



.NET Rocks!
 The Internet Audio Talk Show
 for .NET Developers
 With Carl Franklin **msdn**
 and Richard Campbell
<http://www.dotnetrocks.com>

<http://www.dotnetrocks.com>



Carl Franklin

Carl Franklin and Richard Campbell interview experts to bring you insights into .NET technology and the state of software development. More than just a dry interview show, we have fun! Original Music! Prizes! Check out what you've been missing!



Richard Campbell

Text Transcript of Show # 224
 (Transcription services provided by [PWOP Productions](#))



Oren Eine -NHibernate and Rhino Mocks

March 29 2007

Our Sponsors



**DATA
DYNAMICS**
<http://www.datadynamics.com>



CoDe
 component developer magazine
<http://www.code-magazine.com>



telerik
deliver more than expected
<http://www.telerik.com/>



Geoff Maciolek: The opinions and viewpoints expressed in .NET Rocks! are not necessarily those of its sponsors or of Microsoft Corporation, its partners or employees. .NET Rocks! is a production of FranklinsNet, which is solely responsible for its content. FranklinsNet - 'Training Developers to Work Smarter.'

(Music)

Lawrence Ryan: Hey, Rock heads! Put down the Klingon dictionary and listen up, it's time for another stellar episode of .NET Rocks!, the internet audio talk show for .NET developers with Carl Franklin and Richard Campbell. This is Lawrence Ryan announcing show #224, with guest Oren Eine, recorded live, Thursday, March 1st, 2007. .Net Rocks! is brought to you by FranklinsNet, 'Training Developers to Work Smarter', and now bringing the Just-In-Time Team System class with Joel Semeniuk onsite for your development team online at www.franklins.net. Support is also provided by 'telerik - Combining the best in Windows Forms and ASP.NET controls with first class customer, service online at www.telerik.com. And by 'CoDe Magazine - The leading independent magazine for .NET developers, online at www.code-magazine.com. And now the man who firmly denies ever mooning the mouse at Epcot, Carl Franklin.

Carl Franklin: Thank you very much, and welcome back to .NET Rocks! This is Carl Franklin on the East Coast of the United States of America, coming to you from a booth somewhere on the fifth floor of the Dewart Building in New London, Connecticut, and my partner in crime Richard Campbell out there in Vancouver, British Columbia. Hey man, mow any lawns lately in your bare feet?

Richard Campbell: Not lately, but I think we're still in Orlando when this show is being broadcast.

Carl Franklin: This is true.

Richard Campbell: So, it's still hot.

Carl Franklin: So, I am not sure whether to say I'm in Orlando now or not. I mean, I am from your perspective, dear listener; but from my perspective, I am in New London.

Richard Campbell: It's all very confusing, really.

Carl Franklin: It is confusing. I don't think Einstein's Theory of Relativity explains it enough. I think we have to look to Mr. Stephen Hawking to figure this out.

Richard Campbell: You think so, huh?

Carl Franklin: Yeah.

Richard Campbell: I think he wouldn't take our call, but that's just me.

Carl Franklin: Well, that was just that one time. I've got an email from Andy Weaver.

Richard Campbell: Hit me.

Carl Franklin: "Future Show Suggestion" is the subject. "Hi guys, I'll start off with a confession. I don't listen all of your shows. I am a database administrator and not a programmer nor a developer." Stalker

(laughing).

Richard Campbell: What was that (laughing)?

Carl Franklin: He is a stalker, oh no.

Richard Campbell: He is a DBA. DBAs don't stalk, they sit in bunkers.

Carl Franklin: I was just waiting for a, you know, "Actually, I'm a carpenter."

Richard Campbell: Oh, great.

Carl Franklin: "Yet I do try to listen to shows that don't directly apply to me, so I can at least get a sense of other topics." Well, there you go, he explained himself. "But the main reason I am writing is for a future show suggestion. Since you are now doing two shows a week, we should be able to fit this in somewhere. Business Intelligence is an area that I am soon going to be more involved in, and I would love to hear from somebody from ProClarity, since it is now a subsidiary of Microsoft." Hmm, didn't know that.

Richard Campbell: Yeah, it got bought.

Carl Franklin: "I am sure it will become very more tightly integrated with Microsoft SQL Server and the .NET platform. Getting a sense of what they are up to and where they are going would probably prove valuable to many of your listeners. Keep up the good work and congratulations on taking the show to twice a week." Well, if you keep listening, we'll keep talking.

Richard Campbell: There you go.

Carl Franklin: Thanks, Andy. Yeah, that sounds like a good idea.



Richard Campbell: Well, definitely. I've actually been involved in a ProClarity deployment some years ago and a fan of the product. It was before Microsoft acquired them. So, I don't know what things are like now, but I am sure we can find out.

Carl Franklin: I am sure we can.

Richard Campbell: It's not too far off base, presumably it will get a .NET interface at some point.

Carl Franklin: Yeah, and who knows? We'll just poke around a little bit, we'll ask around, see if we can get somebody on the show.

Richard Campbell: You bet. All right, one for me.

Carl Franklin: Okay.

(00:04:52)

Richard Campbell: This is from Abraham Bayarkhuu, hope I got your name right, Abraham. "Greetings Carl, I am an ASP.NET C# developer and a longtime listener. Many people before me have said how great your show is. So, I'll just say that it's good listening and one of the few knowledge-packed podcasts I listen to." And then he parenthetically said, "(Podcasts can actually give you useful information? Rubbish, I say.) Anyway to the point, I like the video with Ayende on NHibernate" -- he is talking about dnrtv -- "and would like to see more and more on ORMs in general. I think ORMs are a great productivity tool that programmers need to learn and developers need to understand the workings of it. There is a difference between coders and programmers and developers and [insert other similar title here. So, more on ORM, NHibernate in particular.] Please keep up the good work, you help out the community, and the communi-me a lot, Abraham."

Carl Franklin: And the communi-we.

Richard Campbell: There you go.

Carl Franklin: Well, how fortuitous is that, Richard?

Richard Campbell: Yeah. Don't we have a guest on today?

Carl Franklin: I think so.

Richard Campbell: Isn't it Ayende?

Carl Franklin: It is Ayende!

Richard Campbell: Well, how convenient is that. What's he talking about?

Carl Franklin: ORM, specifically NHibernate.

Richard Campbell: Wow! How useful is that?

Carl Franklin: I think he is waiting on the phone right now, so let's get on with it. So, West Michigan Day of .NET is happening. When is that and where, Richard?

Richard Campbell: The West Michigan Day of .NET happening May 19th in Grand Rapids, Michigan at the Davenport University.

Carl Franklin: Very good.

Richard Campbell: And if you wanted more information on that, check shrinkster.com/n1h.

Carl Franklin: Also, we have to talk about MIX, MIX '07, which is happening April 30th through May 2nd. Am I right?

Richard Campbell: That's right. The 72-hour conversation.

Carl Franklin: In Las Vegas.

Richard Campbell: Keynoted by no less than Ray Ozzie, and it's at the Venetian.

Carl Franklin: That's going to be fabulous. There's going to be some special announcement there that's going to "blow people's minds".

Richard Campbell: Looking forward to it.

Richard Campbell: We don't know what it is, but it should be amazing.

Richard Campbell: You can visit them at www.visitmix.com.

Carl Franklin: Also, the Dutch conference, the first Dutch Code Camp.

Richard Campbell: That's right, May 12th, the first Dutch Code Camp, and you can look that up at www.code-camp.nl.

Carl Franklin: Excellent. And getting to the job opportunities that our listeners have sent us, there's the New York tour; Greg Brill at Infusion in New York City is offering living in an apartment rent-free for a year in Manhattan with an appropriate salary for some great software development down there in the financial sector. For information on that, go to shrinkster.com/kh6.



Also, there's a great gig in Washington, D.C. for ASP.NET gurus. If you think you are the stuff in ASP.NET, go to shrinkster.com/mmj. This is also a bunch of show listeners. You got to be located near or be willing to be relocated to Washington D.C., and all the credentials and all the requirements and all the party schedule is all up there at shrinkster.com/mmj. Good stuff, and good luck. We've gotten people -- a continuous stream of people who are interested in these things, so it's working out for the community. All right, Richard, well let's introduce Oren.

Oren Eine is a .NET developer living in Israel and currently working for We! He is heavily involved in the NHibernate project, among other things, and you can read his blog at <http://www.ayende.com>. He is also going to be up at DevTeach in Montreal this coming summer. Welcome, Oren Eine.

Oren Eine: Hi, how do you do?

Carl Franklin: Welcome back, I should say. We had a very good show on dnrTV, dnrtv.com, on NHibernate.

Richard Campbell: I've been getting a lot of email about you lately, actually, and not just around NHibernate, but also some other projects you've worked on or, I think, invented; RhinoMocks is yours, isn't it?

Oren Eine: Yes, RhinoMocks is mine.

Richard Campbell: It's very cool. I don't even know where to begin with you, there's so many choices. Maybe we better start with NHibernate, because I think that's sort of the hub of all the trouble here.

Carl Franklin: Actually, let's start with your name, Ayende. Is Ayende your nickname and Oren is your real name?

Oren Eine: Oren is my name, Ayende is a nickname that I used in '97 on the internet; and I was 17 so everything is allowed because of this, but the reason that I use it, I started blogging again in 2002, I think, and I was in the Army then, and the idea, as our defense thought, there was a very stupid policy about blogging. So, I know at least one case that they put someone to court-martial, because he published a critique about a movie.

(00:10:08)

Carl Franklin: About a movie, wow!

Oren Eine: A movie, yes -- TV show, actually. And this is stupid, so I figured that I don't need it,

and so I just use a nickname, and I used it for, I think, two years, and I was stuck with it.

Richard Campbell: Yeah. Well, I guess you've almost got to split two people out there that are actually you out in the internet that lots of folks know who Ayende is, and lots of folks knew who Oren is and may or may not connect the two together; but they're the same guy.

Carl Franklin: Is there a meaning to the word "Ayende"?

Oren Eine: "Ayende Rahien" is a part of a -- is a fascinating book that I read called "The River Jordan." And the meaning is very stupid. "Ayende" is "freedom," and "Rahien" is "down". So I said I was 17 (laughing).

Carl Franklin: It's good cover.

Oren Eine: Yes. I get accused of everything with it.

Carl Franklin: Okay. Now that we got that out of the way, let's talk NHibernate. So this is the .NET version, I guess you could say, of Hibernate, which is a Java application that is an Open Source ORM project. Is that an accurate statement?

Oren Eine: This is an accurate statement, yes.

Carl Franklin: Okay. And is NHibernate is something that you started, or did you come into the project later?

Oren Eine: No, I didn't start it. Actually, I think there was two or three attempts to port NHibernate before NHibernate came to .NET, and I jumped on the ship around the '06 era; and I wanted to start to understand it. And at that time I had some free time, so I sat down and went over the codes and built a simple tutor that would help me figure out what is actually going on. And eventually, I started using it in the real capacity, not just learning, but in We! projects, and it turns out that I am one of the most visible people in the NHibernate space, because I blog a lot and I blog a lot about NHibernate and issues that I can solve with it.

Carl Franklin: So, you're like the guy.

Oren Eine: Not actually. I know of three, four other guys that I go to ask questions. And the developer, the second developer, and Pierre Henri is currently writing a book about it; but, yes, I can say that I am an expert in it.

Carl Franklin: Okay. And I should also point out that if you want to see NHibernate in action, you



can go to shrinkster.com/mi5, and that points to the dnrTV episode we did on it. As far as feature sets of NHibernate versus the original Hibernate, is everything there, everything and more?

Oren Eine: No. I would say we have a priority in feature to Hibernate 2.1, and we are currently working on releasing the NHibernate1.2 release, which brings support for .NET 2.0 and Generics and lots of other very good stuff. And after that the roadmap calls for mostly priority with Hibernate 3.0. And we do bring features, mostly Sergey, but sometimes I bring features from Hibernate 3.0 that we need now. So I can't tell you where on the map we exactly may be right now, but we are compatible with 2.1 and probably we bring compatibility with 3.0.

Carl Franklin: So, Oren, we're looking at NHibernate. Can you tell us just a little bit about ORM systems in general and where NHibernate fits in the big picture?

Oren Eine: Okay. In the big picture, we divide NHibernate into two. You have the ORM database on the Unit of Work pattern, and Unit of Work basically says that you have... you start in Unit of Work, you do some sort of work, and then you finish. And the responsibility to stay with the implementation is to track what you did while you were in Unit of Work and persist it or do something else with it, but usually it's persistence. So, for instance, I can take an object and inside Unit of Work I can do almost any kind of modification that I want to it, and then I can exit Unit of Work, and the object will be automatically persisted .

(00:15:06)

Carl Franklin: So, it's almost like a transaction in a database.

Oren Eine: Yes, but it is and it isn't.

Carl Franklin: Well, yeah. I mean, it is, because you've got a starting point and an ending point; but it's not, because you're working with the states of objects.

Oren Eine: Yes, because we don't have production of memory yet.

Carl Franklin: Okay.

Richard Campbell: And maybe it's safer to say it's a batch rather than a transaction.

Oren Eine: Yes. It's much that more closely, but the idea is that you don't need to track what you are doing; you don't need to explicitly track what you are doing.

Richard Campbell: I guess that's one of the traps you get into when you start working in ORM is that you're constantly concerned with what code that you use against your objects results in queries to the database.

Oren Eine: That can be a problem if you don't know how to handle it, and it can be a very big problem in certain cases, because developer usually work on small datasets, a thousand rows maximum and when you go to production and you didn't realize what happened under the scenes, sometimes you get into situation where your database is issuing 10,000 calls per page. You should remember that I was called to fix it.

Carl Franklin: So you're saying that the ORMs fall into two categories, those that implement the Unit of Work pattern and those that don't.

Oren Eine: Yes. Those are not usually work in a more disconnected fashion.

Carl Franklin: "Disconnected" meaning not connected to a database?

Oren Eine: Not connected to a database, or maybe they're connected to not exactly a database, but a solutions and objects server.

Carl Franklin: Or even just persisting to XML or something.

Oren Eine: Yes. Yes, will persist, that that's just that..

Carl Franklin: So are you saying that the Unit of Work could be synonymous with persistence?

Oren Eine: Not exactly. You can do more things in Unit of Work than just persist, just you don't usually have... usually have for other patterns. It's usually disconnected work, which has a whole set of patterns which I am not overly familiar with. And maybe you can think about like a CSLA. So, he has the data pattern. In the software work similarly to that. You have an object that you can query for data, and then you can ask it to save; but it doesn't necessarily mean that it will track you. It sometimes seem to work this way; but personally, I obviously like doing Unit of Work pattern.

Carl Franklin: Okay. Oren, do all ORM products have issues with scalability?

Oren Eine: Do all applications have issues with scalability in that?



Carl Franklin: Not all applications, ORM. I mean, when you use ORM, do you inherently have to watch out for performance issues?

Oren Eine: I wouldn't say so. I would say that there are some specific issues with ORM that you usually won't run into in the typical hand-built data layer; but those aren't really scalability issues; they are usually misuse issues. Let me -- okay, here is an example I want to save 5,000 paintings. I can do it in a loop and call in certain things that are recent, and that will be -- that will have horrible scalability. But it would have --

Richard Campbell: Right, because it's going to fire out writes 5,000 times.

Oren Eine: Yes. I can do it in a batch and call it ten times. I can even open a scalable copy and do it once and it will resolve to have 2, say, 50,000 lines both. So it's all about how you do it. ORM has ORM has typically two different issues. One is the Select N+1, where you don't bring the data upfront. So you have to continue to fetch it from the database. The other issue is that you bring too much data. The classic example that you go to user and you go -- or, let's say groups, and in each group, you need to call up all its users, and then all their groups, and then you have all the database in memory.

(00:20:04)

Richard Campbell: And so you end up with query after query to fill in all of that object hierarchy.

Oren Eine: Yes, and even if you don't, even if the ORM is not enough to do it in a minimal amount of queries, you still have 10 times, 100 times more information than you need, than you wanted.

Richard Campbell: Yes. You've loaded a lot more data than you actually needed, and quite often -- I mean, this is where the real battle comes in is, all I did was instantiate the object to get a property out of it, and all that other stuff just happened.

Carl Franklin: So how do you reel it in? How do you get control over that?

Oren Eine: There are ways to handle that. For instance, you can ask NHibernate give me the custom object and all its orders, and you can say give me -- and then it will use the minimal amount of queries, usually one or two, to load the object into memory. And there are more ways you can handle it in... for instance, about two months ago, I added a feature into NHibernate that doesn't exist in Hibernate, which is the ability to batch

several rich statements into the database. For instance, I want to see a page of users; I have 5,000 users, and I want to see the full state and want to see account of them, and I want to do it in a single query. So what I did, I added this function to NHibernate. So what you can do now is say, give me the first ten users, and give me the account of 40 users, and do it in one query.

Richard Campbell: Is NHibernate actually writing the SQL statements for you?

Oren Eine: Yes.

Richard Campbell: So, it's smart enough to know how to combine those things efficiently.

Oren Eine: Yes.

Richard Campbell: Like I am a data guy, so I am going to be looking at profiler and seeing what your nasty ORM is doing to my database. I am just telling you right now, I am going to be the guy banging on your door going, "Don't ever do that again." (Laughing)

Oren Eine: I can tell you that I've seen handwritten SQL that was simply atrocious in so many levels that I can't even begin to...

Richard Campbell: Oh, without a doubt. There's no better way to make ugly SQL than to write it by hand and not know what you're doing. But I guess the strength here, if I am understanding you correctly, is that you guys have really written good quality SQL under the hood to minimize the number of queries.

Oren Eine: Yes. NHibernate gets a lot of its smarts from Hibernate. You've got to realize that in certain databases, for instance, in VB2, if you call a prepared statement and then issue a call, it's not sufficient if you use a... if you don't call up a statement. So this is something that the developer wouldn't know unless they actually, really, really knew well VB2 and have it say how many for you automatically. And taking to -- take, for example, a page in SQL 2005. You have to use common table expression, and you have to do it correctly, and it's at least five lines of code usually. So in 2000, you can't do it at all, you have to select up and everything.

So the ability for NHibernate to know the database, and NHibernate has a set of styles and a set of drivers that it shows how to talk to a database specifically. So it can do a lot more optimization under the hood. For instance, you can go to the config file and, say, when you save to the database, do it in batches, and this uses the Hibernate batch of the current users that are using datasets. So this is something that you



wouldn't have access to; it's internal. No one else can access it except the dataset. And NHibernate did it by hacking the System.Assembly, so it would gain this feature. This is a feature that can bring you 50% more performance.

Richard Campbell : Nice.

Oren Eine: Yes, very nice.

(00:24:41)

Carl Franklin: Have you ever felt envy for the new slick interfaces introduced in Windows Vista? I am sure you want to have something similar in your apps; but unfortunately, that's quite hard to achieve with Windows Forms. There is WPF, of course; but that requires you to adopt a whole new programming model. Wouldn't it be nice if you could have scaling, rotations, animations, alpha blending, complex gradients and all that in classic Windows Forms? How cool would your application be then? Well, it's going to stand out, and it's definitely going to look nice. Stop envying and start delivering great experiences today. Telerik r.a.d. control suite for WinForms offers the first Vista-style controls for Windows Forms.

Pick a Vista piece of UI and try to implement it with the Telerik controls; chances are that you can do it. Join the Telerik WinForms Challenge today to explore the controls in a fun and engaging way. The challenge is a mini-quiz that shows off the unique features of the controls. In just 10 to 15 minutes, you can see how you can make your desktop apps much more appealing, and you can win a product license by simply answering 5 questions correctly. And everybody who completes the challenge is automatically entered into the drawing for the grand prize - get this -- a 50" plasma TV. Check out Telerik r.a.d. controls for WinForms, and join the WinForms Challenge today at www.telerik.com/contest.

Richard Campbell: It makes sense to me, then, that what this really is doing is bringing up the minimum level of data-querying abilities of the average developer that a regular mortal developer doesn't have to know everything there is to know about databases. If they use this, they're going to do respectable querying.

Oren Eine: Yes. And NHibernate has a feature that allows you to say, okay, for this particular class or for this particular query, I don't use SQL, because that is the only way I can get acceptable performance. A bracketed query is the best example, because there is not portable SQL for them. So I can write bracketed queries in SQL using a connect by 4 bracket or, again, common

table expression in this field of 2005 and tell it, here, take this; use the form of the object that you get back, map it to this object and give it back to me; and then I still get all the benefits of using NHibernate, which means that they get the map to persistence, for instance. I get the ability to use an identity map, which means that I don't usually have to worry about concurrency inside the same unit of work.

Carl Franklin: Can you tell us a little bit about the experience of using NHibernate? I mean, this isn't something that's tightly integrated into Visual Studio, is it?

Oren Eine: No. Let us say there are two ways to use NHibernate. The common way is to use NHibernate directly, which means that you have the XML files and you have IntelliSense in the XML files, and you have some sort of a -- if we talk about a Web application, we have the HTTP module that opens in Unit to Work a parent request and closes when the request is ended, and then you just access the decision and load them in the persist object. There isn't support for NHibernate in Visual Studio except the IntelliSense for the XML files.

Carl Franklin: So, really, this is something that you have to do a bit of extra coding that you don't have a lot of help for.

Oren Eine: I wouldn't say so, because you do have a lot of templates for it in code me to have my generation, and I know of at least one person that did a DSL for NHibernate, and there are quite... DSL from the Microsoft designers. So there is a set of designs that you can even drag a table from a ServersPro, and it will create an object for you, it will actually do mapping for you, and it's nice to see that people are doing it.

Carl Franklin: Have you seen any of the integrated tools, like DeclareIt, for example? Have you seen that?

Oren Eine: Yes. I tried DeclareIt. I actually evaluated DeclareIt about two years ago, I think, and it insisted that my object model wasn't formalized. And I said to it, yes, I know it's easy to formalize, I know what I am doing, goodbye.

Carl Franklin: It yelled at you, because it didn't like the way that your data looked.

Oren Eine: Yes, and it didn't let me continue. I have no problem with a tool telling me, hey, you're stupid, stop it. Yes, the tool does it to me all the time; but I have to be able to say, yes, I know what I am doing.

Carl Franklin: Right.



Richard Campbell: And I guess most ORM products tend to insist on a certain structure to your objects.

(00:29:46)

Oren Eine: Not only the objects, also the database, yes. And a lot of the ORM products that I read the documentation about, I haven't actually tried a lot of them. I just read the documentation and say, okay, here's something that I can't live without, even, for instance, I had too many issues for a long time that I don't know if they ever solved it, solved it for the version that they're going to ship. So, okay, interesting, very nice to see that Microsoft is doing the ORM in five different directions; but this is not something that I can use. And I think I have a chat with a Jacob Marnier from the Hibernate team, and their idea about what's extensible in data modalities and what I think was vastly different, because when I need to build an application and... I had a Temporal application, and you know what it is, right?

Richard Campbell: Which application?

Oren Eine: Temporal.

Carl Franklin: Oh, a temporal.

Richard Campbell: A temporal application?

Oren Eine: Yes.

Richard Campbell: Okay.

Carl Franklin: Go ahead and define that for us.

Oren Eine: it's an application where you need to know not only the data, but when did you know the data and when you were supposed to know about the data. So, for instance, the classic example is age application or payroll applications. My salary today is \$1, tomorrow my salary is \$2; but it is effective since yesterday.

Carl Franklin: So, it's in order; you keep track of when things were changed.

Oren Eine: Yes, but not only when things were changed, but when they came into effect, which is a different matter. So, for this, I had to have a very good support in job in service itself for having date times and date ranges capabilities. And we used NHibernate, and what we did is we built a set of custom collections that knew intimately about the logic of traversing the dates. So I could say give me an employee's salary at date and to figure it out; and for all those things from the flexibility points that they have in the

dealings, and later I also evaluated the Add 4 entities that it doesn't -- it just doesn't agree, and we tried the same hood-welded-shut mentality that I see sometimes in Microsoft products.

Richard Campbell: "Hood welded shut," I get it, right; can't get at the engine.

Oren Eine: Yes. And when you get into complex situations, you need to get in the engine, and you need to get into the engine to know what you are doing. And I have no problem with having to do something that they say, look, if you do it, you're on your own. That's fine.

Richard Campbell: Yeah, you're taking your chances if you do this.

Oren Eine: Yes, they will be dropping. Okay, I can handle it; but I need to have the ability to do it, because if I can't, then, okay, I built 80% of my application. That was nice and easy; now, I need to do the complex 20%.

Richard Campbell: Yeah, it's always the exceptions and how they handle that, and it's funny that Microsoft seems to have gotten that right through most of the .NET libraries; but every so often, they still build something that you just can't get at.

Oren Eine: Yes, this is very true. If you take a look at the ASP.NET interrupts, for instance, they are beautiful. They have extension points all over the place. But if you take a look a bit above, then suddenly it becomes black magic again.

Richard Campbell: So I'm sort of building in my head a list of things I need to check whenever I am evaluating an ORM library. Obviously, the many-to-many is one of the key things, how do they handle that. But it's also this extensibility model, how can I make changes to things.

Carl Franklin: That seems to be of utmost importance to me, because that's -- ORM is all about abstraction, it's all about being at a high level, and the classic problem of high-level tools is, like you said, the hood-welded-shut syndrome; something you bump up against a brick wall, you can't tweak it.

Oren Eine: Yes.

Carl Franklin: Oren, how does LINQ fit into the ORM puzzle?

Oren Eine: LINQ doesn't, and it's very interesting, because LINQ isn't actually tied to any ORM.



Carl Franklin: No, it isn't; but, I mean, you have that kind of object-queriability stuff built right in.

Oren Eine: Uh-huh. What LINQ does is actually giving the queriability directly in the language, and this is something that I consider extremely nice to have. It is something that I probably have to implement in NHibernate when I get in projects that use the technologies.

Richard Campbell: So ultimately, LINQ would query NHibernate objects.

(00:35:00)

Oren Eine: Yes, certainly.

Richard Campbell: That makes sense to me.

Oren Eine: If I wouldn't do it, someone else would; but it's certainly that it will help. It isn't that complex.

Carl Franklin: Well, it makes sense. The two are different, right? LINQ is querying that the developer does in the language, and ORM is querying that the tool does below the language to the database. So, they are different.

Oren Eine: The responsibility of the ORM is to bridge the gap between my object and the database.

Carl Franklin: Sure, no. I get it.

Richard Campbell: Right. And I am actually thinking I'd be happier with developers using LINQ against NHibernate objects than I would directly against the database. They have a better chance of not doing harm that way.

Carl Franklin: Or do they? I mean, when you query an object with LINQ, you might have the -- it might trigger the ORM to load the entire database into memory.

Richard Campbell: But I am thinking NHibernate is smart enough that it's not going to do anything really catastrophically dumb.

Oren Eine: I wouldn't bet on it; you can always get a tool to do dumb things.

Richard Campbell: So, it has enough power to shoot your foot off for sure.

Oren Eine: Yes. So LINQ will give you the ability to query NHibernate in a different way. And right now, you have at least two ways to query NHibernate, which is using the Hibernate querying language and using the current API, and there is a tool that they built that gives you

LINQ-like syntax in .NET 2.0, when you can do where.customer.name, equal =4, and you get the results back. And LINQ would simply be another way to query NHibernate.

Richard Campbell: Oren, I know you've worked a great deal on NHibernate, and I know you have a particular project yourself around NHibernate called the Analyzer, which helps folks understand what NHibernate is up to.

Oren Eine: Yes. I originally built it when I didn't understand what NHibernate was doing. So I basically went inside to figure out what I need to do in order to get NHibernate to tell me, and what the driver actually do is composed of two parts. The first part, it lets you run HQL queries and it lets you see what the HQL counterpart is and what the...

Richard Campbell: So, "HQL" is Hibernate Query Language?

Oren Eine: Yes.

Richard Campbell: (Laughing) and that, of course, translates to SQL.

Oren Eine: No, it does not, no. It doesn't translate to SQL, because HQL is an Object Oriented language. Object Oriented Query Language, you can say that; but the idea is that, for instance, I can do a form object and I can do - - and it will get me every single object that was not from the database. It supports for inheritance, it supports for polymorphism; and if you're really crazy, you can do some wild stuff with generate with it that I never comment on, because no one can understand what you're doing then. (laughing)

Richard Campbell: I get that, yeah, over-abstraction.

Carl Franklin: Right.

Richard Campbell: Let's talk about RhinoMocks. I've gotten email from folks saying, "I just love this tool. Could you ask Oren about it?" Maybe we could be -- and this is entirely your project, right, RhinoMocks?

Oren Eine: Yes. This is a project that I wrote because I was reading a book, "*Working Effectively with Legacy Code*", and it's a book it took me nearly two months to finish, because it kept coding with the need to stop reading it.

Richard Campbell: Well, that's like the ultimate compliment to a book is that it makes you want to write code.



Oren Eine: Yes. And it's actually frightening, because I wrote the entire RhinoMocks in, I think, one to two months; and I kept reading the book as I went along.

Richard Campbell: Now, what book is this?

Oren Eine: "Working Effectively with Legacy Code" from Michael Feathers.

Richard Campbell: All right, okay.

Carl Franklin: That sounds like a good .NET Rocks! interview there.

Richard Campbell: Yeah, right there. Write that one down.

Oren Eine: Yes. It's a very fantastic book.

Richard Campbell: Yes. And so RhinoMocks was really built around helping you work with Legacy Code?

Oren Eine: No. Actually, what happened, I was looking on my Analyzer, my NHibernate Query Analyzer, at the time, and I realized that some part of that I wanted to take from the book, I couldn't handle, because the tool wasn't -- I was using NMock at the time, and it wasn't powerful enough to do what I needed. So, I said, okay, I will leave the Analyzer aside for a while, and I will build my own tool. And I did it, and it actually got a lot more attraction than the Analyzer did; and right now, I think it's my favorite Mocking library (laughing).

(00:40:15)

Richard Campbell: I could see why you'd find your Mocking library your favorite.

Oren Eine: Okay, surprisingly.

Carl Franklin: I just want to jump in here and say that I've shrinksterized the link to Amazon.com's listing of that book, "Working Effectively with Legacy Code" at shrinkster.com/mi7.

Richard Campbell: So maybe we should go to the beginning. What's the deal with Mocking? Whether we're talking about NMock or TypeMock or EasyMock or RhinoMocks, why do people want to use these tools?

Carl Franklin: Other than it makes me shudder, because it's more code that I have to write to do nothing.

Oren Eine: Billy Hollis, right?

Carl Franklin: Billy Hollis.

Richard Campbell: Billy Hollis, there you go.

Carl Franklin: Billy Hollis is not a fan of Mock Objects.

Oren Eine: He is not a fan of -- okay, I won't say that. but a Mock Object, before we can understand what a Mock Object is, we need to understand the problem. And the problem is that you were trying to test your code in isolation, and let us say that I have an object that's the coded database. If I want to test this object, I need the database to do this, and if I need the database I need the coded data, and I need the database to be online, and if I am trying to do it in my laptop in the bath, I can't. So another issue is that coding the database costs. It costs several hundred milliseconds. If I'm trying to run several thousand tests, it's starting to get really annoying trying to screen test, test, test, test. So what happened that people said, okay, we will break apart these dependencies, and we will replace it with a Sack one; but when you start to building the Sack Objects, you realize that it's very boring, it's dead code. You write it for a single method or a single test picture, and you're done; and then every time you add something to the original code, you have to go back and change the test code. So people say, this is stupid. We can do it using a tool. So JMock, I think, was the original or a... it started in Java, oddly enough. And in .NET about two years ago, we had NMock, and we had the EMock.net. So the responsibility of these libraries is, you tell them, give me a Mock Object of class 4. And then they hand you back an object, and you can say, expect that the method go to the database will be called with such and such parameters; and when it is called, you should retain these objects. So and I can take this and set up the expectation for the Mock Object appropriately, and then I can call my... call these other tests. And what I've been doing that I know what the call is getting, and I know how it should behave, so I can observe that it behaved correctly.

Richard Campbell: Right. This is the battle that I get into in larger development projects is three or four people all trying to access through to this test database and conflicting with each other or, more relevantly, getting into this battle of what's broken here, was it the database call, or is it the object parameters? Are we not validating correctly?

Oren Eine: Uh-huh. So when you're testing, you want to have a test that when it fails, you know, okay, this test, only such and such class under such and such circumstances can fail; so this has a bug or the test itself has a bug. So you fix it and you say, okay, now can I can run the test again



and I can get the exact same environment, because the environment that you set up for the object is entirely faked. And now, you can test the object in complete isolation, and you say, okay, I have a test that passed for this object under these circumstances, so I know it's not this object. That's a quick summary of it.

Richard Campbell: Yes, it has to be something else.

(00:44:44)

Oren Eine: So the purpose of having Mock library is to make it easy to pass back objects into the object that you are testing, which can be done either using Constructor injection, when the construction has parameters that you pass and then use it later on, or you can pass it to the method directly or you can replace a global -- a single instance or something like that. And what NHibernate does -- what RhinoMocks does is, usually in the Mock libraries, you are working against strings. You said expect once a call rather in quotes the name of the method, and then you can run with the rest of it. What I found that I refactor a lot, and when I refactor, all the tests broke because I changed the method name, and the test called the process method; so it broke. And it got really annoying the 5th time that I had to go and change 17 tests, I think, because I renamed a method. So I decided I want to work against the objects directly. So what RhinoMocks does, you tell it, give me a Mock object of type I4 or you can say, for instance -- because it can mock something, as well. And so it give you an object back, and you say, expect a dot call for Foo.get blah-blah-blah, and then you start to... and when you say "Foo.get, this is the method; you get IntelliSense, you get XML documentation, whatever. This is as far as the environment is concerned, you're calling the method on the object.

Carl Franklin: So you get the benefit of all that written goo without actually having to write the goo.

Oren Eine: Yes.

Richard Campbell: How do you go about creating these Mock Objects in RhinoMocks?

Oren Eine: Black magic.

Carl Franklin: Richard, I think RhinoMocks builds the mock objects from your regular objects.

Richard Campbell: So literally when I specify the call, it creates it on the fly for me.

Oren Eine: Yes. What happen is, you give me a type that you want to mock, I take the type, I look it up, and then what I do in runtime, I build the drive object off this type. So, for instance, if you give me Ni Foo and it's in interface, I build an object that implements its interface, and then I return to you an instance of this object. And when you call a method on this object, the objects know, hey, a method was called; let's tell RhinoMock that it was called. So it does this, and RhinoMocks has a way of going, okay, now I am recording what this object does; so it's setting the expectation list; and later on you say, okay, I finished setting the expectation for this method, let it start replaying the expectations. And from then on, every time that the object comes out OK, I was called with this method with these parameters, it check if this is a valid code; and if it is a valid code, then the expectation is removed, and you can go forward. By the end of the test, you usually call-verify to check that all the expectations were met.

Richard Campbell: I get it, and this is where they come from with the term Expectation-Based Model. All you're doing is defining the expectations.

Oren Eine: Yes, it's all of the codes in Direction-Based testing, because you are testing the interaction of the object, which just repeats collaborators.

Richard Campbell: Right. And what kind of expectations can you set?

Oren Eine: Just about everything, I think. From that, this method should be called with parameters that start with ease of types, start with the string; its property is of this type and is not null in just about everything you can. You can also tell RhinoMocks, look, this is too complex for you to handle, call me when the method is called, and I will tell you if this is the correct method or not.

Richard Campbell: So I can, then, go in and write my own code to say, yeah, that passes or it doesn't.

Carl Franklin: Here's a question for you. When you're building Mock Objects and let's say I want a Mock object of a dataset or something.

Oren Eine: Oh, boy. Yes.

Carl Franklin: Yeah, "oh, boy." Now, are you going to inherit from a dataset object and give me an object?

Oren Eine: Yes.



Carl Franklin: So what if I am actually writing code that's looking at the type and validating the type using Reflection and all that kind of stuff.

Oren Eine: This depends on what you're trying to do, because if you're trying to check the types that you got, it is actually an ease of dataset.

(00:49:55)

Carl Franklin: Well, what if I am just trying to cast it? What if I am trying to cast it, and there's like shadows somewhere, you know, a shadowed method somewhere, and I cast it and I get different behavior based on whether I am using the original, you know, the top level type or derive type?

Oren Eine: Yes. What happen, this is a bit complex, because it can have a virtual shadowed method virtual shadow is viewing the...

Carl Franklin: As if shadowed wasn't abstract and obscure enough, now we have virtual shadow. Wow.

Oren Eine: Yes, and you can give, more complex when you go to the IL level. But what happen is, RhinoMock will overwrite each and every virtual method in the object. In the case of datasets, it basically mean every method, I think. And what happen is, let us say you have a MML in doc, and you try to Mock doc, and doc has a shadowed property that the MML has, as well. So what happens is, if you call into the MML property -- assuming that port is virtual, by the way -- if you called the MML property, RhinoMock will set an expectation for a call on the MML property. And if you call on the doc property, it will set an expectation on doc property. And the reason for this is because on the IL level, there are two separate methods in the method table of the object, and RhinoMock intercept that and it can understand what is happening. Under normal circumstances, for instance, if it's done at the virtual call, then it's not a problem. But I actually get people that even when we start questions on very real scenarios that they're having, this makes for very interesting docs to follow.

Carl Franklin: Yeah, I can imagine (laughing).

Oren Eine: I know more about these internals than I want (laughing).

Richard Campbell: So I know you've gotten some traction from this, because the listeners have asked about it; but how many people are using RhinoMocks, do you know?

Oren Eine: I can't actually tell. I know that there are several thousand downloads, I think, for the

various versions; but right now, I think I released Beta 2 of the RhinoMocks. RhinoMocks 3.0 Beta 2 a week ago and I plan on doing the Betas until I run out of numbers or out of bugs; but I think that I've got around the 150 area adopters for RhinoMocks 3.0 Betas. And I am getting letters of questions. There's quite a few.

Carl Franklin: Wow. That's great. How far do you plan on taking this? Is this like a side hobby now versus what you're doing full-time?

Oren Eine: No, this is side hobby. RhinoMocks is a free Open Source product. So I just do it because I am usually working on very -- RhinoMocks is probably the clearest code that we ever produce.

Carl Franklin: Wow.

Richard Campbell: Ah, I see. Well, it's got a very specific domain to focus on, and it looks like you've got a real vision around it. So it's probably pretty tight.

Oren Eine: Yes, except that I have to deal with some very funky serialization of stuff. Where it stops, I cannot know (laughing); but I had recently added a generic method support to RhinoMocks, and I started getting some very, very weird bugs back. What happen, when you have a generic method that call to a generic type and return it. So the actual type isn't known in runtime, and even then, the meter would lie to you about the time that you get. So, that was a fun week (laughing).

Richard Campbell: Yeah. You're inside the space of objects analyzing objects and deciding who they are trying to impersonate, that can get pretty weird.

Carl Franklin: I was going to say the weirdness factor is just going to get more and more pronounced, and do you ever foresee it just becoming too complex to continue being a hobby?

Oren Eine: I actually got several requests about commercial in RhinoMocks, and people want commercial support for it. I may be able to provide commercial support for it; I don't think that I will commercial the library, because frankly I don't -- the amount of investment that's needed to produce a commercial library is too much for me to want to deal with. And there are commercial libraries such as TypeMock that exist in this space, and TypeMock is actually mocked off of the RhinoMocks if you want to mock just about everything.

(00:55:15)



Richard Campbell: Wow.

Carl Franklin: TypeMock, you said?

Oren Eine: TypeMock, yes. Basically, what TypeMock does, it uses the profiler APIs to intercept each and every call that are made under the applications; so it can actually mock everything. RhinoMocks work by overwriting every method of property that it can. So it can mock interfaces, it can mock classes or ops classes; but if you have an un-virtual method, it can't mock that. There is a property that won't allow it.

Carl Franklin: I am sorry. That's at TypeMock.com.

Oren Eine: Yes. A little PS product.

Carl Franklin: So what's next for your products here? What's next for NHibernate, and what's next for RhinoMocks?

Oren Eine: RhinoMocks is going to have a final 3.2 release in I hope the next month or so, depending on how many weird conditions people put it under.

Richard Campbell: for people doing generic calls to generics, you mean?

Oren Eine: Yes, very, very ugly, freaky stuff. Even speaking about someone who implemented the -- implemented compiler and also queries using generics. Yes, this is very, very freaky stuff.

Richard Campbell: U g h . Well, the correct answer is "Don't do that" (laughing).

Carl Franklin: But, you know they're going to.

Oren Eine: Yes, but some people get it a lot, quick response. Don't do that (laughing).

Carl Franklin: If it can be done, they will do it, and they will want your tool to work with it.

Oren Eine: Yes (laughing).

Carl Franklin: Or else I am going to give my money to someone else.

Oren Eine: Uh-huh. And I actually got some very interesting response because of that. Why aren't you supporting that, because this is very freak scenario? Here's an extension point, go ahead, have fun with it. This is literally what I did when several people came and say I want to do this or that, and I said I don't think that is the appropriate thing for RhinoMocks to do. So why don't you go

ahead? Here is a way for you to hook into the RhinoMocks internal. I built a demo for them to see how it can be done. It's working perfectly. They're so happy; I don't get bugged (laughing).

Richard Campbell: Yeah, you don't have to write it.

Carl Franklin: Yeah.

Oren Eine: And about NHibernate, we have the 1.2 release coming I hope very soon. We are now in the -- you will call it RC for some reason it's confidential for some reason, Sergey redeveloped our quality candidate before release -- no idea why -- and the general availability release is the next release, so to speak. It's probably a month or two away.

Carl Franklin: Excellent.

Oren Eine: Yes, but of course, we have all the products, which then we can decide will we release it tomorrow or release it in six months. No promises made about release dates.

Richard Campbell: So I know you're coming to DevTeach in Montreal in May. Any other conferences we could see you at?

Oren Eine: Actually, no. I recently did a methods academy in Israel. We had roughly 2,500 people, and I gave to talk about the vision of control, which is a very hot topic to talk about, apparently (laughing); and then beyond that, just a feature, I think. A friend of mine is trying to build a user group here in Israel. There are several already existing user groups, and a friend of mine is trying to build a new one, and I'll probably help him with it. That is just about it (laughing).

Richard Campbell: Okay. And what are you going to be talking about at DevTeach?

Oren Eine: I am going to give three talks. One about RhinoMocks, one about advance usages of vision of control, and if I don't just introducing vision of control is hard; but to say how I can handle some advanced usages, and I am going to talk about MonoRail, which is a...

Richard Campbell: Oh, really?

Oren Eine: Yes.

Richard Campbell: Yeah, we didn't mention MonoRail at all so far.

Oren Eine: A fantastic framework, simply a joy to work with.

(00:59:55)



Richard Campbell: Really?

Oren Eine: Yes.

Carl Franklin: You know Ted Neward is going to be there at DevTeach. Perhaps we could convince you and Ted to go head-to-head on ORM on a panel or something.

Oren Eine: Okay, but they need to have it first.

Richard Campbell: Because I don't know if you know this, but Ted is not a big fan of ORM.

Oren Eine: Really? You couldn't tell (laughing).

Carl Franklin: But it will be great to actually pull apart some of these points, and it would certainly be great for the attendees to see the fur fly.

Oren Eine: That would be interesting discussion.

Carl Franklin: Yeah, maybe we could make that happen. The website is Ayende.com, and the website for NHibernate is NHibernate.org. It's also at SourceForge.net/projects/NHibernate. Oren Eine, thank you very much for being on the show and explaining your product and your work. It's very much appreciated.

Oren Eine: Thank you very much for having me.

Carl Franklin: Okay, and we'll see you next week on .NET Rocks!

(Music)

Carl Franklin: .NET Rocks! is recorded and produced by PWOP Productions -providing professional audio, audio mastering, video, post production and podcasting services, online at www.pwop.com.

.NET Rocks! is a production of FranklinsNet - 'Training Developers to Work Smarter' and offering custom onsite classes in Microsoft development technology with expert developers, online at www.franklins.net.

For more .NET Rocks! episodes and to subscribe to the podcast feeds, go to our Website at www.dotnetrocks.com.

(Music)