

j-Interop Value Proposition

j-Interop implements the Microsoft© DCOM protocol in pure java. This protocol forms the backbone for all Windows© COM operations performed in a distributed environment. Since j-Interop does not carry any platform specific native code it can be effortlessly integrated into an existing java application and will run on any operating system where java VMs are available. This enables java based non-Windows systems to directly interact with native COM applications from their hosted platforms.

The source code of j-Interop is available freely and is open to review and comment by thousands of developers across the globe. They, put together, have use-cases which by far exceed our own test framework. The library has been successfully deployed on hardware platforms and operating systems which no single enterprise could ever accumulate. It undergoes rigorous quality control on such heterogeneous environment each time a release is made. Furthermore j-Interop is a continuously evolving product, closely guided by the user community. This increases applicability of it to a wide number of opportunities which surface in a constantly dynamic market. More information on its advantages can be obtained from <http://www.j-interop.org/advantages.html>.

We believe that this cross-platform connectivity provided by j-Interop and its open-source foundation guarantees flexibility for your current and future IT strategies.

About Our Offerings

Level 1

“Breaking-the-ice”

It is quite natural for anyone to feel a certain degree of hesitation and risk in adopting a new technology into their product or solution. We understand this predicament and assist you sail past this critical phase by making you feel at home with j-Interop.

1. Technology evaluation and adoption

We enable you to understand the technology behind the library through active communication (by email, phone or VoIP) and provide consultancy to carry out preliminary investigations into the feasibility of using the library as a solution. We try to offer best advise to you on how to proceed even if it means, at times, not promoting our products.

2. Rapid prototyping

"Action speaks louder than Analysis" (we changed the original one a bit). At times, it may be required to go beyond simple samples and implement a few use cases which functionally address the issues encountered by you. This exercise aids in covering the core capabilities of the library and also provides you with sufficient hands-on experience. We assist you in building prototypes rapidly, leaving adequate time to consider alternate options.

Level 2

"j-Interop is the way to go!"

1. Aid or undertake architecture and/or design of solutions based on j-Interop

If the feasibility analysis confirms successful evaluation of the library, we often assist the users with architecture and design of solutions based on j-Interop. Through the years, we have covered a lot of ground in Windows COM and COM based systems, this proves quite invaluable during our involvement in this aspect of software life cycle.

2. Aid or undertake software development over j-Interop

Independently, we also assist you in the next logical step of software development. Over the last few years, we have been actively involved with customers from different domains. We do not claim to be experts in those fields, but working with them has definitely provided us with unique insights into the niche areas where j-Interop can be applied.

3. Review and optimize solutions built over j-Interop

We also help clients by reviewing and suggesting optimizations to their solutions built on j-Interop; advising them in areas such as performance enhancements, the "right-way" usage, refactor for a "better fit", feature show stoppers etc.

4. Aid or undertake porting of solutions based on other libraries to j-Interop

We have experience with other native solution\technologies and alternate products. We help you port from one of these alternatives to j-Interop. Naturally, we take utmost care that no "level-downs" or "unfortunate" loss of functionality happens due to this porting.

Level 3

"Could we do this with the library?"

1. Customize, Enhance j-Interop for special requirements of the user

At times we have been approached to enhance the library, to add certain features to it which the customer requires for a j-Interop based solution and we have gladly done so. After due testing these changes are incorporated into the public repository.

2. Reverse Engineer existing COM Products for a compatible solutions

By far the most challenging work for us is in the area of reverse engineering an existing COM server and determining its IDL. This requires a lot of traffic analysis and in depth knowledge of COM data structures. It is used mostly by clients trying to build specialized solution over custom COM Servers (such as Microsoft SQL Server Debugger) for which the IDL is not published on purpose.

Level 4

"Thank You, but stay with us..."

1. Provide premium support

We provide yearly and hourly support (only during business hours) to our existing customer which covers all the work we have done so far (and will do in the duration when the contract is in place). This guarantees a turn around time for the issues reported with little or no cap on the number of issues which can be reported (as long as they are classified\diagnosed as "bugs" and not enhancements). These contracts also provide a commitment to availability in case of critical issues (or during customer requested time frames such as major releases) for longer period of time or on stand-by from the j-Interop development team.

About Dimentrix

Product and Engineering Services organization founded with following intents -

1. Professionally support, enhance and maintain its flagship product j-Interop.
2. Assure quality, responsiveness and commitment to the open source foundation of its product lines with a focus on community driven development.
3. Enabling customers to realize the full potential of their ideas using our high-end engineering services through an open, transparent and accountable engagement model.