

## **The Need for Program Development Flexibility In Machine Vision Application Software**

Machine vision, the process of visually inspecting a component or assembly for defects and/or measurement criteria, has become an accepted and often critical part of today's manufacturing process. Demands for increased quality and lower product cost - coupled with ever increasing manufacturing throughput requirements - have made machine vision systems a vital component for meeting these increasingly stringent demands.

Machine vision inspection applications, by their nature, are very diverse - covering an extremely wide assortment of applications. These applications can range from performing robotic guidance and positioning, inspecting date codes on pharmaceuticals, and reading bar codes on electronic assemblies, to inspecting pin-stamped letters on automotive parts, verifying print registration on product packaging, and checking for correct colors on consumer goods.

Such a diverse application range requires an equally diverse machine vision software package as the foundation for a visual inspection system. The software must not only include the many software tools and vision algorithms needed to do a wide array of inspection tasks, but also allow for flexibility in the way the package is programmed and implemented. In addition, since the program developers using the software package can range from novices to experienced integrators and code-based programmers, it must also provide suitable program development levels catering to each programmer's skill and expertise.

### **Application Diversity Requires Software Flexibility**

As inspection applications have become more and more demanding, increased pressure has been placed on machine vision developers (OEMs, integrators, and end-users) to produce inspection systems that are faster, more robust, and that can be quickly designed, implemented, and integrated into the line. Unfortunately, since each installation may have an entirely different set of application and integration requirements, it is difficult for developers to know exactly what each new application will entail. In some cases developers may need to rapidly create graphical programs so that they can perform quick feasibility studies, while in other cases they may need to develop customized high-speed programs entirely at the code level. And in still other cases they may need to create custom vision tools and algorithms to address certain application-specific tasks. In order to address all of these issues, a developer needs a machine vision software package that not only provides the ability to choose the programming environment that best suits the application requirements, but also allows the easy creation of custom vision tools and custom applications. Most importantly, the developer needs this level of software flexibility and sophistication while also remaining easy to use.

### **Development Environments**

Most machine vision software packages provide a single development environment in which programs and vision tools are created. One common type is the *graphical* environment, in which a mouse is used to create a program by selecting icons, drawing, pointing and clicking, or using a similar technique. The key advantage of this type of environment is that it allows programs to be generated quickly, and since there is no code writing involved, it can be performed by developers who do not have coding experience. The key disadvantage of this environment is that flexibility is limited - especially if a great deal of customization is required or if application-specific vision tools must be developed.

Another common development environment is the *code-based* type, in which programs and tools are created by writing lines of code. Common languages used for this technique include C, Microsoft® Visual Basic®, Visual C++®, and others. The key advantage of this type of environment is flexibility and portability, since writing at the code level allows everything to be

completely customized, and the written code can be used by other compatible software packages (provided that they are written in the same language). The key disadvantage of this environment is that it requires code knowledge and takes longer to develop programs and tools due to the code writing process.

Each of these programming environments has its merits, and in general work well when the application requirements match the advantages that each has to offer. However, since machine vision applications can differ vastly from one another, a machine vision software package that provides only one development environment can quickly become unsuitable due to the limits and disadvantages imposed by that environment.

To address this problem, developers often revert to using both graphical and code-based software packages – switching between them depending on which is best suited the task at hand. Unfortunately dealing with two different packages becomes a complex compromise, since the developer must become very proficient with both software packages, deal with two different software vendors, and most importantly – support both packages in the field. This, at best, is difficult to manage and even harder to maintain.

### **The Development Solution**

The ideal solution is to use a machine vision software package that includes multiple development levels in one integrated solution. This allows the development level to be selected based on the application requirements, the amount of time available, and the expertise of the program developer. It also allows the programs to be migrated to different development environments within the same software package when the application requirements or programming expertise change. One such machine vision package that offers this integrated solution is DT Vision Foundry<sup>®</sup>. This unique architecture provides both graphical and code-level programming environments, as well as a large suite of vision tools - allowing developers to select the environment they need for each application and to standardize on one integrated solution for all of their application needs.

DT Vision Foundry provides three different development levels:

**1. Rapid Program Development** – DT Vision Foundry can be used as an out-of-the-box stand-alone Machine Vision application. An extensive library of vision and image-processing tools is provided, and the “Point and Click” Script Tool is used to quickly build comprehensive ready-to-run test programs – all without writing code. Or, if desired, a Custom Script Tool is available to write macros in a Basic-like language.

Ideal uses for this level include developing quick programs for performing feasibility studies prior to investing additional time and effort into an application. This level also allows the use of customized tools developed in the Custom Tool Development level (described below) for Point and Click implementation. And since there is no coding necessary, developers without code experience can use this level for developing both prototype as well as production-ready programs.

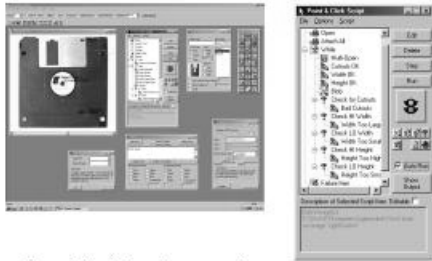
**2. Custom Application Development** – Using Microsoft Visual C++, DT Vision Foundry supplies Object-Oriented API (Application Programming Interface) calls to allow comprehensive, customized vision applications to be developed.

Ideal uses for this level include creating compact high-speed applications, custom applications, as well as developing applications requiring custom user interfaces.

**3. Custom Tool Development** – DT Vision Foundry’s tools can be modified and new custom vision tools created using Microsoft’s Visual C++ and DT Vision Foundry’s underlying API function calls. Custom graphical user interfaces can also be added using Microsoft Foundation Classes (MFC).

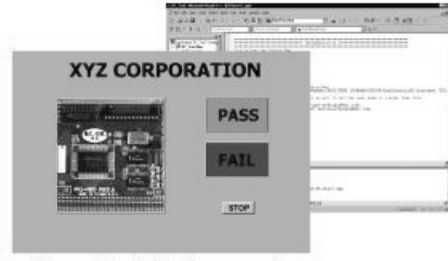
Ideal uses for this level include using the provided DT Vision Foundry Tool API calls to create custom vision tools. In addition, tools created in this level can be added to the Rapid Program Development level, allowing them to be used in a Point & Click graphical environment.

**Rapid Program Development**



- Graphical Environment
- Point and Click Scripting
- Over 30 Tools Provided
- Quickly Generate Ready-to-Run Programs

**Custom Application Development**



- VC++ Code Environment
- Use DTVF Application API Calls
- Create Custom Application Interfaces
- Create Fast, Compact VC++ Programs



**Custom Tool Development**



- VC++ Code Environment
- Use DTVF Tool API Calls
- Create Custom Vision Algorithms
- Create Custom Tool GUI Interfaces
- Use New Tools in the Rapid Program Development Environment for Point and Click Implementation

DT Vision Foundry leverages the advantages of each development level to address a wide range of application requirements. And since the DT Vision Foundry API function calls are the same as those used by the tools, identical performance is guaranteed when moving from the graphical level to code development levels.

**Summary**

No longer do developers need to use multiple machine vision software packages to address their varied application requirements. DT Vision Foundry, with its integrated development levels,

provides the flexibility needed to best address the wide range of applications that developers encounter. DT Vision Foundry is the ideal solution for machine vision OEMs, integrators and end-users that require the features of a graphically-based rapid program development system and the flexibility of a code-level development system – all in a single, completely integrated package.

**Data Translation**

Machine Vision Products  
100 Locke Drive  
Marlboro, MA 01752 USA

(800) 525-8528

[www.datatranslation.com](http://www.datatranslation.com)