Cloud Connect Serviceability

This section covers the serviceability information related to Cisco Web Proxy, Cloud Connect management, Digital Routing service, and DataConn service.

Cloud Connect Platform

This chapter provides steps to collect, download and view service logs using RTMT, and purge log files using CLIs.

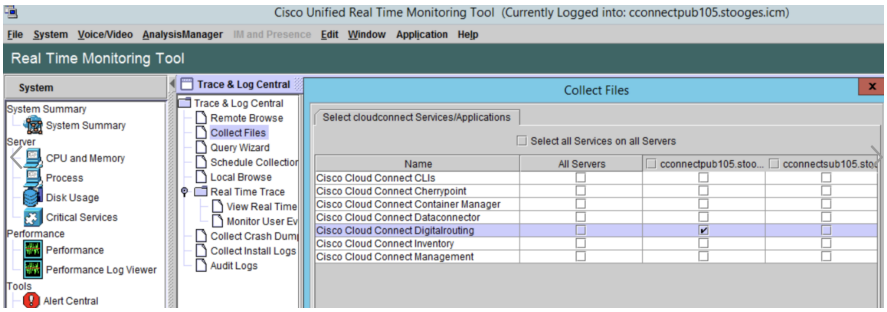
Collect service logs using RTMT

You can collect and download the service logs from the Real Time Monitoring Tool (RTMT) interface. For instructions, see the *Download Trace and Log Files* section in the *Serviceability Guide for Cisco Unified ICM/Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html

Use the **Cisco Cloud Connect Digital Routing** option in the **Collect Files** page in the RTMT interface to download the Digital Routing logs.

Use the **Cisco Cloud Connect Management** option in the **Collect Files** page in the RTMT interface to download the Cloud Connect Management service logs.

Use the **Cisco Cloud Connect DataConnector** option in the **Collect Files** page in the RTMT interface to download the DataConn logs.



The service logs are downloaded to the "Downloads directory".

View Real-time service logs using RTMT

To view the real-time logs,

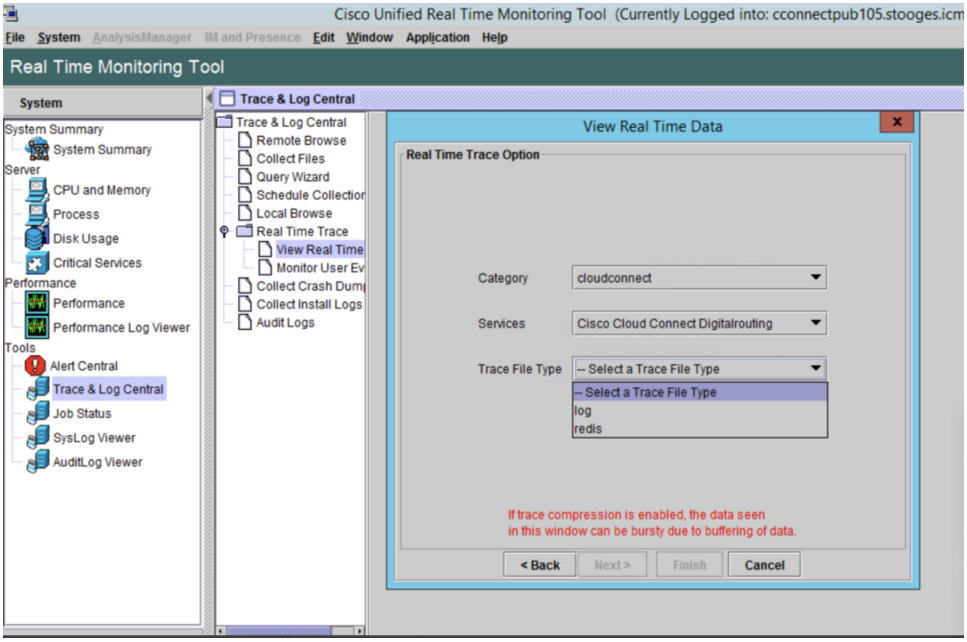
Procedure

Step 1 Run RTMT to connect to the target server.

Step 2 Choose **Tools > Trace & Log Central > Real Time Trace > View Real Time Data** in the **System** pane.

Step 3 Select the **Trace File Type**. For example, select log or redis for Cloud Connect Digitalrouting.

Figure 1: Trace File Type



Step 4 Click **Finish**.
The list of real-time log appears.

Purge log files using CLIs

You can use the CLIs to purge the hprof files created during the heap dump. Following are the CLI commands to list and delete the file:

Table 1: Purge log files using CLIs

Command	Description	Sample
file list activelog <log location that consists the files>	This lists the files in the location specified.	file list activelog hybrid\log\digitalrouting*.hprof
file delete activelog <file name >	This deletes the file in the location specified.	file delete activelog hybrid\log\digitalrouting\<name>.hprof

Serviceability for Web Proxy

This section provides serviceability information for the Web Proxy service. You can set up trace levels and collect Web Proxy log information.

Set up trace levels

The admin sets the Web Proxy access-log-level.

Syntax

set webproxy access-log-level [option]

Following are the options:

- **off:** turns off the log-level access
- **info:** sets log-level access to information
- **debug:** sets log-level access to debug

Example:

admin:set webproxy access-log-level info

Output response:

Successfully set webproxy access log-level to info

Collect Web Proxy logs

You can download the logs using RTMT or file command in the admin console. You can review these logs to verify the status or any problem in the Web proxy services.

Serviceability for Cloud Connect Management

This section provides serviceability information for the Cloud Connect Management. You can configure service logging and remote syslog destinations using APIs and Command-line Interfaces (CLIs). You can view or download the log files that are stored in the Directory listing using CLI or the built-in Real Time Monitoring Tool (RTMT). You can also monitor the status of the Cloud Connect Management.

Set up trace levels

This section outlines the setting up of the trace levels for Cloud Connect Management.

To view the list of containers in the Cloud Connect, run the following command:

```
admin:utils cloudconnect list
```

To set the list of trace level for the Cloud Connect, run the following command:

```
set cloudconnect log_level [container-name] [log-level]
```

- container-name must be valid
- log-level must be a valid log level [trace|debug|info|warn|error]

For example,

admin:set cloudconnect log_level cloudconnectmgmt trace

The result for the command is as follows:

The log level will be changed to TRACE for the container cloudconnectmgmt within 30 seconds

Note The container-name and log-level must be valid. The valid log levels are [trace|debug|info|warn|error].

Download Cloud Connect Management logs

To download logs, run the following command:

```
file get activelog hybrid/log/cloudconnectmgmt/cloudconnectmgmt.log
done.
Sub-directories were not traversed.
Number of files affected: 1
Total size in Bytes: 2472833
Total size in Kbytes: 2414.876
Would you like to proceed [y/n]? y
FTP server IP: 192.168.1.105
                  FTP server port [22]:
                  Jser ID: root
                  Password:
                  *********
                  Download directory:
                  Transfer completed
```

View the status of Cloud Connect Management service

The status API provides the internal status of the Cloud Connect node and functional modules of the Cloud Connect Management service. The API can be used to fetch the internal state and verify status.

Cloud Connect Management services are accessed through the following status API:

- URL: https://cloudconnectfqdn:8445/cloudconnectmgmt/status
- Method: GET
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).


```
{
    "status":"IN_SERVICE",
    "timestamp":1668416327203,
    "cluster":{"nodes":
    [
    {
       "address":"cconnectpub105.stooges.icm",
       "status":"MemberUp","statusSince":1668416327201,
       "statusUrl":"https://cloudconnectFQDN/cloudconnectmgmt/status"
    },
    {
        "address":"cconnectsub105.stooges.icm","status":"MemberUp",
        "statusSince":1668416327201,
        "statusUrl":"https://cloudconnectFQDN/cloudconnectmgmt/status"
    }
    ]
},
"isConfigWriter":true,"description":"cloudconnectmgmt Service Status Snapshot",
"details":
{
"components":
[
{
    "name":"cloudconnectmgmt","status":"IN_SERVICE",
    "statusSince":1668416327202,
    "fusion":{"registration":
{
    "status":"REGISTERED",
    "registeredSince":1665377645055,
    "orgId":"4a2a8302-56f0-47b4-8800-45dd68104bd7",
    "clusterId":"0ddbf6d5-f949-4e2a-ab59-7dc3466ba6ea",
    "idBrokerHost":"idbrokerbts.webex.com",
    "fmsHost":"hercules-intb.ciscospark.com",
    "u2cHost":"u2c-intb.ciscospark.com"
},
"connector":
{
    "connectorType":"cjp_hybrid",
    "connectorVersion":"12.6.2-1.0.389",
    "status":"STARTED",
    "heartbeatFailureCount":0,
    "lastHeartbeatTime":1668416323571,
    "lastHeartbeatFailureTime":-1
}}}]}}
```

Serviceability for Digital Routing

This section provides serviceability information for the Digital Routing service. You can configure service logging and remote syslog destinations using APIs and Command-line Interfaces (CLIs). You can view or download the log files from the directory listing using CLI or the built-in Real Time Monitoring Tool (RTMT). You can also monitor the status of the Digital Routing service.

Additionally, you can get insights into how to access and collect the Java Management Extensions (JMX) counters using APIs or Java Monitoring and Management Console (JConsole).

Configure Service Logging

The Digital Routing service contains multiple modules. The service provides options to configure the logging levels for each of those modules independently. The default out-of-box service logging for all the modules is INFO. By default, the maximum size of each log file is 100 MB . The log files are rotated or overwritten when the accumulated file size reaches to 500 MB.

The following are the log levels for the Digital Routing service:

Log Level	Indicates
info	Informational log
debug	Debugging log
warn	Warning log
error	Error log
notice	Notification log

Configure service logging using API

You can configure the Digital Routing service log modules using API.

The following are the descriptions of the Digital Routing service log modules:

Log Module	Description
api	Logging layer for REST controller events for all the exposed APIs.
db	Logging layer for handling data sources like awdb (local database that contains configuration and real-time data) and Media Routing Domain (MRD) configuration.
security	Logging layer for handling authentication and authorization related events.
mr	Logging layer for interaction with CCE Media Routing Peripheral Gateway (MR-PG).
persistence	Logging layer for persistence to file system for configurations including replication.
logging	Logging layer for Java Management Extensions (JMX) monitoring related events.
state	Logging layer for events related to Digital Routing state machine related events.
mbeans	Logging layer for Digital Routing mbeans related traces for service counters.

servicecounter	Logging layer for Digital Routing servicecounter traces.


```
{
    "configuredLevel": [
        {
            "logModule": "logging",
            "logLevel": "info"
        },
        {
            "logModule": "org.springframework.security",
            "logLevel": "info"
        },
        {
            "logModule": "org.springframework",
            "logLevel": "info"
        },
        {
            "logModule": "api",
            "logLevel": "info"
        },
        {
            "logModule": "db",
            "logLevel": "info"
        },
        {
            "logModule": "mr",
            "logLevel": "info"
        },
        {
            "logModule": "security",
            "logLevel": "info"
        },
        {
            "logModule": "state",
            "logLevel": "info"
        },
        {
            "logModule": "persistence",
            "logLevel": "info"
        },
        {
            "logModule": "jmx",
            "logLevel": "info"
        },
        {
            "logModule": "redisclient",
            "logLevel": "info"
        },
        {
            "logModule": "mbeans",
            "logLevel": "info"
        },
        {
            "logModule": "servicecounter",
            "logLevel": "info"
        }
    ]
}
```


Digital Routing trace

The following are the API details to view the Digital Routing service trace:

- URL: https://hostname/drapi/v1/config/trace/dr

- Method: GET, PUT
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).

GET Response:

```
{
    "syslogConfig": {
    "primary-host": "",
    "secondary-host": "",
    "logLevel": "info"
}
}
```

PUT Payload

```
{
    "syslogConfig": {
    "primary-host": "<Primary SyslogIP>”
    "secondary-host": ""<Secondary SyslogIP>””,
    "logLevel": "debug"
}
}
```

Redis service trace

The following are the API details to view the Redis service trace:

- URL: https://host_fqdn:8445/drapi/v1/config/trace/redis
- Method: GET,PUT
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).

GET Response:

```
[
    "loglevel",
    "notice"
]
```

PUT Payload:

```
{
    "redisConfiguredLevel": "<notice|debug|verbose|warning>"
}
```

Configure service logging using CLI

To configure the service logging for the Digital Routing service and the Redis service traces using CLI:

Digital Routing service trace

To configure the Digital Routing service trace, run the following command:

```
utils cloudconnect digitalrouting logging trace set drapi <info|debug|warn|error>
```

To view the Digital Routing service trace list, run the following command:

```
utils cloudconnect digitalrouting logging trace list drapi
```

The list of Digital Routing service trace appears.

Redis service trace

To configure the Redis service trace, run the following command:

```
utils cloudconnect digitalrouting logging trace set redis <notice|warning|verbose|debug>
```

To view the Redis service trace, run the following command:

```
utils cloudconnect digitalrouting logging trace list redis
```

The list of Digital Routing service trace appears.

Configure syslog

Digital Routing service provides the option to configure remote syslog destinations. It allows configuration for up to two remote destinations. The default out-of-box syslog remote destinations are empty and on deployment, it requires accurate configuration.

Configure syslog using API

The following are the API details to configure the destination where the syslogs are stored:

- URL: https://host_fqdn:8445/drapi/v1/config/trace/syslog
- Method: GET, PUT
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).

GET Response

```
{
    "syslogConfig": {
    "primary-host": "",
    "secondary-host": "",
    "logLevel": "info"
}
}
```

PUT Payload

```
{
    "syslogConfig": {
    "primary-host": "<Primary SyslogIP>”
    "secondary-host": ""<Secondary SyslogIP>”",
    "logLevel": "debug"
}
}
```

Configure syslog using CLI

Digitalrouting provides the ability to configure remote syslog servers through which specified system logs are generated. You can access the syslog from the syslog server.

To configure syslogs, run the following command:

```
utils cloudconnect digitalrouting logging syslog set <primary-host|secondary-host> %host address%
```

- primary-host: Use this keyword when you are setting the primary syslog server hostname.
- secondary-host: Use this keyword when you are setting the secondary syslog server hostname.
- host address: Refers to the remote syslog server IP address or hostname or fqdn.

Once you run the command to configure syslogs, you will get the message in your command prompt as below:

The syslog configurations are successfully updated.

To view the syslog list, run the following command:

```
utils cloudconnect digitalrouting logging syslog list
```

The list of syslogs that are logged for the Digital Routing service appears.

To view the trace list syslog, run the following command:

```
utils cloudconnect digitalrouting logging trace list syslog
```

The trace list syslogs that are logged for the Digital Routing service appears.

To update log trace set, run the following command:

```
utils cloudconnect digitalrouting logging trace set syslog %loglevel%
```

The Cloud Connect Digital Routing logging trace set syslog debug successfully updated.

Note Log level options are error, warn, info, and debug.

Monitor the status of the Digital Routing service

Digital Routing services are monitored through the log collection and Java Management Extensions (JMX) counters.

The following are the API details to monitor the service status:

- URL: https://host_fqdn:8445/drapi/v1/status
- Method: GET
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).

The status is INACTIVE when the connector is started, and all services are initializing and making connections. The status is ACTIVE when the connector is started and connected with all the components.

```
{
  "status": "ACTIVE",
  "cluster": {
    "nodes": [
      {
        "address": "CloudConnect local FQDN or IP",
        "status": "MemberReachable",
        "statusSince": 1664305403815,
        "statusURL": "https://<CloudConnect local FQDN or IP>:8445/drapi/v1/status"
      },
      {
        "address": "CloudConnect remote FQDN or IP",
        "status": "MemberReachable",
        "statusSince": 1662189184290,
        "statusURL": "https://<CloudConnect remote FQDN or IP>:8445/drapi/v1/status"
      }
    ]
  },
  "description": "digitalrouting Service Status Snapshot",
  "details": {
    "components": [
      {
        "name": "digitalrouting",
        "status": "ACTIVE",
        "statusSince": 1664305429343
      },
      {
        "name": "redis",
        "status": "MASTER",
        "statusSince": 1664305423670
      }
    ]
  },
  "apiLatencyResponseinMs": {
    "avgCurrentResponse": 0,
    "avgPreviousResponse": 0,
    "avgTodayResponse": 0,
    "avgServiceUpResponse": 0
  }
}
```


Download logs using CLI

To download logs, run the following command:

```
admin: file get acti

admin: file get activelog hybrid/log/digitalrouting/dr-app-*.log

lease wait while the system is gathering files info

Get file: active/hybrid/log/digitalrouting/dr-app-2022-08-04-051932.log

Get file: active/hybrid/log/digitalrouting/dr-app-2022-08-04-051934.1og

Get file: active/hybrid/log/digitalrouting/dr-app-2022-08-04-092115.log

Get file: active/hybrid/log/digitalrouting/dr-app-2022-08-04-092116.1og

done.

subdirectories were not traversed.

Number of files affected: 12
```

```
Total size in Bytes: 2434450

Total size in Kbytes: 2377.3926

Would you like to proceed [y/n]?

Y

FTP server IP: 192.168.1.105

FTP server port [22]:

Jser ID: root

Password:

*********

Download directory:

Transfer completed

admin•
```

Directory listing

The directory listing page provides the path to the log files with the file name, file size, and last modified details. You can click on a file name to access the log file.

The path to access the log directory for Digital Routing service is as follows:

https://host_fqdn:8445/drapi/log/digitalrouting

The path to access the log directory for Platform files is as follows:

https://host_fqdn:8445/drapi/common/log

Access JMX counters

You can access the Java Virtual Machine (JVM) counters, Service counters, and Redis counters using either APIs or JConsole.

Access JMX Counters using API

To access the JMX counters using API:

JVM counters

The following are the API details to access the JVM counters:

- URL: https://host_fqdn:8445/drapi/v1/metrics/jvmcounters
- Method: GET
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).

```
{
  "java.lang:type=Memory": {
    "name": "java.lang:type=Memory",
```

```
    "attribute": {
      "HeapMemoryUsage": "{committed=1073741824, init=1073741824, max=1719664640,
used=365429248}",
      "NonHeapMemoryUsage": "{committed=141688832, init=2555904, max=1056964608,
used=134275968}"
    }
  },
  "java.lang:type=OperatingSystem": {
    "name": "java.lang:type=OperatingSystem",
    "attribute": {
      "ProcessCpuTime": "63140000000",
      "SystemCpuLoad": "0.1756357022315391",
      "CommittedVirtualMemorySize": "2839908352",
      "AvailableProcessors": "2",
      "FreePhysicalMemorySize": "1289973760",
      "ProcessCpuLoad": "0.0",
      "TotalPhysicalMemorySize": "2147483648"
    }
  },
  "java.lang:type=Threading": {
    "name": "java.lang:type=Threading",
    "attribute": {
      "ThreadCount": "210",
      "PeakThreadCount": "210",
      "CurrentThreadCpuTime": "221284760"
    }
  }
}
```


Service counters

The following are the API details to access the JMX Service counters:

- URL: https://host_fqdn:8445/drapi/v1/metrics/servicecounters
- Method: GET
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).


```
{
  "DrapiServiceCountersPreviousInterval30Minutes": {
    "name": "DrapiServiceCountersPreviousInterval30Minutes",
    "attribute": {
      "TasksAbandonedAgent": "0",
      "TasksAbandonedCustomer": "0",
      "TasksAccepted": "0",
      "TasksCompleted": "0",
      "TasksReceived": "0",
      "TasksRejected": "0",
      "TasksRejectedCCERouter": "0",
      "TasksRouted": "0",
      "TasksTransferred": "0",
      "WebhookNotify": "0",
      "WebhookNotifyFailed": "0"
    }
  },
  "DrapiServiceCountersToday": {
    "name": "DrapiServiceCountersToday",
    "attribute": {
      "TasksAbandonedAgent": "0",
      "TasksAbandonedCustomer": "0",
      "TasksAccepted": "0",
```

```
      "TasksCompleted": "0",
      "TasksReceived": "0",
      "TasksRejected": "0",
      "TasksRejectedCCERouter": "0",
      "TasksRouted": "0",
      "TasksTransferred": "0",
      "WebhookNotify": "0",
      "WebhookNotifyFailed": "0"
    }
  },
  "Time": {
    "requestReceivedTime": "22 Nov 2022 03:11:03.574 UTC"
  },
  "DrapiServiceCountersSinceServiceUp": {
    "name": "DrapiServiceCountersSinceServiceUp",
    "attribute": {
      "TasksAbandonedAgent": "0",
      "TasksAbandonedCustomer": "0",
      "TasksAccepted": "0",
      "TasksCompleted": "0",
      "TasksReceived": "3",
      "TasksRejected": "3",
      "TasksRejectedCCERouter": "0",
      "TasksRouted": "0",
      "TasksTransferred": "0",
      "WebhookNotify": "0",
      "WebhookNotifyFailed": "0"
    }
  },
  "DrapiServiceCountersCurrentInterval30Minutes": {
    "name": "DrapiServiceCountersCurrentInterval30Minutes",
    "attribute": {
      "TasksAbandonedAgent": "0",
      "TasksAbandonedCustomer": "0",
      "TasksAccepted": "0",
      "TasksCompleted": "0",
      "TasksReceived": "0",
      "TasksRejected": "0",
      "TasksRejectedCCERouter": "0",
      "TasksRouted": "0",
      "TasksTransferred": "0",
      "WebhookNotify": "0",
      "WebhookNotifyFailed": "0"
    }
  },
  "DrapiServiceCountersRealTime": {
    "name": "DrapiServiceCountersRealTime",
    "attribute": {
      "ChatTasksInDrapiQueue": "0",
      "EmailTasksInDrapiQueue": "0",
      "RunApplicationScriptThrottleCount": "0",
      "SocialTasksInDrapiQueue": "0",
      "TasksInCCEQueue": "0",
      "TasksInDrApiQueue": "0",
      "TasksInWebhookNotificationQueue": "0",
      "TelephonyTasksInDrapiQueue": "0"
    }
  }
}
```


Redis counters

The following are the API details to access the Redis counters:

- URL: https://host_fqdn:8445/drapi/v1/metrics/rediscounters

- Method: GET
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).


```
{
  "redis_version": "6.2.7",
  "redis_mode": "standalone",
  "os": "Linux 3.10.0-1160.53.1.el7.x86_64 x86_64",
  "arch_bits": "64",
  "gcc_version": "10.3.1",
  "server_time_usec": "1664456513263005",
  "uptime_in_seconds": "94",
  "uptime_in_days": "0",
  "connected_clients": "9",
  "maxclients": "10000",
  "blocked_clients": "1",
  "used_memory_human": "1.26M",
  "used_memory_rss_human": "4.54M",
  "used_memory_peak_human": "1.28M",
  "used_memory_peak_perc": "98.98%",
  "used_memory_overhead": "992439",
  "used_memory_startup": "846815",
  "used_memory_dataset": "333195",
  "used_memory_dataset_perc": "69.59%",
  "total_system_memory_human": "9.61G",
  "used_memory_lua_human": "77.00K",
  "used_memory_scripts_human": "9.01K",
  "maxmemory_human": "0B",
  "maxmemory_policy": "noeviction",
  "mem_fragmentation_ratio": "4.28",
  "mem_fragmentation_bytes": "3645653",
  "mem_replication_backlog": "0",
  "mem_clients_slaves": "0",
  "mem_clients_normal": "136138",
  "total_connections_received": "89",
  "total_commands_processed": "942",
  "instantaneous_ops_per_sec": "9",
  "total_net_input_bytes": "145705",
  "total_net_output_bytes": "48704",
  "instantaneous_input_kbps": "0.24",
  "instantaneous_output_kbps": "0.07",
  "rejected_connections": "0",
  "evicted_keys": "0",
  "role": "master",
  "connected_slaves": "0",
  "master_failover_state": "no-failover",
  "master_replid": "400fbf197d0ac2375ee9eafddc4fcf539fa8e7e0",
  "master_replid2": "feb0f97258906cad43eba1657bdf28419d1348af",
  "master_repl_offset": "0",
  "second_repl_offset": "1",
  "repl_backlog_active": "0",
  "repl_backlog_size": "1048576",
  "repl_backlog_first_byte_offset": "0",
  "repl_backlog_histlen": "0",
  "used_cpu_sys": "0.117309",
  "used_cpu_user": "0.077111",
  "errorstat_ERR": "count=52"
}
```

Access Counters using JConsole

Following are the JMX counters you can access using the JConsole:

- JVM counters
- Service counters
- Redis counters

For instruction about how to access JConsole, see the *Using JConsole* section available at https://openjdk.org/tools/svc/jconsole/

JVM counters

You can access JVM counters of the following mbeans type: Memory, Operating System, and Threading.

To access the JVM **Memory** counter attributes:

1. Navigate to **java.lang** > **java.lang** > **Memory** > **Attributes**. Jconsole opens to provide the IP address of cloud-connect box and port 10006.
2. Navigate to the **MBeans** tab > **java.lang** > **Memory** > **Attributes**to view the list of attributes.

To access the JVM **Operating System** counter attributes:

1. Navigate to **java.lang** > **java.lang** > **Operating System** > **Attributes**. Jconsole opens to provide the IP address of cloud-connect box and port 10006.
2. Navigate to the **MBeans** tab > **java.lang** > **Operating System** > **Attributes**to view the list of attributes.

To access the JVM **Threading** counter attributes:

1. Navigate to **java.lang** > **java.lang** > **Threading** > **Attributes**. Jconsole opens to provide the IP address of cloud-connect box and port 10006.
2. Navigate to the **MBeans** tab > **java.lang** > **Threading** > **Attributes**to view the list of attributes.

Service counters

To access the Digital Routing Service counters:

1. Navigate to **com.cisco.ccbu.dr.mbeans** > **Drapi Mxbean Counter**.

   Following are the **MxBean** counters available:

   a. drapicountersCurrentIntervalMXBean
   b. drapicountersPreviousIntervalMxBean
   c. drapicountersRealTimeMXBean
   d. drapicountersServiceUpMXBean
   e. drapicoutnersTodayMXBean

2. Click **Attributes** to view the Service counter attributes.
3. Click on an *Attribute* to view the attribute value.

Redis counters

To access the Digital Routing specific Redis counters:

1. Navigate to **com.cisco.ccbu.dr.mbeans** > **RedisMXBean** > **Attributes**.
2. Click on an *Attribute* to view the attribute value.

Access Digital Channel Statistics in CCE Administration Portal

To access the statistics for Digital Channel and Redis service running on the Cloud Connect:

Procedure

In **Unified Contact Center Enterprise Management**, navigate to **Overview** > **Digital Channels** > **Digital Channel Statistics**.

The Digital Channel Statistics page displays the Digital Routing and Redis service Host, Status, Role, and Up Since.

The Task Information section displays the Realtime tasks, Historical tasks with duration of each task.

The Realtime Tasks displays the Tasks in Digital Routing Queue and the Tasks in CCE Routing Queue.

The Historical Tasks section displays the duration of the current and previous task.

The historical task statuses are *Received*, *Rejected*, *Rejected by CCE*, *Queued*, *Route requests*, *Close requests*, *Transfer requests*, *Abandoned by customer*, and *Failed webhook requests*.

For more information, refer to the descriptions for JMX Service counter definitions, on page 17 in the *Cloud Connect Serviceability* section of the Serviceability Guide for Cisco Unified ICM/Contact Center Enterprise..

JMX Service counter definitions

The following table provides the Java Management Extensions (JMX) Service counter definitions:

Table 2: JMX Service counter definitions

JMX Counters	Label in CCE Administration Portal	Current Interval (default 30 minutes interval)	Previous Interval	Today (24 hours)	Since Service Up	Real-time
TasksReceived	Received	The counter increments for every incoming task. The counter resets after every interval.	The number of incoming tasks in the current interval is updated to the previous interval at the interval cutover.	The total number of tasks of all the previous interval for the day + the number of injected tasks in the current interval.	The total number of all the previous interval incoming tasks of the day + the number of task received in the current interval.	**TasksInCCEQueue/ CallsQueued** The counter increments when a new task changes from CREATED to QUEUED state. The counter decrements when the task changes from QUEUED to ROUTED or CLOSED.
TasksAccepted	Queued	The counter increments for the tasks in the CREATED state and continues to increment for every incoming task updated to CREATED state. The counter resets after every interval.	The number of current interval task in the CREATED state updated to task accepted in the previous interval at the interval cutover.	The total number of tasks of all previous interval in the CREATED state for the day + the number of task accepted in the current interval.	The total number of all previous interval incoming task in the CREATED state + the number of tasks accepted in the current interval.	**TasksInDrApiQueue** The *ClientRequestQueue* size provides the real-time counter value of incoming task accepted.

JMX Counters	Label in CCE Administration Portal	Current Interval (default 30 minutes interval)	Previous Interval	Today (24 hours)	Since Service Up	Real-time
TasksCompleted	Close requests	The counter increments the number of normally closed tasks and continues to increment for every closed task. The counter resets after every interval.	The number of current interval tasks in the CLOSED state with normal disposition updated to TaskCompleted in the previous interval at the interval cutover.	The total number of tasks in CLOSED state with normal disposition for the day + the TasksCompleted in the current interval.	The total number of all previous interval incoming task in the CLOSED state with normal disposition + the number of tasks completed in the current interval.	**TasksInWebhookNotificationQueue** The *RedissonQueues* size provides the real-time counter value of the task in *WebhookNotificationQueue*.
TasksRouted	Route requests	The counter increments the number of tasks in the ROUTED state and continues to increment the counter for every task updated to ROUTED state. The counter resets after every interval.	The number of tasks in the ROUTED state in the current interval is updated to TasksRouted in the previous interval at the interval cutover.	The total number of all previous interval tasks which are updated to ROUTED for the day+ the number of TasksRouted in the current interval.	The total number of all previous interval incoming task in the ROUTED state + the number of tasks routed in the current interval.	**ChatTasksInDapQueue** The *ClientRequestQueue* size provides the real-time counter value of incoming accepted task for media type chat.

JMX Counters	Label in CCE Administration Portal	Current Interval (default 30 minutes interval)	Previous Interval	Today (24 hours)	Since Service Up	Real-time
TasksTransferred	Transfer requests	The counter increments incoming tasks in the TRANSFERRED state and continues to increment for every incoming transferred task. The counter resets after every interval.	The number of incoming Tasks in TRANSFERRED state in the current interval is updated to TasksTransferred in the previous interval at the interval cutover.	The total number of all previous interval incoming task in the TRANSFERRED state for the day + the number of task transferred in the current interval.	The total number of all previous interval incoming task in the TRANSFERRED state + the number of tasks transferred in the current interval.	EmailTasksInClientRequestQueue The *ClientRequestQueue* size provides the realtime counter value of incoming accepted task for media type email.
TasksAbandonAgent	NA	The counter increments closed task with disposition 37 and continues to increment for every closed task with the disposition 37. The counter resets after every interval.	The number of closed tasks with disposition 37 in the current interval is updated to TasksAbandonAgent in the previous interval at the interval cutover.	The total number of closed task of all previous interval with disposition 37 for the day + the number of TasksAbandonAgent in the current interval.	The total number of all previous interval closed tasks with disposition 37 + the number of TasksAbandonAgent in the current interval.	SocialTasksInClientRequestQueue The *ClientRequestQueue* size provides the real-time counter value of incoming accepted task for media type social.

JMX Counters	Label in CCE Administration Portal	Current Interval (default 30 minutes interval)	Previous Interval	Today (24 hours)	Since Service Up	Real-time
TasksAbandonCustomer	Abandoned by customer	The counter increments the number of closed tasks with disposition 29 and continues to increment the counter for every closed task with disposition 29. The counter resets after every interval.	The number of closed Tasks with disposition 29 in the current interval is updated to TasksAbandonCustomer in the previous interval at the interval cutover.	The total number of closed tasks all the previous interval with disposition 29 for the day + the number of TasksAbandonCustomer in the current interval.	The total number of all previous interval closed tasks with disposition 29 + the number of TasksAbandonCustomer in the current interval.	TelephonyTaskDropQueue The *ClientRequestQueue* size provides the real-time counter value of incoming accepted task for media type telephony.
TasksRejected	Rejected	The counter increments the number of incoming tasks rejected due to the maximum limit of *ClientRequestQueue* size and failure of authentication. The counter resets after every interval.	The number incoming Tasks rejected due to maximum limit of *ClientRequestQueue* size and failure of authentication. The number of tasks in the current interval is updated to TasksRejected in the previous interval at the interval cutover.	The total number of all the previous interval incoming tasks rejected for the day due to maximum limit of *ClientRequestQueue* size and failure of authentication + the number of rejected task in the current interval.	The total number of all previous interval tasks dropped due to max limit of ClientRequestQueue size and Authentication failure + the number of TasksRejected in the current interval.	RunApplicationSpillinCount The DRAPI service can receive RUN_APPLICATION_REQ message from Media Routing (MR) Protocol Independent Multicast (PIM) and this counter increments whenever service receives more than six RUN_APPLICATION_REQ

JMX Counters	Label in CCE Administration Portal	Current Interval (default 30 minutes interval)	Previous Interval	Today (24 hours)	Since Service Up	Real-time
TaskRejectedCCERouter	Rejected by CCE	The counter increments the number of incoming tasks rejected at the CCE router. The counter resets after every interval.	The number of incoming tasks rejected at the CCE router. The counter resets after every interval.	The number of incoming tasks rejected at the CCE router. The counter resets after every interval.	The total number of all previous interval incoming tasks rejected in CCE due to new task failure + the number of TaskRejectedCCERouter in the current interval.	
WebhookNotify	NA	The counter increments the number of Webhook notifications sent successfully to the Webex Connect. The counter resets after every interval.	The number of Webhook notifications sent successfully to the Webex Connect is updated to WebhookNotify in the previous interval at the interval cutover.	The total number of all the previous interval Webhook notifications sent successfully to the Webex Connect for the day + the number of WebhookNotify in the current interval.	The total number of all previous interval Webhook notifications sent successfully to WebexConnect + the number of WebhookNotify in the current interval.	
WebhookNotifyFailed	Failed Webhook requests	The number of Webhook notifications failed to send successfully to WebexConnect. The counter resets after every interval.	The number of Webhook notifications failed to send successfully to WebexConnect is updated to WebhookNotifyFailed in the previous intervalat the interval cutover.	The number of all previous interval Webhook notifications failed to send successfully to WebexConnect for the day + number of WebhookNotifyFailed in the current interval.	The number of all previous interval Webhook notifications failed to send successfully to WebexConnect + the number of WebhookNotifyFailed in the current interval.	

Serviceability for DataConn

This section provides serviceability information for the Cloud Connect DataConn. You can view or download the log files that are stored in the directory listing using the Real Time Monitoring Tool (RTMT). You can also monitor the status of the cloud connect DataConn.

Download DataConn logs

To download logs, run the following command:

```
file get activelog hybrid/log/cloudconnectmgmt/dataconn.log
```

The downloaded log details are displayed.

```
done.
Sub-directories were not traversed.
Number of files affected: 1
Total size in Bytes: 2472833
Total size in Kbytes: 2414.876
Would you like to proceed [y/n]? y
FTP server IP: 192.168.1.105
                   FTP server port [22]:
                   Jser ID: root
                   Password:
                   *********
                   Download directory:
                   Transfer completed
```

Monitor the status of DataConn service

DataConn services are monitored through status API.

The following are the API details to monitor the service status:

- URL: https://cloudconnectfqdn:8445/dataconn/status
- Method: GET
- Content-Type: application/json
- Authentication: Basic authentication using platform credentials (username and password).

The following is the output of status API:

```
{
    "status": "IN_SERVICE",
    "timestamp": 1681814206313,
    "cluster": {
        "nodes": [
            {
                "address": "cconnectpub105.stooges.icm",
                "status": "MemberUp",
                "statusSince": 1681814206299,
                "statusUrl": https://cconnectpub105.stooges.icm:8445/dataconn/status
            },
            {
                "address": "cconnectsub105.stooges.icm",
                "status": "MemberUp",
```

```
                "statusSince": 1681814206299,
                "statusUrl": https://cconnectsub105.stooges.icm:8445/dataconn/status
            }
        ]
    },
    "isConfigWriter": true,
    "description": "Service Status Snapshot",
    "details": {
        "components": [
            {
                "name": "UserSync",
                "status": "IN_SERVICE",
                "statusSince": 1681784668359,
                "userSync": "STARTED",
                "syncEnabled": true
            }
        ]
    }
}
```