

# **ICEBOX**

## **Getting Started Guide**

**Used for Arrival Checkout  
and Confidence Recheck**

**September 1, 2001**

**ICE Peripheral Interconnect Components(PIC)  
Innovative Computer Engineering  
[www.ice-online.com](http://www.ice-online.com)**

**High-End "ICEPIC-Ready" Servers  
for demanding DSP Applications  
LowNoiseFloor  
[www.LowNoiseFloor.com](http://www.LowNoiseFloor.com)**

**RI(ICE) Additional Hardware and  
Software Options  
rumel Incorporated  
[www.rumel-online.com](http://www.rumel-online.com)**

## **Errata**

Recent improvements in technology have allowed upgrades for the servers delivered with ICEBOX's to be a standard 64-bit configuration in all motherboard slots. There are 4 5Volt 64-bit slots and 2 3.3Volt 64-bit slots. Also an AGP slot is provided for the latest video card technology.

The operating system is currently LINUX 7.2 and comes pre-installed and configured. There is a supplemental document provided that details the configuration of the server with all the necessary media configuration details.

# Contents

Introduction .....	3
STEP 1: Initializing X-Midas .....	4
STEP 2: Starting the X-Midas Shell .....	4
STEP 3: Establishing Health of PIC1 .....	5
STEP 4: Establishing Health of PIC2 (optionally installed) .....	5
STEP 5: Testing module loopback for PIC1 .....	6
STEP 6: Testing module loopback for PIC2 (Optionally installed) .....	6
STEP 7: Testing onboard tuners for PIC1 .....	7
STEP 8: Testing onboard tuners for PIC2 (Optionally installed) .....	7
STEP 9: Testing the specs of PIC1 .....	8
STEP 10: Testing the specs of PIC2 (Optionally installed) .....	8
STEP 11: Online Macro Help .....	8
STEP 12: Online Documentation .....	8
ICEBOX Quick Reference Sheet .....	9
Shell Initialization Page .....	10
For MATLAB Users .....	11
Sample .cshrc file .....	12

# ICEBOX

## Getting Started Guide

This guide provides easy to use step-by-step procedures for performing initial arrival and checkout of your new ICEBOX with any installed modules that were ordered from the factory. These procedures can also be used for reestablishing correct operations when changes occur in the software or hardware installed in the system.

The procedures should be followed in the order they are given. An abbreviated “cheat-sheet” is provided for quick reference once the procedure has been run through once and confidence is already established.

Press the tab key to freeze this the above screen for reading

For initial checkout please login using the xmidas account.

Following login and successful desktop activation open a terminal window and perform the following steps:

[Note: for maximum compatibility it is recommended that the Enlightenment desktop manager be chosen. Follow the on screen activity to learn the necessary menus. A left mouse click on the Enlightenment Desktop followed by the choice of the GNOME Terminal will bring up the necessary window to invoke the following steps.

A terminal window will activate similar to the one shown below:

### **STEP 1: Initializing X-Midas**

At the “**icebox>**” prompt type “xms” (short for “xmidas start”)

```
icebox>xms
```

X-Midas will initialize.

### **STEP 2: Starting the X-Midas Shell**

Next type “xm”

```
X-Mi das>xm
```

This command puts you into the X-Midas shell.

### **STEP 3: Establishing Health of PIC1**

Next type the following command to test PIC1.

[PIC - Peripheral Interconnect Component] number one

```
X-Mi das>pi c test pi c1
```

```
X-Mi das> pi c test pi c1
Testing RESET      ... 0 errs
Testing MBX/REG    ... 0 errs
Testing SHARC mem  ... 0 errs
Testing FIFO       ... 0 errs
Testing IOC Reg    ... 0 errs
Testing SHARC DMA  ... 0 errs
Testing PCI -32b bus DMA ...
Maxsjump = 27  SRate = 128.104294 Mby/s  Ndata = 1017613
Maxbjump = 4914  BRate = 110.026863 Mby/s  Blk = 4096
Testing Checksum  ... 0 errs
X-Mi das>
```

On the Normal Server Version the rate are:

Note: The performance of SRate should be above 125Mby/s and the performance of BRate should be above 105Mby/s

One the High end Server Version the rate are:

Note: The performance of SRate should be above 260Mby/s and the performance of BRate should be above 260Mby/s

### **STEP 4: Establishing Health of PIC2 (optionally installed)**

Next type the following command to test PIC2

```
X-Mi das>pi c test pi c2
```

```
X-Mi das> pi c test pi c2
Testing RESET      ... 0 errs
Testing MBX/REG    ... 0 errs
Testing SHARC mem  ... 0 errs
Testing FIFO       ... 0 errs
Testing IOC Reg    ... 0 errs
Testing SHARC DMA  ... 0 errs
Testing PCI -32b bus DMA ...
Maxsjump = 27  SRate = 125.564285 Mby/s  Ndata = 997437
Maxbjump = 4914  BRate = 110.026863 Mby/s  Blk = 4096
Testing Checksum  ... 0 errs
X-Mi das>
```

On the Normal Server Version the rate are:

Note: The performance of SRate should be above 125Mby/s and the performance of BRate should be above 105Mby/s

One the High end Server Version the rate are:

Note: The performance of SRate should be above 260Mby/s and the performance of BRate should be above 260Mby/s

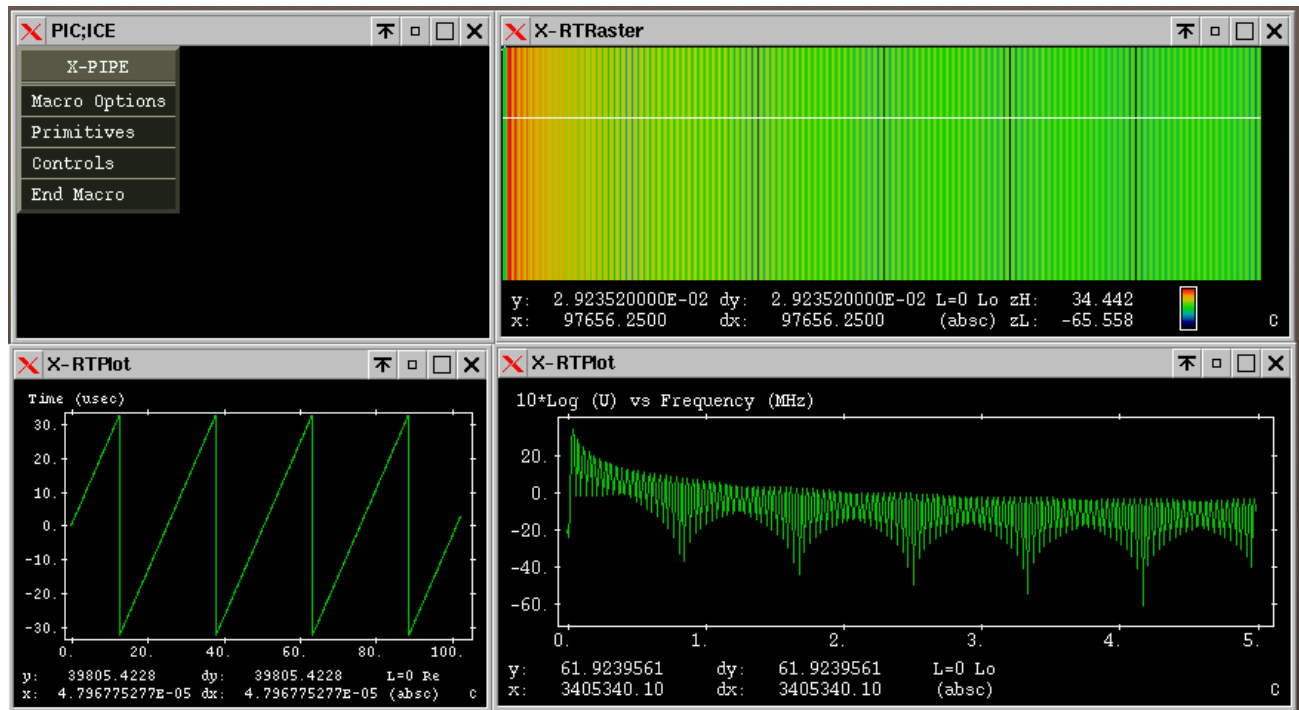
## STEP 5: Testing module loopback for PIC1

Next type the following command to test module loopback of PIC1

```
X-Mi das> pic test/mod/rt pic1t
```

This tests the module loop back of pic1. The “t” in the command above is a test alias used for specific tests that do not require the use of any installed modules.

The result of this command is an activation of 4 windows showing controls, real-time waterfall (xrtraster), real-time ramp (xrtplot), and spectrum (xrtplot).



The falling raster window should be waterfalloing down at a regular rate. Click on EndMacro in the controls window to step this test procedure.

## STEP 6: Testing module loopback for PIC2 (Optionally installed)

Next type the following command to test module loopback of PIC2

```
X-Mi das> pic test/mod/rt pic2t
```

The results should be the same as STEP 5.

Click on EndMacro in the controls window to step this test procedure.

## STEP 7: Testing onboard tuners for PIC1

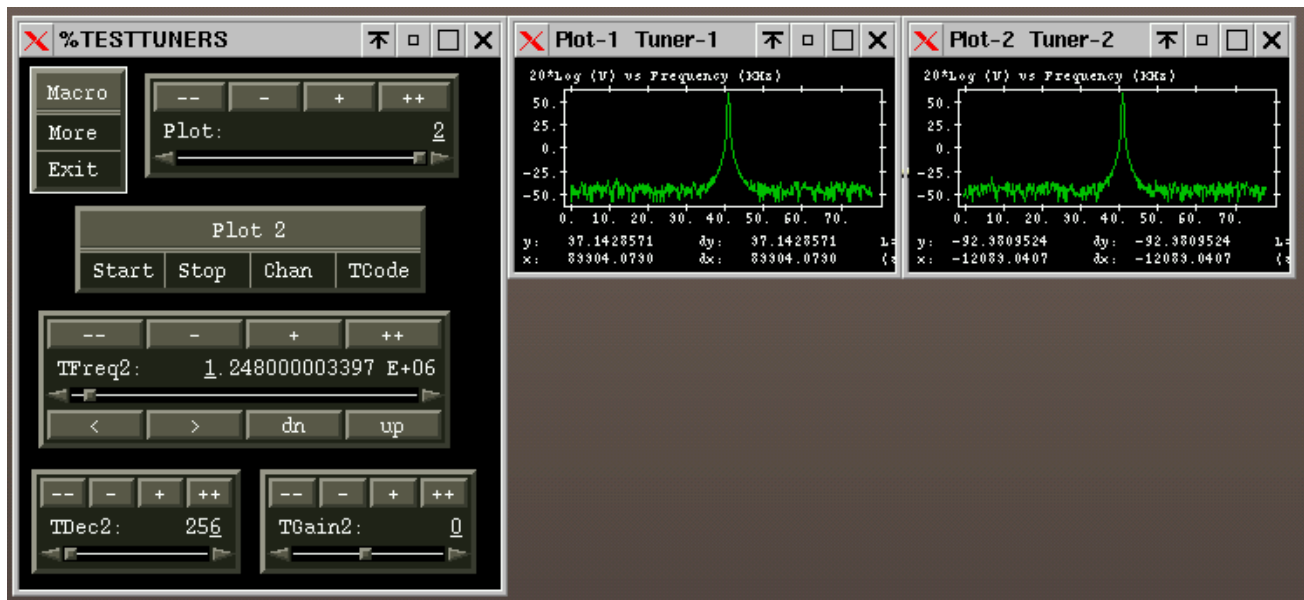
Next type “pic home” to change to the PIC home directory

```
X-Mi das> pic home
```

Next type “%testtuners pic1t” to test the tuners on PIC1

```
X-Mi das> %testtuners pic1t
```

Note that a “. . . I n i t i a t e d D a t a s t r e a m C l i e n t ” will display in the window. This tests the onboard tuners of pic1. The “t” in the command above is a test alias used for specific tests that do not require the use of any installed modules.



## STEP 8: Testing onboard tuners for PIC2 (Optionally installed)

Next type “pic home” to change to the PIC home directory

```
X-Mi das> pi c home
```

Next type “%testtuners pic2t” to test the tuners on PIC2

```
X-Mi das> %testtuners pi c2t
```

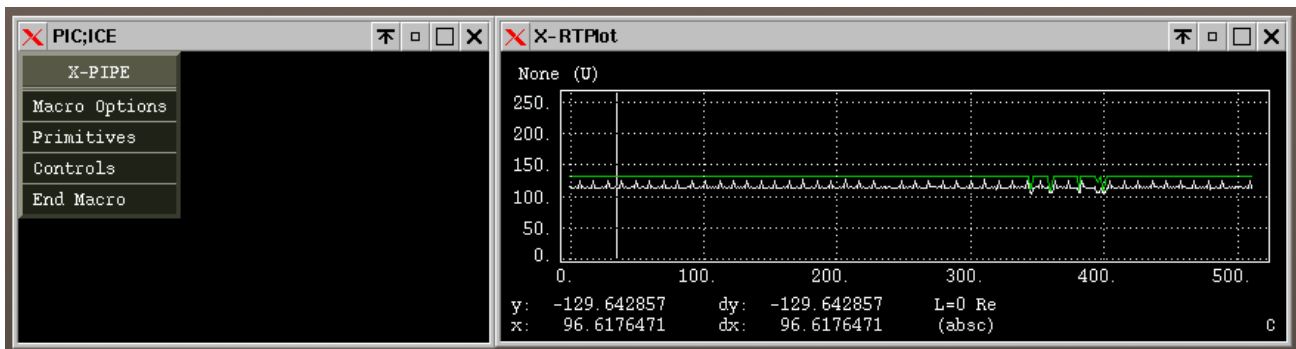
Note that a “. . . I n i t i a t e d D a t a s t r e a m C l i e n t ” will display in the window. This tests the onboard tuners of pic2. The “t” in the command above is a test alias used for specific tests that do not require the use of any installed modules. Test results should be the same as shown in step 7.

## STEP 9: Testing the specs of PIC1

Next type “pic test/spec pic1t” to change to the PIC home directory

```
X-Mi das> pi c test/spec pi c1t
```

This tests the specifications of pic1 over the PCI bus. The bus will be 32bit or 64bit depending on your configuration. The “t” in the command above is a test alias used for specific tests that do not require the use of any installed modules.



## STEP 10: Testing the specs of PIC2(Optionally installed)

Next type “pic test/spec pic2t” to change to the PIC home directory

```
X-Mi das> pi c test/spec pi c2t
```

This tests the specifications of pic2 over the PCI bus. The bus will be 32bit or 64bit depending on your configuration. The “t” in the command above is a test alias used for specific tests that do not require the use of any installed modules. Test results should be similar to those shown in Step 9.

## **STEP 11: Online Macro Help**

Next type "pic test/spec pic2t" to change to the PIC home directory

```
X-Mi das> expl ai n pi c
```

Follow the screen help that activates. The macros that are used for testing are very good starting points for beginning to build your own application.

## **STEP 12: Online Documentation**

Next type "pic test/spec pic2t" to change to the PIC home directory

```
X-Mi das> pi c hel p
```

It is recommended that all information related to the modules that you have installed be reviewed. This documentation is brief but very informative. Spending a short amount of time reading it will very often eliminate the need for trouble shooting later.

## ICEBOX Quick Reference Sheet

### STEP 1: Initializing X-Midas

```
i cebox>xms
```

### STEP 2: Starting the X-Midas Shell

```
X-Mi das>xm
```

### STEP 3: Establishing Health of PIC1

```
X-Mi das>pi c test pi c1
```

### STEP 4: Establishing Health of PIC2 (optionally installed)

```
X-Mi das>pi c test pi c2
```

### STEP 5: Testing module loopback for PIC1

```
X-Mi das> pi c test/mod/rt pi c1t
```

### STEP 6: Testing module loopback for PIC2 (Optionally installed)

```
X-Mi das> pi c test/mod/rt pi c2t
```

### STEP 7: Testing onboard tuners for PIC1

```
X-Mi das> pi c home
```

```
X-Mi das> %testtuners pi c1t
```

### STEP 8: Testing onboard tuners for PIC2 (Optionally installed)

```
X-Mi das> pi c home
```

```
X-Mi das> %testtuners pi c2t
```

### STEP 9: Testing the specs of PIC1

```
X-Mi das> pi c test/spec pi c1t
```

### STEP 10: Testing the specs of PIC2

```
X-Mi das> pi c test/spec pi c2t
```

### STEP 11: Online Macro Help

```
X-Mi das> expl ai n pi c
```

### STEP 12: Online Documentation

```
X-Mi das> pi c hel p
```

## Shell Initialization Page

### **Initialize X-Midas and start Xmidas Shell**

```
i cebox>xms  
X-Mi das>xm
```

### **Initialize NeXtMidas and start NeXtMidas Shell**

```
i cebox>nms  
X-Mi das>nm
```

### **Starting MATLAB**

```
i cebox>ml s  
i cebox>matl ab
```

## For MATLAB Users

### Starting MATLAB

```
icebox>ml s
```

```
icebox>matlab
```

### MATLAB sample M-file (ice.m)

```
function [status] = ice (func, port, fname, ...
    rate, size, bits, clks, dec, freq, gain)

prog = '$ICEROOT/test/ice'
if exist('func','var') == 0
    func = 'ACQUIRE';
end
func = upper(func);
if exist('port','var') == 0
    port = 'MODULE1';
end
port = upper(port);
devty = 'PIC'
devno = '0'
if exist('fname','var') == 0
    fname = 'mydata.dat';
end
if exist('clks','var') == 0
    clks = 'I';
end
if exist('rate','var') == 0
    rate = 10000000;
end
if exist('size','var') == 0
    size = 4*1024*1024;
end
if exist('bits','var') == 0
    bits = 16;
end
if exist('dec','var') == 0
    dec = 64;
end
if exist('freq','var') == 0
    freq = rate/4;
end
if exist('gain','var') == 0
    gain = 0;
end

[cmdstr,errstr] = sprintf('%s %s %s %s %s %s %d %s %d %d %d %f %d', ...
    prog,func,devty,devno,port,fname,bits,clks,rate,size,dec,freq,gain);
status = unix( cmdstr );

file = fopen(fname,'r');
[data,count] = fread(file,2048,'int16');
fclose(file);

plot(data);

return
```

## Sample .cshrc file

The c-shell resource file is setup during the login procedure. There are a number of path and environment details that are set during the execution of this file. If something does not initially work correctly this is the first place to look for configuration errors.

```
setenv JAVAHOME /user/java/IBMJava2-13
setenv JAVAHOME /user/java/jdk1.2.2
setenv TZ "MST 7"
#
setenv XMDISK /user/xm361
alias xms `source $XMDISK/xm/unix/xmstart`
setenv NMROOT /user/nxmxxx
alias nms `source $NMROOT/os/unix/nmstart`
setenv MATLABROOT /user/matlab5
alias mls `$MATLABROOT/etc/lmstart`
setenv ICEROOT /user/xmopts/icexxx
#
alias dir `ls -al | more`
set prompt=`hostname`> `
```