



# MIMIK

U.S. Patent No. 7,280,955

## Application Suite 3.0

### **EXTREMELY** rapid test tool development

Cut the development time of custom test tools by a factor of **5 to 20**, enabling your organization to:

- Compress program budgets and schedules
- Decrease time to market
- Increase customer satisfaction

### Finally, a protocol emulator at the **application layer...**

Generally, the Mimik Application Suite applies to the development of embedded systems, though many exceptions to this rule can be found.

Mimik can stimulate or monitor any system that:

1. Communicates with other computers, systems or devices in a binary format (many text formats as well)
2. Carries your application's messages using one or more of the following protocols
  - TCP
  - UDP
  - MIL-STD-1553
  - CAN/J1939
  - GPIB
  - Serial Ports

### **RAPIDLY DEVELOP TEST TOOLS FOR THE SYSTEMS YOUR ORGANIZATION DEVELOPS**

These test tools can stimulate their system-under-test with incoming messages, and automatically respond to its outgoing messages. Alternatively, these tools can silently monitor the communications between your systems. In either case, all message data is displayed and defined in a human-readable format, logged and graphed.

### **GENERATE YOUR TEST TOOLS USING AN INTERFACE CONTROL DOCUMENT (ICD)**

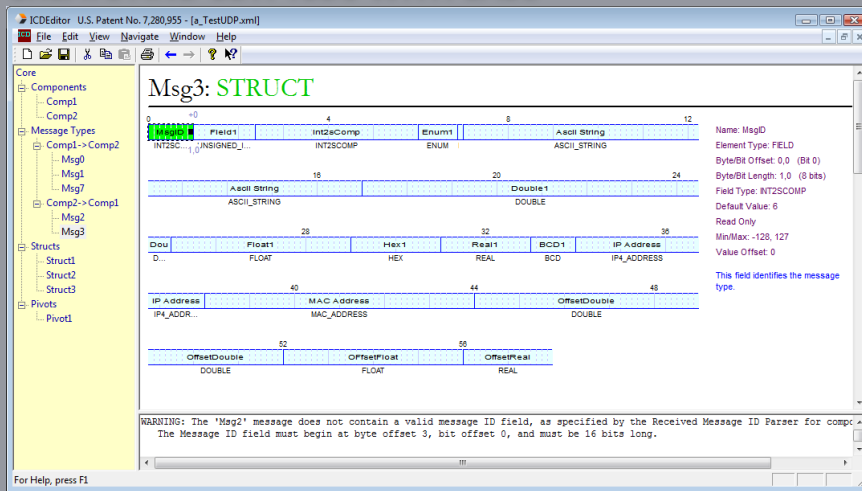
Mimik's breakthrough in productivity involves its most basic, fundamental feature: its ability to read and consume an Interface Control Document (ICD), enabling Mimik to configure itself to emulate the messaging interface of the systems that your system communicates with. The ICD gives Mimik the ability to display message data in a human readable format, while communicating with the system under test in the proper binary format.

Instead of developing an emulator tool from scratch, simply define the ICD for Mimik to consume, and load the ICD into Mimik. This immediately gives you a test tool you can use to stimulate and monitor the system-under-test. For an intermediate-level user, it generally takes several hours to define the ICD, whereas it takes weeks or months to create emulation tools from scratch.



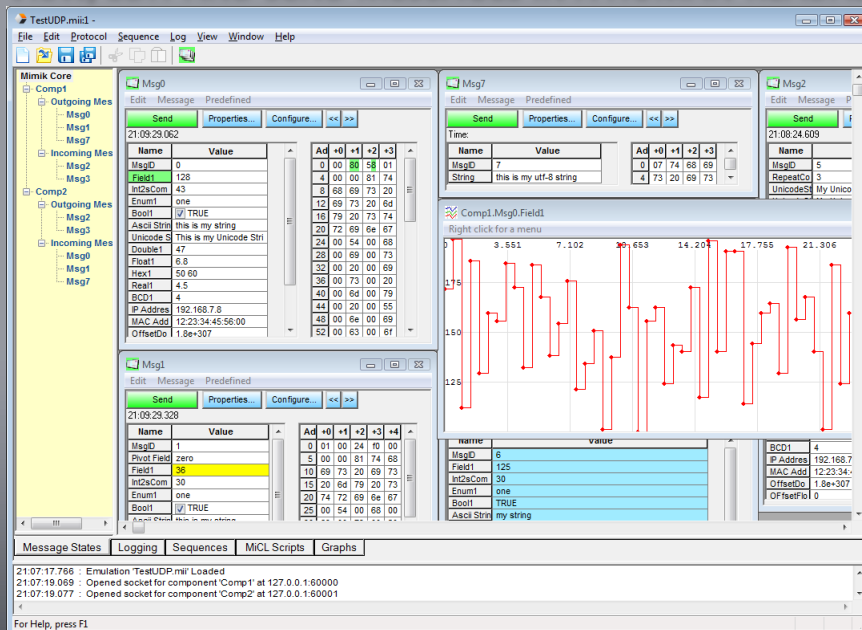
## How does it work?

### CREATE AND EDIT ICDs WITH EASE



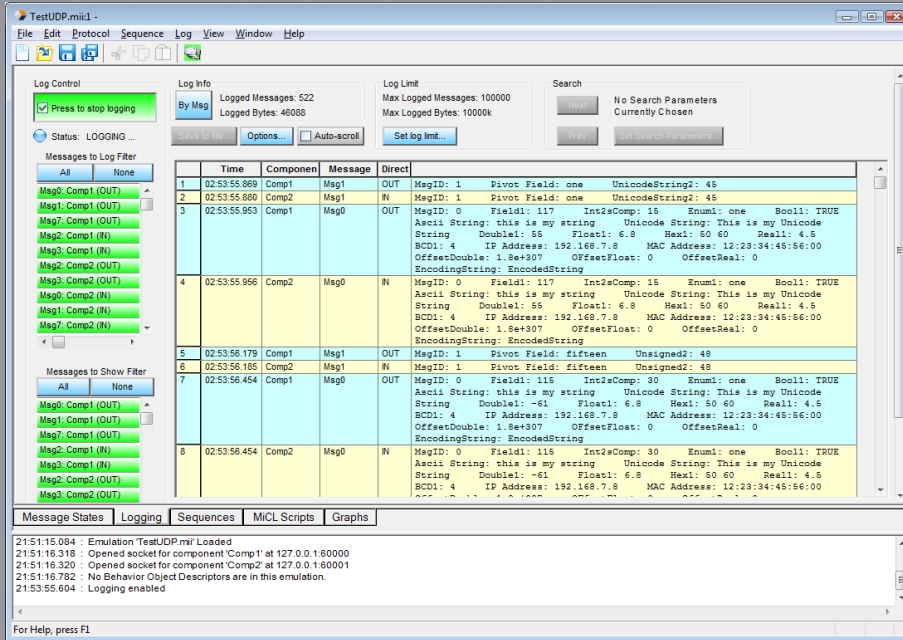
Thanks to the ICD Editor, Mimik's sister application, you can **define ICDs in an easy, visual, point-and-click manner**. While the ICD is stored in an XML format, you will never need to write any XML directly – the ICD Editor hides these formatting details from you, enabling you to create Mimik-consumable ICDs quickly and easily.

### VIEW, EDIT AND SEND MESSAGES IN A HUMAN-READABLE FORMAT



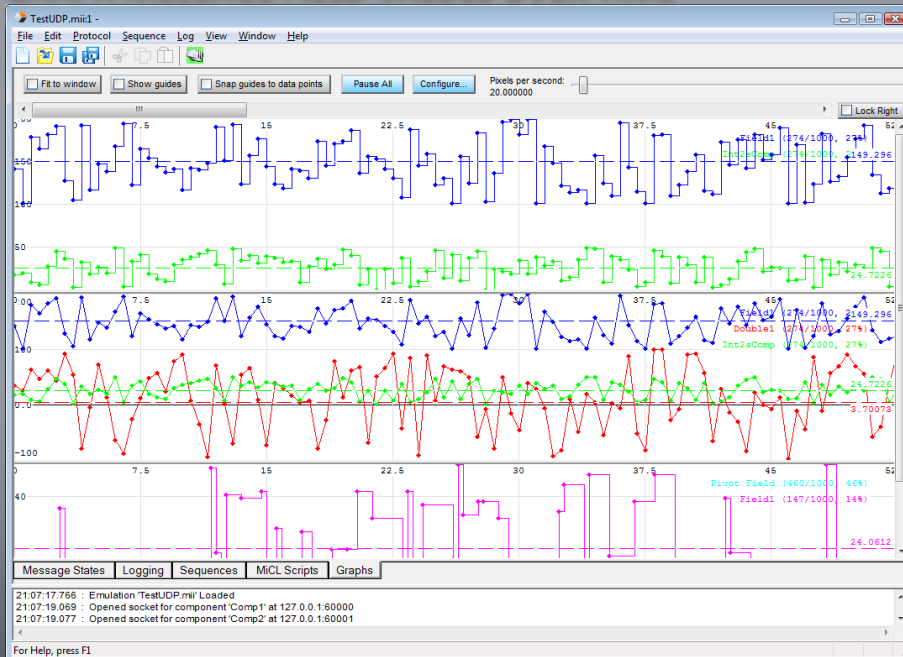
- View and edit message field data in a **human readable format**. All message fields are displayed and editable in name/value tables. Combo boxes are provided for defining enumerated fields, check boxes for boolean fields.
- View and edit message data in **Raw, "packed" format** – hexadecimal or binary.
- **Graph** the values of any message field to view trends in the data over time. Add other fields to the same graph for easy, visual comparison of data points and trends.

## LOG TRANSMITTED AND RECEIVED MESSAGES



- **Log** any or all transmitted and received message types
- **Filter** message types from the display
- **Search** the log for particular field values
- **Save** the log to plain text, spreadsheet or browser-friendly files
- Optionally show **protocol headers** and **raw data**
- **Limit** the log to a certain number of bytes or messages

## GRAPH MESSAGE FIELD VALUES OVER TIME



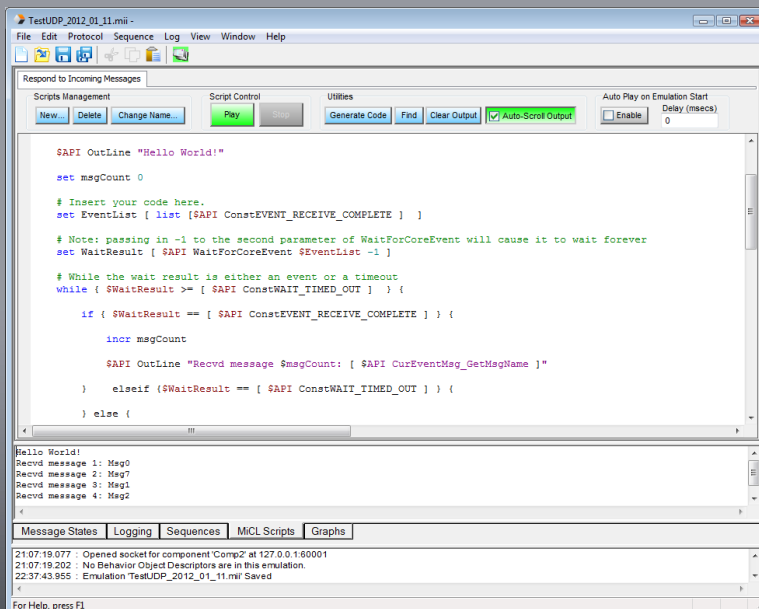
- Create as many **graphs** as you wish
- Add as many **fields** to each graph as you wish – each field gets its own color
- Generate **statistics snapshots** to view average, median, mode, population distributions, etc.
- **Pause** graph updates without data loss
- **Scroll** across time, or **fit to window**
- **Snap** guides to data points with the mouse, showing exact data point values

## WRITE SCRIPTS TO CUSTOMIZE YOUR EMULATION

Develop custom scripts to control the behavior of your emulation or monitoring solution, using Mimik's embedded scripting language, "MiCL." MiCL consists of two components:

1. The TCL language in its entirety, an industry-standard, commercial-quality scripting language, and TCL's associated GUI toolkit, Tk
2. The MiCL API, a set of over 400 Mimik-specific extensions to TCL, which enables script writers to:

- Get and set field data (in a human-readable format)
- Get and set raw message data
- Send messages
- Wait for and respond to incoming messages and other internal Mimik events
- Create your own GUI using Tk, and hook your GUI events into MiCL API functions
- Access Mimik's command line parameters, enabling highly-configurable automated testing
- Start and stop other MiCL scripts
- Share data with other concurrently-running MiCL scripts
- Implement "Custom Streaming" protocols
- Create and play sequences of outgoing messages
- Control and access Mimik's Message Log
- Output text in Mimik's GUI



```
SAPI OutLine "Hello World!"
set msgCount 0
# Insert your code here.
set EventList [ list [SAPI ConstEVENT_RECEIVE_COMPLETE ] ]
# Note: passing in -1 to the second parameter of WaitForCoreEvent will cause it to wait forever
set WaitResult [ SAPI WaitForCoreEvent $EventList -1 ]
# While the wait result is either an event or a timeout
while { $WaitResult >= [ SAPI ConstWAIT_TIMED_OUT ] } {
    if { $WaitResult == [ SAPI ConstEVENT_RECEIVE_COMPLETE ] } {
        incr msgCount
        SAPI OutLine "Recvd message $msgCount: [ SAPI CurEventMsg_GetMsgName ]"
    } elseif { $WaitResult == [ SAPI ConstWAIT_TIMED_OUT ] } {
    } else {
    }
}
```

Hello World!  
Recvd message 1: Msg0  
Recvd message 2: Msg?  
Recvd message 3: Msg1  
Recvd message 4: Msg2

Message States | Logging | Sequences | MiCL Scripts | Graphs

21:07:19.077 : Opened socket for component 'Comp2' at 127.0.0.1:80001  
21:07:19.202 : No Behavior Object Descriptors are in this emulation.  
22:37:43.955 : Emulation TestUDP\_2012\_01\_11.msi Saved

For Help, press F1

- Create as many scripts as you wish
- Simultaneously run as many scripts as you wish
- Configure scripts to start automatically when the emulation opens
- Leverage MiCL API code completion to increase productivity