The MathWorks

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098
USA

Tel: 508.647.7000
Fax: 508.647.7001
info@mathworks.com
www.mathworks.com

Via Email – rule-comments@sec.gov

August 2, 2010

Ms. Elizabeth M. Murphy
Secretary
Securities and Exchange Commission
100 F Street NE
Washington, D.C. 20549-1090

Re: File Number S7-08-10

Dear Ms. Murphy:

The MathWorks Inc. submits these comments on the proposed rule issued on April 7, 2010 concerning revisions to Regulation AB and other rules regarding the offering process, disclosure, and reporting for asset-backed securities. These comments focus on the new disclosure requirements, in particular, the proposed requirements that entities filing a prospectus submit source code for a contractual cash flow ("waterfall") model in the Python programming language. These comments do not address other proposed requirements and changes to the regulations.

As discussed further below, MathWorks believes that the requirement to use a single programming language is unnecessary, inappropriate, and counter-productive to achieve the Commission's disclosure objectives of greater transparency and efficiency through viewable source code and a shareable, executable waterfall cash flow model. Mandating a single programming language in this situation will restrict the ability of the market to promote more user-friendly solutions and to respond to changes in technology. Mandating a single programming language is also contrary to sound procurement policy established by the Office of Management and Budget and inconsistent with regulations governing federal standards of impartiality. At a minimum, the rules should set forth technical requirements for allowing viewable computer code and shareable waterfall models, which other computer programs and packages may also be able to satisfy.

Background – The MathWorks, Inc.

The MathWorks Inc. is a leading creator and provider of software for technical computing and Model-Based Design, including for computational finance. Financial

professionals worldwide use the interactive programming environment and prebuilt computational libraries of MathWorks' MATLAB® software to develop quantitative applications used in many different areas of the financial services industry, including to develop and implement increasingly complex financial models. These financial models include the waterfall cash flow models mandated by the proposed rule.

In the financial services market, MathWorks modeling and analysis software is used by

- Each of the top fifteen asset-management companies
- Nine out of the top ten U.S. commercial banks.
- Eleven out of the top fifteen hedge funds
- All OECD Central Banks, including the Federal Reserve and European Central Bank

MathWorks MATLAB software has over 1,000,000 users, in more than 100 countries. The MathWorks software products have become fundamental tools for work at the world's most innovative technology companies, financial institutions, government research labs, and at more than 5000 universities. MathWorks is headquartered in Natick, Massachusetts, was founded in 1984, and currently employs more than 2000 people worldwide, with the vast majority located in the United States.

General Comments:

The proposed regulations, as they relate to the waterfall computer program (§ 229.1113 and § 232.314), have as at least the following objective – to make it easier for the smaller institutional investor to have transparency into the cash flow analysis that is currently described narratively in the prospectus. April 7, 2010 Notice of Proposed Rules ("NOPR") at pp. 206-211. To meet this objective, the proposed rules require the offering entity to submit source code to a waterfall computer program that models the contractual ("waterfall") cash flow described in the prospectus. Further, the potential investor must be able to run the computer model reflected in the source code and substitute its own values for certain values in the issuer's program. Proposed § 229.1113; NOPR at p. 210. These two technical requirements – that the waterfall program code be viewable and be shareable/executable -- can be satisfied by multiple software programs or packages other than the Python programming language and at lesser burden to the potential investor and to the issuer of the asset backed security.

Proposed Rule Should Set Forth Technical Requirements: Rather than require a single programming language, the SEC should establish technical requirements and allow the market and the programmers to determine how best to meet those needs. Mandating a single program such as Python, or any other program, runs the risk of locking users into a program that might not be the best, might change or be superseded by better programs, and unfairly biases the

market in favor of a particular software vendor when it does not need to and when procurement rules indicate it should not.

The proposed rules do not make allowance for the burden on the investor, especially a smaller investor, to acquire and learn and manage the Python programming language. Python is perceived by many to be a more complex or advanced programming language, requiring a greater sophistication in computer science and computer programming. Many financial quants, who do not have the depths of programming skills that some Python users have, prefer other software programs as easier to use, such as MATLAB. And although Python may be available without license fee because it is available under an open source software license that does not mean it is "free," as in no cost to the potential investor. Any investor, as well as issuer, that does not currently use Python will have to invest resources to learn or hire staff that are skilled in that programming language to be able to install, build, use and maintain tools for using the programming language. Or they will choose to pay for a commercial distribution of a supported version of Python.

Other Software can Satisfy the Requirements: Mandating industry wide use of Python is not necessary. The investor might already be using a financial software application such as MathWorks' MATLAB or one of any number of other available financial analysis and modeling software that is able to satisfy the technical disclosure requirements of the rule from a number of sources, *see, e.g.,* (1) Fincad (www.fincad.com) ; (2) Numerix (www.numerix.com); (3) SAS (www.sas.com); (4) Tibco (www.tibco.com) ; (5) Wolfram (www.wolfram.com) and (8) r-project (www.r-project.org), or more general programming languages or applications such as C/C++, Visual Basic, Java, or Excel®.

MATLAB has viewable code and shareable models: As an example, the MATLAB source code that would be used in creating a waterfall program is viewable, human readable, and could be submitted along with the prospectus to the EDGAR website. MATLAB code is standard ASCII formatted text. MATLAB is also an interpreted programming language, as is Python; to the extent the Commission believes being an interpreted programming language is relevant or preferable. Further, the MATLAB code for a waterfall program can be made available to be run by the investor in three different ways, all of which meet the SEC's objectives of being available for free, and can be made available without having executable code on the EDGAR website.

First, if the investor already has a MATLAB license, the investor can download the waterfall program code for the model and run the MATLAB code using its existing MATLAB software.

Second, MathWorks offers software that will allow the issuer to compile the waterfall program into a distributable application. That distributable version of the waterfall program may then be made available to the potential investor for download from the issuer's website, free of charge. The investor does not need to have MATLAB and will be able to run the waterfall model. See http://www.mathworks.com/desktop-web-deployment/ and http://www.mathworks.com/desktop-web-deployment/deploying-code-executable-software-component.html (printed copies attached). When the issuer submits the source code for the waterfall program to the EDGAR website, the issuer may provide a pointer to its website or some other location where the investor may download an executable version of the waterfall program.

Third, the issuer may host the MATLAB software waterfall model on its server, allowing the investor to come to the issuer's website and run the model in a web browser, much like individuals may prepare their tax returns using Turbo Tax® on the web. See http://www.mathworks.com/desktop-web-deployment/deploying-code-web-application.html (printed copy attached). Again, the issuer may provide a pointer to the appropriate website or URL to access such a hosted version of the waterfall program when it submits the source code for the waterfall program to the EDGAR website.

These scenarios should also be available to a greater or lesser extent with other financial modeling software available on the market. And the programmer of the waterfall program may create interfaces to allow the users to modify different values in the program.

Let the Market Decide What Software to Use: Under the proposed rule, if the investor is going to be able to run the Python code, the investor will have to download, install, and "build" the Python software environment, including finding compatible versions and installing any needed supporting libraries. This process of building the open source software can be complicated, especially for those not well-versed in computer programming. Further, certain competing software offerings, such as MATLAB, can be simpler to use, and in terms of the compiled application or hosted model mentioned above, the pre-packaged waterfall programs remove the complexity of building the application so the investor can just use it. The issuer can also provide interfaces to input different variables in the model. That the investor may obtain the compiled application or the browser version from a website maintained by the issuer is no different from requiring the user to go find the necessary versions and associated libraries for Python. Indeed, it is likely to be easier, because in these scenarios the application can presumably be easy to run and there will be a dedicated location maintained by the issuer, which is not the case with Python.

The Proposed Rule is contrary to procurement policy and rules

Before addressing specific NOPR questions set out in connection with the disclosure requirements, MathWorks wants to comment briefly on how this rule would impact the financial software market and is contrary to procurement policy.

Market Competition -- Financial Software

By endorsing a single programming language, the proposed rule is picking winners and losers in an active commercial market for competitive financial software. It is not in the best interest of the government nor its citizens, particularly the investors/issuers most directly effected, to limit the number of options available to meet the general disclosure requirements. Rather, if the SEC will only require viewable source code and shareable, executable models, it will allow the commercial market place to determine which software program best serve the issuers'/investors' needs.

The software language Python competes with MATLAB, as well as a number of other financial software programs including but not limited to programs developed and offered by: (1) Fincad (www.fincad.com); (2) Numerix (www.numerix.com); (3) SAS (www.sas.com); (4) Tibco (www.tibco.com); (5) Wolfram (www.wolfram.com); and (6) r-project (www.r-project.org). Competition occurs based on a wide variety of factors including ease of use, maintenance, overall cost, and security. The proposed rule, by mandating a particular software language, would give an unfair advantage to a particular brand regardless of whether the actual users determined that it was in their best interest and most efficient to use.

Proposed Rule Is Contrary to OMB Procurement Policy

By a memorandum of July 1, 2004, the Office of Federal Procurement Policy within the Office of Management and Budget, clearly stated that the regulations regarding agency investments in information technology "are intentionally technology and vendor neutral, and to the maximum extent practicable, agency implementation should be similarly neutral." M-04-16. This sound direction from OMB should be followed by the SEC in this instance rather than pre-selecting a technology and brand that is neither the government's nor the industry's standard in this instance.

Prohibition on Preferential Treatment And Product Endorsement

As sound government policy, all government employees are governed by the "Standards of Ethical Conduct for Employees of the Executive Branch. 5 C.F.R. § 2635. Those standards,

while not directly applicable to official agency action, nevertheless articulate principles that should be followed in this instance. As stated in those regulations, a "basic obligation of public service" is for each and every employee to "act impartially and not give preferential treatment to any private organization or individual." 5 C.F.R. § 2635.101(b)(8). This obligation properly ensures that personal preferences and biases of federal employees are to have no place in public decision making—including which technology solution will best serve public purposes. Similarly, those same regulations prohibit a federal employer from endorsing "any product, service or enterprise." 5 C.F.R. § 2635.702. The proposed regulations would accomplish for the agency what the regulations prohibit for any government employee—the endorsement and preferential selection of a particular product.

Specific Questions from the proposed rulemaking

MathWorks provides comments to the below questions raised in the NOPR regarding the use of a single computer programming language or application.

1. Is it appropriate to require issuers to submit the waterfall computer program in a single programming language, such as Python, to give investors the benefit of a standardized process. If so, is Python the best choice or are there other open source programming language alternatives (such as PERL) that would be better suited for these purposes?

   Response

   MathWorks believes that it is not appropriate to require a single programming language be used to satisfy the disclosure requirements. There is a competitive marketplace for financial software and for policy and other reasons the government should not be mandating a particular technology (open source versus proprietary) or a particular brand. See OMB policy and regulations mentioned above. The regulations should be technology and vendor neutral and focus on meeting the specific functional and performance requirements that are stated in proposed regulation § 229.1113.

   There has been no indication that Python is the best programming language or software for the purpose, nor the easiest for investor or issuers to use. To the contrary, given the nature of open source software it is quite possible that it will be more difficult for small investors to get the benefit of greater disclosure. As currently proposed, the regulations would require each and every investor interested in utilizing the proposed waterfall program to select, install, maintain, and train on Python. As MathWorks understands the general purpose of the proposed regulations, the intent of the regulations is to make it easier, particularly for smaller investors, to make better and more informed investment decisions. The SEC should require that the information be made available in

one or more commercially available formats that can be utilized and manipulated by investors at no additional cost to the investor. Issuers who are seeking investors have the economic incentive to make the mandated information available in a variety of formats or in formats that are easier to use so that their prospective investors can select the format that best meets their needs at the most efficient cost. Because Python is highly technical in nature, small investors would need to expend resources to acquire technical expertise to both learn it and maintain it which would duplicate, in many instances, an already existing investment in other formats. If the SEC goal is to make the mandated information available to as many investors as possible, the requirement should not limit the mandated information to a single, non-market dominant programming language but rather should require that the mandated information be made available so that it meets the functional and performance requirements established by the SEC as necessary to allow investors to make better informed judgments.

An even better solution might be for the regulations to require only that the investor-useful data be made available in a raw, machine readable format and uniformly tagged with sufficient metadata such that the date could be readily imported in by investors into the broadest range of existing financial analysis tools – chosen by the investor rather than the government. Rules of this nature would allow the widest possible used of the data for its intended purpose and would allow for innovation in a variety of existing and new programs, rather than a single programming language and be consistent with technology neutral procurement policies and regulations mentioned above.

2. Should more than one programming language be allowed? If so, which ones and why?

Response

The SEC should adopt regulations that establish functional and performance requirements that allow both the issuer of asset based securities and the investors in such instruments to make the most efficient decision on how to meet those requirements. The regulations should not be based on any particular programming language. Indeed, the proposed regulations puts the SEC in a position of lagging behind technological improvements in the commercial marketplace by writing into the regulations a particular solution that would necessitate formal rulemaking to change in the future, thereby denying users the timely advantages of technical and programming advancements. As stated above, issuers will have the incentive to make the mandated information available to prospective investors in the formats that can best be utilized by the investors.

3. Should we restrict ourselves to only open source programming languages or allow fully commercial or partly-commercial languages (such as C-Sharp or Java) to be used? If so, what factors should be considered?

Response

First, the proposed regulations indicate a misunderstanding with regard to "commercial" languages. The software language currently being proposed is "commercial" with a number of variations available in the commercial marketplace. As with many open source projects, it can be difficult to identify, install, and configure the appropriate combination of libraries to create a usable software package. Because of the need for easier installation, verified packages (with known versions of each library), and ongoing support, most investors (and certainly most small investors) are likely to choose to purchase access to a commercial distribution package of the Python programming language, see, e.g., www.activestate.com/activepython ($999 per server/year); www.enthought.com ($199-990 per user per year).

Thus to the extent one goal of the proposed rule is to limit out-of-pocket costs to investors, it is unlikely to do so. Further, the proposed rule and its costs/benefits analysis makes no provision for the time that can be involved in installing a no-license-fee open source software program, nor the need to develop or hire new programming expertise to be able to understand, run, and manipulate the proposed Python based program when the investor has no such expertise. Second, as stated above, the proposed regulations already identify functional and performance requirements for the waterfall program that are sufficient for use by the marketplace. Those functional and performance requirements are sufficient to allow issuers and investors the opportunity to select the software that best meets their needs and is the most efficient for them.

4. Are there other requirements we should impose on the possible computer programming languages that are used to satisfy this requirement, other than such languages be open source and interpreted?

Response

The SEC should not require that the computer programming languages be open source but rather should follow OMB guidance with regard to being technology neutral. The SEC has set forth basic functionality and performance requirements, and it should allow the market and computer programmers to determine the best way to satisfy those requirements. The requirements appear to be, (i) viewable source code for the waterfall computer program, submitted to EDGAR for public access, and (ii) a mechanism

whereby the investor is able, with no out of pocket license fees required, to run the waterfall program and experiment with how different assumptions might change the waterfall program outputs. These objectives can be met through means other than mandated open source and interpreted software.

5. Under our proposal, issuers would be required to file the waterfall computer program in the form of downloadable source code on EDGAR. Prior to filing, the code would not be tested by the Commission. Would downloading the code onto a local computer give rise to any significant risks for investors? If so, please identify those risks and what steps or measures we should take to address the risks, if any?

   Response

   Whether the unreviewed source code holds risks depends on whether it has been in anyway corrupted, such to introduce bugs or viruses when installed on an investor's computer. Corrupted or risky code could be received from the issuer due to some failed testing by the issuer or if something is hidden in the underlying software code the issuer is using to create the waterfall program. An issue with open source code is that there is not a single source that necessarilycontrols the source code. Under the SEC proposed rules, the investor would download the source code and run it on their computer using a computer programming language the investor may have never used before, obtained from a currently undetermined and unspecified location.

   One step that could be taken to reduce this risk would be allowing more flexibility in how the market/issuer meets the SEC policy objective of having waterfall program source code available and shareable. For instance, rather than downloading the waterfall program and the necessary code to build and run the waterfall program, some issuers might make the waterfall program available to an investor to run in a browser, hosted by the issuer. This would not require the investor to download the waterfall program or whatever software program is needed to run the program.

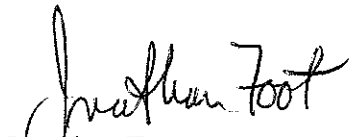
6. Are the proposed input and output requirements for the waterfall computer program appropriate? If not, what type of output and tests should be required for the waterfall computer program? Should the outputs of the waterfall computer program be specified in detail by rule, or broadly defined to afford flexibility to ABS issuers?

   Response

   The commercial marketplace for financial management software is constantly producing new and innovative means and processes for the analysis of financial information. A

minimum requirement should be established by rule, but the marketplace should be allowed to be innovative and competitive with regard to the information available.

Respectfully submitted,



Jonathan Foot
Assistant General Counsel

The MathWorks™
Accelerating the pace of engineering and science

Products & Services | Solutions | Academia | Support | User Community | Company

Share the work you do in MATLAB with others

MathWorks products provide several ways to share individual algorithms or complete applications that you develop using MATLAB® and add-on toolboxes. You can distribute your code directly to others to use in their own MATLAB sessions. To distribute to people who do not have MATLAB, the MathWorks application deployment products enable automatic generation and royalty-free distribution of turnkey applications and components that can be easily integrated into a larger IT infrastructure.

For **embedded systems** and real-time applications, you can use MathWorks products to automatically generate C or HDL code from MATLAB and Simulink® algorithms for microcontrollers, DSP chips, and simulators.

Learn more:

- Deploying MATLAB Code as an Executable or Software Component
- Deploying MATLAB Code as a Web Application

FREE Desktop and Web Deployment Technical Kit

Desktop and Web Deployment Trial Software

Contact Sales
Trial Software
Free Technical Kit

Applied Biosystems
MATLAB has always been ideal as an algorithm prototyping tool, but MATLAB Compiler adds a whole new dimension, allowing rapid delivery of sophisticated solutions.

» Read the story

Related Products

- MATLAB Builder EX™ (for Microsoft Excel)
- MATLAB Builder JA™ (for Java language)
- MATLAB Builder NE™ (for Microsoft .NET Framework)
- MATLAB Compiler™

Deploying MATLAB Code as an Executable or Software Component

- Desktop and Web Deployment Overview
- **Deploying MATLAB Code as an Executable or Software Component**
- Deploying MATLAB Code as a Web Application

Using MathWorks application deployment products, you can eliminate the costly and error-prone work of recoding your MATLAB algorithms in another programming language. Because you maintain your source code in MATLAB, you can easily develop and update your algorithms and automatically package them as standalone executables or software components for integration in environments such as C, C++, Java™, .NET, and Excel®.

MATLAB Compiler packages your MATLAB applications as encrypted standalone executables or C/C++ shared libraries. MATLAB builder products work in conjunction with MATLAB Compiler to create standard components for use with Java, .NET, or Excel. These executables and components can be deployed royalty-free on operating systems supported by MATLAB.

Deploying MATLAB Components to C/C++, Java, .NET, and Excel

Robeco Develops Quantitative Stock Selection and Portfolio Optimization Models

Free Desktop and Web Deployment Technical Kit

Learn how companies deploy MATLAB applications royalty-free to desktops, intranets, and the Web.

» **Get free kit**

Trials Available

Try the latest versions of desktop and Web deployment products.

» **Get trial software**

- **Contact Sales**
- **Trial Software**
- **Free Technical Kit**

Intuitive Analytics

"MATLAB Builder NE gives us a huge advantage in moving from prototyping, testing, to deploying our applications because we avoid the enormous cost of rewriting everything in C/C++."

» **Read the story**

Related Products

- **MATLAB Builder EX™** (for Microsoft Excel)
- **MATLAB Builder JA™** (for Java language)
- **MATLAB Builder NE™** (for Microsoft .NET Framework)
- **MATLAB Compiler™**

Deploying MATLAB Code as a Web Application

- Desktop and Web Deployment Overview
- Deploying MATLAB Code as an Executable or Software Component
- **Deploying MATLAB Code as a Web Application**

Using MathWorks application deployment products, you can develop MATLAB based components for the Web that execute mathematical computations and generate interactive graphics. After developing an algorithm in MATLAB, you can automatically create a standard component designed to integrate in a Web application using MATLAB builder products for either Java or .NET.

Once you place the component on a Web server, your users access the application through a Web browser and do not need to install additional software on their desktop computers.

The Java and .NET components created by the deployment tools can be used in conjunction with standard Web technologies such as ASP.NET, SOA, SaaS, JavaScript, and HTML.

Application Deployment with MATLAB

Rendering High Dynamic Range Images on the Web

Free Desktop and Web Deployment Technical Kit

Learn how companies deploy MATLAB applications royalty-free to desktops, intranets, and the Web.

» Get free kit

Trials Available

Try the latest versions of desktop and Web deployment products.

» Get trial software

Contact Sales

Trial Software

Free Technical Kit

UniCredit Bank Austria

"With MATLAB, we can focus on business logic instead of implementation details. We can deploy an algorithm in a Java environment the same day, without any additional coding. This approach enabled us to cut our development time in half, if not more."

» Read the story

Related Products

- MATLAB Builder EX™ (for Microsoft Excel)
- MATLAB Builder JA™ (for Java language)
- MATLAB Builder NE™ (for Microsoft .NET Framework)
- MATLAB Compiler™