

The background features a dark, textured surface with a glowing blue and purple orb in the center. The orb has a white Apple logo on its top. To the right, there are faint, glowing lines and shapes that resemble a globe or a network. The overall aesthetic is futuristic and tech-oriented.

Worldwide

Developers

Conference





Mac OS User Experience II

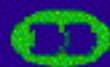


User Experience Directions

*Arno Gourdol,
User Experience
Tech Lead*

User Experience Roadmap

-  Available now
-  Available this summer
-  Coming up



System Management

**Extensions
Manager 4**

Upgrader 1.0



Desktop Management



Information Access

AIAT SDK



Customization

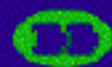


Assistance and Help

Apple Guide

User Experience Roadmap

- Available now
- Available this summer
- Coming up



System Management

Extensions Manager 4

Upgrader 1.0

Upgrader 1.1

Hard Disk Layout

Setup Assistant



Desktop Management

Threaded Finder

Pop-up windows and SLF

Filesharing

View Options



Information Access

AIAT SDK

Contextual Menus

ADD



Customization

Appearance 1.0

Improved Toolbox

Desktop Pictures



Assistance and Help




Apple Guide

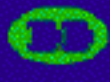


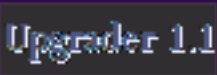

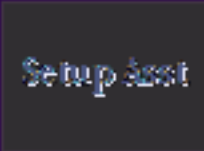

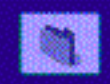
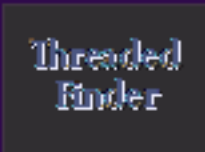
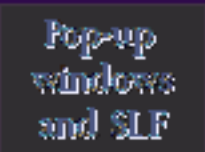
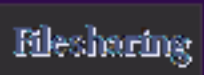
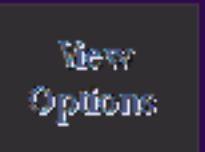
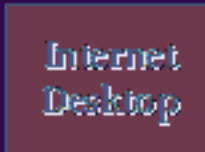



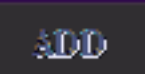



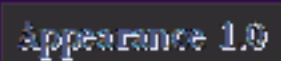
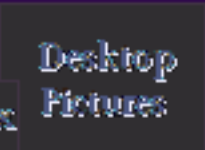
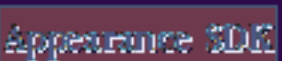
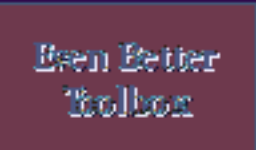
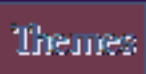



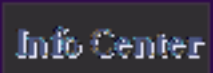
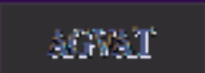

Better Error Messages

Info Center

AGWAT

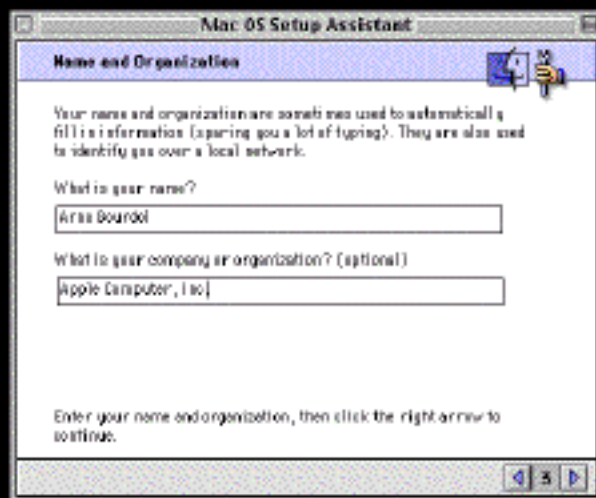
User Experience Roadmap

-  Available now
-  Available this summer
-  Coming up

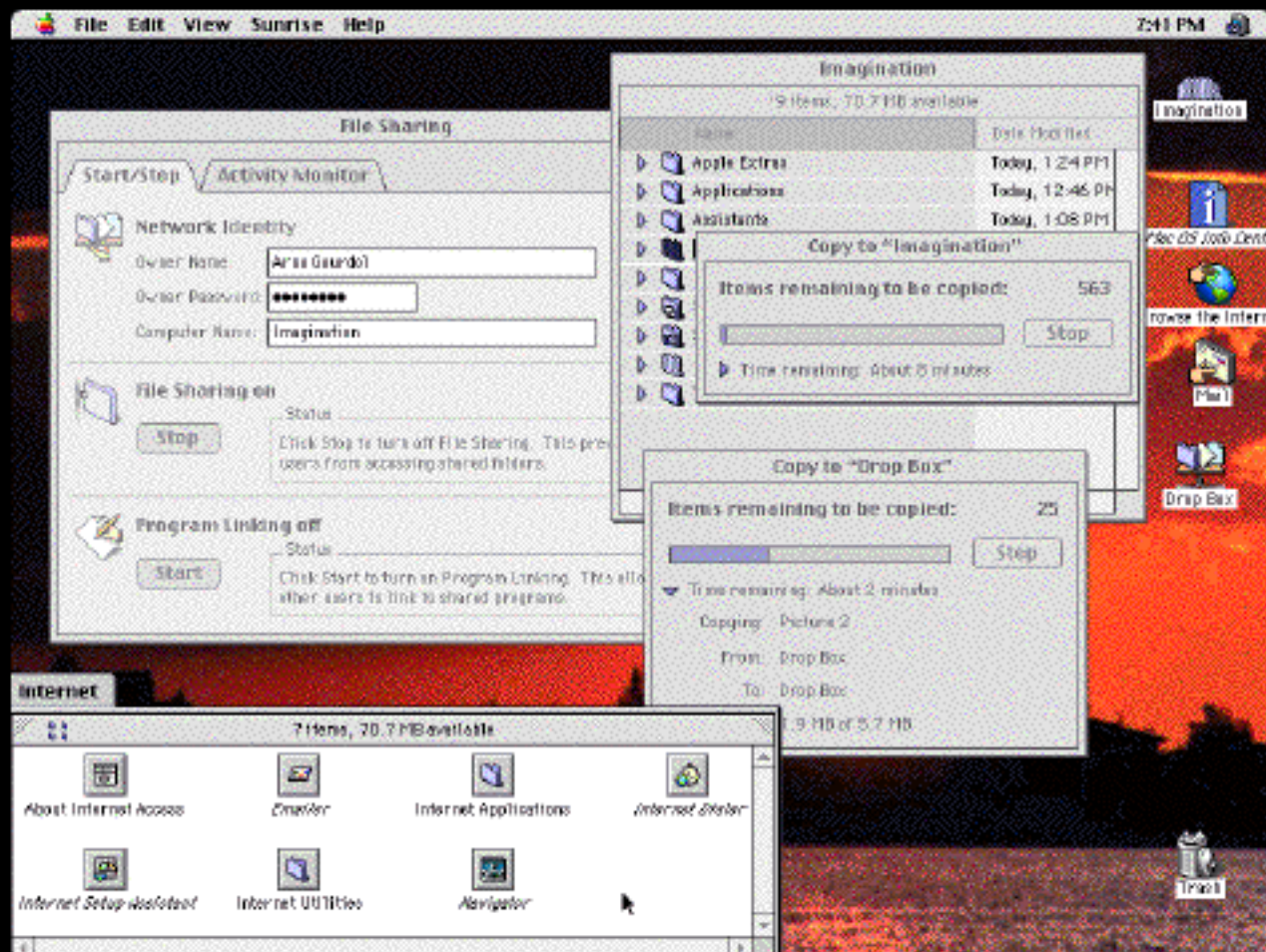
 System Management	 Extensions Manager 4	 Upgrader 1.0  Upgrader 1.1	 Hard Disk Layout	 Setup Assistant	 Application Management
 Desktop Management	 Threaded Finder	 Pop-up windows and SLF	 Filesharing	 View Options	 Internet Desktop
 Information Access	 AIAT SDK	 Contextual Menus	 ADD	 Navigation Services	 Find by Content
 Customization	 Appearance 1.0	 Desktop Pictures	 Appearance SDF	 Even Better Toolbox	 Themes
 Assistance and Help	 Apple Guide	 Better Error Messages	 Info Center	 ASWAT	 Tips

System Management

- **Extensions Manager**
 - ‘CCI™’ and ‘hfdv’—5696 resources
 - See TN 1091
- **Hard Disk Layout**
 - New FindFolder selectors
- **Assistants**
 - See article in *develop 27*



Desktop Management



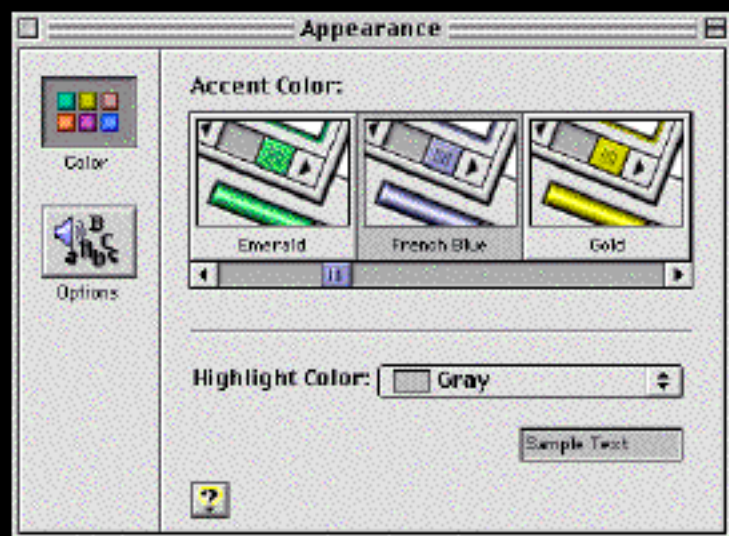
Information Access

- **Apple Information Access Toolkit**
 - A.k.a V-Twin
 - Session 104
- **Contextual menus**
 - Use in your applications
 - Plug-ins
- **Apple Data Detectors**



Customization

- Appearance 1.0
 - Session 102



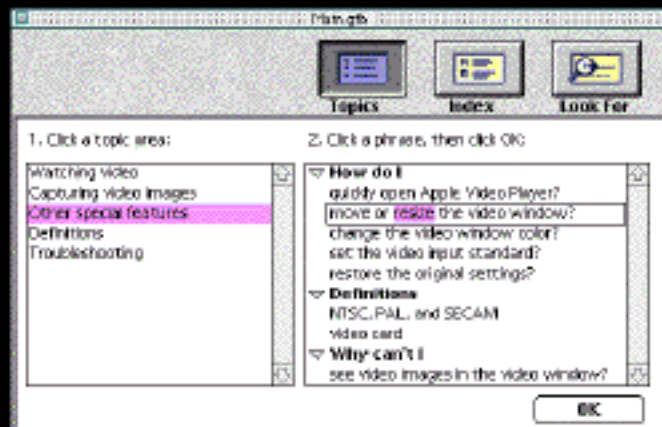
Assistance

- **Apple Guide Visual Authoring Tool**
 - Compatible with existing Guide source files
 - Improved compiler speed
 - Improved Localization support
 - Seed in June 1997

Before AGVAT

```
<define nav button set> "GoBack Only","GoStart"  
<default nav button set> "GoBack Only"  
  
<define event> "GoBack", 's***', 'help', 'gobk'  
<define event> "GoStart", 's***', 'help', 'stac'  
  
# Access window  
<define sequence> "Shortcuts Access: Macintosh S"  
  <seq nav button set> NONE  
  <sequence prompt set> NONE  
  <panel> "Shortcuts Access"  
<end sequence>
```

AGVAT





Contextual Menu Plug-ins

Guy Fullerton

**Contextual Menu Engineer,
User Experience Team**

Contextual Menu Plug-in:

- Standalone code module
- Augments any application's Contextual Menu content
- SOM based
 - C++ , Direct to SOM
 - IDL Compiler
- PowerPC only

Help Commands

Help

Application
Commands

Cut
Copy
Paste

Plug-in Commands

Check Spelling...
Define...
Speak It!





**Contextual
Menu Plug-ins**

Demo

Inheritance

- **Your plug-in must be a subclass of the AbstractCMPlug-in SOM Class:**



MyPlug-in

Initialize

ExamineContext

HandleSelection

PostMenuCleanup



MyPlug-in::Initialize()

- **First method called after an instance of your plug-in is created**
- **Returning an error means your plug-in will not be used**
- **Keep a small footprint!**
 - **Avoid allocating large persistent buffers here**

```
OSStatus Initialize(FSSpec* inFileSpec);
```



MyPlug-in::ExamineContext()

- **Called before the Contextual Menu Manager displays a menu**
- **Gives Plug-ins the opportunity to enhance the menu content**

```
OSStatus ExamineContext(AEDesc* selectionDesc,  
                        Sint32 timeoutInTicks,  
                        AEDesclList* commands,  
                        Boolean* needMoreTime);
```



MyPlug-in::ExamineContext()

- Plug-ins don't modify the menu directly
- Report back commands via an AEDescList of AppleEvent Records

AEDescList of Plug-in commands

Command AERecord:

keyAENName: "Check Spelling ..."

keyContextualMenuCommandID: 1

Command AERecord:

keyAENName: "Define ..."

keyContextualMenuCommandID: 2

Etc ...



Want Submenus?

- Just add `keyContextualMenuSubmenu` to the supercommand's `AERecord`:

AEDescList of Plug-in commands

Command AERecord:

keyAEName: "Text Commands"

keyContextualMenuSubmenu:

AEDescList of subcommands

Command AERecord:

keyAEName: "Check Spelling ..."

keyContextualMenuCommandID: 1

Command AERecord:

keyAEName: "Define ..."

keyContextualMenuCommandID: 2

Etc ...



Respect the Time Limit!

- **Number of ticks available for your Plug-in to do its processing**
- **Report back via the `outNeedMoreTime` parameter when you need to**

```
OSStatus MyPlug-in::ExamineContext(...)
{
    while (true) {
        ...
        if (myTimeUsed >= timeOutInTicks) {
            *needMoreTime = true;
            break;
        }
    }
}
```



MyPlug-in::HandleSelection()

- Called when one of your Plug-in commands is selected
- Given the same selection descriptor as in `ExamineContext()`
- Carry out your command on the descriptor data

```
OSStatus HandleSelection(  
    AEDesc* selectionDesc,  
    SInt32 commandID);
```



MyPlug-in::PostMenuCleanup()

- Called after each Contextual Menu has been torn down
- Always called for every plug-in
 - Dispose of any buffers you allocated in ExamineContext()

```
OSStatus PostMenuCleanup(void);
```



SOM Bug Workaround

- **Create an instance of your SOM class in your initialization routine**
 - Otherwise CMM won't be able to instantiate it by name

```
OSErr MyCFragInitRoutine(  
    CFragInitBlockRec* block)  
{  
    MyPlug-in* Plug-in = new MyPlug-in;  
    delete Plug-in;  
    return noErr;  
}
```



Make your Plug-in Available

- File type 'cmpi'
- Must live in the "Contextual Menu Items" subfolder of the System Folder
- Need an Extended 'cfrg' resource



Cool Opportunities

Across all applications

- **Finder Enhancement**
- **File Compression**
- **Encryption/Encoding**
- **Spelling Checker**
- **Text Matching**
- **Optical Character Recognition**





Apple Data Detectors: Intelligent Integration

Dave Wright

**Technical Lead,
User Experience Team and
Apple Technology Group**

Apple Data Detectors

Agenda

- **Benefits**
- **Overview**
- **Extend what Apple Data Detectors finds!**
- **Extend what Apple Data Detectors does!**



Benefits

How it started!

- **User study from Apple Technology Group**
 - Study focused on information access
 - No problem with finding and filing data
 - Problem with handling data
 - Too many steps. Computer should know what I do to this data
- **Productization in OS Technology Group**
 - Cooperative effort between research and product division

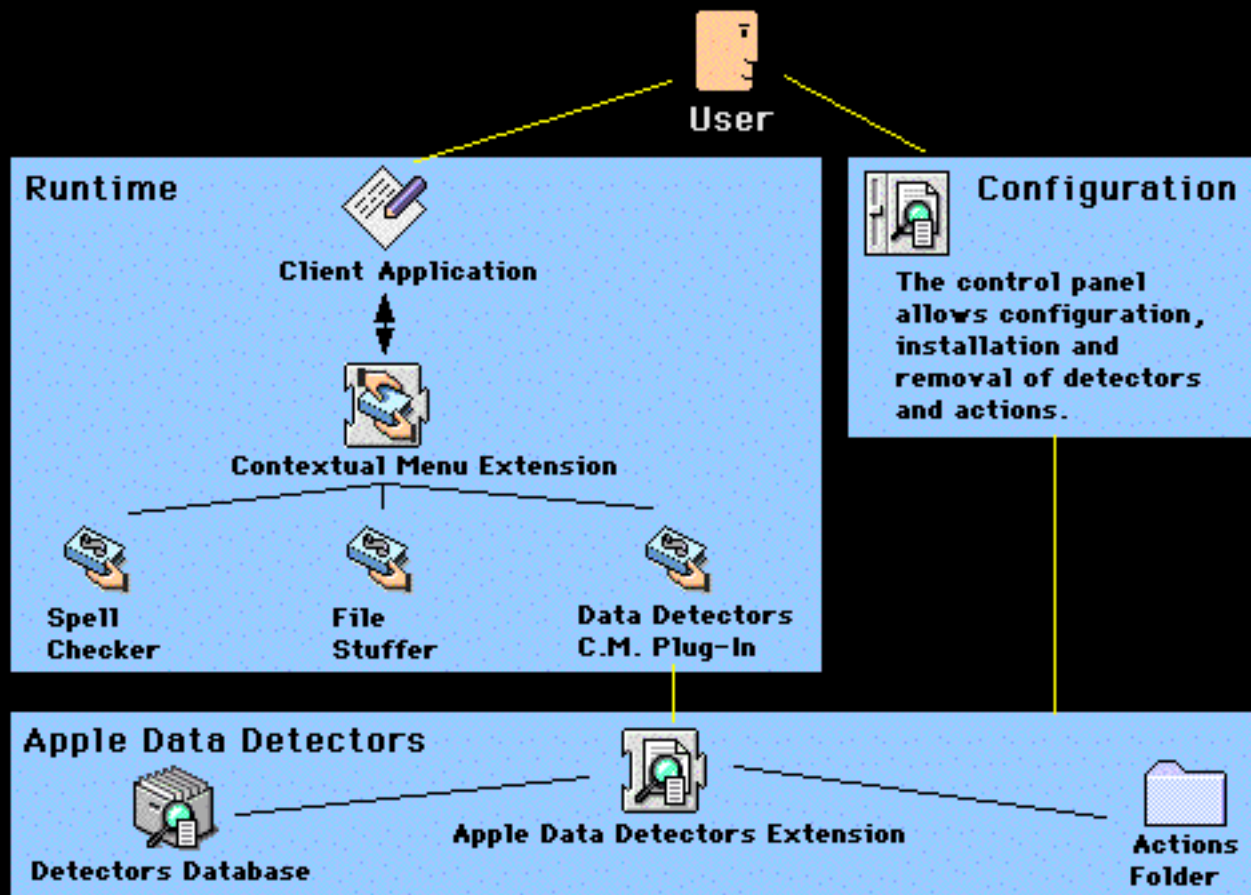




**Apple Data
Detectors**

Demo

Overview





Authoring Detectors

*Extend what Apple
Data Detectors Finds!*

Authoring Detectors

Remember compilers class?

- Regular expressions
- Lots of other fun stuff
 - Parsers
 - Parse Trees
 - Ambiguity



Authoring Detectors

Beefed up regular expression language

- **Syntax**

- `DetectorName = <Description>, Expression`

- **Example**

- `My Holiday = <This detector finds my favorite holiday.>, "July 4"`



Authoring Detectors

Expressions

- **Atomic Expressions**
 - Character Set, example: [a-zA-Z]
 - Character Sequence, example: “abc”



Authoring Detectors

Expressions

- **Atomic Expressions**

- Character Set, example: [a-zA-Z]
- Character Sequence, example: “abc”

- **Compound Expressions:**

- Expression Set, example: { [a-z], “abc” }
- Expression Sequence, example:
([a-z], “abc”)



Authoring Detectors

Expressions

- **Atomic Expressions**

- Character Set, example: [a-zA-Z]
- Character Sequence, example: "abc"

- **Compound Expressions:**

- Expression Set, example: { [a-z], "abc" }
- Expression Sequence, example:
([a-z], "abc")

- **Detector References:**

- Reference to other detectors, example date
= (DayOfWeek, Month ...)



Authoring Detectors

Expression modifiers

- * 0 or more, Example: [A-Z]*
- + 1 or more, Example: {"abc", "def"}+
- ? 0 or 1, (DayOfWeek?, Month, ...)



Authoring Detectors

Examples

- **Date = “April 5, 1997”**
- **Date = ({“January”, “February”}, “5, 1997”)**
- **Date = ({“January”, “February”}, [0-3]?, [0-9], “1997”)**
- **Date = (Month, [0-3]?, [0-9], “199”, [0-9])**
- **Date = (Month, Day, Year?)**



Authoring Detectors

Interaction with actions

- **Export Symbols:**
 - **Syntax:**
 - **Label:Expression**
- **Example:**
 - **EEmailAddress =**
(theUser:User “@” theHost:Host)



Authoring Detectors

Don't clobber each other

- **Name spaces:**
 - Detector file name spaces
 - Global name space
 - \wedge - Export to global name space
 - Example:
 - $\text{Date}^{\wedge} = (\text{DayOfWeek}, \text{Month}^{\wedge})$



Authoring Detectors

User Interface Control

- **Visibility in Control Panel**
 - ! —Explicitly visible in control panel
 - Example:
 - Date ![^] = (...)



Authoring Detectors

User specific data

- Customizable Detectors



Authoring Detectors

Context sensitive data

- Customizable Detectors
- #—Customizable Detectors =
{ “string1”, “string2”, ... }
- Example:
 - Host Domain# = { “.com”, “.net” }



Authoring Detectors

Guidelines

- **Performance**

- Avoid ambiguity!!!!!!!!!!!!

- Don't do this: $[A-D]^*$, $[.A-Z]^*$

- Do this instead: $[A-D]^*$, $(\ "." [A-Z]^*)^*$

- Detectors can be too complex

- Regular Expression Engine vs. Chart Parser





Authoring Actions

*Extend what Apple
Data Detectors Does!*

Authoring Actions

What is an action?

- **One of these in the Actions folder in the Apple Data Detectors Folder:**
 - AppleScript script
 - AppleScript applet
 - AppleEvent aware application
- **With both of these:**
 - Required resource
 - Required Apple Event Handler



Authoring Actions

What is an action?

- **Required resource—‘TEXT’, 1128**
- **Format:**
 - Supported Detector
 - Contextual Menu String
 - Description for control panel
- **Example:**
 - Apple::Email Address
 - Send e-mail with Claris Mailer
 - This action will start a new message to the detected address in Claris Mailer



Authoring Actions

What is an action?

- Same 'TEXT' resource as descriptions of AppleScript scripts
- Why?
 - Discoverability
 - One tool for authoring actions



Authoring Actions

What is an action?

- **Required Apple Event Handler**

- Handle detected data

- **Example:**

```
on handle detected data inData
  set myText to detected text of inData
  tell application "Claris Mailer"
    set recipientlist to {{name_old:""},
                          address_old:myText, type_old:0}}
    create mail subject "" body ""
    recipients recipientlist
  end tell
end handle detected data
```



Authoring Actions

Apple Data Detectors does more than recognize! It's a parser too!

- **Example:**

```
on handle detected data inData
    set myParseTree to parse tree of inData
    set myStartDate to
        findNamedRecord("StartDate",
            myParseTree)
end handle detected data
```



Authoring Actions

A Tool! Really!! 

- **Detector Editor**
 - Provides syntax checking
 - Provides results useful for testing
 - Provides easy view of detector hierarchy



Apple Data Detectors

How to start!

- **Application Developers**
 - Adopt Contextual Menu
 - Make your application scriptable
- **ADD Plug-In Developers**
 - Write detectors for new types of data
 - Our wish list: phone numbers, chemical symbols, dates, times, meetings, mailing addresses, ...
 - Write actions that operate on the detected data



Apple Data Detectors

The innovation continues!

- **Task centered activity**
- **Intelligent integration**
- **How to stay in touch**
 - <http://www.research.apple.com/research/tech/AppleDataDetectors>
 - Gordon Garb, ggarb@apple.com
- **Feedback welcome!**





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 Developers Conference" is overlaid on the image. The word "Worldwide" is in a gold, serif font. The word "Developers" is in a white, serif font and is enclosed in a white rectangular border. The word "Conference" is in a gold, serif font.

Worldwide

Developers

Conference