data structures in the model, in this case two tuples representing different types of gas and different types of oil. And then I can connect those tuples. In OPL, we use a concept called model data separation. So you can build a model completely parameterized without any reference to the actual underlying data and then you can populate the model with specific data instances to run individual problems. That's the purpose of what we call run configuration in the OPL IDE. You can create combinations of data.

So for example, if you want to do a small test problem while you're debugging and then in deployment you want to use a very large dataset, you can do the debugging with the small test dataset and then migrate to the large dataset just by changing the data source, no change to the code required.

Now, there are three ways you can provide data to an OPL model. The first one at the top right is text data. You simply create a text file and the text file defines all the data that gets read into the model. And that's good for small problems for debugging. But in most cases, people have enterprise data sources or at least large data sources that are kept in other format. So to access that kind of data, we actually have the ability to read data from spreadsheets or relational databases. And those are illustrated in the other two boxes on this slide.

You create a connection to the data source and then you use a read function to read the data. And in the spreadsheet, you can read it either by the column identifier or the rolling column identifier or by named ranges. In a database, you can actually issue SQL commands to create the data that you want to use. So it's very simple to create models that rely on external data sources in OPL.

Now, in addition to OPL, we have several other ways that you can build optimization models in CPLEX Optimization Studio. First of all, we have a component library for creating these models called the ILOG Concert Technology. These are object-oriented APIs that are available in C++, Java and .NET, and .NET is C#, and then Microsoft Visual Basic. In addition to the object-oriented APIs, we also have a callable library, which many of you are familiar with because it was the original way in which people were able to develop models for CPLEX. The callable library is a matrix-oriented library using the C language.

So you can create models with these component libraries. You can use a complete sparse matrix representation or build the problem up by adding constraints and variables using a natural algebraic notation. You can control the optimization process and the results. You can access the complete solution formulation. You can perform sensitivity analysis. You can diagnose infeasibility. You can customize the optimization process and you can implement your own solution algorithms. And you have access to all of the CPLEX Optimizer messages. You can use callback routines to interrupt the optimization. And you can retrieve status information anytime.

So these APIs give you the ability to modify problems, to query problems, to read and write files and to set and query parameters in the two solvers. The APIs require you to write code in one of the languages, C++ or Java or .NET. And so generally speaking, they provide you with a much more detailed control, but at a cost of much more complexity over using OPL.

Let's talk a little bit about the two optimization engines, starting with the CPLEX Optimizer. We've created continuing improvements in performance. It's the world's most trusted solver. It's fast, robust and reliable. So CPLEX Optimizer obviously is the engine which solves mathematical programming problems. So you can use it to solve problems that have been modeled mathematically, and you can use powerful algorithms to produce precise and logical decisions.

IBM's mathematical programming technology enables analytical decision support for improving efficiency, reducing costs and increasing profitability. We have fundamental algorithms in the CPLEX Optimizers that provide high performance for linear programming, mixed integer programming, quadratic programming and quadratically constrained programs. We have robust algorithms that solve demanding problems. We solve problems routinely with millions of constraints and decision variables. We have the power to solve very large real-world optimization problems as well as the speed that's required for today's interactive analytical decision support applications.

We have a very large installed base with the CPLEX Optimizer and we build on that base to provide a very large benchmarking library, which enables us to improve performance with every release. Every new feature is tested on the biggest, most diverse model library in the world.

And we have flexible interfaces. We have multiple ways to enable you to interact with the Optimizer during the development and deployment of your applications. As I mentioned, we have OPL and CPLEX Optimization Studio IDE. We have the Concert APIs and we have the callable library.

So what have we done with version 12.3? Well, we've increased the ability to solve large mixed integer programming problems by about a factor of 2 on very hard problems. And we define the hard problems as those that require 1,000 seconds or more. So the factor of 2 increase is compared to the previous release, which is our version 12.2.

We have added the ability to solve very large models, more than 2 billion non-zero elements. We now have created the ability to solve indefinite quadratic programs, non-convex of quadratic objective functions. And I want to point out since someone asked that these find a local optimum. We are still working on the ability to solve globally, but right now the indefinite quadratic programming finds a local optimum. And this is particularly useful when you have portfolio optimization problems with messy data. So you can't guarantee positive definiteness of the correlation matrix. Also, you have the ability to solve bilinear problems that involve both price and quantity in the decision variables.

Some of the other enhancements we've made in the CPLEX Optimizer: we now have the ability to write out an incumbent feasible solution. We can support non-English language variables and constraint names. We give you a greater control over callbacks. And we now provide support for 64-bit MATLAB on the MAC OS. It's already available for Windows and Linux. And we have a continuing history of performance enhancements on CPLEX. The latest version, as I said before, version 12.3, shows a 20% speed-up overall with the factor 2 speed-up on problems that require 1,000 seconds or more.

IBM has a continuing commitment to performance leadership in mathematical programming. We have improved the performance to a number of features. We've improved the heuristics. We've got more cuts and better cuts. We've reduced the parallel overhead for solving multi-processor machines. And we've done a lot of work with general software engineering to try and tune the performance of CPLEX Optimizer.

Let's talk a little bit more about the constraint programming technology, which is the CPLEX CP Optimizer. This is the alternative solver engine that we have. And one of the great features when we introduced the CP Optimizer is the ability to use a model-and-run philosophy, the ability to declare a CP model and then solve it automatically. And most dramatically, this is used for in developing complex -- solving complex scheduling problems.

So CP Optimizer handles the complexity of real-world scheduling problems, which are used for scheduling personnel, machines or process steps. Our constraint programming solver is robust and it enables a model-and-run philosophy so that you don't have to define the search algorithm. The search algorithm -- the robust search algorithm already exists. If you need to improve the performance, you can also build your custom search using the C++ API.

You can support business goals in scheduling, such as optimizing earliness and tardiness costs, duration costs and non-execution costs. You can model the work breakdown structure of the schedule, task dependencies as well as multiple production modes. You can model finite capacity resources in reservoirs. You can model setup times to compute schedules that define the best possible sizes for batches. And you can find optimized solutions to combinatorial optimization problems as well as scheduling problems.

What's new in version 12.3 of CPLEX CP Optimizer? We've increased the performance on multi-core parallel machines. By default, the CP Optimizer now uses multiple threads to more quickly solve problems, and the parallel mode has been improved both in terms of speed and robustness.

We've got some generalized objective functions for scheduling. So you can now use semi-convex. We now can -- we've lifted the restriction on the piecewise linear functions being semi-convex. You can now represent very general preference functions for the start and end times of activities. You now have the ability to solve multi-criterion optimization problems. You can have multiple objective functions which the criteria are lexicographically ordered and they can be solved directly within a single model. And the way this works is you have multiple objectives you solve for the first objective function, the optimized first objective function. And then within the set of optimal solutions for the first objective function, you optimize the second and so on. This is what we mean by lexicographic ordering. So for example, you can minimize the number of machines and then minimize the schedule tardiness within the set of machines that you've identified.

You also have the ability to model transition-based scheduling. You can model costs and constraints for task transitions. And as you see in this diagram, you can actually display the results in a Gantt chart.

We've developed an API for search. So when the problem requires an algorithm that's more robust or has better performance than the default search, you can write your own custom search and propagation algorithms for the CP Optimizer. And this slide demonstrates some of the features of the search API.

As I said before, you can display schedules using Gantt charts. And this is based on the ILOG product called JViews, which is a set of components for developing visualizations in Java. And you can tie the Gantt chart directly to the solution variables, the decision variables in your scheduling problem. Or if you properly formulate a mixed integer program, you can also display the results of a mixed integer program on a Gantt chart.

Now, in addition to all the components that are available within the CPLEX Optimization Studio, we also have connections to third-party platforms, and those enable you to retain your familiar modeling interfaces. Some of the connectors that we include, first of all we have the new connector the IBM SPSS Modeler. So you can use modeler streams as OPL data sources and Ferenc's demonstration will show you that in just a minute. We have an add-in to Microsoft Excel that enables you to use the power of the CPLEX Optimizer as an alternative to the default software that's included with Excel. And you can now solve from the spreadsheet interface or alternatively you can invoke a macro to solve models.

We have a toolbox that enables you to access CPLEX Optimizer from MATLAB. So you can access the industry-leading mixed integer and quadratic solvers from a numerical computing environment. It's a blackbuck approach to modeling and solving. You have full API access and CPLEX Optimizer allows for sophisticated solution algorithms to be invoked. And we now, in version 12.3, support 64-bit MATLAB on the MAC OS.

We also have an interface called Python, which is an alternative to using the CPLEX callable library. It bridges the need for both interactivity and programming flexibility. We have the ability to interface with Microsoft Solver Foundation 2.0, and we continue to support access to AMPL for mathematical programming models.

At this point, I am going to play a video of the demonstration we have of SPSS Modeler CPLEX connection. And then we will continue with the presentation.

Ferenc Katai: Hello, everyone. I want to give you a demo on how you can use SPSS Modeler strength in ILOG Studio. To use this connection, this feature, you need CPLEX User Studio SPSS Modeler. You can download this [free]. You need also license for this SPSS Modeler. If it's just the first time you are using the trial license, then that is valid for 90 days. In license renewal, you can go to this page, this link. Then of course you need a stream, what you can use with the Studio.

What is SPSS Modeler? And this is a visual modeling environment. You have -- you are constructing a stream developed by adding and connecting nodes. Every node is kind of a data transformation of entity, simple things like reading data or displaying data, or complex ones like datamining and statistics.

Here is a screen of SPSS Modeler. This stream I am going to use in my demo. This one is coming from -- directly from the SPSS Modeler distribution. The only difference is the time (inaudible) here a node called Filter1. This one is necessary, because I wanted to have only data which is necessary for Studio, the alternatives from other development, not the whole data, (inaudible) data.

So we are going to see campaign optimization based on SPSS Modeler's pm_selflearn stream. Just imagine there is a bank which has four products to start, car loan, savings, mortgage and pension. These are the data the bank can deduce a certain number of segments, customer segments, using the SPSS predictive analytics tools. A segment has different attributes like income, family background, what product has been offered to that segment and what was the response and so on.

Also, from historical data, the bank can predict again based on SPSS predictive analytic tools for each customer segment what are the probabilities for 2 products that the customer would take or buy them. For instance, if you have a segment A and the savings is 13%, it means that segment A would very likely buy savings. He should offer with 13% probability. On the other hand, pension would be purchased by the segment A customer with 86% probability.

Here we have the data, what you could get -- the bank's historical data. You can see first is custom ID that was what kind of obvious data; then a campaign, what's the response; the purchase; what is the age of the youngest child; the average balance feed index; whether the customer belonged to branch and so on.

Actually this one is the raw data. We don't need all this. You will see that what we need is just the customer ID. And at the bottom, we have string as a campaign -- the name of the campaign, float as the probability, whether the customer would buy if he is offered to him or not. And the second one is the same.

Each segment has a certain size. It means that each segment can only store a certain number of people. We have the model [in that part]. It comes into play when we want to compute the cost. Each product has a certain cost when advertised or offered. Lastly, at the moment, we just assume that the product has one cost to be offered or advertised. There might be some cases where you can have more than one channel, like speaking of the phone, mailing paper or e-mailing. And these may have different cost factors. Each product has different value for the bank. Definitely there might be some differences making money on the savings accounts than mortgage, for instance.

What kind of decisions do we have to make? Or this application, this demo behind what kind of application should be made? Definitely, for each segment, we have to decide whether that segment should be offered a certain product or not. The bank would like to sell maximum two products to all customers having the same -- being in the same segment. This one is cost range which you expect to be set. A maximum two products will be offered to one segment.

Then the bank has a certain budget for advertisement. Here it comes the size of the segment, because you have to -- if you decide to offer the products to a certain segment, then the cost will be multiplied by the number of people in those segments.

Then the bank definitely doesn't want to just offer to anyone without any consideration what kind of probability the customer may have to accept a certain offer. The bank would set a certain threshold upon which a customer would be offered the product. And below that, the customer is not approached at all. For instance, if you say that the segment is -- the threshold is 75% and the savings probability is 13%, meanwhile the pension is 86%, then customer in this segment that needs to be offered savings definitely will be not offered the savings and will be offered the pension. And there is also some other constraints.

The goal here is straightforward. It is a demo. Basically the bank would like to maximize the sum of their values, the products' values, what they can sell to the customer. Finally, you can see a screen dump. You can see it in light. The first customer is not for them. You can see the second customer. Both products will be offered. And so on.

Let's take a view, a very quick view of how it looks like in Modeler. You can see these are the three nodes, are the data. You can have different datasets. We are using just this one. Then the datamining activity, if I can say, in Modeler we take that here. This is the Filter1 mode, what I use are the tier-2 filter certain unnecessary data, you can see here that actually I just used the customer ID and the campaign name and probability campaign. This one is a table kind of display.

Okay. Let's jump to Studio itself. So how does it look like? Basically, in Studio you want to get the data from the [stream] by very simple mechanism. If somebody is very familiar the Excel sheet connect and sheet read. This would be not any problem for that person. First, you have to put the prepared statement to sort of connect anything SPSS to the Studio. Then you have comment, effectively express what kind of logic you then want to use in your [dom.doc] data file. That's one is done by SPSS connection. Then you effectively take the data by providing the SPSS feed. You have the possibility to decide which node you consider, which is the final node of the string for you for your optimization model. And here is the second document is expressing that. You have to put the name of the node where you want to draw the data.

Okay. After that, simply you just execute as you would execute anyway an optimization run configuration. And at the back, the SPSS stream will be execute. And then at that node where you write, what you put into the SPSS [feeds], that node will give to the optimization elements. And then the optimization part will start and provide the results.

Here we go. We have a solution. You can see [inaudible] formats. And here what I show -- so I showed in the screen dump this is what it looks like in real life. The first customer segment doesn't gets any offer or were not given any offer. The second segment, they will be given savings and pension and so on.

There are two other interesting features I would like to talk about. One of them is showing how a data looks like. So for instance, we are here in the stream. So you can display the stream in the Studio, the result of this stream in Studio in table format. Why is it useful? Probably it's useful before you start writing your model. But definitely given debugging is very important, if you have any trouble with the data. You can see here is the customer ID and the first campaign names and the probability, second campaign names and probability.

Okay. The third and final interesting feature is let's assume that you have a very long-running stream. If you do have to compute all the time when you [enter your] optimization model, you have to execute the stream as well. Then it's actually waste of time for you to wait for a long-running stream. So what you can do is take a snapshot in a vertical [slime] of Modeler. So what you can do is generate data, and actually it's kind of extra feature, you can generate the topology of the data. So if you click on run SPSS stream and generate all your files, you will get two files. And these two files are one with the data and the other one is -- this is what you can see on the screen.

So basically, what I did here is generate the data in a top-down format, in OPL format. In that case, if you use this one, this data, then you don't have to execute the SPSS stream again and again. Just input the red and the final node once you specify that - in your SPSS feed, this data will be read, speeding up the optimization process.

Thank you very much.

Jeremy Bloom: Just to wrap up here, I wanted to reiterate the value of the IBM ILOG Optimization technology. Optimization allows you to make smarter decisions. You get better performance for lower cost. True optimization finds non-obvious solutions that maximize your value and minimize your costs while observing many complex requirements and strategies of modern businesses. An optimization produces quantifiable benefits on your bottomline. And if you want to hear more about quantifiable benefits, let us know. We can talk to about very many used cases and success stories that we've had.

Optimization also permits you to make faster decisions. You can automate decision processes to increase the speed of your response in today's accelerating markets. And you can allow your managers and your planners to focus on critical complexities rather than on routine issues.

If you're developing an optimization application, you can get faster and lower cost development and maintenance using the high-level modeling tools that are available in CPLEX Optimization Studio. They enable your analytic code and validate your model in less time and with less effort than traditional programming languages. And by increasing the transparency of your model, it makes maintaining and upgrading your system easier and more reliable.

And finally, optimization turns information into action. IBM ILOG Optimization technology leverages the investments you are making in enterprise information technology.

And with that, I'm going to close the presentation and I'm going to turn to the questions that we received. If you have a question, please submit it and we will address it as soon as we can.

I'll take the first question, which is, is ODM a part of CPLEX Optimization Studio now?

No. The answer is it is not. ODM is a separate product; however, ODM includes a version of CPLEX Optimization Studio. That is, if you want to build an ODM application, it's built on top of CPLEX Optimization Studio. But in order to get ODM, you actually have to purchase it as a product -- the bundled product inclusive of CPLEX Optimization Studio.

I'll address the next question to my colleague, Ferenc Katai. Do we have any in-depth training options, preferably online, available for licensed users to make use of the new version 12.3 features? Ferenc?

Ferenc Katai: Hello there. Yes. Well, actually anybody could take training courses, and some of them are already converted to -- I simply call it the Without Trainer or kind of -- I don't remember the exact term for them. But we have online trainings, definitely. And we have even training where you don't even have to have a trainer.

That one is -- has some separately purchased and it doesn't seem that you need a license. You can just acquaint to a prospect who wants to explore through the training the product. And so it does not [require them to state] any license.

Jeremy Bloom: And you can review the training options on our website, which is ibm.com. Search for optimization. And if you have specific questions about training, then e-mail to me or to Ferenc and we'll put you in touch with our training people.

Next question. Can CPLEX solve non-convex quadratic problems with quadratic constraints? Ferenc?

Ferenc Katai: At the moment, no, but it's very high on the agenda. We are working on it.

Jeremy Bloom: Okay. Next question. Is CPLEX version 12.3 designed to be used in the cloud? In other words, can it be deployed in, say, a Windows Azure or Amazon? Ferenc?

Ferenc Katai: Well, the same kind of answer -- unfortunately, it's to say that -- if I say more concrete things, I will be shot in the head, so I cannot. So actually it's also on the agenda very high. At the moment, you cannot use XML resource (inaudible) but again, we are working on it.

Jeremy Bloom: Next question. Is the callable library the next -- instead of Concert -- available to be used from .NET?

Ferenc Katai: No, you have to use the Concert to access (technical difficulty)

Jeremy Bloom: Right. The callable library is a C library. Can we read XML files?

Ferenc Katai: Well, this one uses -- please send an email what is exactly your need, because you want to have data, you want to sort of communicate to XML the models, it's not very clear what you mean. In (inaudible) wherever our ID is based on, we could have XML [pressure]. But again, what is exactly you want to use for is not very clear.

Jeremy Bloom: Okay. So the person who asked that question can communicate with us by e-mail either to me or to Ferenc, and we will give you a more detailed answer.

Another question about the indefinite QP. Does it escape local optima?

Ferenc Katai: I would say yes, but actually the engine guys should be here and update – to answer this.

Jeremy Bloom: Okay. So again, if you want more details on that question, send us an e-mail, and we'll get our developers and our high-level support level to discuss the options there.

And the final question was one we addressed before that CPLEX Optimization Studio is not currently available on the MAC OS.

Ferenc Katai: There is just one comment on this. The problem is not really the IDE. The IDE is a sort of community, and I have to tell you that internally one of our guys, one of our developers, is using MAC OS, and he is a developer. So he is using the IDE on MAC OS. It's not just the IDE what is the blocking or locking. What [does you have to imagine?]

The Studio is providing you tool for you to (things] we have to think on and go to a new port like MAC OS. One of them is the database. Any kind of things when you want to use IDE, definitely you need some database and not just database, but spreadsheets. For instance, if we want to easily go to MAC OS, we should support OpenOffice.

Then the other one is the Java script part. So we need Javascript engine on MAC OS. So you can use for flow control Java screen component. So all these kinds of things when we have -- and we have to have to be [performant]. It's not just that people want the (inaudible) Java screen engine. It should be as (inaudible) as we are on Windows.

So these are the -- I'm not complaining. As I said before, we don't have -- but this one is also quite high on the agenda.

Jeremy Bloom: Okay. Got a few more questions that have come in. Is the new version of 12.3 available for the academic initiative program?

The answer is we are in the process of uploading it now. It's probably not available quite yet, but it will be available shortly, and it will also be available shortly in the downloadable trial.

Next question. Has there been any improvement in the LP barrier solve using multiple cores for large dense problems? Ferenc?

Ferenc Katai: Yes, we've improved the barrier quite a lot. And I think this one is very important, because barrier is a part of our software. If you want to go, for instance, just reframe that to cloud, that one should -- if you want to improve the [kernel solver] then the cloud would be kind of a filler for us, I would say. So definitely, the improvement already, and we are putting there quite a lot of efforts to improve even more.

Jeremy Bloom: Next question. Can the version 12.3 CPLEX DLL be bundled with our custom application? If so, does each use of a custom application needs to be a licensed CPLEX user?

All right, maybe we have to take that one offline, Ferenc. We'll respond directly to that person.

Ferenc Katai: Right.

Jeremy Bloom: Okay. And another question. Does Concert work with Visual Studio 2010? Ferenc?

Ferenc Katai: It does. I think it does -- all these kinds of questions are very clear if you go to ibm.com and you look for the Optimization -- under Optimization Studio. And we have at the left tab something like "system requirements." And if Visual Studio 2010 is not there, then I was not correct. I believe it's there.

Jeremy Bloom: Okay. That being the case, I believe that's all the questions, and we are at the top of the hour. I want to thank you all for participating.