Unified Logging and Activity Tracing

Logging for the future

Session 721

Steven Szymanski Core OS Engineering
Matthieu Lucas System Applications

Agenda

Introduction

Logging Concepts

Demo

Using the Unified System

Tools

Best Practices

Gathering Logs

Deprications

Introduction

Background

Background

In 2014 Apple introduced Activity Tracing

Background

In 2014 Apple introduced Activity Tracing

We also introduced the concept of Faults and Errors

Background

In 2014 Apple introduced Activity Tracing

We also introduced the concept of Faults and Errors

We recognize that Apple has several logging APIs

Goals

Goals

One common, efficient logging mechanism for both user and kernel mode

Goals

One common, efficient logging mechanism for both user and kernel mode

Maximize information collected while minimizing observer effect

Goals

One common, efficient logging mechanism for both user and kernel mode

Maximize information collected while minimizing observer effect

- Compressing data

Goals

One common, efficient logging mechanism for both user and kernel mode

Maximize information collected while minimizing observer effect

- Compressing data
- Deferring work and data collection

Goals

One common, efficient logging mechanism for both user and kernel mode

Maximize information collected while minimizing observer effect

- Compressing data
- Deferring work and data collection
- Managing log message lifecycle

Goals

One common, efficient logging mechanism for both user and kernel mode

Maximize information collected while minimizing observer effect

- Compressing data
- Deferring work and data collection
- Managing log message lifecycle

We want as much logging on all the time as possible

Goals

One common, efficient logging mechanism for both user and kernel mode

Maximize information collected while minimizing observer effect

- Compressing data
- Deferring work and data collection
- Managing log message lifecycle

We want as much logging on all the time as possible

Design privacy into the system

Features

Features

Improved categorization and filtering of log messages

Features

Improved categorization and filtering of log messages

Logging system collects caller information for you

Features

Improved categorization and filtering of log messages

Logging system collects caller information for you

New builtin type specifiers - simplifies log message preparation

Features

Improved categorization and filtering of log messages

Logging system collects caller information for you

New builtin type specifiers - simplifies log message preparation

New Console application and command-line tool

Features

Improved categorization and filtering of log messages

Logging system collects caller information for you

New builtin type specifiers - simplifies log message preparation

New Console application and command-line tool

Supported on macOS, iOS, tvOS, watchOS, and Simulators

Features

Improved categorization and filtering of log messages

Logging system collects caller information for you

New builtin type specifiers - simplifies log message preparation

New Console application and command-line tool

Supported on macOS, iOS, tvOS, watchOS, and Simulators

Support for Objective-C, C++ and C

Features

Improved categorization and filtering of log messages

Logging system collects caller information for you

New builtin type specifiers - simplifies log message preparation

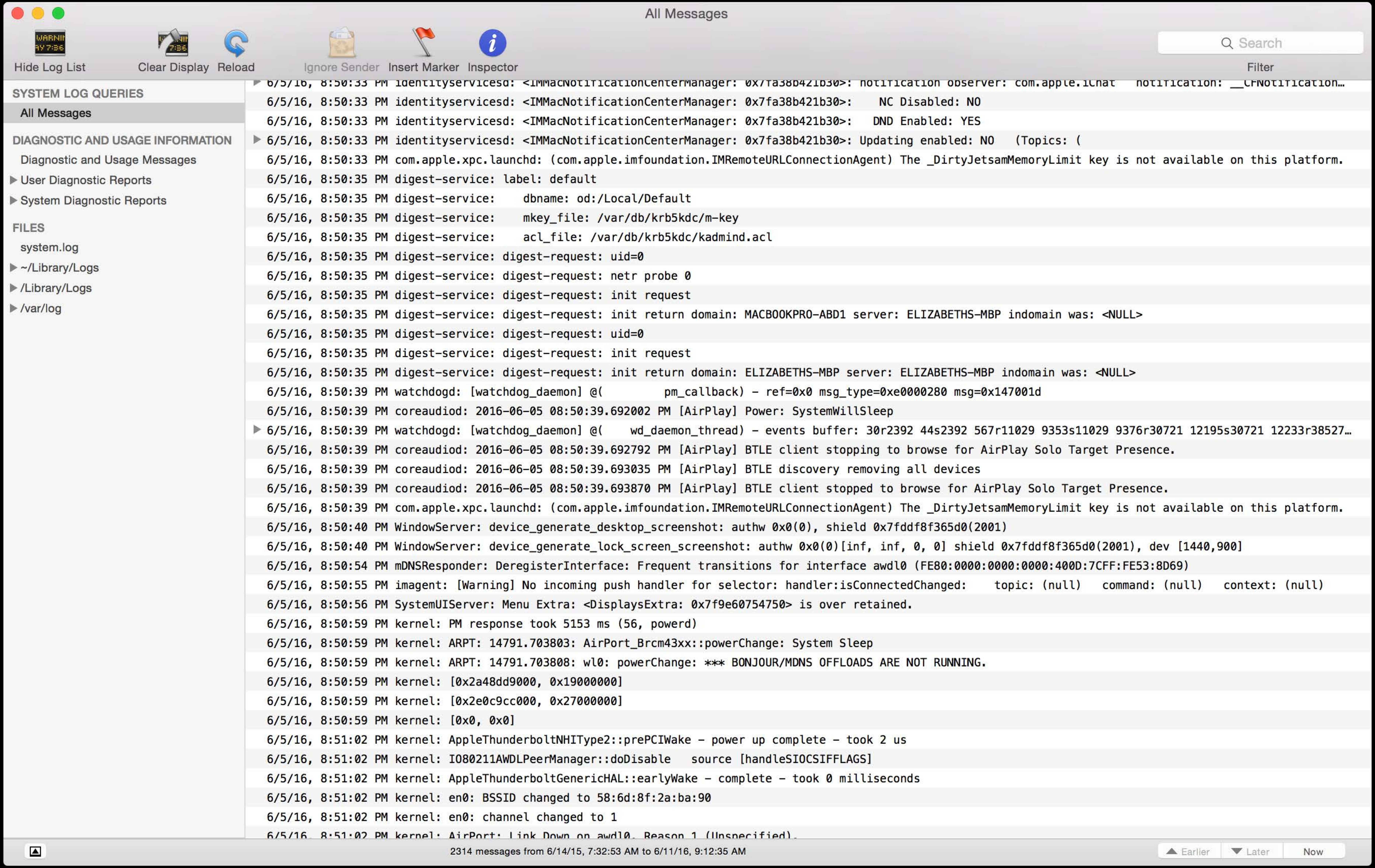
New Console application and command-line tool

Supported on macOS, iOS, tvOS, watchOS, and Simulators

Support for Objective-C, C++ and C

Swift support in upcoming seed

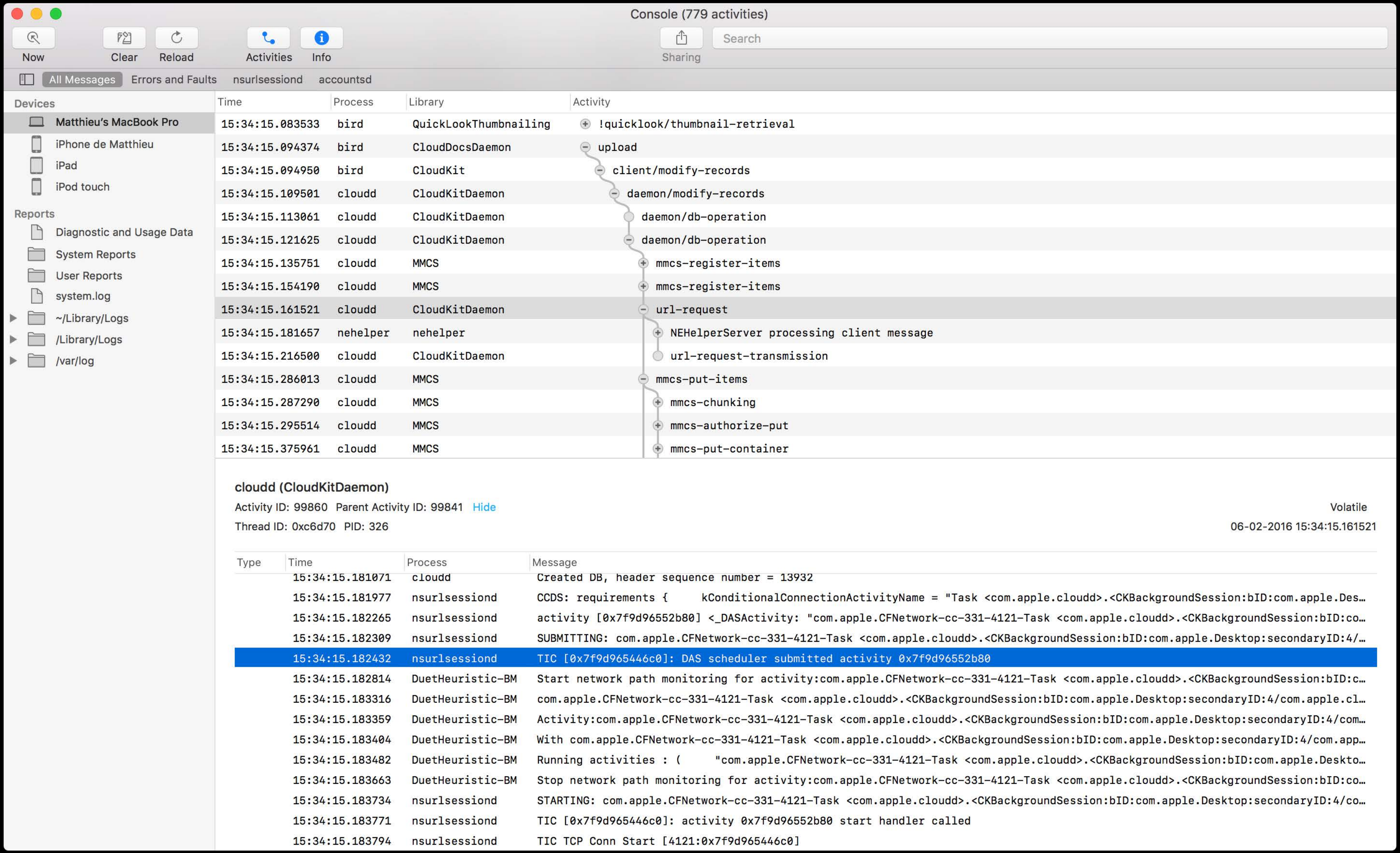
Current Console



```
▶ 6/5/16, 8:50:33 PM identityservicesd: <IMMacNotificationCenterManager: 0x7fa38b421b30>: notification observer: com.apple.iChat   notification: __CFNotification…
  6/5/16, 8:50:33 PM identityservicesd: <IMMacNotificationCenterManager: 0x7fa38b421b30>:    NC Disabled: NO
  6/5/16, 8:50:33 PM identityservicesd: <IMMacNotificationCenterManager: 0x7fa38b421b30>:   DND Enabled: YES
▶ 6/5/16, 8:50:33 PM identityservicesd: <IMMacNotificationCenterManager: 0x7fa38b421b30>: Updating enabled: NO   (Topics: (
  6/5/16, 8:50:33 PM com.apple.xpc.launchd: (com.apple.imfoundation.IMRemoteURLConnectionAgent) The _DirtyJetsamMemoryLimit key is not available on this platform.
  6/5/16, 8:50:35 PM digest-service: label: default
  6/5/16, 8:50:35 PM digest-service:    dbname: od:/Local/Default
  6/5/16, 8:50:35 PM digest-service:    mkey_file: /var/db/krb5kdc/m-key
  6/5/16, 8:50:35 PM digest-service:    acl_file: /var/db/krb5kdc/kadmind.acl
  6/5/16, 8:50:35 PM digest-service: digest-request: uid=0
  6/5/16, 8:50:35 PM digest-service: digest-request: netr probe 0
  6/5/16, 8:50:35 PM digest-service: digest-request: init request
  6/5/16, 8:50:35 PM digest-service: digest-request: init return domain: MACBOOKPRO-ABD1 server: ELIZABETHS-MBP indomain was: <NULL>
  6/5/16, 8:50:35 PM digest-service: digest-request: uid=0
  6/5/16, 8:50:35 PM digest-service: digest-request: init request
  6/5/16, 8:50:35 PM digest-service: digest-request: init return domain: ELIZABETHS-MBP server: ELIZABETHS-MBP indomain was: <NULL>
  6/5/16, 8:50:39 PM watchdogd: [watchdog_daemon] @(         pm_callback) - ref=0x0 msg_type=0xe0000280 msg=0x147001d
  6/5/16, 8:50:39 PM coreaudiod: 2016-06-05 08:50:39.692002 PM [AirPlay] Power: SystemWillSleep
▶ 6/5/16, 8:50:39 PM watchdogd: [watchdog_daemon] @(    wd_daemon_thread) - events buffer: 30r2392 44s2392 567r11029 9353s11029 9376r30721 12195s30721 12233r38527…
  6/5/16, 8:50:39 PM coreaudiod: 2016-06-05 08:50:39.692792 PM [AirPlay] BTLE client stopping to browse for AirPlay Solo Target Presence.
  6/5/16, 8:50:39 PM coreaudiod: 2016-06-05 08:50:39.693035 PM [AirPlay] BTLE discovery removing all devices
  6/5/16, 8:50:39 PM coreaudiod: 2016-06-05 08:50:39.693870 PM [AirPlay] BTLE client stopped to browse for AirPlay Solo Target Presence.
  6/5/16, 8:50:39 PM com.apple.xpc.launchd: (com.apple.imfoundation.IMRemoteURLConnectionAgent) The _DirtyJetsamMemoryLimit key is not available on this platform.
  6/5/16, 8:50:40 PM WindowServer: device_generate_desktop_screenshot: authw 0x0(0), shield 0x7fddf8f365d0(2001)
  6/5/16, 8:50:40 PM WindowServer: device_generate_lock_screen_screenshot: authw 0x0(0)[inf, inf, 0, 0] shield 0x7fddf8f365d0(2001), dev [1440,900]
  6/5/16, 8:50:54 PM mDNSResponder: DeregisterInterface: Frequent transitions for interface awdl0 (FE80:0000:0000:0000:400D:7CFF:FE53:8D69)
  6/5/16, 8:50:55 PM imagent: [Warning] No incoming push handler for selector: handler:isConnectedChanged:    topic: (null)   command: (null)   context: (null)
  6/5/16, 8:50:56 PM SystemUIServer: Menu Extra: <DisplaysExtra: 0x7f9e60754750> is over retained.
  6/5/16, 8:50:59 PM kernel: PM response took 5153 ms (56, powerd)
  6/5/16, 8:50:59 PM kernel: ARPT: 14791.703803: AirPort_Brcm43xx::powerChange: System Sleep
  6/5/16, 8:50:59 PM kernel: ARPT: 14791.703808: wl0: powerChange: *** BONJOUR/MDNS OFFLOADS ARE NOT RUNNING.
  6/5/16, 8:50:59 PM kernel: [0x2a48dd9000, 0x19000000]
  6/5/16, 8:50:59 PM kernel: [0x2e0c9cc000, 0x27000000]
  6/5/16, 8:50:59 PM kernel: [0x0, 0x0]
  6/5/16, 8:51:02 PM kernel: AppleThunderboltNHIType2::prePCIWake - power up complete - took 2 us
  6/5/16, 8:51:02 PM kernel: IO80211AWDLPeerManager::doDisable   source [handleSIOCSIFFLAGS]
  6/5/16, 8:51:02 PM kernel: AppleThunderboltGenericHAL::earlyWake - complete - took 0 milliseconds
  6/5/16, 8:51:02 PM kernel: en0: BSSID changed to 58:6d:8f:2a:ba:90
  6/5/16, 8:51:02 PM kernel: en0: channel changed to 1
  6/5/16, 8:51:02 PM kernel: AirPort: Link Down on awdl0. Reason 1 (Unspecified).
```

2314 messages from 6/14/15, 7:32:53 AM to 6/11/16, 9:12:35 AM

Console Revisited…



```
Time             Process    Library                Activity
15:34:15.083533  bird       QuickLookThumbnailing  !quicklook/thumbnail-retrieval
15:34:15.094374  bird       CloudDocsDaemon        upload
15:34:15.094950  bird       CloudKit                 client/modify-records
15:34:15.109501  cloudd     CloudKitDaemon             daemon/modify-records
15:34:15.113061  cloudd     CloudKitDaemon               daemon/db-operation
15:34:15.121625  cloudd     CloudKitDaemon               daemon/db-operation
15:34:15.135751  cloudd     MMCS                           mmcs-register-items
15:34:15.154190  cloudd     MMCS                           mmcs-register-items
15:34:15.161521  cloudd     CloudKitDaemon                 url-request
15:34:15.181657  nehelper   nehelper                         NEHelperServer processing client message
15:34:15.216500  cloudd     CloudKitDaemon                   url-request-transmission
15:34:15.286013  cloudd     MMCS                           mmcs-put-items
15:34:15.287290  cloudd     MMCS                             mmcs-chunking
15:34:15.295514  cloudd     MMCS                             mmcs-authorize-put
15:34:15.375961  cloudd     MMCS                             mmcs-put-container
```

cloudd (CloudKitDaemon)

Activity ID: 99860 Parent Activity ID: 99841 Hide

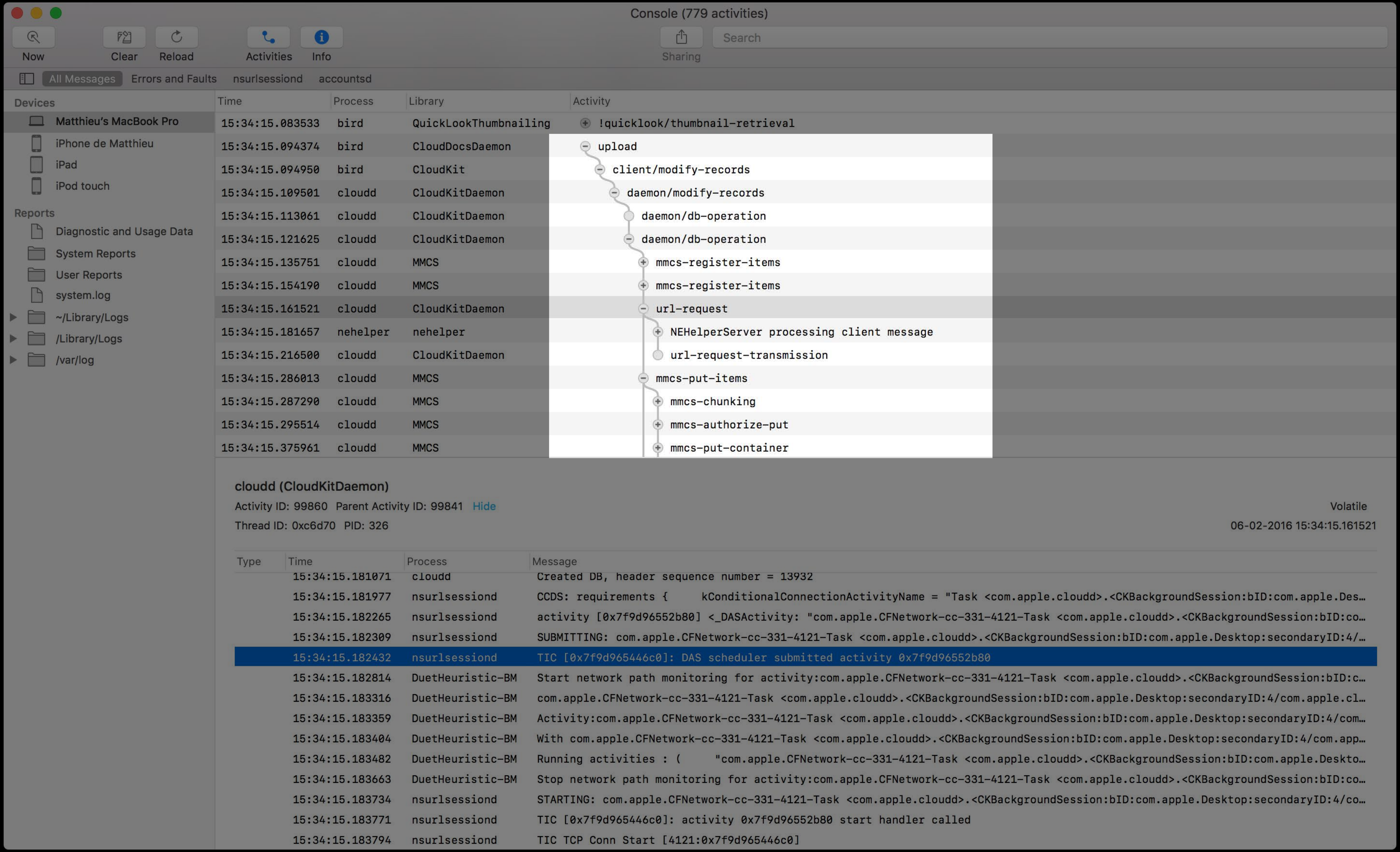
Thread ID: 0xc6d70 PID: 326

Volatile

06-02-2016 15:34:15.161521

```
Type  Time             Process           Message
      15:34:15.181071  cloudd            Created DB, header sequence number = 13932
      15:34:15.181977  nsurlsessiond     CCDS: requirements {     kConditionalConnectionActivityName = "Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Des…
      15:34:15.182265  nsurlsessiond     activity [0x7f9d96552b80] <_DASActivity: "com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:co…
      15:34:15.182309  nsurlsessiond     SUBMITTING: com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/…
      15:34:15.182432  nsurlsessiond     TIC [0x7f9d965446c0]: DAS scheduler submitted activity 0x7f9d96552b80
      15:34:15.182814  DuetHeuristic-BM  Start network path monitoring for activity:com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:c…
      15:34:15.183316  DuetHeuristic-BM  com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/com.apple.cl…
      15:34:15.183359  DuetHeuristic-BM  Activity:com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/com…
      15:34:15.183404  DuetHeuristic-BM  With com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/com.app…
      15:34:15.183482  DuetHeuristic-BM  Running activities : (     "com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Deskto…
      15:34:15.183663  DuetHeuristic-BM  Stop network path monitoring for activity:com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:co…
      15:34:15.183734  nsurlsessiond     STARTING: com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/co…
      15:34:15.183771  nsurlsessiond     TIC [0x7f9d965446c0]: activity 0x7f9d96552b80 start handler called
      15:34:15.183794  nsurlsessiond     TIC TCP Conn Start [4121:0x7f9d965446c0]
```

Console Revisited…



```
Time              Process    Library              Activity
15:34:15.083533   bird       QuickLookThumbnailing  !quicklook/thumbnail-retrieval
15:34:15.094374   bird       CloudDocsDaemon      upload
15:34:15.094950   bird       CloudKit               client/modify-records
15:34:15.109501   cloudd     CloudKitDaemon           daemon/modify-records
15:34:15.113061   cloudd     CloudKitDaemon             daemon/db-operation
15:34:15.121625   cloudd     CloudKitDaemon             daemon/db-operation
15:34:15.135751   cloudd     MMCS                         mmcs-register-items
15:34:15.154190   cloudd     MMCS                         mmcs-register-items
15:34:15.161521   cloudd     CloudKitDaemon               url-request
15:34:15.181657   nehelper   nehelper                       NEHelperServer processing client message
15:34:15.216500   cloudd     CloudKitDaemon                 url-request-transmission
15:34:15.286013   cloudd     MMCS                         mmcs-put-items
15:34:15.287290   cloudd     MMCS                           mmcs-chunking
15:34:15.295514   cloudd     MMCS                           mmcs-authorize-put
15:34:15.375961   cloudd     MMCS                           mmcs-put-container
```

cloudd (CloudKitDaemon)

Activity ID: 99860 Parent Activity ID: 99841 Hide

Thread ID: 0xc6d70 PID: 326

Volatile

06-02-2016 15:34:15.161521

```
Type  Time              Process            Message
      15:34:15.181071   cloudd             Created DB, header sequence number = 13932
      15:34:15.181977   nsurlsessiond      CCDS: requirements {     kConditionalConnectionActivityName = "Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Des…
      15:34:15.182265   nsurlsessiond      activity [0x7f9d96552b80] <_DASActivity: "com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:co…
      15:34:15.182309   nsurlsessiond      SUBMITTING: com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/…
      15:34:15.182432   nsurlsessiond      TIC [0x7f9d965446c0]: DAS scheduler submitted activity 0x7f9d96552b80
      15:34:15.182814   DuetHeuristic-BM   Start network path monitoring for activity:com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:c…
      15:34:15.183316   DuetHeuristic-BM   com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/com.apple.cl…
      15:34:15.183359   DuetHeuristic-BM   Activity:com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/com…
      15:34:15.183404   DuetHeuristic-BM   With com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/com.app…
      15:34:15.183482   DuetHeuristic-BM   Running activities : (     "com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Deskto…
      15:34:15.183663   DuetHeuristic-BM   Stop network path monitoring for activity:com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:co…
      15:34:15.183734   nsurlsessiond      STARTING: com.apple.CFNetwork-cc-331-4121-Task <com.apple.cloudd>.<CKBackgroundSession:bID:com.apple.Desktop:secondaryID:4/co…
      15:34:15.183771   nsurlsessiond      TIC [0x7f9d965446c0]: activity 0x7f9d96552b80 start handler called
      15:34:15.183794   nsurlsessiond      TIC TCP Conn Start [4121:0x7f9d965446c0]
```

Logging Concepts

Adoption

Adoption

If you want to use the new Unified Logging system

- Build with the macOS 10.12, iOS 10.0, tvOS 10.0 or watchOS 3.0 SDK
- Legacy APIs (NSLog, asl_log_message, syslog…) redirected into new system
- Log data will be in new format and location

Adoption

If you want to use the new Unified Logging system

- Build with the macOS 10.12, iOS 10.0, tvOS 10.0 or watchOS 3.0 SDK
- Legacy APIs (NSLog, asl_log_message, syslog…) redirected into new system
- Log data will be in new format and location

If you don't want to use the new Unified Logging system

- Build with macOS10.11, iOS 9.0, tvOS 9.0 and watchOS 2.0 SDK
- No changes

New File Formats

New File Formats

Log data is kept in a compressed binary format: .tracev3 files

New File Formats

Log data is kept in a compressed binary format: .tracev3 files

Stored under /var/db/diagnostics/ with support in /var/db/uuidtext

New File Formats

Log data is kept in a compressed binary format: .tracev3 files

Stored under /var/db/diagnostics/ with support in /var/db/uuidtext

New tools to access the stored and live log messages

New File Formats

Log data is kept in a compressed binary format: .tracev3 files

Stored under /var/db/diagnostics/ with support in /var/db/uuidtext

New tools to access the stored and live log messages

- Because data is stored in binary format, you MUST use new tools to access files

New File Formats

Log data is kept in a compressed binary format: .tracev3 files

Stored under /var/db/diagnostics/ with support in /var/db/uuidtext

New tools to access the stored and live log messages

- Because data is stored in binary format, you MUST use new tools to access files

New .logarchive format for portability of logs

New File Formats

Log data is kept in a compressed binary format: .tracev3 files

Stored under /var/db/diagnostics/ with support in /var/db/uuidtext

New tools to access the stored and live log messages

- Because data is stored in binary format, you MUST use new tools to access files

New .logarchive format for portability of logs

Subsystems and Categories

NEW

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Can be used to control how log messages are filtered and displayed

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Can be used to control how log messages are filtered and displayed

A subsystem can contain multiple categories

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Can be used to control how log messages are filtered and displayed

A subsystem can contain multiple categories

You can use as many subsystems and categories as needed

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Can be used to control how log messages are filtered and displayed

A subsystem can contain multiple categories

You can use as many subsystems and categories as needed

Example:

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Can be used to control how log messages are filtered and displayed

A subsystem can contain multiple categories

You can use as many subsystems and categories as needed

Example:

Subsystem	Category
com.your-company.your-application	setup, inprogress, teardown

Subsystems and Categories

NEW

Log messages can be associated with a subsystem and category

Can be used to control how log messages are filtered and displayed

A subsystem can contain multiple categories

You can use as many subsystems and categories as needed

Example:

Subsystem	Category
com.your-company.your-application	setup, inprogress, teardown
com.your-company.test.your-application	test

Logging Behavior

NEW

Logging Behavior

NEW

Each log message has a level determined by the API used

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Each basic level has two characteristics that can be set for system, subsystem, or category

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Each basic level has two characteristics that can be set for system, subsystem, or category

- Is is enabled? (Default messages are always enabled)

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Each basic level has two characteristics that can be set for system, subsystem, or category

- Is is enabled? (Default messages are always enabled)
- Is it stored to disk or memory?

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Each basic level has two characteristics that can be set for system, subsystem, or category

- Is is enabled? (Default messages are always enabled)
- Is it stored to disk or memory?

The levels are hierarchical

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Each basic level has two characteristics that can be set for system, subsystem, or category

- Is is enabled? (Default messages are always enabled)
- Is it stored to disk or memory?

The levels are hierarchical

- So setting Debug to go to disk implies that Info will also go to disk

Logging Behavior

NEW

Each log message has a level determined by the API used

- Three basic levels—Default, Info, Debug
- Two special levels—Fault, Error

Each basic level has two characteristics that can be set for system, subsystem, or category

- Is is enabled? (Default messages are always enabled)
- Is it stored to disk or memory?

The levels are hierarchical

- So setting Debug to go to disk implies that Info will also go to disk

Behavior can be customized by installing profiles or, on macOS, via log command

Standard Behavior

NEW

Standard Behavior

NEW

Message Level	Enabled	Destination
DEFAULT LEVEL	ALWAYS	DISK
INFO LEVEL	YES	MEMORY
DEBUG LEVEL	NO	N/A

Standard Behavior

NEW

Message Level	Enabled	Destination
DEFAULT LEVEL	ALWAYS	DISK
INFO LEVEL	YES	MEMORY
DEBUG LEVEL	NO	N/A
ERROR	ALWAYS	DISK
FAULT	ALWAYS	DISK

Privacy

NEW

Privacy

NEW

Prevent accidental logging of Personally Identifiable Information (PII)

Privacy

NEW

Prevent accidental logging of Personally Identifiable Information (PII)

Dynamic strings, collections, arrays, etc. are assumed to be private

Faults and Errors

NEW

Faults and Errors

NEW

We do extra work saving additional information on Fault or Error

Faults and Errors

NEW

We do extra work saving additional information on Fault or Error

Errors represent issues discovered within a given application/library

Faults and Errors

NEW

We do extra work saving additional information on Fault or Error

Errors represent issues discovered within a given application/library

Faults represent more global problems in the system

Faults and Errors

NEW

We do extra work saving additional information on Fault or Error

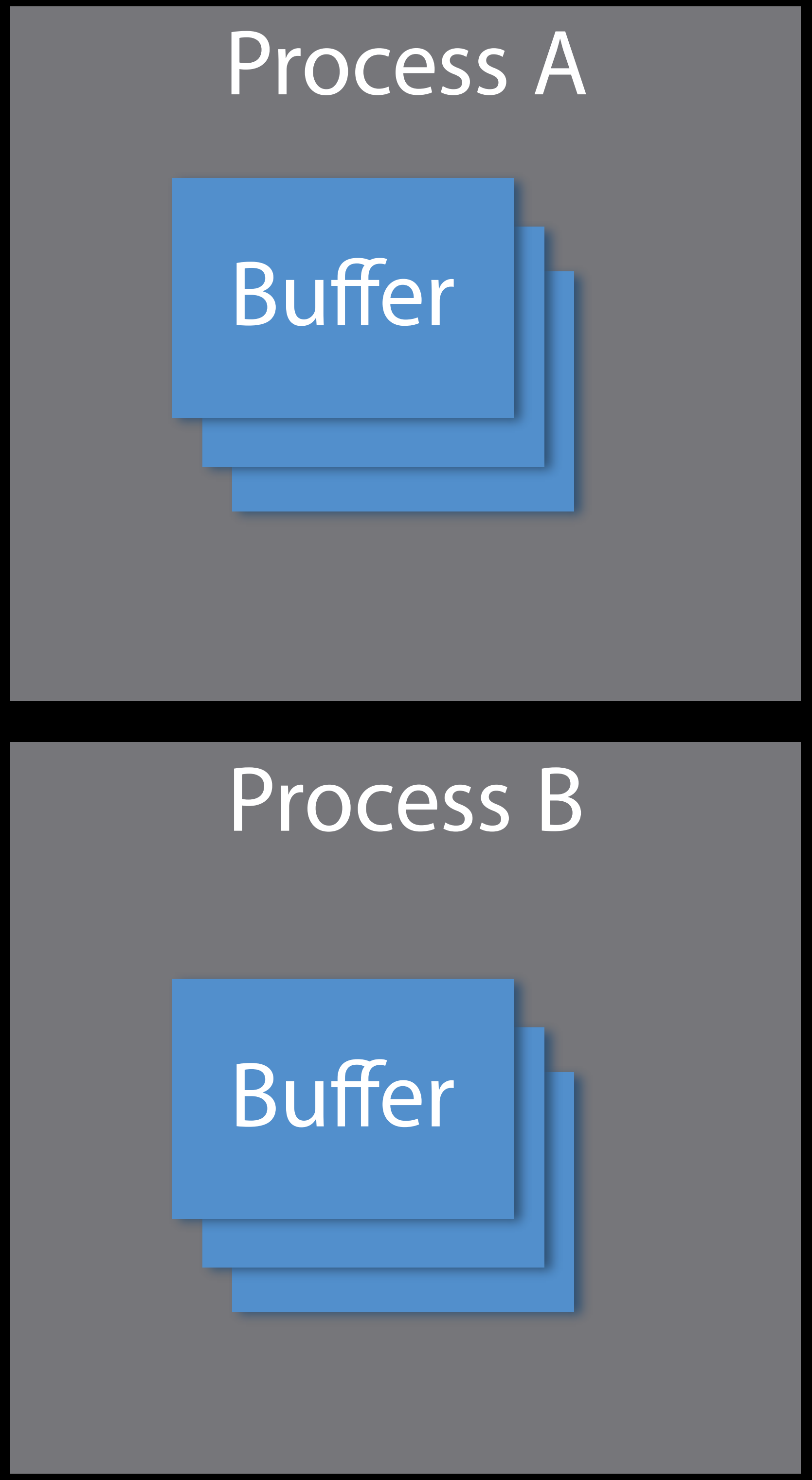
Errors represent issues discovered within a given application/library

Faults represent more global problems in the system

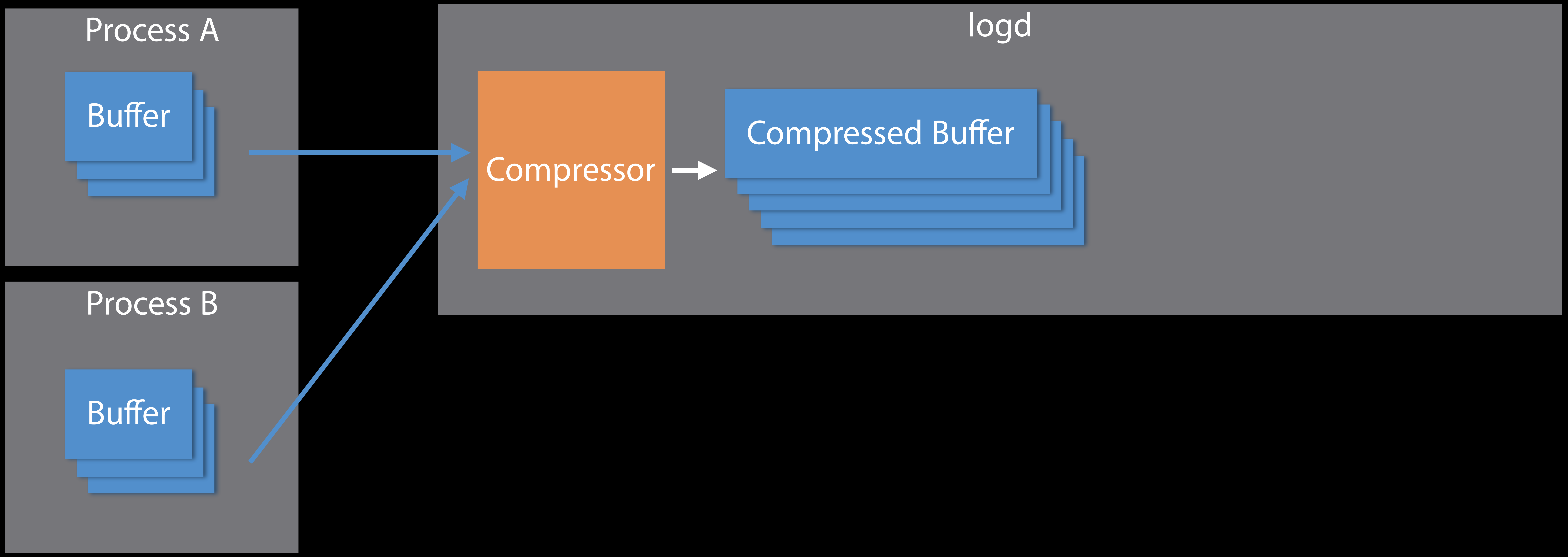
Faults and Error log information is captured into a separate set of log files

Architecture

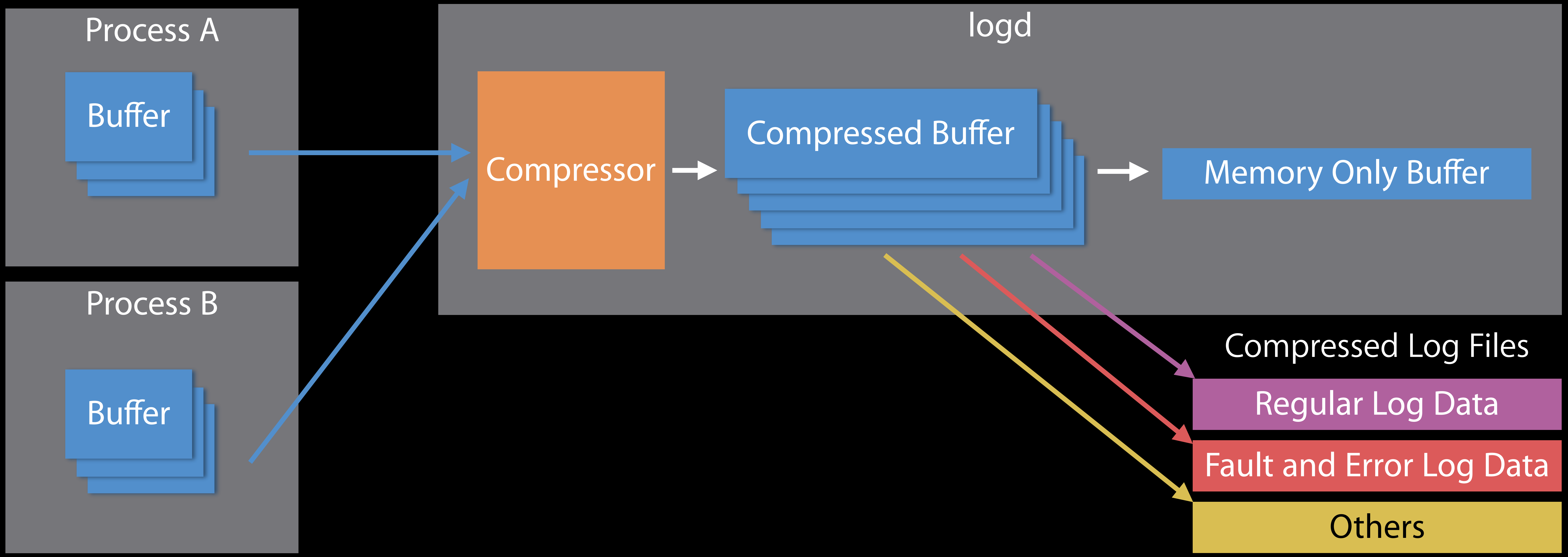
Architecture



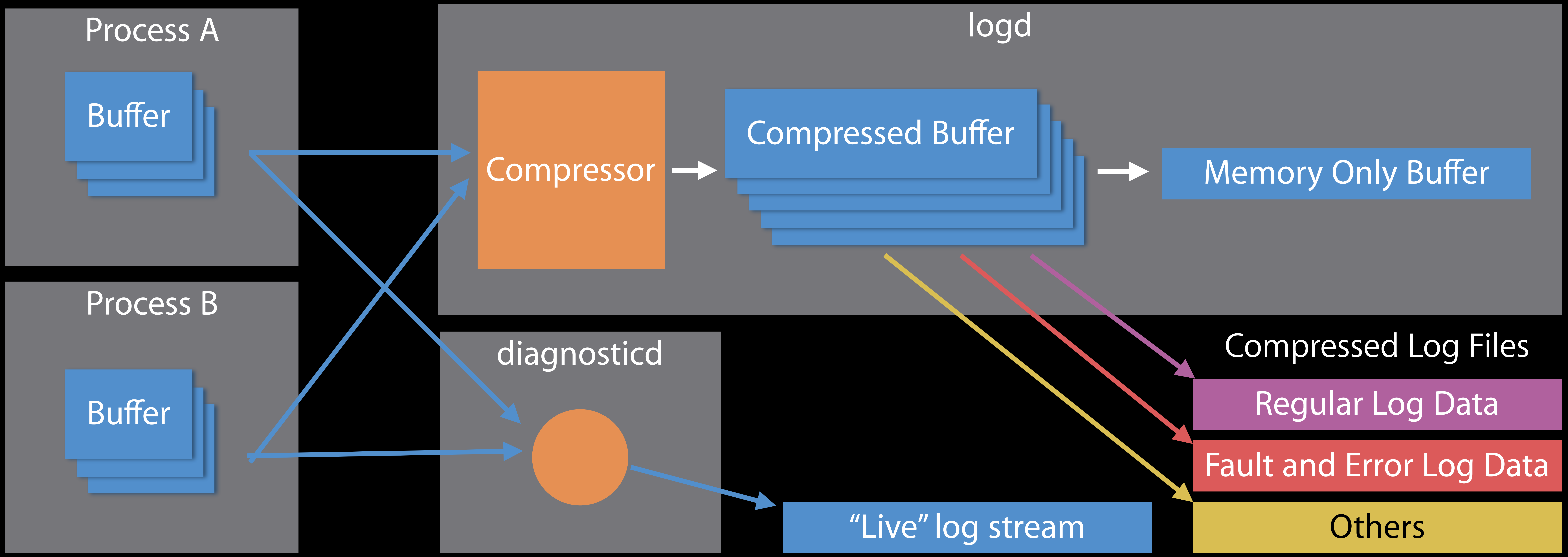
Architecture



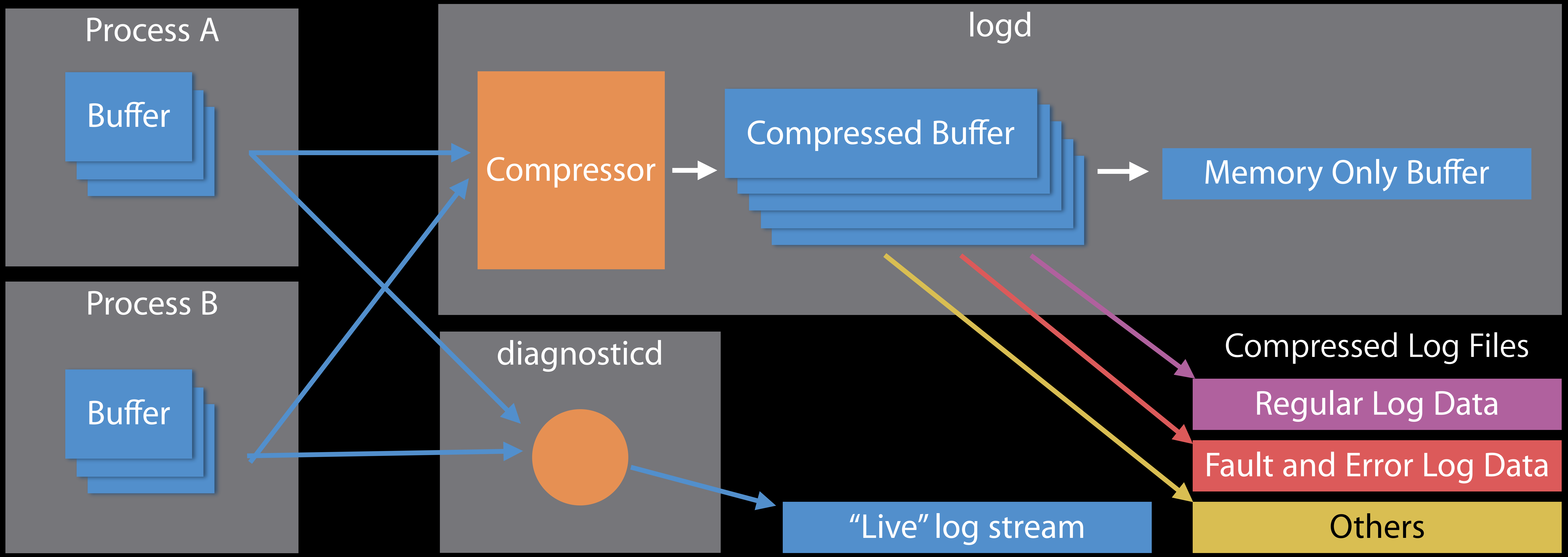
Architecture



Architecture



Architecture



Profile can change routing and rules for given applications or subsystems

Demo

Console demo

Using the Unified System

Summary of New APIs

NEW

API	Destination	Description
os_log	Disk	Default logging level that is always captured
os_log_info	Memory	Additional information (defaults to memory-only buffers)
os_log_debug	Off	Debug level content (off-by default)
os_log_error	Disk	Process local error
os_log_fault	Disk	System-level error (usually involves multiple processes)
os_log_create	n/a	Create a log object for custom behaviors

Creating a Log Object

NEW

Creating a Log Object

NEW

```
os_log_t log = os_log_create("com.your_company.subsystem", "network");
```

Create thread-safe singleton object that controls behavior of log messages

Creating a Log Object

NEW

```
os_log_t log = os_log_create("com.your_company.subsystem", "network");
```

Create thread-safe singleton object that controls behavior of log messages

Defaults to system-behavior

Creating a Log Object

NEW

```
os_log_t log = os_log_create("com.your_company.subsystem", "network");
```

Create thread-safe singleton object that controls behavior of log messages

Defaults to system-behavior

Usage

```
os_log(log, "This happened");
```

Creating a Log Object

NEW

```
os_log_t log = os_log_create("com.your_company.subsystem", "network");
```

Create thread-safe singleton object that controls behavior of log messages

Defaults to system-behavior

Usage

```
os_log(log, "This happened");
```

Reference to category and subsystem stored with every log message

Creating a Log Object

NEW

```
os_log_t log = os_log_create("com.your_company.subsystem", "network");
```

Create thread-safe singleton object that controls behavior of log messages

Defaults to system-behavior

Usage

```
os_log(log, "This happened");
```

Reference to category and subsystem stored with every log message

Or use `OS_LOG_DEFAULT`

- For messages not associated with subsystem/category

Creating a Log Object

NEW

```
os_log_t log = os_log_create("com.your_company.subsystem", "network");
```

Create thread-safe singleton object that controls behavior of log messages

Defaults to system-behavior

Usage

```
os_log(log, "This happened");
```

Reference to category and subsystem stored with every log message

Or use `OS_LOG_DEFAULT`

- For messages not associated with subsystem/category

Built-in Type Formatters

NEW

We all spend too much code converting binary information to strings to log

Built-in Type Formatters

NEW

We all spend too much code converting binary information to strings to log

Built-in decoding for common values

`"%{time_t}d"` or `"%{errno}d"`

Built-in Type Formatters

NEW

We all spend too much code converting binary information to strings to log

Built-in decoding for common values

`"%{time_t}d"` or `"%{errno}d"`

Arbitrary binary data using a new format type

`"%.*P"`

Built-in Type Formatters

NEW

We all spend too much code converting binary information to strings to log

Built-in decoding for common values

`"%{time_t}d"` or `"%{errno}d"`

Arbitrary binary data using a new format type

`"%.*P"`

Built-in decoding for binary-types

`“%{uuid_t}.16P"`

Example Type Formatters

NEW

Type	Format String	Example Output
time_t	%{time_t}d	2016-01-12 19:41:37
timeval	%{timeval}.*P	2016-01-12 19:41:37.774236
timespec	%{timespec}.*P	2016-01-12 19:41:37.774236823
errno	%{errno}d	Broken pipe
uuid_t	%{uuid_t}.16P %{uuid_t}.*P	10742E39-0657-41F8-AB99-878C5EC2DCAA
sockaddr	%{network:sockaddr}.*P	fe80::f:86ff:fee9:5c16 17.43.23.87
in_addr	%{network:in_addr}d	17.43.23.87
in6_addr	%{network:in6_addr}.16P	fe80::f:86ff:fee9:5c16

Per Parameter Privacy

NEW

Per Parameter Privacy

NEW

Privacy is handled on a parameter by parameter basis

Per Parameter Privacy

NEW

Privacy is handled on a parameter by parameter basis

Scalars and static strings are assumed to be public

Per Parameter Privacy

NEW

Privacy is handled on a parameter by parameter basis

Scalars and static strings are assumed to be public

Dynamic strings, collections, and objects are assumed to be private

Per Parameter Privacy

NEW

Privacy is handled on a parameter by parameter basis

Scalars and static strings are assumed to be public

Dynamic strings, collections, and objects are assumed to be private

Can be overridden on a per-parameter basis

`"%{public}@"` or `"%{private}d"`

Per Parameter Privacy

NEW

Privacy is handled on a parameter by parameter basis

Scalars and static strings are assumed to be public

Dynamic strings, collections, and objects are assumed to be private

Can be overridden on a per-parameter basis

`"%{public}@"` or `"%{private}d"`

Combine privacy and formatting

```
"%{public, uuid_t}.16P”
```

```
/*
 * Log Message Simplification
 */
```

```
/*
 * Log Message Simplification
 */

// Old way:
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
    free(addr_desc);
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
    free(addr_desc);
}
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
    free(addr_desc);
}

// New way:
```

```
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
    free(addr_desc);
}

// New way:
os_log_info(OS_LOG_DEFAULT,
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
    free(addr_desc);
}

// New way:
os_log_info(OS_LOG_DEFAULT,
            "fd: %d, uuid: %{uuid_t}.16P, IP: %{network:sockaddr}.*P”,
/*
 * Log Message Simplification
 */

// Old way:
if (LogLevelEnabled(info)) {
    uuid_string_t uuid_str;
    uuid_unparse_upper(uuid, uuid_str);
    char *addr_desc = _convert_sockaddr(&sa);
    NSLog(@"%s (%s:%d) - fd: %d, uuid: %s, IP: %s”,
__PRETTY_FUNCTION__, __FILE__, __LINE__, fd, uuid_st, addr_desc);
    free(addr_desc);
}

// New way:
os_log_info(OS_LOG_DEFAULT,
            "fd: %d, uuid: %{uuid_t}.16P, IP: %{network:sockaddr}.*P”,
                fd, uuid, sa->sa_len, &sa);
```

```
/*
 * Example Code
 */
```

```
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
```

```
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
os_log_t time_log = os_log_create("com.apple.logging.example", "timestamp");
```

```
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
os_log_t time_log = os_log_create("com.apple.logging.example", "timestamp");

os_log(general_log, "running example code”);
```

```c
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
os_log_t time_log = os_log_create("com.apple.logging.example", "timestamp");

os_log(general_log, "running example code”);

os_log_info(general_log, "processing file %{public}s”, filename);
```

```
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
os_log_t time_log = os_log_create("com.apple.logging.example", "timestamp");

os_log(general_log, "running example code”);

os_log_info(general_log, "processing file %{public}s”, filename);

int fd = open(filename, O_RDONLY);
if (fd < 0) {
    os_log_error(general_log, "Cannot open file %{public}s – %{errno}d", filename, errno);
    . . .
}
```

```
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
os_log_t time_log = os_log_create("com.apple.logging.example", "timestamp");

os_log(general_log, "running example code”);

os_log_info(general_log, "processing file %{public}s”, filename);

int fd = open(filename, O_RDONLY);
if (fd < 0) {
    os_log_error(general_log, "Cannot open file %{public}s – %{errno}d", filename, errno);
    . . .
}

struct stat sb;
if (fstat(fd, &sb) < 0) {
    os_log_fault(general_log, "Failed to fstat %{public}s – %{errno}d", filename, errno);
    . . .
}
```

```
/*
 * Example Code
 */

os_log_t general_log = os_log_create("com.apple.logging.example", "general");
os_log_t time_log = os_log_create("com.apple.logging.example", "timestamp");

os_log(general_log, "running example code”);

os_log_info(general_log, "processing file %{public}s”, filename);

int fd = open(filename, O_RDONLY);
if (fd < 0) {
    os_log_error(general_log, "Cannot open file %{public}s - %{errno}d", filename, errno);
    . . .
}

struct stat sb;
if (fstat(fd, &sb) < 0) {
    os_log_fault(general_log, "Failed to fstat %{public}s - %{errno}d", filename, errno);
    . . .
}

os_log_info(time_log,
     "status for file %{public}s, atime:%{timespec}.*P, mtime:%{timespec}.*P, ctime:%
     {timespec}.*P",
     filename,
     sizeof(struct timespec), &sb.st_atimespec,
     sizeof(struct timespec), &sb.st_mtimespec,
     sizeof(struct timespec), &sb.st_ctimespec);
```

Activity API Improvements

Activity API Improvements

Activities are now objects that can be stored and re-used

Activity API Improvements

Activities are now objects that can be stored and re-used

- Direct control of activity relationships during creation

Activity API Improvements

Activities are now objects that can be stored and re-used

- Direct control of activity relationships during creation

New API to auto-scope activities within your code

Improved Activity APIs

NEW

os_activity_create	Creates a new activity object
os_activity_scope	Makes an execution block a part of an activity
os_activity_apply	Invokes a block scoped to a given activity
os_activity_label_useraction	Label an activity as a user action (UI-based activities)

```
/*
 * New Activity API Example
 */
```

```
/*
 * New Activity API Example
 */

os_activity_t init_activity = os_activity_create("Init", OS_ACTIVITY_CURRENT,
OS_ACTIVITY_FLAG_DEFAULT);
```

```
/*
 * New Activity API Example
 */

os_activity_t init_activity = os_activity_create("Init", OS_ACTIVITY_CURRENT,
OS_ACTIVITY_FLAG_DEFAULT);

os_activity_t verify_activity = os_activity_create("Verify", init_activity,
OS_ACTIVITY_FLAG_DEFAULT);
```

```
/*
 * New Activity API Example
 */

os_activity_t init_activity = os_activity_create("Init", OS_ACTIVITY_CURRENT,
OS_ACTIVITY_FLAG_DEFAULT);

os_activity_t verify_activity = os_activity_create("Verify", init_activity,
OS_ACTIVITY_FLAG_DEFAULT);

if (isReady) {
    os_activity_scope(verify_activity);
    // All of the following work is done under “verification activity scope”
    . . .
}
```

```
/*
 * New Activity API Example
 */

os_activity_t init_activity = os_activity_create("Init", OS_ACTIVITY_CURRENT,
OS_ACTIVITY_FLAG_DEFAULT);

os_activity_t verify_activity = os_activity_create("Verify", init_activity,
OS_ACTIVITY_FLAG_DEFAULT);

if (isReady) {
    os_activity_scope(verify_activity);
    // All of the following work is done under “verification activity scope”
    . . .
}
    // This is not part of that activity
```

```
/*
 * New Activity API Example
 */

os_activity_t init_activity = os_activity_create("Init", OS_ACTIVITY_CURRENT,
OS_ACTIVITY_FLAG_DEFAULT);

os_activity_t verify_activity = os_activity_create("Verify", init_activity,
OS_ACTIVITY_FLAG_DEFAULT);

if (isReady) {
    os_activity_scope(verify_activity);
    // All of the following work is done under “verification activity scope”
    . . .
}
    // This is not part of that activity



os_activity_apply(init_activity, ^{
    // do some work based on the “init_activity”
    . . .
});
```

Tools

Console

NEW

Console

NEW

View live content from a system

Console

NEW

View live content from a system

Open log archives

Console

NEW

View live content from a system

Open log archives

New Activity centric view of logging and tracing

Console

NEW

View live content from a system

Open log archives

New Activity centric view of logging and tracing

Advanced filtering and searching

Console

NEW

View live content from a system

Open log archives

New Activity centric view of logging and tracing

Advanced filtering and searching

Device support

log Command Line Tool

NEW

log Command Line Tool

NEW

Same functionality as Console from the command line

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
```

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
$ log stream --predicate 'eventMessage contains "my message"'
```

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
$ log stream --predicate 'eventMessage contains "my message"'
```

Display a log file or archive

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
$ log stream --predicate 'eventMessage contains "my message"'
```

Display a log file or archive

```
$ log show system_logs.logarchive
```

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
$ log stream --predicate 'eventMessage contains "my message"'
```

Display a log file or archive

```
$ log show system_logs.logarchive
```

Enable debug for your subsystem on macOS

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
$ log stream --predicate 'eventMessage contains "my message"'
```

Display a log file or archive

```
$ log show system_logs.logarchive
```

Enable debug for your subsystem on macOS

```
$ log config --mode "level:debug" --subsystem com.mycorp.myapp
```

log Command Line Tool

NEW

Same functionality as Console from the command line

Stream live log messages

```
$ log stream
$ log stream --predicate 'eventMessage contains "my message"'
```

Display a log file or archive

```
$ log show system_logs.logarchive
```

Enable debug for your subsystem on macOS

```
$ log config --mode "level:debug" --subsystem com.mycorp.myapp
```

Coming Soon

NEW

Coming Soon

NEW

Tools for accessing new log information from 10.11 are coming soon

Coming Soon

NEW

Tools for accessing new log information from 10.11 are coming soon

But… in the meantime

Coming Soon

NEW

Tools for accessing new log information from 10.11 are coming soon

But… in the meantime

```
$ xcrun simctl spawn booted log show system_logs.logarchive
```

Coming Soon

NEW

Tools for accessing new log information from 10.11 are coming soon

But… in the meantime

```
$ xcrun simctl spawn booted log show system_logs.logarchive
```

Best Practices

Logging Etiquette

Logging Etiquette

Ensure messages contain only information useful for debugging

Logging Etiquette

Ensure messages contain only information useful for debugging

Let us do the formatting for you—leverage built-in formatters

Logging Etiquette

Ensure messages contain only information useful for debugging

Let us do the formatting for you—leverage built-in formatters

Avoid creating wrapper functions for `os_log*` APIs

Logging Etiquette

Ensure messages contain only information useful for debugging

Let us do the formatting for you—leverage built-in formatters

Avoid creating wrapper functions for `os_log*` APIs

Log only what you need from collections (Dictionaries, Arrays, etc.)

Logging Etiquette

Ensure messages contain only information useful for debugging

Let us do the formatting for you—leverage built-in formatters

Avoid creating wrapper functions for `os_log*` APIs

Log only what you need from collections (Dictionaries, Arrays, etc.)

Avoid logging in tight code loops

Using os_log Family of APIs

Using os_log Family of APIs

Use `os_log` to log critical details to help debug issues

Using os_log Family of APIs

Use `os_log` to log critical details to help debug issues

Use `os_log_info` for additional info that will be captured during error or fault

Using os_log Family of APIs

Use `os_log` to log critical details to help debug issues

Use `os_log_info` for additional info that will be captured during error or fault

Use `os_log_debug` for high-volume debugging during development

Using os_log Family of APIs

Use `os_log` to log critical details to help debug issues

Use `os_log_info` for additional info that will be captured during error or fault

Use `os_log_debug` for high-volume debugging during development

Use `os_log_error` to cause additional information capture from app

Using os_log Family of APIs

Use `os_log` to log critical details to help debug issues

Use `os_log_info` for additional info that will be captured during error or fault

Use `os_log_debug` for high-volume debugging during development

Use `os_log_error` to cause additional information capture from app

Use `os_log_fault` to cause additional information capture from system

Gathering Logs

Using sysdiagnose

NEW

Using sysdiagnose

NEW

sysdiagnose is the preferred method to capture data for bug reports

Using sysdiagnose

NEW

sysdiagnose is the preferred method to capture data for bug reports

- Unified Logging data in system_logs.archive

Using sysdiagnose

NEW

sysdiagnose is the preferred method to capture data for bug reports

- Unified Logging data in system_logs.archive

You can use key-chord to trigger

Using sysdiagnose

NEW

sysdiagnose is the preferred method to capture data for bug reports

- Unified Logging data in system_logs.archive

You can use key-chord to trigger

sysdiagnose on Apple Watch will trigger on both Apple Watch and iPhone

Using sysdiagnose

NEW

sysdiagnose is the preferred method to capture data for bug reports

- Unified Logging data in system_logs.archive

You can use key-chord to trigger

sysdiagnose on Apple Watch will trigger on both Apple Watch and iPhone

Transfer from device using iTunes

Using sysdiagnose

NEW

sysdiagnose is the preferred method to capture data for bug reports

- Unified Logging data in system_logs.archive

You can use key-chord to trigger

sysdiagnose on Apple Watch will trigger on both Apple Watch and iPhone

Transfer from device using iTunes

This is the file to send to Apple (either Radar or Developer Technical Support)

Key-chords for sysdiagnose

NEW

Mac OS	Shift + Control + Option + Command + Period (.)
iOS	Volume Up + Volume Down + Power Slight vibration on iPhone to indicate start
watchOS	Press and hold Digital Crown + Side Button for 1 second A screen shot is triggered if not held long enough. Slight haptic to indicate start.
tvOS	Play/Pause + Volume Down On older remote controls for Apple TV, must be held for 5 seconds

Deprecations

Deprecation of Legacy logging APIs

Deprecation of Legacy logging APIs

All ASL logging APIs are superseded by the new APIs

Deprecation of Legacy logging APIs

All ASL logging APIs are superseded by the new APIs

New APIs for searching new log data will not be made public this release

Deprecation of Legacy logging APIs

All ASL logging APIs are superseded by the new APIs

New APIs for searching new log data will not be made public this release

- No equivalent asl_search functionality

Deprecated Activity APIs

os_activity_start	Use os_activity_create and os_activity_scope / os_activity_apply
os_activity_end	Use os_activity_create and os_activity_scope / os_activity_apply
os_activity_set_breadcrumb	Use os_activity_label_useraction
os_trace_with_payload	Use os_log*

Summary

Summary

The new Unified Logging system is

Summary

The new Unified Logging system is

- Faster

Summary

The new Unified Logging system is

- Faster
- Easier to use

Summary

The new Unified Logging system is

- Faster
- Easier to use
- Gives you more control

Summary

The new Unified Logging system is

- Faster
- Easier to use
- Gives you more control

But requires using new APIs and new tools

Related Sessions

Fix Bugs Faster Using Activity Tracing	WWDC 2014

More Information

https://developer.apple.com/wwdc16/721

WWDC16