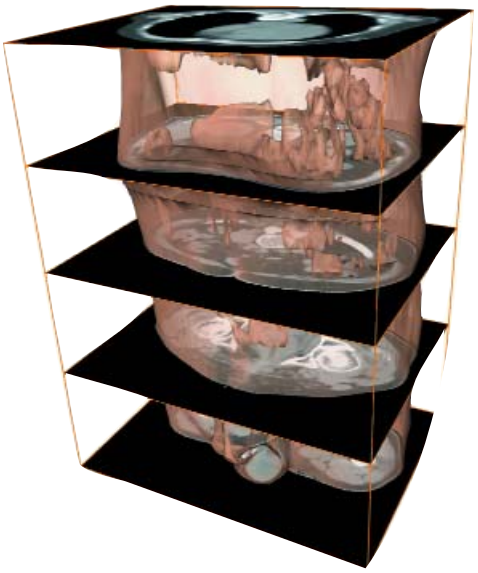
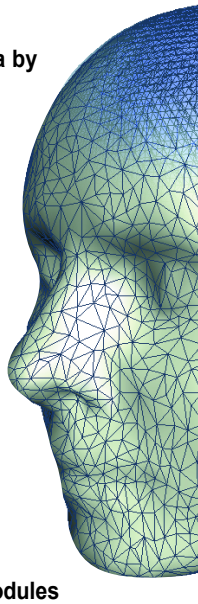
amiraDev™

The Ideal Development Platform for 3D Visualization



amira® is an advanced tool for 3D visualization, data analysis and geometry reconstruction. The amira Developer Edition – amiraDev™ for short - allows you to further extend the capabilities of the amira system. This makes amira the ideal development and research platform for 3D data processing and visualization tasks.

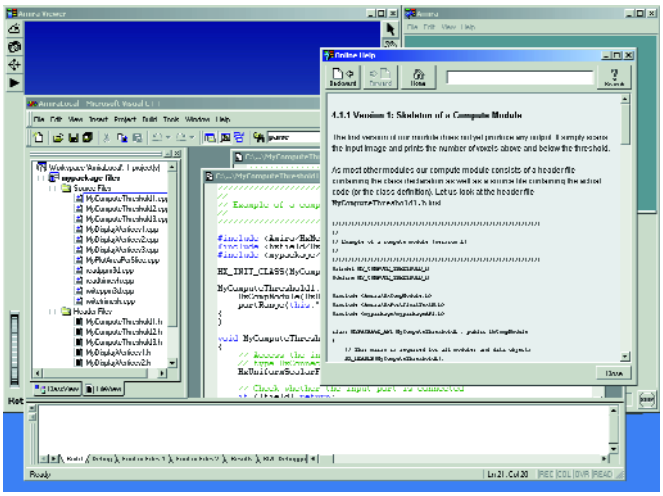


amiraDev allows the user to add new read or write routines, new modules for visualizing data or new modules for processing data by C++ programming. The amira development wizard automatically creates a ready-to-compile skeleton for your new components. Due to amira's object-oriented design, even existing modules operate on newly defined data classes. The graphical user interface for your new modules is generated automatically in a consistent and platform-independent way.

Converting existing code into amira components is very easy. Due to the Open Inventor™ scene graph layer, even complex 3D visualization modules can be created in surprisingly short time.

The most complete 3D visualization workshop on the market

Based on a number of industry standard libraries, such as Open Inventor, OpenGL®, Qt™, and Tcl, amiraDev is delivered with the Open Inventor from Mercury graphic toolkit (SDK version). The unique object-oriented interface provides maximum flexibility. It makes it as easy as never before to add custom I/O-routines, compute and visualization modules. Writing new visualization modules for amiraDev essentially means creating an Open Inventor scene from the input data. If you already have code doing this, it will be straightforward to turn it into an amira module. Open Inventor is a C++ library using OpenGL which provides an object- oriented scene description layer. The Open Inventor library is a complete C++ developer suite for creating enhanced 2D/3D charting, high-end 2D/3D representations on 2D & 3D meshes.



Application Areas

Medical Visualization and Simulation
- Development of systems for treatment planning and diagnosis
- Add custom feature extractors for geometry reconstruction

Microscopy and Biology
- Add special-purpose analysis tools, e.g. in functional imaging experiments
- Add special-purpose filters for data processing

Education
- Use as platform for teaching in computer science classes
- Environment for diploma or master theses

3D Data Visualization and Simulation
- Write custom I/O routines to make your every-day's work easier
- Create your own integrated simulation and visualization environment by turning your collection of standalone programs into Amira modules

Virtual Reality, Immersive Applications
- Add support for custom input/output devices
- Development of new 3D interaction techniques

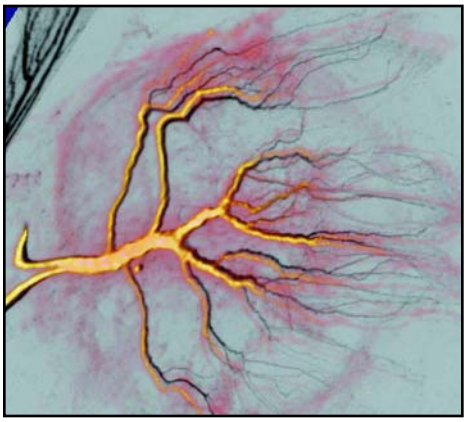
Industry, Industrial Tomography
- Add special-purpose tools for data analysis

Computer Systems, Inc.
MERCURY

```
// Demo for simple point reader.
// Little error checking is done in this example !

#include <Amira/HxMessage.h> //For output in Amira console
#include <hxcluster/HxCluster.h> //Data class for pointclouds
#include <mypackage/mypackageAPI.h>

MYPACKAGE_API int readsimplepoints(const char* filename)
{
    FILE* f = fopen(filename, "r"); // Open the file
    if (!f) {
       theMsg->ioError(filename);
       return 0; // Indicate error
    }

    // Skip header (first line).
    // We could do some checking here.
    char buf[80];
    fgets(buf, 80, f);

    // Create Amira data object.
    HxCluster* cluster = new HxCluster();

    // Read points.
    while (!feof(f)) {
        McVec3f p;
        fscanf(f,"%f %f %f",&p[0],&p[1],&p[2]);
        int idx = cluster->points.append(p);
        cluster->ids.append(idx);
    }
    fclose(f); // Close the file.

    // Register data object to make it visible in the object pool.
    HxData::registerData(cluster, filename);
    return 1; // Indicate success.
}
```

Extending amira

Modules, data classes, and I/O routines are organized in packages. Each package is a shared library (DLL), which can be loaded at run-time. In order to extend amira, the development wizard can be used to create C++ template.



Example: Adding a Reader

As an example, we will write a reader that reads clusters of points from a file. The example file format shall consist of a header-line followed by lines with three coordinates in each line. Each line shall correspond to one point in space, like this

```
# Example Point File
2.3 0 123.1
12 10 -6.7
33 80 20
```

Implementing the reader is quite easy, you have to complete the template with:

- Open the file
- Create an amira data object
- Read the data into the data object
- Register the data object

To make the new file format known to amira, a command must be added to a resource file:

```
dataFile -name "Example Point Format" \
-load "readsimplepoints" \
-header "Example Point File" \
-package "mypackage"
```

The header option lets amira automatically determine the file format.

As you see in the code example, the implementation is straightforward. The core part is adding the data to the amira data class. This, of course, is specific to the kind of data you are reading.

Data File Formats

amira supports a wide variety of data file formats for data input or output, including:

TIFF (r/w 2D,3D), PGM/PPM (r/w), SGI-RGB (r/w), Dicom (r), ACR-Nema (r), Iris Explorer (r,SGI), HyperMesh (r/w), Leica LSM data, Binary Raw Data (r/w), IEEEIO (r/w), Open- Inventor (r/w), UCD (r), STL (w), PLY (r/w), HyperMesh (r/w), DXF (r,SGI), OBJ (r/w, SGI), Fluent/UNS (w)

Other file formats can be easily added by writing new I/O routines using amiraDev.

About amira®

amira is a 3D visualization and modeling system originally developed at the Department for Scientific Visualization of ZIB (Zuse Institute Berlin, Germany). Indeed - Visual Concepts GmbH, a spin-off from ZIB, continued to develop and maintain the product, based on the Open Inventor toolkit from Mercury. In 1999 Indeed and Mercury entered into a partnership that made Mercury the exclusive world-wide distributor for amira. Since then the two companies have worked together to make amira one of the most efficient and best-selling visualization solutions in the world.

Mercury Corporate Background

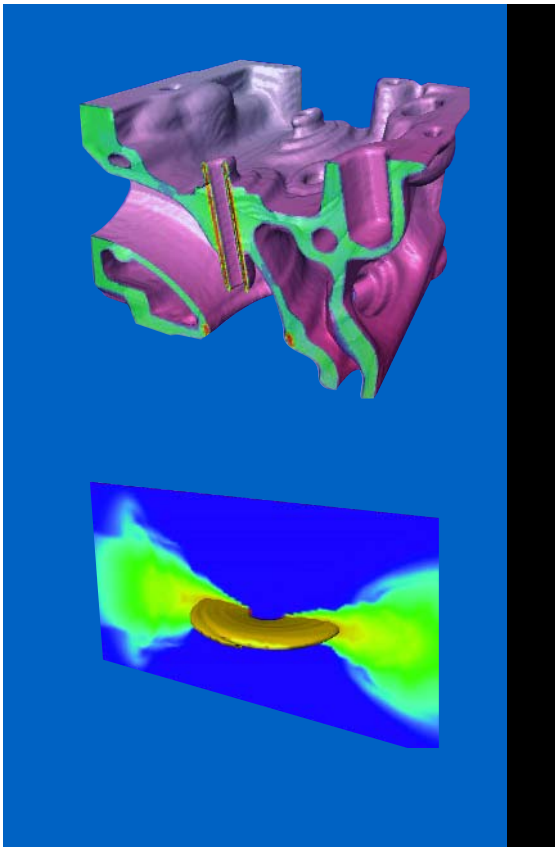
For over 20 years, Mercury has been providing open systems-based 3D graphics software tools for application developers and end users.

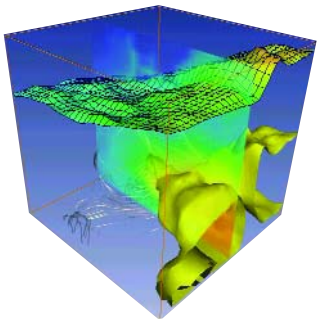
Mercury offers standards-based cross-platform graphics and is the leading independent software supplier of graphics software tools and utilities.

Technical Information

More information and additional resources are available on the web at www.mc.com/tgs.

Look up frequently asked questions, application examples, customer spotlights, technical papers, and much more.





Computer Systems, Inc.
MERCURY

Europe, Middle East, Africa and Asia
Mercury Computer Systems SA
Parc Activités Kennedy 1, BP 227
F-33708 Mérignac Cedex - FRANCE
+33 (0) 556 13 37 77
3D_info_europe@mc.com

USA, Canada, South America
Mercury Computer Systems, Inc.
199 Riverneck Road
Chelmsford, MA 01824 - USA
+1 (978) 967-1300
3D_info@mc.com

TGS-DEV-3002-1202-6M