

From: "Beth Steinbock (Kelly Services Inc)" <a-beths@microsoft.com>  
Sent: Monday, November 22, 1999 4:18 PM  
To: Phil Shoemaker/HQ/3Com  
Subject: VSIP information



VSIP Q&A.doc



Visual Studio  
Integrators SDK...

<<...>>

Phillip,

Thank you for returning the NDA. Attached you'll find the additional information you've requested on the Visual Studio Integrator Program. If you are interested in joining the VSIP please email vsip@microsoft.com requesting the Visual Studio Integration and Distribution Agreement. Thanks for your continued interest.

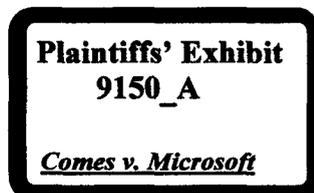
Sincerely,  
Beth Steinbock

<<VSIP Q&A.doc>> <<Visual Studio Integrators SDK.htm>>

<<...>> <<VSIP Q&A.doc>>

<<...>> <<Visual Studio Integrators SDK.htm>>

PALM-000071

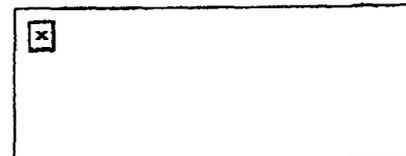


CONFIDENTIAL

MS-CCPMDL 000000153955

## Visual Studio Integrators SDK

White paper Rev 1.0



The Microsoft® Visual Studio® (VS) Integrators SDK is available to all VS Integrator Partners. The VS Integrators SDK is a collection of SDKs that include these and more:

- [Visual Studio Environment SDK](#)
- [VS Debugging SDK](#)
- [HH2.0 SDK - Extending VS7 Help](#)
- [DDS SDK](#)

The various SDKs are created by our individual product and feature teams to enable advanced integration in the Visual Studio Integrated Development Environment (IDE).

The VS Integrator SDK is the way for companies to get the tightest level of integration with Visual Studio as possible.

### Overview of Extensibility in Visual Studio

In Visual Studio 7.0 (VS7), developers will have all the components necessary to write tools, add-ins, and wizards that complement those tools already offered in Visual Studio. This makes Visual Studio more extensible than ever before. Each Microsoft product that ships in VS7 will expose some form of extensibility to our customers. Each subsequent feature within Visual Studio will expose some level of extensibility, too. Visual Studio 7.0 will set the standard for delivering an open and flexible, integrated development environment as well as providing the complete set of development tools it has in the past.

There are two ways to extend Visual Studio: 1) consuming the *automation model* by providing add-ins, wizards or macros; and 2) creating *VS Packages* that create new project types, add a new designers and editors, or add a language service. Each of these approaches is valid. The decision to choose one or the other is driven by your need to enhance the design-time experience of the user of Visual Studio.

The two approaches are represented by two SDKs: the public [Visual Studio SDK](#), available to anyone who purchases Visual Studio; and the more tightly held [Visual Studio Integrators SDK](#), available to anyone in the Visual Studio Integrator Program.

### Visual Studio SDK

The Visual Studio SDK is public program, targeted primarily at the add-in market. It contains reference for the VS7 automation model, macro recording, a collection of SDKs that ship with VS7 and white papers/technical articles.

## Visual Studio Integrators SDK

The VS Integrators SDK is designed for companies that want even tighter integration into the VS7 platform than what is offered in the VS7 box. The VS Integrator SDK is a way for companies to get the tightest level of integration with VS7 as possible. It is a contractual agreement between any company and Microsoft for the use of our advanced Interfaces.



## Visual Studio Environment SDK

The Visual Studio Environment SDK supplies tools and documentation that enable you to use and extend the Microsoft Visual Studio IDE through VSPackages. A VSPackage is the term for an installable piece of software that extends the Visual Studio environment by contributing one or more of the following: user interface, services, project types, or editor/designer.

The Visual Studio IDE, also known as the environment, provides the user interface (UI) to the traditional components found in IDEs: compilers, editors, and debuggers. The Visual Studio Environment can host a wide variety of extensions. With the Environment SDK, you can create a simple tool that adds a pane to the output window or an application as complex as Microsoft Visual InterDev. To extend the environment with this SDK you create a VSPackage.

The Visual Studio IDE also provides wizards, macros, and add-ins that help you develop applications for Microsoft® Windows®-based desktop platforms, without the burden of building or assembling the required infrastructure for distributed applications. See Automation model for more information.

The Visual Studio Environment SDK enables these advanced integration scenarios over the public automation model:

- Create a project type (like a Visual C++ or Visual Basic project)
- Create editors and designers (mdi child windows)
- Proffer information to the property browser (required for project types)
- Perform advanced text buffer manipulations and other language services such as creating red squiggles

## VSPackages

A VSPackage is a Component Object Model (COM) object that implements an interface called IVsPackage. A VSPackage is registered in such a way that the Visual Studio Environment knows it exists and understands the functionality the VSPackage provides. In general, VSPackages

require more advanced programming than writing an add-in that calls the automation model interfaces. Most of the VSPackage-level interfaces do not implement IDispatch, and thus you must create VSPackages using tools that support COM, such as Visual C++.

A VSPackage gives you greater flexibility than add-ins that use the Automation model. With VSPackages, you can accomplish a wide range of tasks, including building new project types, adding new designers or editors, and adding new tool windows. In fact, the products you currently see in the Environment, such as Visual J++ 6.0 and Visual InterDev 6.0, are VSPackages (or a set of VSPackages) that use the same interfaces documented by the Environment SDK.

The Visual Studio Environment exposes nearly all its basic user interface elements for customization, automation, or augmentation. You can extend these basic UI features with VSPackages.

## VS Debugging SDK

As a Visual Studio Integrator, you can integrate with the VS debugger in two ways: custom languages and run times.

The Active debugger SDK provides debugging support for languages and run times of integrators within the Visual Studio environment. An integrator writes components called debug engines that act as the debugger's backend for a particular language or run time. The debug engines can do the following:

- Allow expression evaluation for the watch and locals windows
- Enable source-level debugging, even for repositories that are not file-based
- Show local variables, threads and stack frames
- Set and handle breakpoints
- Manage multiple threads of execution
- Debug applications over the network.

The Visual Studio debugger and your debug engine communicate through an event model implemented using DCOM. Events, both synchronous and asynchronous, are expressed as COM objects that describe the event.

### Hooks for Debug Events

The Active Debugger SDK contains an interface for an event hook that allows integrators to monitor and intercept all debugging events. The events may be consumed or passed on at the discretion of a software component that uses the event hook interface.

This feature enables a partner application to interact with breakpoints that they have set in a running program, independent from the VS

PALM-000074

CONFIDENTIAL

debugger.

## User Interface Hooks by Datatype

The Visual Studio SDK also provides hooks that create property viewers and UI elements for specific data types such as hash tables and record sets. Partners can extend the Visual Studio user interface with custom property grids or even graphical views.

## HH2.0 SDK - Extending VS7 Help

This section provides an overview of Help extensibility mechanisms in Visual Studio 7.0.

### Integration with Visual Studio Documentation

The online documentation architecture in Visual Studio 7.0 will allow VARs to fully integrate their online documentation with that of Visual Studio. These documents will be fully integrated into the Visual Studio Help experience, including F1, table of contents, index, and full-text search navigation.

### Viewer Integration

Like Visual Studio 6.0, Visual Studio 7.0 will allow users to view their online documentation in an external viewer. However, users will also have the option of viewing the Help content directly within the Visual Studio shell (similar to Microsoft® Visual C++® 5.0). Both the integrated mode and the external viewer will provide improved usability over all previous versions.

### Attribute-based Filtering

One of the most exciting enhancements in the new Help system is attribute-based filtering. Each Help topic can be tagged with attribute name/value pairs (e.g., Development Language=VB) so that authors can categorize the information they are producing. At run time, users can select from a list of predefined filters based on those attributes to reduce the scope of the information presented. Users can also modify and create filters to fully customize their Help experience.

Attribute filters reduce the items listed in the table of contents, index, and full-text search. Unlike the subset feature in Visual Studio 6.0, filtered keywords in the index are removed, rather than being grayed out. A separate filter can also be set to apply to F1 lookups.

### Authoring

Authoring online documentation for integration with Visual Studio is very similar to authoring for HTML Help 1.x. A simple conversion tool will be provided to convert existing HTML Help 1.x sources into a format compatible with Visual Studio 7.0 documentation. A new Visual Help Workshop (hosted in the Visual Studio shell) will be provided to facilitate documentation project management, TOC and index authoring, and compilation.

## DDS SDK

DDS is a generic diagramming surface. DDS is designed to make application development with a diagramming interface as easy as possible, while not limiting the complexity of the application.

With DDS you can develop a complex diagrammatic UI to the data that your application needs to manipulate or display. DDS will supply you with a great deal of functionality with very little implementation cost.

By using DDS, you can focus on building your application and not have to worry about designing or implementing a UI. With DDS you can easily build complex representations of interconnected data. It will allow your user to manipulate the data through the diagram. DDS will also give you all of the standard UI functionality for free (Zooming, Scrolling, Drag-Drop, Printing, Clipboard, etc). On top of standard UIs, DDS also provides a set of layouts for your diagram. You can use one of our existing layouts or build your own using our extensible layout engine.

## Proven Technology

DDS has been shipping since Visual Studio 97. We are currently developing our third release with even better functionality. DDS has been accepted as the design surface of choice by more than 10 shipping products and has shipped with almost every Developer Tool and Microsoft® BackOffice® offering.

If your application shows or manipulates complex data, DDS can add value to your application and please your customers even more.

## Splash Screen

The Visual Studio Splash Screen - interfaces will enable a VSIP to add its own company logo onto the splash screen. After your product is installed, the splash screen gets updated with your logo. Additionally, the Visual Studio "About box" is fully extensible. You can add your company information to it.

## Visual Studio Integration Details

The interfaces that ship in the VSI SDK are fully supported by Product Support Services. As a customer of Microsoft, you have many options for support. Microsoft's support ranges from a per-incident to a premier basis. These plans give you access to the key people that can help troubleshoot problems and provide assistance.

**Will these interfaces ever become public?**

Yes, there is a good chance that the technologies included in the VS Integrators SDK may be available to the public. It is our goal to eventually have our Automation model support all extensibility in Visual Studio.

**Is there a Logo Program?**

The Visual Studio team is currently looking into building a logo program around the VS Integrators program. We currently plan to have some level of testing required by Integrators to ensure your product offerings do not break any Visual Studio or Microsoft tools.

**Will there be UNIX version of Visual Studio?**

There are no plans to provide a Unix version of Visual Studio.

Last Edit: 07/09/99

PALM-000077

CONFIDENTIAL