Volume 3 Issue 4

PLM WORLD™

BACK TO INDEX

PLM WORLD NEWS

- Letter from PLM World Chairperson
- CAD/CAM/CAE Student Competition
- News from the Drafting and Dimensional Management Technical Committee

UNIGRAPHICS

- NX 3 Supports More Languages!!

I-DEAS

- Bolted Connections
- NX Manager I-DEAS via Teamcenter Engineering
- First of Two Simple Applications of Using Part Equations in I-DEAS

TEAMCENTER

- NX Manager I-DEAS via Teamcenter Engineering
- Repeatable Digital Validation: Teamcenter's integrated solution for digital product validation

THIS EDITION

- Volume 3 No 4 PDF Version (731 Kb)

PREVIOUS EDITIONS

- Volume 3 No 1
- Volume 3 No 2
- Volume 3 No 3

WEBSITE LINKS

- Become a PLM World Citizen
- PLM World Member Conferencing Area
- ICCON Legacy Data
- PLM World Citizens Area

NX 3 Supports More Languages!!

by Elango Ramanathan Programming tools – Technical track chair

Grrrrrrr! Yes! That is exactly how I felt when I heard UGS is adding more languages. Oh my God, GRIP, GRIP NC, C, C++, TCL, Intent! and now C#, VB.net. None of the existing language covers everything I need. We have been asking UGS to added more coverage in CAM area. What is wrong with these UGS people? Why are they wasting their resources in adding more languages? I was so curious when I attended beta testing at Cypress CA, I could not resist asking for a presentation on what was happening. But the more I found out about UGS' strategy, the more impressed I became.

Initially I thought UGS was just going to add a few more language to their list. But UGS has a well thought out strategy to help developers in the long run. UGS is providing a new recording tool called Journaling, and, by NX 3, will support VB.NET, C# and C++ for traditional automation. By NX 4 Java will be supported also. Any developer working with the NX Open for .NET API can use all the functionalities of the Microsoft® .NET environment. NX 3 will support full object oriented programming concepts. Also, with the .NET API a developer can do remote and distributed programming, as well as create Graphical User Interfaces (GUIs) using WinForms.

Journaling is more like macro recording in Word or Excel. Journals will record VB.NET commands instead of menu clicks. They can also record in C++, but journal replay is limited to VB.NET only. Since journals are capturing functional code, and are not recording GUI or screen picks, they will work across multiple versions of NX. And journaling supported automation code provides the added benefit of recording actual automation commands. In the past, I used to make a lot of calls to GTAC to find out whether a function was available in Open C and how it worked. Now, with journals, it is so easy to find out. I can just record a journal and I get my sample program in no time. I can cut and paste directly into an automation program, then add only a few variables and logics to make the program work.

My next worry was that I would have to learn yet another new language. I never had a need to fill my brain with one more language. But I saw it as opportunity to beat my son. Since he already knows C#, I borrowed his C# book and started to go through it. I was really surprised to know C# is more like C++ than what I had originally thought. A lot of the syntax is very similar to C++ except for arrays and the fact that there are no pointers. There are differences like delegates etc., but I felt comfortable with C# very fast. And, since I could add the NX .NET classes in the Visual Studio development environment, I got all the benefits of 'intellisense'. Wow!

Software Tips

NX Open for .NET supports not only internal/external programs but now it supports remote programming as well. Remoting allows an NX user to execute an automation program from the same or a different machine from where the NX session is running. Via remoting, a user can connect to another system running NX within a network. This addition is available through the .NET version of the NX Open API only. This mode is not available in the legacy Open C API.

Okay! The future is very bright, but what about the past? I have about 150K lines of code to support. As usual, UGS is committed to supporting their customers and has promised to continuing to support legacy Open C (User Function) libraries and GRIP. But no new functions will be added to Open C and GRIP. All the new functions will be added to the NX Open for .NET API and to the new NX Open C++ API. Both the .NET API and the new C++ API are derived from the same Common API kernel, so any new functionality will be provided equally for the new APIs. Customers with existing Open C licenses will automatically receive the new Open C++ API free of charge. As for my existing code, I recompiled and linked my code with the NX 3 Open C library. Everything worked fine except for few minor changes like the unit conversion function in expression is inch(....) instead of in(....).

Not all NX functionality is available for automation through the native.NET API yet. To provide complete programming coverage in .NET, UGS is providing an additional library of .NET-wrapped Open C functions. This added library provides a .NET programmer with nearly the same automation access to NX functionality that the existing Open C and Open C++ libraries provide. Although UGS has plans to provide all existing and new functionality through the .NET API in future releases, the .NET-wrapped Open C library will not be removed.

UGS is continuing to support C-based programming by releasing a new C++ library built from the same object and class structure of the new .NET library. Although this library is different than the existing Open C++ library, I can compile and link my existing C and C++ programs with this library. But if I want to use any of the Microsoft .NET classes I have to follow a book full of protocols and conventions. I tried and failed miserably. So I have decided to stick with C# because I want to program with .NET classes.

NX Open for .NET comes with a new set of documents that are presented in a standard Microsoft help file format. The documents are user friendly and more like Visual Studio docs. And details about the NX Open for .NET classes and methods show up automatically in the Visual Studio Object Browser. It should be noted, though, that a user needs to be running Visual Studio .NET 2003 (Version 7.1) if they want to work with the .NET API. They also need to have the Microsoft .NET Framework 1.1 loaded on their workstation

I need to buy a new NX Open for .NET authoring license if I want to program in C# or VB.net, but no additional execution license is required. No new licenses are required if I program in C++ and link with the new library.

Over all I am quite impressed with UGS' strategy to support more object oriented programming languages. In the long run, I believe this new direction will help developers a lot.

Elango Ramanathan
Technical Track Chair

BACK TO INDEX