

EPICS with Cocoa

Thomas Pelaia II, Ph.D.

Application Programming

EPICS Collaboration Meeting

June 12-16, 2006

What?

- **Cocoa is a core set of Mac OS X frameworks for building applications, tools and frameworks**
 - **Language is Objective-C**
 - **Mature (inherited and extended from NextStep)**
 - **Allows for rapid development of full featured applications with Mac OS X look and feel**
 - **Integrated with a powerful IDE and GUI builder**
- **GNUStep is an open source project that ports Cocoa to other operating systems (see <http://www.gnustep.org/>)**

Goals

- Provide an EPICS **client** framework in **Objective-C**
 - Simple, object oriented interface to EPICS
 - Integrate with **Cocoa** to facilitate rapid development of applications, tools, ...
 - Reduce EPICS learning curve
- Provide an EPICS installer for **easy installation** of EPICS libraries and the EPICS framework in standard locations

Objective-C

- **Mature - Developed by Brad Cox in 1980s**
- **100% pure object oriented extension to the C language**
- **Minimal language additions to C**
- **Easy to read**
- **Available as part of gcc distribution**
- **Unique bracket notation**
- **Extensive and mature Foundation and GUI frameworks**

Language Comparison

Feature	Objective-C	Java	C++
Inheritance	Single	Single	Multiple
Forwarding	Yes	No	No
Formal Protocols	Yes	Yes	No
Typing	Weak	Strict	Strict
Java Morphing	Yes	N/A	No
C++ Morphing	Yes	No	N/A
Informal Protocols	Yes	No	No
Class Inheritance	Yes	No	No
Dynamic	Yes	Yes	No
Named Arguments	Yes	No	No
Memory Management	Assisted	Automatic Garbage Collection	Manual and Assisted
Complexity	Minimal	High	High

EPICS on Mac OS X

- Only possible through the EPICS support of POSIX and the community of developers who ported EPICS to Darwin. **Thank You!**
- This new framework takes the existing Darwin libraries and headers and wraps them in an Objective-C framework that integrates with Apple's Cocoa development frameworks and tools

EPICS Features of the Framework

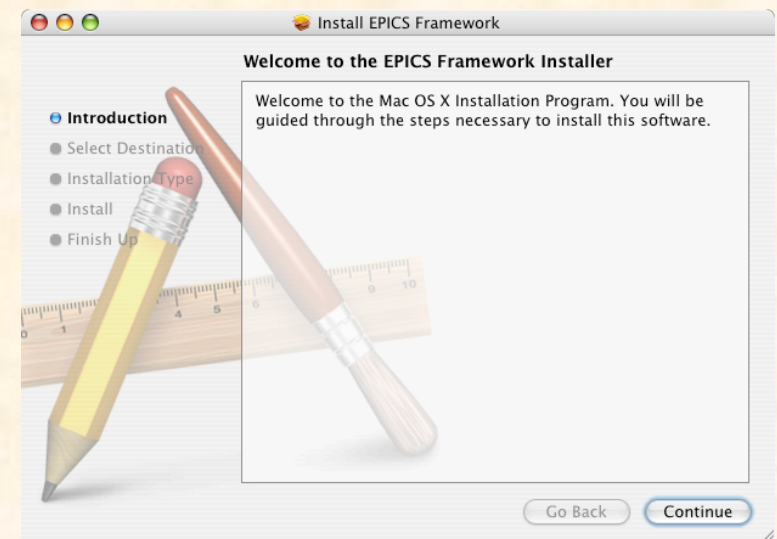
- **Supported EPICS client features**
 - Channel connections (blocking and event)
 - Get requests (blocking and event)
 - Put requests (blocking)
 - Monitors (value, status, severity, time stamp)
- **Extensible native type support via adaptors**
 - Char, double, float, int, long, short, string
 - Includes array support
- **Exception handling**

Cocoa Features of the Framework

- **Standard Cocoa style Framework**
 - Objective-C Object Oriented Language
 - Standard framework deployment for easy integration
- **Cocoa integration supporting foundation types**
 - NSArray
 - NSNumber
 - NSString
 - NSDate
- **Objective-C style events**
 - Target selectors
 - NSNotification

Installation

- **Standard, Familiar and Easy Mac OS X PowerPC Installer (Download at <http://ics-web1.sns.ornl.gov/> and navigate to Software->EPICS Framework)**
- **Installs**
 - **EPICS License and README files**
 - **EPICS base headers and libraries**
 - **Versions 3.14.6, 3.14.7 and 3.14.8.2 (default)**
 - **EPICS Framework (Objective-C)**
 - **Sample applications demonstrating how to use the framework**
 - **Simple command line tool**
 - **Cocoa application**
 - **Safari internet plug-in for displaying EPICS fields in the Safari web browser**
 - **EDM application (requires additional installation of Open Motif and libungif)**



Sample Command Line Tool Code

```

/Library/EPICS/Examples/EpicsTest/ChannelTester.m
//
// ChannelTester.m
// EpicsTest
//
// Created by Thomas Pelaia on 10/25/05.
// Copyright 2005 Oak Ridge National Lab. All rights reserved.
//
#import "ChannelTester.h"

@implementation ChannelTester

// Constructor
- (id)init {
    [super init];
    return self;
}

// primary procedure - run the test to monitor specified PVs
- (void)runTest {
    NSAutoreleasePool *pool = [NSAutoreleasePool new];

    // monitor each of the specified PVs
    [[self monitorPV:@"CF_CU:CF_AlarmActiveRed:Sts.INPA"] retain];
    [[self monitorPV:@"Ring_Diag:BLM_A01:mpsAlarm1"] retain];
    [[self monitorPV:@"MEBT_Diag:BP04:yAvg"] retain];
    [[self monitorPV:@"Physics_Test:Scope:ch1"] retain];

    // perform a get with callback for both a value and an array and a String
    [CAGetRequest requestDoubleArrayForChannel:[CACHannel
        channelWithName:@"Physics_Test:Scope:ch1"] connect]
        withSelector:@selector(handleGetRequest:values:) toTarget:self count:100];
    [CAGetRequest requestDoubleValueForChannel:[CACHannel
        channelWithName:@"Physics_Test:Scope:ch1"] connect]
        withSelector:@selector(handleGetRequest:value:) toTarget:self];
    [CAGetRequest requestStringValueForChannel:[CACHannel
        channelWithName:@"CF_CU:CF_AlarmActiveRed:Sts.INPA"] connect]
        withSelector:@selector(handleGetRequest:value:) toTarget:self];

    // flush all requests
    [CACHannel flushIO];

    // monitor the PVs for 2 seconds so we can observe a few events
    [NSThread sleepUntilDate:[NSDate dateWithTimeIntervalSinceNow:2.0]];

    [pool release];
}

// create a monitor for the specified PV
- (CAMonitor *)monitorPV:(NSString *)name {
    // create a new channel for the PV
    CACHannel *channel = [CACHannel channelWithName:name];

    // create a monitor for the channel and add an observer of monitor events

```

Printed: 5/23/06 2:11 PM

Page 1

```

/Library/EPICS/Examples/EpicsTest/ChannelTester.m
CAMonitor *monitor = [CAMonitor monitorWithChannel:channel];
[monitor addMonitorObserver:self selector:@selector(handleMonitor:)];

// add channel connect/disconnect observers, request the connection and flush the re
quest to the servers
[channel addConnectObserver:self selector:@selector(channelConnected:)];
[channel addDisconnectObserver:self selector:@selector(channelDisconnected:)];
[channel connect];

// return the new monitor
return monitor;
}

// handle the connection request
- (void)channelConnected:(NSNotification *)notification {
    NSLog(@"Channel %@ connected...", [notification channel] );
}

// handle the disconnect request
- (void)channelDisconnected:(NSNotification *)notification {
    NSLog(@"Channel %@ disconnected...", [notification channel] );
}

// handle the monitor event by printing the event
- (void)handleMonitor:(NSNotification *)notification {
    NSLog(@"Monitor event: %@", [notification monitorEvent] );
}

// handle a get request for values
- (void)handleGetRequest:(CAGetRequest *)request values:(NSArray *)values {
    NSLog(@"Get Event for channel %@, values: %@", [[request channel] name], values );
}

- (void)handleGetRequest:(CAGetRequest *)request value:(id)value {
    NSAutoreleasePool *pool = [NSAutoreleasePool new];
    NSString *channelName = [[request channel] name];
    NSLog(@"Get Event for channel %@, value: %@", channelName, value );
    [pool release];
}

@end

```

Printed: 5/23/06 2:11 PM

Page 2



Sample Application Code

```

/Library/EPICS/Examples/EPICS Demo/ChannelMonitor.m
//
// ChannelMonitor.m
// EPICS Demo
// Created by Thomas Pelaia on 11/21/05.
// Copyright 2005 Oak Ridge National Lab. All rights reserved.
//
#import "ChannelMonitor.h"

@implementation ChannelMonitor

// cleanup due to faulting of object
- (void) didTurnIntoFault {
    [_monitor release];
    [_latestEvent release];
}

// get the PV
- (NSString *)pv {
    NSString * tmpValue;

    [self willAccessValueForKey: @"pv"];
    tmpValue = [self primitiveValueForKey: @"pv"];
    [self didAccessValueForKey: @"pv"];

    return tmpValue;
}

// set a new PV and start monitoring it
- (void) setPv:(NSString *)xvalue {
    [self willChangeValueForKey: @"pv"];
    [self setPrimitiveValue: xvalue forKey: @"pv"];
    [self didChangeValueForKey: @"pv"];
    [self monitorPV];
}

// validate the PV
- (BOOL) validatePv: (id *)valueRef error:(NSError **)outError {
    // Insert custom validation logic here.
    return YES;
}

// get the color to display based on connection status - grey for unconnected and black
// for connected
- (NSColor *)statusColor {
    NSColor *color;
    CAMonitorEvent *event = [_latestEvent retain];
    [self willAccessValueForKey: @"statusColor"];
    if ( ![[[_monitor channel] isConnected] ] ) {
        color = [NSColor lightGrayColor];
    }
}

/Library/EPICS/Examples/EPICS Demo/ChannelMonitor.m
else if ( event != nil ) {
    switch ( [event severity] ) {
        case NO_ALARM:
            color = [NSColor greenColor];
            break;
        case MINOR_ALARM:
            color = [NSColor orangeColor];
            break;
        case MAJOR_ALARM:
            color = [NSColor redColor];
            break;
        case INVALID_ALARM:
            color = [NSColor blueColor];
            break;
        default:
            color = [NSColor blueColor];
            break;
    }
} else {
    color = [NSColor blackColor];
}
[self didAccessValueForKey: @"connectionColor"];
[event release];
return color;

// advertise the updated color to observers
- (void) updateStatusColor {
    [self willChangeValueForKey: @"statusColor"];
    [self didChangeValueForKey: @"statusColor"];
}

// get the latest cached monitor event
- (CAMonitorEvent *)latestEvent {
    return _latestEvent;
}

// cache the latest monitor event
- (void) setLatestEvent:(CAMonitorEvent *)event {
    [self willChangeValueForKey: @"latestEvent"];
    id oldEvent = _latestEvent;
    _latestEvent = [event retain];
    [self didChangeValueForKey: @"latestEvent"];
    [oldEvent release];
}

// begin monitoring the PV when awaking from the document fetch
- (void) awakeFromFetch {
    [self monitorPV];
}

/Library/EPICS/Examples/EPICS Demo/ChannelMonitor.m
// create a channel for the PV, request a connection and begin monitoring as soon as the
// channel is connected
- (id) monitorPV {
    // stop listening to the old monitor, clear the cached monitor event, and dispose of
    // the old monitor
    [_monitor removeMonitorObserver:self];
    [self setLatestEvent:nil];
    id oldMonitor = _monitor;
    _monitor = nil;
    [oldMonitor release];

    // check that the PV is not empty
    if ( [self pv] != nil & ![self pv] isEqualToString:@""] ) {
        // create a new channel and observe connection status
        id channel = [[CAMonitor channelWithName:[self pv]];
        [channel addConnectObserver:self selector:@selector(handleConnect)];
        [channel addDisconnectObserver:self selector:@selector(handleDisconnect)];

        // create a monitor for the new channel which will start when the channel is con
        // nected and observe new monitor events
        _monitor = [[CAMonitor monitorWithChannel:channel] retain];
        [_monitor addMonitorObserver:self selector:@selector(handleMonitor)];

        // request a channel connection and flush the request to the server
        [channel connect];
        [CAMonitor flushIO];

        return self;
    }

    // handle new monitor events by caching the latest event
    - (void) handleMonitor:(NSNotification *)notification {
        [self setLatestEvent:[notification monitorEvent]];
        [self updateStatusColor];
    }

    // handle a channel connection event
    - (void) handleConnect:(NSNotification *)notification {
        [self updateStatusColor];
    }

    // handle a channel disconnect event
    - (void) handleDisconnect:(NSNotification *)notification {
        [self updateStatusColor];
    }

@end
Printed: 5/23/06 2:18 PM Page 1 Printed: 5/23/06 2:18 PM Page 2 Printed: 5/23/06 2:18 PM Page 3
```



Sample Application

EPICS Demo File Edit Window Help

EPICS Demo

Tue 2:28:00 PM Thomas Pelaia

Channel Monitor

Q- All

Pv	Value	Status	Severity	Time Stamp
FE_MPS:MIOC1A:status_sum_LOLO	2.5000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.391
Physics_TestScope:ch1	0.0000	NO_ALARM	NO_ALARM	Dec 31, 1989 19:00:00.000
Physics_TestScope:ch2	0.0000	NO_ALARM	NO_ALARM	Dec 31, 1989 19:00:00.000
Ring_Diag_BLM_A06:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
Ring_Diag_BLM_A07:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:11.741
Ring_Diag_BLM_A08:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
Ring_Diag_BLM_A09:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:11.741
Ring_Diag_BLM_B01:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
Ring_Diag_BLM_B02:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:11.741
Ring_Diag_BLM_B03:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
Ring_Diag_BLM_B04:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:11.741
Ring_Diag_BLM_B05:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
Ring_Diag_BLM_B06:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:11.741
Ring_Diag_BLM_B07:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
Ring_Diag_BLM_B08:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:11.741
Ring_Diag_BLM_B09c:mpsAlarm1	1.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:27:12.207
SCL_LLRF:HPM01a:H80	2,228,257.0000	NO_ALARM	NO_ALARM	May 23, 2006 14:28:00.822

1 out of 17

PV: SCL_LLRF:HPM01a:H80

Fetch Remove Add

Selection Detail

Values

2,228,257.00000
2,359,351.00000
7,536,821.00000
15,466,767.00000
19,792,198.00000
22,610,283.00000
24,838,538.00000
26,542,497.00000
27,984,309.00000
29,295,047.00000
30,409,175.00000
31,261,155.00000
32,178,673.00000
32,899,578.00000
33,554,949.00000
34,275,853.00000
34,865,687.00000
35,324,447.00000
35,848,742.00000
36,307,502.00000
36,766,259.00000
37,225,018.00000
37,552,706.00000
37,880,388.00000
38,208,073.00000
38,535,759.00000
38,928,979.00000
39,125,591.00000
39,453,277.00000
39,715,424.00000
39,977,572.00000
40,174,183.00000
40,501,868.00000
40,698,479.00000
40,960,627.00000
41,091,701.00000
41,353,849.00000
41,484,922.00000
41,747,070.00000
41,878,146.00000
42,074,755.00000
42,205,830.00000
42,467,976.00000
42,533,515.00000
42,730,125.00000
42,861,199.00000
42,992,274.00000
43,123,346.00000
43,254,421.00000
43,319,959.00000
43,516,570.00000
43,582,106.00000
43,778,716.00000
43,844,255.00000
43,975,328.00000



OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY
EPICS Collaboration Meeting, June 12-16, 2006



Alarm Annunciator

Developed by Curtis Dunn

- Monitors Alarm summary PVs
- Speaks and displays Warnings and Alarms as they occur and repeats as necessary
- Deployed in the SNS Control Room
- Uses the EPICS Framework

Time Stamp	PV	Message	Status	Severity
05/30/06 19:31:13	CCL_RCCS:Summary_Skid3:Alarm	CCL_RCCS Skid 3	LINK	MAJOR
05/30/06 16:19:11	CCL_RCCS:Summary_Skid3:Alarm	CCL_RCCS Skid 3	LINK	MAJOR
05/30/06 14:22:03	CCL_DIWS:Summary:Alarm	CCL_DIWS	LOLO	MAJOR
05/30/06 14:21:55	ICS_MPS:Timing:Alarm	ICS_MPS:Timing:Alarm	HIHI	MAJOR
05/30/06 00:52:29	CF_PM:MB_FDR3014:Watts	Power Feeder 3014	LOLO	MAJOR
05/28/06 07:37:21	SCL_DIWS:Summary:Alarm	SCL_DIWS	LINK	MAJOR
05/27/06 03:54:50	CCL_RCCS:Summary_Skid3:Alarm	CCL_RCCS Skid 3	LINK	MAJOR
05/26/06 21:05:14	DTL_Gly:Summary:Alarm	DTL Glycol Cooling Systems	LOLO	MAJOR
05/23/06 18:27:11	CF_PM:Summary:Alarm	Power Monitoring Systems	LINK	INVALID
05/23/06 09:47:19	CF_CU:CF_AlarmActive:Red:Sts	CF Summary Alarm	HICH	MAJOR
05/19/06 13:37:46	PPS_Lin:Chmk_Summary:Alarm	Linac Chipmunk	LINK	INVALID
05/19/06 13:37:46	PPS_HEBT:Chmk_Summary:Alarm	HEBT Chipmunk	LINK	INVALID
05/19/06 13:37:46	PPS_Ring:Chmk_Summary:Alarm	Ring Chipmunk	LINK	INVALID
05/19/06 13:37:46	PPS_RTBT:Chmk_Summary:Alarm	RTBT Chipmunk	LINK	INVALID
05/19/06 13:37:46	PPS_Tgt:Chmk_Summary:Alarm	Target Chipmunk	LINK	INVALID
05/30/06 16:14:28	CCL_RCCS:Summary_Skid3:Alarm	CCL_RCCS Skid 3	HIGH	MINOR
05/30/06 15:44:30	CCL_DIWS:Summary:Alarm	CCL_DIWS	HIGH	MINOR
05/30/06 14:03:20	ICS_MPS:Timing:Alarm	ICS_MPS:Timing:Alarm	HIGH	MINOR
05/30/06 03:54:11	CF_PM:MA_D73:Watts	Ring Power Substation	LOW	MINOR
05/30/06 01:51:00	CF_PM:MA_D73:Watts	Ring Power Substation	LOW	MINOR
05/29/06 23:53:50	CF_PM:MA_D73:Watts	Ring Power Substation	LOW	MINOR
05/29/06 12:03:40	CCL_RCCS:Summary_Skid3:Alarm	CCL_RCCS Skid 3	LOW	MINOR
05/29/06 06:58:30	CHL_Util:AlmMod1:Alarm	CHL Building Systems	LINK	MINOR
05/27/06 03:27:20	PPS_HEBT:Gate:HEBT:RingSts	HEBT/Ring Door is Open	STATE	MINOR
05/26/06 17:31:09	CF_Cool:Summary:Alarm	Cooling Water Systems	LINK	MINOR
05/25/06 23:24:30	SCL_QMCS:Summary:Alarm	SCL QMCS	LOW	MINOR
05/25/06 09:22:46	CF_PM:MB_FDR3074:Watts	Power Feeder 3074	LOW	MINOR
05/25/06 08:34:07	SCL_DIWS:Summary_KL2:Alarm	SCL DIWS Pump KL-2	LOLO	MINOR
05/25/06 01:21:03	ICS_MPS:Timing:Alarm	ICS_MPS:Timing:Alarm	HIGH	MINOR
05/24/06 21:32:10	CHL_Util:AlmMod2:Alarm	Transfer Line and Cryo Systems	LINK	MINOR
05/24/06 03:59:20	CF_PM:Summary:Alarm	Power Monitoring Systems	LINK	MINOR
05/23/06 06:31:40	CCL_RCCS:Summary:Alarm	CCL_RCCS	LINK	MINOR
05/23/06 05:59:13	SCL_DIWS:Summary_KL2:Alarm	SCL DIWS Pump KL-2	LOLO	MINOR
05/23/06 00:12:38	PPS_Lin:PLC_C:Chmk204Alarm	Linac Chipmunk 204	STATE	MINOR
05/22/06 10:19:33	PPS_Lin:PLC_C:Chmk204Alarm	Linac Chipmunk 204	STATE	MINOR
05/20/06 08:51:29	FE_MPS:MIOC1A:status_sum	MPS	LOW	MINOR
05/20/06 02:21:10	HEBT_Coll:Summary:Alarm	HEBT Collimator Cooling Skid	HIGH	MINOR
05/19/06 20:21:09	Ring_Mag:Summary_Ovr:Alarm	Ring Magnet Alarm	LINK	MINOR
05/18/06 11:47:06	ICS_MPS:Timino:Alarm	ICS MPS:Timino:Alarm	HIGH	MINOR

Screenshot is Courtesy of Curtis Dunn

EPICS Internet Plug-in

- Internet Plug-in for Apple's Safari web browser
- Prototype
- Written using the EPICS Framework
- Displays EPICS PV values along with color for severity using simple HTML
- Allows JavaScript access to EPICS for more power

PV Status Demo

MEBT - Ring

Ring Chicane Magnets

Ring_Mag:DH_A10	9.9356E-5
Ring_Mag:DH_A11	3.5316E-6
Ring_Mag:DH_A12	2.8179E-5

Ring Diagnostics A		Ring Diagnostics B		Ring Diagnostics C	
PV	Value	PV	Value	PV	Value
Ring_Diag:BPM_A01:xAvg	-3259.336	Ring_Diag:BPM_B01:xAvg	-0.3200	Ring_Diag:BPM_C01:xAvg	1.6423
Ring_Diag:BPM_A02:xAvg	0.1282	Ring_Diag:BPM_B02:xAvg	waiting	Ring_Diag:BPM_C02:xAvg	-2.0371
Ring_Diag:BPM_A03:xAvg	-0.9090	Ring_Diag:BPM_B03:xAvg	2.6381	Ring_Diag:BPM_C03:xAvg	-0.7609
Ring_Diag:BPM_A04:xAvg	1.2721	Ring_Diag:BPM_B04:xAvg	1.0427	Ring_Diag:BPM_C04:xAvg	1.0326
Ring_Diag:BPM_A05:xAvg	-1.1598	Ring_Diag:BPM_B05:xAvg	0.6910	Ring_Diag:BPM_C05:xAvg	-1.7413
Ring_Diag:BPM_A06:xAvg	0.3239	Ring_Diag:BPM_B06:xAvg	-0.4678	Ring_Diag:BPM_C06:xAvg	1.8748
Ring_Diag:BPM_A07:xAvg	-1.1167	Ring_Diag:BPM_B07:xAvg	-2.7596	Ring_Diag:BPM_C07:xAvg	-1.1174
Ring_Diag:BPM_A08:xAvg	-1.6967	Ring_Diag:BPM_B08:xAvg	1.9601	Ring_Diag:BPM_C08:xAvg	-49.7097
Ring_Diag:BPM_A09:xAvg	-2.8040	Ring_Diag:BPM_B09:xAvg	-1.7262	Ring_Diag:BPM_C09:xAvg	-1.0062
Ring_Diag:BPM_A10:xAvg	0.6781	Ring_Diag:BPM_B10:xAvg	0.9362	Ring_Diag:BPM_C10:xAvg	-6.5008

Ring Diagnostics D	
PV	Value
Ring_Diag:BPM_D01:xAvg	-2.8210
Ring_Diag:BPM_D02:xAvg	-1.9560
Ring_Diag:BPM_D03:xAvg	-2.8702
Ring_Diag:BPM_D04:xAvg	52.0223
Ring_Diag:BPM_D05:xAvg	-3.4890
Ring_Diag:BPM_D06:xAvg	-0.6575
Ring_Diag:BPM_D07:xAvg	-2.8558
Ring_Diag:BPM_D08:xAvg	-1.5296
Ring_Diag:BPM_D09:xAvg	-133390.5
Ring_Diag:BPM_D10:xAvg	424.0814

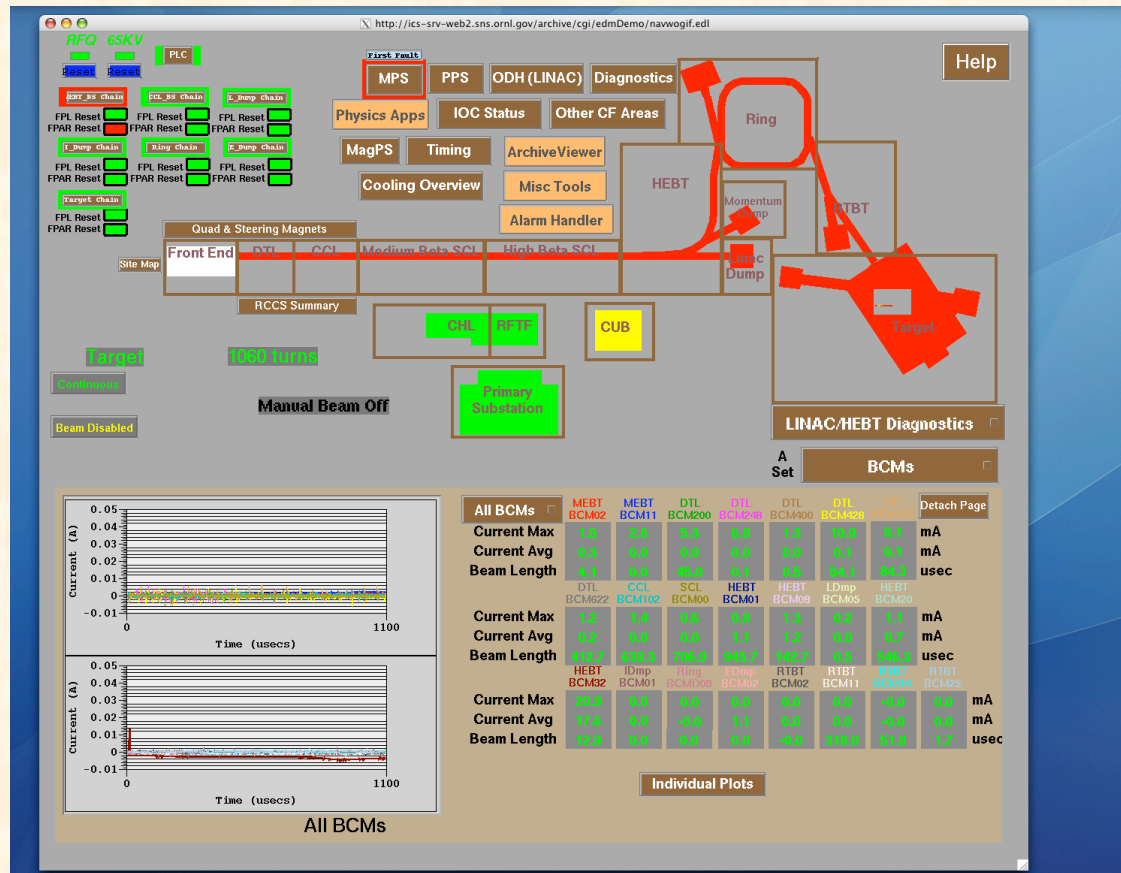
MEBT_Diag:BPM01:amplitu Monitor PV 0.1567

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY



EDM on the Mac

- EDM has been ported to Mac OS X with the help of Ernest Williams and John Sinclair
- Much faster than remotely using X
- Native application but not Cocoa and not based on the EPICS Framework
- Gets installed with EPICS Framework Installer



Summary

- **EPICS is alive and well on Mac OS X thanks to the work of many people in the EPICS community who have adopted POSIX standards and provided a Darwin port of EPICS**
- **EDM has been ported to the Mac**
- **A Cocoa based EPICS framework has been developed for Mac OS X PowerPC**
- **An internet plug-in for EPICS has been developed**
- **Need a Universal Binary for EPICS**