

# Introduction to

MicrolImages SDK  
is FREE with  
every TNT product.

# Using TNTsdk



with the  
**TNT Products**

---

## Before Getting Started

This booklet introduces the use of the TNT Software Development Kit (TNTsdk) with the TNT products from MicroImages, Inc. TNTsdk is a software development kit for computers equipped with a full TNTmips license (not TNTview or TNTedit) to create custom processes for TNTmips, TNTedit, or TNTview. TNTsdk provides programming libraries of several thousand C functions and hundreds of C++ classes. Your development project will involve writing C++ programs that invoke the MicroImages library functions to present the user interface, perform geodata manipulation, and present the process results.

**Prerequisite Skills** The TNTsdk is not for everyone. The programmer who uses TNTsdk should already know how to program. This booklet contains no instruction on installing and using your OS, Programming in C++, or compiling programs. Your project team should include at least one expert C++ programmer. The C++ programmer working with the TNTsdk must have access to certain basic support and reference works. MicroImages recommends the printed manuals published by O'Reilly & Associates.

This booklet also assumes that your project team has at least one expert in the type of geospatial application you are trying to develop. The GIS specialist who works on your project should already know exactly how your geodata will be processed and analyzed.

**Resources Needed** TNT products version 7.0 or above. C++ programming tools and environment. Compilers: For Windows: Visual Studio .NET; For Linux, Unix, and Mac OS X: the current version of GCC. You must also have a reliable internet connection to access the online documentation.

**More Documentation** Documentation for all the functions in the MicroImages programming libraries is generated from comments in the C++ header files. Keep an online link open to the TNTsdk documentation, which can be opened from

<http://www.microimages.com/product/tntsdk.htm>

*Keith Ghormley, 30 December 2004*

It may be difficult to identify the important points in some illustrations without a color copy of this booklet. You can print or read this booklet in color from MicroImages' Web site. The Web site is also your source for the newest tutorial booklets on other topics. You can download an installation guide, sample data, and the latest version of TNTlite:

**<http://www.microimages.com>**

---

## Plan Your Project

You can write one set of C++ code using the X, LessTif and MicroImages library functions and then compile it once on each machine where it will be used. MicroImages has maintained a design goal that TNTmips (and thus your customized processes also) be completely portable across all platforms.

**Project Concept** Define your project as narrowly, specifically, and simply as possible. Avoid ambitious "do-everything" projects, which take much longer to develop, and which are much more difficult to create and maintain. Focus on the end user (a specialist? a novice? computer literate?). Consider the type of interface you need (start with the simplest, least complex design possible). Ask what kinds constraints you need (how do you prevent the user from skipping necessary inputs or making out-of-bounds choices that will produce useless results?).

**Project Team:** As you begin your project, give careful thought to the composition of your project team. Your team should include a **user representative**, who is typical of the specialist who will actually sit at the computer and use the application. Only such a user can ask the obvious questions that often don't occur to the programmer and designer.

Another team member is the **GIS design supervisor**. This is the expert who understands the input data, the processing needs, and the purpose of the project. The design supervisor must keep asking the large questions, "What are we doing?", "How will we do it?", and "How will we apply the results?"

Finally, your team needs a good C++ **programmer**. The programmer works with the user and the designer to make the computer show them what they want to see, and do the work they want it to do. (And then gladly changes everything when they change their minds.)

This booklet assumes that you are an accomplished programmer who is comfortable with C++. If you do not already understand how to create, debug, and compile C++ programs, then **TNTsdk is not for you.**

Careful planning in the early stages saves much time and work later on. Assemble a good project team and ask lots of questions.

## License, Protection, Distribution

The TNT products Software License Key protects your custom process and adds both security and flexibility.

The TNT products Software License Key will be required to run your new application. Routines in the MicroImages software libraries periodically check for the presence of the key in order to insure that the user has a legitimate product license. Putting your custom process under the TNT products Software License Key has several advantages:

1. Protects your proprietary algorithms and processing models. You need never fear that the marketplace advantage that your special process gives you will find its way out the door to a competitor's shop.
2. Offers complete cross-platform flexibility. Your process will work on any platform in your organization, so many departments and users can share the benefit.
3. Your new process can work seamlessly with all the existing TNTmips processes. You don't have to build a full-featured system from the ground up; you can use any number of integrated TNT processes.
4. The speed of a compiled process saves time over the same work done with SML. You gain a speed advantage especially for computationally complex processing.
5. Gives you a choice of distribution strategies. You can add your process to any existing TNT products menu, for use in a shop where users are already familiar with TNTmips.

Or you can create a direct stand-alone process. Your one-task end user is never required to find a path through the many menus and features of the entire TNTmips product.

## Prototype with SML

For the initial development of your project concept, use SML for testing and proof of concept. SML development is simpler, so initially it will prove to be faster and easier. You can quickly write a section of code, see how it runs with known sample data, examine the results, and make modifications. Since SML is interpreted, not compiled, you don't have to go through the extra steps involved in compiling a complete executable program.

**Proof of concept.** An SML script will show you quickly the kind of process results you can expect. Give your SML prototype script some sample data and examine the results carefully. Does your processing model give you accurate and useful results?

**User Interface.** SML also offers a quick way to develop dialogs, menus, messages, and other interface elements. Let the user representative on your team make suggestions on ways to make the design clearer and easier to use.

**New Feature Requests.** SML prototyping can give you a quick sense of what new library functions to seek from MicroImages, Inc. If you find yourself working through an awkward set of steps again and again, it may be possible to accomplish what you need with a single, generalized function. Talk with the software engineers at MicroImages and describe the function you want. The MicroImages software engineers may be able to suggest another way to accomplish what you want with existing functions, or MicroImages may be able to add the new function you need to the TNTsdk libraries.

For help with SML, refer to the tutorial booklets from MicroImages:

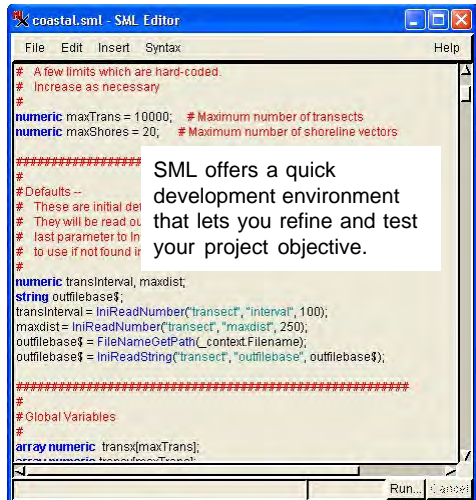
*Writing Scripts in SML*

and

*Building Dialogs in SML*

It is also instructive to examine the large library of sample SML scripts online at

[www.microimages.com/sml/](http://www.microimages.com/sml/)



# Getting Support

First: keep current with the latest TNTsdk

Second: work with the MI Software Support Engineers and Software Development Engineers.

On-line documentation for the TNTsdk is generated automatically by Doxygen. (See page 8.)

As your development team works on your project, you will want to take advantage of the support that MicroImages can provide.

**Step One.** Keep current with the latest patches. Develop the regular habit of keeping up with the latest version of the TNTsdk. The libraries are updated every week along with the patches to the TNT products Release Version (RV) and Development Version (DV). Get the latest TNTsdk from

**[www.microimages.com/product/tntsdk.htm](http://www.microimages.com/product/tntsdk.htm)**

**Step Two.** Work with the MicroImages Support staff. Of course, MicroImages will not design your project for you, or write the programming code that you need, but you can benefit greatly from the support that is available.

Begin by contacting the MicroImages technical support team. They can help you with many non-programming issues.

For support in programming, the technical support team will put you in touch with one or more of the MicroImages Software Engineers, depending on the type of project you have. You can make steady progress by email. You can include sections of program code for the MicroImages Software Engineers to examine. They may be able to suggest improvements or new approaches. You will do well to take advantage of the years of expertise they have in working with the MicroImages software libraries.

Email communication with the MI Software Engineers is also the best way to determine the need for new or modified library functions. Only after the MI Software Engineers understand the need and purpose of a new function, class, or member, will they be able to modify the TNTsdk libraries accordingly.

# Installation and Setup

Download the TNTsdk from the TNTsdk product page on the MicroImages web site. Links to the current version are provided on the page:

**[www.microimages.com/product/tntsdk.htm](http://www.microimages.com/product/tntsdk.htm)**

Unzip the file that you download. You can select a target directory so that TNTsdk fits logically with the existing elements of your programming environment.

Modify your programming environment to include the TNTsdk directories.

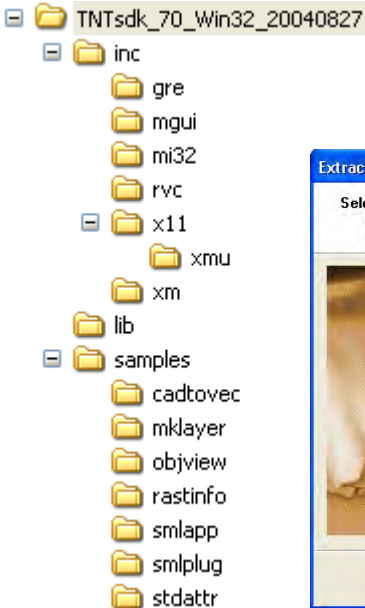
The two main parts of the TNTsdk are the /inc and /lib directories. Also note the /samples directory, which provides working programs for study.



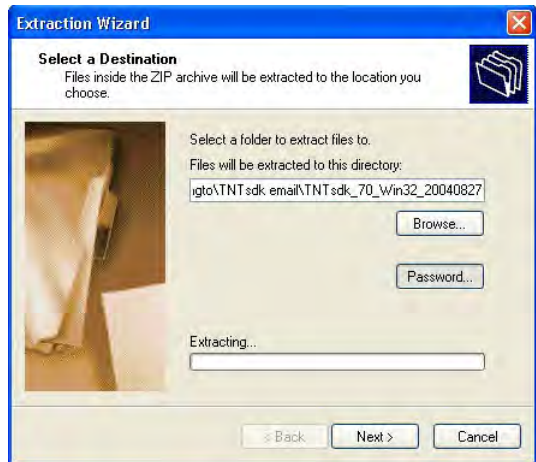
TNTsdk\_70\_Win32\_20040827.zip

Download the sdk file from [www.microimages.com/product/tntsdk.htm](http://www.microimages.com/product/tntsdk.htm)

When you unzip the TNTsdk file, it creates the directories illustrated:



**NOTE:** The TNTsdk for V6.9 will not work with V7.0, and the V7.0 TNTsdk will not work with V6.9. The change from Motif to LessTif prevents cross-version library calls.



# TNTsdk Online Documentation

Use the online documentation to learn the purpose and use of the functions and structures in the TNTsdk. The better you know the contents of the documentation, the more quickly you will be able to develop a clean and direct project solution.

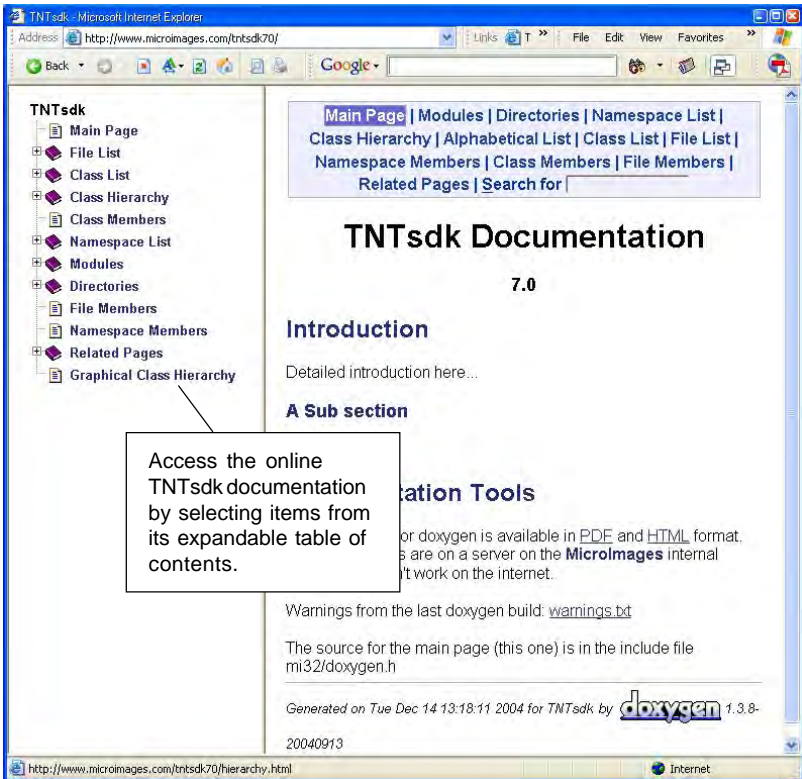
Documentation is available from the TNTsdk product page on the MicroImages website:

**[www.microimages.com/product/tntsdk.htm](http://www.microimages.com/product/tntsdk.htm)**

Documentation includes full definitions and samples. If you try to access and get a “Document not Found” error, you have probably come during a one of our regular site updates, so just try again later.

Spend some time browsing the online documentation so that you become completely familiar with how it is organized, and how to navigate in it.

Beginning with V6.9, documentation for TNTsdk is generated by Doxygen ([www.doxygen.org](http://www.doxygen.org)).





## Sample Programs

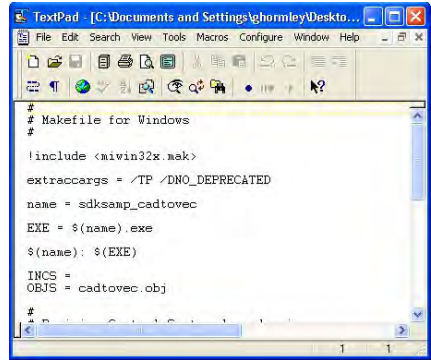
Sample programs provided with TNTsdk are included for learning and testing. You should examine the source code in each program:

1. Look at the program structure. Be sure you can follow the logic and flow of the code.
2. Open each include file (\*.h) to see what it provides.
3. Use the online documentation to examine the classes, functions, and other elements that are used.
4. Look up the documentation for the functions that the program calls.

After you are familiar with the content, structure, and function of a sample program, compile it and then run the resulting executable file. As you use the features of the sample program, look back at the program code in order to understand how the code produces the program behavior.

Sample programs include:

cadtovec.c	convert a CAD object to a vector object
mklayer.c	sample for Mdisp layer creation from a "fixed" file and object
objview.c	demonstrate use of object display functions. This program allows the user to view one or more spatial data layers
rastinfo.c	very simple application allowing user to select a raster and display basic information about it
smlapp.c	(obsolete - use smlplug.c) an older method for extending SML functions
smlplug.c	a better method for extending SML functions by creating plugin modules to be called from SML scripts
stdattr.c	computes "standard attributes" for CAD/TIN/Vector objects



A complete makefile is provided for each of the sample programs. You will have to modify the makefile according to the way you have set up your programming environment.

Consider using sections of code from the sample programs in your own programs. You may be able to use the code directly, or with only slight modification.

## cadtovec.c

Much of the task in even a simple program involves knowing what libraries to include, what data structures to use, and what functions to call. Learn to search and browse the TNTsdk online documentation.

Open cadtovec.c in your editor and examine the first part of the program.

- Always begin with a comment block (lines 1-3)
- Note the comment definitions with each #include (lines 7-15). The include statements indicate the directory and filename of the include file in the form  
#include <mgui/appmain.h>

Be sure that your compiler environment is configured so that the compiler can find the TNTsdk directories. In this example, the full path might be c:\programming\tntsdk\inc\mgui\appmain.h

- Examine the CADTOVECDATA class definition in line 19. Notice that the ctrl.h #include in line 13 is provided to support the use of that class. Consult the online TNTsdk documentation for MGUI::CTRL.

```

/**
 * CADTOVEC - Convert a CAD object into a vector object
 **/

#define RCSID "$Id: cadtovec.c,v 1.5 2004/07/06 22:30:38 scowan Exp $"

#include <mgui/appmain.h>
#include <rvcl/dialogs.h>
#include <rvcl/combine.h>
#include <mi32/mstatus.h>
#include <mi32/simplear.h>
#include <mgui/dlgshell.h>
#include <mgui/ctrl.h>
#include <mgui/formobj1.h>
#include <mgui/combobox.h>

// For RVC::DlgGetObjectSet()
// For RVC::COMBINE::ExtractToVector()
// For STATUSDIALOG and STATUSCONTEXT
// For SIMPLE_ARRAY<>
// For MGUI::DLGSHELL
// For MGUI::CTRL_TOGGLEBUTTON
// For MGUI::FORM_OBJECT_LIST<T>
// For MGUI::FORM_COMBOBOX

namespace {

class CADTOVECDATA : public MGUI::DLGSHELL {
public:
    CADTOVECDATA (
        TEXTID Title,
        HELPID HelpID
    ) :
        DLGSHELL(Title, HelpID)
    {}

    ~CADTOVECDATA () {}

private:
    MGUI::FORM_OBJECT_LIST<CADTOVECDATA> m_ObjectList;
    MGUI::CTRL_TOGGLEBUTTON m_Optimize;
    MGUI::FORM_COMBOBOX m_TopType;
    RVC::GEOMETRIC::GETOBJFILTER_ISNOTEMPTY m_IsEmptyFilter;

    virtual void v_OnApply ();
}

```

Look for MGUI::CTRL in the TNTsdk online documentation to see the inheritance for MGUI::CTRL\_TOGGLEBUTTON. (see next page...)

# MGUI::CTRL in cadtovec.c

Find the TNTsdk online documentation for the MGUI::CTRL class reference and examine its inheritance diagram and detailed description.

## Detailed Description

Base class for "single" graphical user interface controls.

There are two basic types of 'controls' which can be used in user interface implementation, controls which the user can interact with and 'static' controls which the user cannot change directly.

All controls which the user can manipulate also encapsulate information about the current 'state' of the control (except for MGUI::CTRL\_PUSHBUTTON as explained below). There are three ways to use each of these controls in a program (in order of convenience):

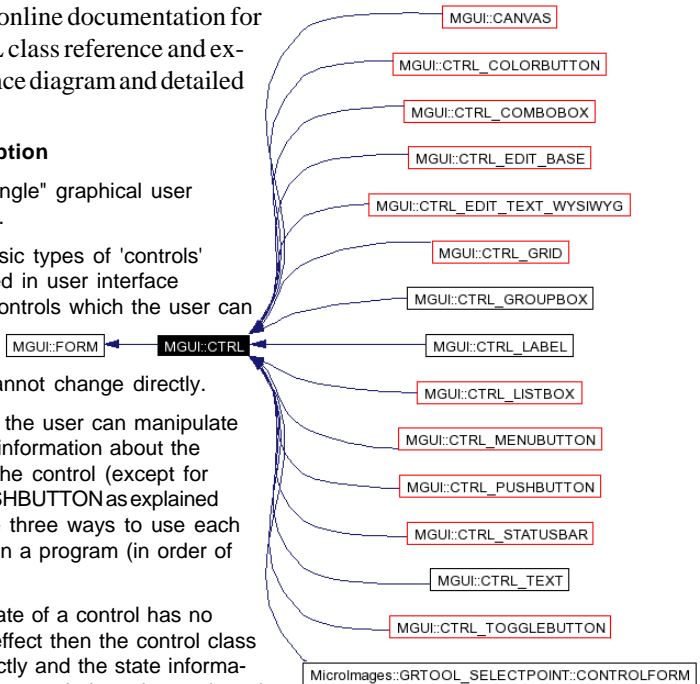
If changing the state of a control has no other immediate effect then the control class may be used directly and the state information retrieved when needed, such as when the user presses 'OK' on the dialog. This option is not available for MGUI::CTRL\_PUSHBUTTON as pressing a button must always have some other affect and MGUI::CTRL\_PUSHBUTTON contains no 'state' information.

Use the template version of the control and specify a callback 'container' and callback method pointer for the control. This allows the callback to be a method of the 'container' class (usually the dialog, page or main window) thus allowing access to other controls in the container. This is similar to MFC message maps and avoids the common need to include a reference or pointer to the container in a subclass. Templates for controls with a single On... virtual method will generally include the container and method pointer as parameters to the Create() methods. Templates for controls with multiple On... virtual methods will generally have a SetFunc... method for each corresponding On... virtual method in their base class and will not override the Create() methods.

Inherit from the control class and override the On... virtual method to perform task specific processing. This is simplest if no reference to the containing dialog or window is necessary. Also for some control types, the template version itself may be inherited from to override other virtual methods, such as MGUI::CTRL\_EDIT ... validation methods.

Definition at line 227 of file ctrl.h.

Click on the link to go to the definition in ctrl.h



## Localization

If your language does not have a current localization for the TNT products, contact MicroImages about becoming an Official Translator for your language.

[www.microimages.com/i18n/](http://www.microimages.com/i18n/)

Creating a custom reference file is not a trivial programming task. Refer to the online documentation for the `TEXTRESClass` Reference. Your sdk program will need to load your resource file with `TEXTRES::AddFile()`. MGUI controls have a "name" parameter which is looked up in the reference file, and the localized string from the reference file is used.

MicroImages has carefully designed the TNT products so that they can be used in the language and script of any locale in the world.

Existing locales are defined by a locale resource file (`textres.xxx`, where `xxx` is the language code, such as `textres.enu` for U.S. English, `textres.rus` for Russian, or `textres.ell` for Greek). MicroImages software engineers put all interface text into the language resource files and never put interface text directly into the program code. The interface files can thus be maintained and translated easily.

If your project includes the use of standard TNT interface components, then your users will be able to select an existing locale and see the components that depend on the `textres` file in their language.

There are two ways to add localized text to the interface for your own programs.

1. If your intended user base is confined to a known group in your own company or organization, the easiest thing to do is to put the interface text directly into your source code. This practice makes the text harder to maintain, and prevents it being translated for other languages, but it is straightforward and easy.
2. Create your own external reference file. Use text string variables in the code, and put all your actual interface text into an external reference file.

Note: It is generally not practical to add your text variables and interface text to MicroImages' `textres` files. Frequent updates and revisions make coordinating such an effort impractical.

**<http://www.microimages.com/documentation/cplates/69textlocalization.pdf>**

---

## Platform-specific Issues

### Windows

Use Visual Studio .NET from Microsoft Corporation. Visual Studio offers a complete development environment and supports integration with XML. Refer to the Visual Studio home page:

<http://msdn.microsoft.com/vstudio/>

### LINUX, UNIX

Use the most recent version of GCC (the GNU Compiler Collection), which is published by the Free Software Foundation, Inc. The GCC release and documentation is available from the GCC home page:

<http://gcc.gnu.org>

Pre-compiled versions of GCC are also available for some platforms.

### Mac OS X

Mac OS X also uses the GCC compiler. Apple's developer tool Xcode (and previously, Project Builder) uses GCC under the hood for building executable images from source code. The GNU Debugger (GDB), a companion to GCC, comprises the foundation of the Xcode debugger. Apple's version of GCC is based on standard GCC releases and adds features that support Mac OS X.

You can download the GCC compiler and other tools for Mac OS X at the Developer Connection on Apple's website:

<http://developer.apple.com>

Look elsewhere on Apple's site for other GCC documentation, support, and information, such as

[http://developer.apple.com/releasesnotes/  
DeveloperTools/GCC3.html](http://developer.apple.com/releasesnotes/DeveloperTools/GCC3.html)

the GNU Compiler Collection (GCC) from the Free Software Foundation is a free software project that has been used for many years to create software for UNIX and other platforms.

Since GCC is an open source project it depends on its developer community for enhancements and fixes. Developers are encouraged to participate so that GCC can continue to evolve to support new languages, new processors, better optimization techniques, and so on.

# How To Build with Visual Studio

The makefiles that are included with the samples in the SDK will need to be modified for your specific environment before you build with them. You must consider how you have set up your programming environment, and how you have installed the TNTsdk. The differences between platforms and operating systems make giving specific details difficult. For example, for VisualStudio, you need to define the following compiler values and flags:

```
-DWIN32 -DWINNT -DNT_INTEL -DNOMINMAX -DVC7 /G5 /GF /GX /Gs  
-D_MT -D_DLL /Zc:forScope -DUSE_MGUI -DX_NATIVE  
-DXTSTRINGDEFINES -DXMSTRINGDEFINES  
-DXmHTML_STRINGDEFINES -DNeedFunctionPrototypes
```

Also, add the TNTsdk /inc folder to the INCLUDE environment path (or equivalent).

Libraries to link (Windows names shown; similar names for other platforms):

erdashfa.lib	zlib.lib
tiff.lib	lcms.lib
png.lib	libmix.lib
compress.lib	xlib.lib
freetype13.lib	xt.lib
misystem.lib	xmhtml.lib
geom.lib	msvcprt.lib (Windows only)
rvcsys.lib	

---

## A Note on Deprecated Functions

There are many functions or classes in the SDK that are old and remain only to support older code in the TNT products that will eventually be rewritten. Functions that are DEPRECATED should not be used in any new code, since these functions will eventually be removed.

In addition, there are a number of old functions that should be avoided even though they are not yet marked at "deprecated." These groups of functions can usually be recognized by the first few letters of the function name:

Old name	New class
Mf	RVC
Mx	MGUI
Mgd	MGD
Mie	MIE (starting in V7.1)

## Frequently Asked Questions

### **Can you show me some sample code that does \_\_\_\_?**

Maybe. MicroImages wants to support you in your development efforts. Our software engineers may be able to show you how to accomplish your purpose from similar program components in the TNT products. Be sure you also carefully examine the sample programs included with the TNTsdk (see page 9). However there are limits to how much help the MicroImages software engineers can give you. We are not able to write your code for you or teach you how to program.

### **Do I have to rename the sample programs from .c to .cpp?**

Renaming the file to end in .cpp will work. If you use Visual Studio, you can also add the /TP flag to the compiler options. This tells the compiler to compile as C++ even though the file ends in .c

### **What are these compiler warnings about "deprecated functions"?**

There are many functions or classes in the SDK that are old and remain only to support older code in the TNT products that will eventually be rewritten. Functions that are DEPRECATED should not be used in any new code, since these functions will eventually be removed.

### **Will my TNTsdk application developed with the current libraries work with an older TNTmips license?**

In most cases, probably not. MicroImages does not guarantee that an application built with a later version of TNTsdk will work on an earlier TNTmips license.

### **How do I add my SDK functions and/or classes to SML?**

Look at the sample program smlplug.c. It shows how to extend SML functions by creating plugin modules to be called from SML scripts. You just create a DLL containing your code, drop it into the right directory, and SML will automatically make it available.

### **Can TNTsdk use XML for GUI elements?**

Yes. Visual Studio supports integration with XML. XML extensions for GCC are also available.

### **Can I create a TNTsdk application that uses "TNTmips" functions and run it on a platform that is licensed only for TNTview?**

Yes. Anything in the SDK is available for any application you create. However, this is not as helpful as it might seem. Many processes that are not in TNTview, for example, have little in the SDK. Thus things like the editor and Hyperspectral analysis have little in the SDK that would make it easy to add such capabilities to a TNTview license. You would end up writing nearly the entire application, duplicating the code in TNTmips. But of course, you are free to do so if you wish.

# Advanced Software for Geospatial Analysis

MicroImages, Inc. publishes a complete line of professional software for advanced geospatial data visualization, analysis, and publishing. Contact us or visit our web site for detailed product information.

**TNTmips** TNTmips is a professional system for fully integrated GIS, image analysis, CAD, TIN, desktop cartography, and geospatial database management.

**TNTedit** TNTedit provides interactive tools to create, georeference, and edit vector, image, CAD, TIN, and relational database project materials in a wide variety of formats.

**TNTview** TNTview has the same powerful display features as TNTmips and is perfect for those who do not need the technical processing and preparation features of TNTmips.

**TNTatlas** TNTatlas lets you publish and distribute your spatial project materials on CD-ROM at low cost. TNTatlas CDs can be used on any popular computing platform.

**TNTserver** TNTserver lets you publish TNTatlases on the Internet or on your intranet. Navigate through geodata atlases with your web browser and the TNTclient Java applet.

**TNTlite** TNTlite is a free version of TNTmips for students and professionals with small projects. You can download TNTlite from MicroImages' web site, or you can order TNTlite on CD-ROM.

## Index

building .....	14	online documentation .....	8, 10, 11
C++ programming tools .....	2	patching .....	6
cadtovec.c .....	10	prerequisite skills .....	2
CADTOVECDATA class .....	10	programmer .....	2, 3
compiler flags .....	14	programming environment .....	7
compilers .....	2	project concept .....	3
cross-platform support .....	4	project team .....	2, 3
deprecated functions .....	14, 15	proprietary protection .....	4
design supervisor .....	3	prototype with SML .....	5
documentation .....	2, 8	sample code .....	15
download TNTsdk .....	7	sample programs .....	9
FAQ .....	15	SML .....	5, 15
GCC .....	13	Software License Key .....	4
GIS specialist .....	2	support .....	6
inheritance diagram .....	11	textres file .....	12
installation .....	7	TEXTRES::AddFile() .....	12
LessTif .....	3	updates .....	6
libraries to link .....	14	user representative .....	3
Linux, UNIX .....	13	V6.9 and OSF/Motif .....	7
localization .....	12	VisualStudio .....	13
Mac OS X .....	13	Windows .....	13
makefile .....	9	X .....	3
MGUI::CTRL .....	10, 11	XML .....	15



**MicroImages, Inc.**

Voice: (402)477-9554

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