

The background of the image is a collage of various items: a magnifying glass with an Apple logo on its handle, a green pen holder with several pens, a globe, and some papers. The text is overlaid on this background.

Worldwide

Developers

Conference



**Building Network
Savvy Applications
with
MacApp Release 13**

Mike Rossetti

**MacApp Underground
Cell Chief**

Session Description

What we will cover in this session

- **Details on some new features**
 - Networking
 - Platinum appearance
 - Threading
- **And some words on**
 - Transition to Release 13





Networking Suite

Mike Fullerton

ACS's Networking Suite

Impress your co-workers with little effort

- **Designed for networking newbies**
- **Designed for networking gurus**
- **Abstracted API**
- **Flexible object-oriented subsystem**





Webinator
Demo

Webinator — ctor

```
CWebinatorApp::CWebinatorApp()  
    : TApplication('????', '????'),  
      fReceivingData(0)  
{  
    ...  
    fConnections  
        = TH_new CMultiTCPConnectionManager_AC;  
    CReactor_AC* aReactor  
        = fConnections->GetReactor();  
    aReactor->AcquireIdleTimeEngine(  
        TH_new TIdleTimeEngine);  
  
    MTCPCConnectionEventReceiver_AC::  
        RegisterForEventsFrom(fConnections);  
  
    fConnections->ListenTo(  
        CInternetAddress_AC::kHTTP);  
}
```



Receiving Data — Preparation

```
void CWebinatorApp::PrepareToReceiveData(  
    CTCPCConnection_AC* inHandler)  
{  
    inHandler->Receive(TH_new CDataReceiver_AC);  
}
```



Receiving Data

```
void CWebinatorApp::DoReceiveData(
    CTCPCConnection_AC* inSession,
    CNetworkTransaction_AC* inRcvr
{
    CFSSpec_AC spec; CTextParser_AC parser;
    CTextParser_AC::CToken token =
        parser.Start(
            inRcvr->GetStream().GetReadPositionPtr());

    if(token == "GET")
    {
        token = parser.GetNextToken();

        if(token == "/")
            spec.SetName(CStr63_AC("index.html"));
        else
            spec.SetName( CStr63_AC(token+1));
        // Advance over length byte
    }
    inSession->Send( TH_new FileSender_AC(spec,
        CFileSender_AC::kDataFork ));
    inSession->Send(TH_new CDisconnect_AC );
}
```



ACS's Networking Suite

For simple tasks

- **Simple API inspired by Java**
- **Example: CTCPSocket_AC**
- **Blocking, synchronous calls**
- **Thread savvy**



ACS's Networking Suite

For complex tasks

- **Asynchronous API**
- **Polymorphic network abstractions**
- **Manage multiple connections or sessions simultaneously**
- **Helper classes for managing single or multiple connections or sessions**
- **Queueable net transactions**



ACS's Networking Suite

Platform, framework and provider agnostic

- **Chooses provider at run-time**
 - MacTCP
 - OpenTransport
 - WinSock
 - BSD Sockets
- **Not tied to the Macintosh platform**
- **...we'd prefer that you use MacApp!**



ACS's Networking Suite

Uses modern C++'isms

- Auto pointers
- Reference counting auto pointers
- Templates
- Native exceptions
- Several *Design Patterns*
- Modern casting operators



ACS's Networking Suite

Stream-based I/O

- Streams
- Sources/Sinks
- All compatible with
MStreamable_AC



ACS's Networking Suite

Features available today

- The “plumbing”
- Internet aware



ACS's Networking Suite

Today: The "Plumbing"

- **Asynchronous event handling**
- **Service handlers**
- **Service handler managers**
- **Streams, sources and sinks**
- **Buffer objects**
- **Threading savvy**



ACS's Networking Suite

Today: The "Plumbing" (cont.)

- **Transaction classes for sending and receiving files and data**
- **Text parsing and byte iterators**
- **Timers**
- **Abstracted addressing**
- **Strategy classes**



ACS's Networking Suite

Today: Internet aware

- Internet DNR services
- TCP connections
- OpenTransport/MacTCP



ACS's Networking Suite

Future features

- WinSock
- BSD Sockets
- AppleTalk
- UDP
- TCP Multicasting
- FTP, SMTP, POP, &c.
- Supporting views in MacApp
- Let us know what you'd like





Dropener
Demo

Receive Data — Preparation

```
...  
CFileReceiver_AC* receiver  
    = TH_new CFileReceiver_AC(spec,  
        CFileReceiver_AC::kEntireFile);  
  
receiver->SetErrorHandler(TH_new  
    CReceiverErrorHandler);  
  
inHandler->Receive(receiver);  
  
inHandler->Send(TH_new CDisconnect_AC);  
...
```



Receive Data

```
...
if(inTransaction->IsComplete())
{
    CFileReceiver_AC* fileReceiver
        = dynamic_cast<CFileReceiver_AC*>
            (inTransaction);

    ::ThrowIfNULL_AC(fileReceiver);

    TFile* file = new TFile;
    file->Specify(fileReceiver->GetFileSpec());
    HandleReceivedFile(file);
}
...
```



Sending Data

```
...
TemplateIterator_AC<TFile*> iter(aFileList);
TFile * aFile = iter.First();

while (aFile)
{
    CRefCountingPtr_AC<CServiceHandler_AC>
        newConnection = fConnections->ConnectTo(
            CInternetAddress_AC(remoteAddressStr,
                               remotePort));

    CFileSender_AC* sender
        = TH_new CFileSender_AC(aFile->GetFileSpec(),
                                CFileSender_AC::kEntireFile);

    newConnection->Send(sender);
    aFile = iter.Next();
}
...
```





MacApp's Threading and Appearance Manager Support

Brian Arnold

Appearance Manager

Platinum Appearance

- ACS appearance classes
- New MacApp views
- Ad Lib templates





Appearance
Demo

ACS's Threading Suite

Overview

- **Multitasking within an application**
- **Improved responsiveness**
- **Synchronization of shared resources**



ACS's Threading Suite

Quick and easy threaded applications

- **Simplified CommonPoint threads**
- **CThread base class**
- **Convenience classes**
 - CThreadRunner
 - CWhileSynchronized and CMonitorLock
- **Platform agnostic**
- **Flexible object-oriented subsystem**
- **Thread Manager, MP, Win32 support**



ACS's Threading Suite

CThread

- Base class for your threads
- Implements
 - Run(), and
 - A bool for shutdown



ACS's Threading Suite

CThreadRunner

- Convenience class
- Accepts thread and runs it
- Provides
 - State inquiry
 - Shutdown request
 - Join (wait for) its thread



ACS's Threading Suite

CWhileSynchronized and CMonitorLock

- Stack-based, accepts CMonitorLock
- Acquires lock during ctor
- Releases lock in dtor





Simple Threads
Demo

Pardon the Interruption

Contextual menus

- Expose selection content to plug-ins
- Data Detectors plug-ins
 - Maps content and patterns to actions



ACS's Threading Suite

```
// Thread implementation
```

```
class TSimpleThread : public CThread_AC
{
public:
    TSimpleThread(TSimpleThreadView*
                 itsView = NULL);
    virtual ~TSimpleThread();
    virtual void Run();

private:
    TSimpleThreadView* fView;
};
```



ACS's Threading Suite

```
// Thread implementation

void TSimpleThread::Run()
{
    int iter = 0;
    if (fView != NULL)
    {
        while (!ShutdownRequested())
        {
            fView->ThreadedDraw(iter);
            Yield();
        }
    }
}
```





FracAppMP
Demo

FracAppMP

```
class CFracAppEngine
{
public:
    bool          CalcCity()
    ...
protected:
    CMonitorLock_AC fLock;
    CPoint_AC       fCurrentLocation;
    ...
}
```



FracAppMP

```
bool CFracAppEngine::CalcCity()
{
    ...
    {
        // Scoped to unlock as soon as able.
        // CalcCity is called from within
        // multiple threads, so synchronize
        // when updating the current location.

        CWhileSynchronized_AC entry(fLock);
        fCurrentLocation.v++;

        ...
    }
    ...
}
```



ACS's Threading Suite

Threads, MacApp and the Future

- **Use threads with any framework**
 - On any platform
- **Threaded idle and reactor patterns**
- **Threaded MacApp event handling**
- **Rhapsody thread support**





TidyHeap

Mike Fullerton

ACS's TidyHeap Tools

Diagnostic tool

- Suite of C++ classes and operators
- Tracks and verifies memory blocks
- Reports undeleted allocations
- Flexible and extensible



ACS's TidyHeap Tools

Diagnostic tool

- Detects double deletions
- Garbage-in action class
- Garbage-out action class
- Force *new* to fail
- Verify selected or all blocks
- More...



ACS's TidyHeap Tools

Support Tools

- TidyHeap Director
- NewDropper





TidyHeap
Demo

Transitioning to Release 13

Here are some thoughts

- One or two weeks typical
- MacApp 3.3 or R12 as a base
- Manual intervention required
- Conversion scripts and tools



Transitioning to Release 13

Fundamental changes

- I-methods -> ctors
- Free -> dtors
- Fail -> Throw/Catch
- ACS naming convention
- Scope qualification





Brian Arnold
Mary Boetcher
David Clancy
Steven Friedrich
Mike Fullerton
Gary Little
Shaowei Mao
Mike Rossetti

Q&A

The background features a dark, textured surface with a glowing blue and purple sphere in the center. The sphere has a white Apple logo on its top. A magnifying glass is positioned over the sphere, and a pen is visible on the right side. The text "Worldwide" is written in a golden, serif font across the top.

Worldwide

Developers

Conference