



ESC のシステムログ

- [ESC ログメッセージの表示 \(1 ページ\)](#)
- [ESC ログファイルの表示 \(7 ページ\)](#)

ESC ログメッセージの表示

ログメッセージは、VNF ライフサイクル全体にわたって ESC イベント用に作成されます。これらには、外部メッセージ、ESC から他の外部システムへのメッセージ、エラーメッセージ、警告、イベント、障害などがあります。ログファイルは、`/var/log/esc/escmanager_tagged.log` にあります。

次に、ログメッセージの形式を示します。

```
date=<time-date> [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

次に、ログの例を示します。

```
date=15:43:58,46022-Nov-2016]
[loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7] [cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:CSCvd94541,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0] [msg=sleepingfor5seconds
to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7
name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

要求を受信すると、一意のトランザクション ID を自動生成する `RequestDetails` オブジェクトが作成されます。この値は、すべてのスレッドで転送されます。分類とタグは任意です。これらは、読みやすくするためにログメッセージに追加されたプレフィックスであり、デバッグに役立ちます。分類とタグを使用すると、ログメッセージを簡単に解析し、ログ分析ツールでフィルタリングすることができます。

次に、サポートされている分類を示します。

NBI	「com.cisco.esc.rest」 「com.cisco.esc.filter」 (ノース バウンド インターフェイス : 証明書)
-----	---

SBI	「com.cisco.esc.rest」：ソースはコールバックハンドラまたは「EventsResource」（サウスバウンドインターフェイス、ESC と VIM 間）
SM	「com.cisco.esc.statemachines」は StateMachine を意味します。この分類は、StateMachine カテゴリのログを示します。
MONITORING	「com.cisco.esc.monitoring」 「com.cisco.esc.paadaptor」（MONA 関連ログ）
DYNAMIC_MAPPING	「com.cisco.esc.dynamicmapping」 「com.cisco.esc.db.dynamicmapping」（MONA 関連ログ）
CONFD	「com.cisco.esc.confid」
CONFD_NOTIFICATION	「com.cisco.esc.confid.notif」 「com.cisco.esc.confid.ConfidNBIAdapter」
OS	「com.cisco.esc.vim.openstack」
LIBVIRT	「com.cisco.esc.vim.vagrant」
VIM	「com. esc. vim」
REST_EVENT	「ESCManager_Event」 「com.cisco.esc.util.RestUtils」。ログ内の REST 通知を示します。
WD	「com.cisco.esc.watchdog」
DM	「com.cisco.esc.datamodel」 「com.cisco.esc.jaxb.parameters」（データモデルとリソースオブジェクト）
DB	「com.cisco.esc.db」（データベース関連ログ）
GW	「com.cisco.esc.gateway」
LC	「com.cisco.esc.ESCManager」（スタートアップ関連ログ）
SEC	「com.cisco.esc.jaas」
MOCONFIG	「com.cisco.esc.moconfig」（MOCONFIG オブジェクト関連ログ。これは ESC 開発者用の内部ログです）
POLICY	「com.cisco.esc.policy」（サービス/VM ポリシー関連ログ）
TP	「com.cisco.esc.threadpool」
ESC	「com.cisco.esc」上記にないその他のパッケージ

次に、サポートされているタグを示します。

- **ワークフロー [wf:]** : RequestDetails オブジェクトのアクションとリソースを使用して生成されます。例 : 「wf: create_network」
- **イベントタイプ [eventType:]** : 現在のアクションをトリガーしたイベント。例 : 「eventType:VM_DEPLOY_EVENT」
- **リソースベース** : これらの値は、イベントで使用されるパラメータのタイプに基づいて生成されます。階層 (テナント、VM グループなど) がログに追加されます。

テナント	[tenant:<tenant name>]
ネットワーク	[tenant:<tenant id>, network:<network name>] (注) テナントは、該当する場合にのみ表示されます。
サブネット	[tenant:<tenant name or id>, network:<network name or id>, subnet:<subnet name>] (注) テナントは、該当する場合にのみ表示されます。
ユーザ	[tenant:<tenant name>, user:<user name or id>] (注) テナントは、該当する場合にのみ表示されます。
イメージ	[image:<image name>]
フレーバ	[flavor:<flavor name>]
配置	[tenant:<tenant name or id>, depName:<deployment name>]
DeploymentDetails	[tenant:<tenant name or id>, depName:<deployment name>, vmGroup:<vm group name>, vmName:<vm name>]
スイッチ	[tenant:<tenant name or id>, switch:<switch name>]
音量	[volume:<volume name>]
サービス	[svcName:<Service Registration name>]

さらに、分析とログの管理を促進するため、ESC ログを rsyslog サーバに転送することもできます。

ConfD API を使用したログのフィルタリング

ConfD API に導入されたログフィルタを使用して、ESC でログ (展開ログやエラーログなど) を照会および取得できます。テナント、展開名、および VM 名の新しいフィルタが導入されました。これにより、ConfD API のログフィルタを使用して、最新のエラーログの ESC ログをさらに照会することができます。ESC と OS 間の通信に関連する ESC ログを取得することもできます (分類タグを「OS」に設定します)。

次に、ConfD API ログを取得するためのログ形式を示します。

```
date=<time-date> [loglevel=<loglevel>] [tid=<transactionid>] [cl=<classifications>]
[tags=<tags>] [msg=<message>
```

次に、サンプルログの例を示します。

```
date=15:43:58,46022-Nov-2016] [loglevel=ERROR ] [tid=0793b5c9-8255-47f3-81e6-fbb59f6571f7]
[cl=OS ]
[tags=wf:create_vm,eventType:VM_DEPLOY_EVENT,tenant:test,depName:test-dep,vmGrpName:test-VNF,
vmName:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0]
[msg=sleepingfor5seconds to allow vm to become ACTIVE instance id:
162344f7-78f9-4e45-9f23-34cf87377fa7
name:test-dep_test_0_dc3f406c-05ca-43b3-af21-0841e3b029a0
```

ログレベル、分類、およびタグのパラメータは、ログを取得するために相互に依存します。次の組み合わせを使用してログを正常に取得できます。

- log_level=ERROR, classifications=OS, tags=(depName:test-dep)
- log_level=ERROR, classifications=OS, tags=(tenant: test)

ログフィルタは、次の条件がすべて満たされたときに値を返します。

- ログ レベル
- 分類 (指定されている場合)
- タグ (指定されている場合)



(注) 複数の分類がリストされている場合は、1つ以上の分類に一致する必要があります。同じことが、タグにも適用されます。

たとえば、次のログフィルタ条件では、前述のログサンプルを返しません。

```
log_level=ERROR, classifications=VIM, tags=(depName:test-dep)
```

ログレベルとタグが一致していても、分類の VIM が一致していないので値は返されません。

次に、データモデルを示します。

```
rpc filterLog {
  description "Query and filter escmanager logs using given parameters";
  tailf:actionpoint esrcpc;
  input {
    leaf log_level {
      mandatory false;
      description "One of DEBUG / INFO / WARNING / ERROR / TRACE / FATAL. Results will
include all logs at and
      above the level specified";
      type types:log_level_types;
      default ERROR;
    }
    leaf log_count {
      mandatory false;
      description "Number of logs to return";
      type uint32;
      default 10;
    }
  }
  container classifications {
    leaf-list classification {
      description "Classification values to be used for the log filtering. For
example: 'OS', 'SM'.
      Logs containing any of the provided classification values will be
```

```
returned.";
    type types:log_classification_types;
  }
}
container tags {
  list tag {
    key "name";
    leaf name {
      mandatory true;
      description "Tag name to be used for the log filtering. For example: 'tenant',
'depName'."
      Logs containing any of the provided tag name plus the tag values
will be returned.";
    }
    type types:log_tag_types;
  }
  leaf value {
    mandatory true;
    description "Tag value pairs to be used for the log filtering. For example:
'adminTenant', 'CSRDeployment'";
    type string;
  }
}
}
}
output {
  container filterLogResults {
    leaf log_level {
      description "Log level used to filter for the logs.";
      type types:log_level_types;
    }
  }
  list logs {
    container classifications {
      leaf-list classification {
        description "Classifications used to filter for the logs.";
        type types:log_classification_types;
      }
    }
    container tags {
      list tag {
        key "name";
        leaf name {
          mandatory true;
          description "Tag name used to filter for the logs.";
          type types:log_tag_types;
        }
        leaf value {
          mandatory true;
          description "Tag value used to filter for the logs.";
          type string;
        }
      }
    }
  }
  leaf log_date_time {
    description "Timestamp of the log.";
    type string;
  }
  leaf log_message {
    description "The log message.";
    type string;
  }
}
}
}
```

```
}

```

NETCONF コンソールまたは `esc_nc_cli` を使用して、ConfD API ログを照会できます。

- NETCONF コンソールを使用して、次のクエリを実行します。

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin
--privKeyFile=/home/admin/.ssh/confd_id_dsa --privKeyType=dsa --rpc=log.xml
```

- `esc_nc_cli` を使用して、次のクエリを実行します。

```
esc_nc_cli --user <username> --password <password> filter-log log.xml
```

次に、`log.xml` の例を示します。

```
<filterLog xmlns="https://www.cisco.com/esc/esc">
  <log_level>INFO</log_level>
  <log_count>1</log_count>
  <classifications>
    <classification>OS</classification>
    <classification>SM</classification>
  </classifications>
  <tags>
    <tag>
      <name>depName</name>
      <value>CSR_ap1</value>
    </tag>
    <tag>
      <name>tenant</name>
      <value>admin</value>
    </tag>
  </tags>
</filterLog>
```

応答は次のとおりです。

```
<rpc-reply xmlns="urn:iETF:params:xml:ns:netconf:base:1.0" message-id="1">
  <filterLogResults xmlns="https://www.cisco.com/esc/esc">
    <log_level>INFO</log_level>
    <logs>
      <classifications>
        <classification>OS</classification>
        <classification>SM</classification>
      </classifications>
      <tags>
        <tag>
          <name>depName</name>
          <value>CSR_ap1</value>
        </tag>
        <tag>
          <name>tenant</name>
          <value>admin</value>
        </tag>
      </tags>
      <log_date_time>13:06:07,575 31-Oct-2016</log_date_time>
      <log_message> No pending work flow to start.</log_message>
    </logs>
  </filterLogResults>
</rpc-reply>
```



(注) ログイン API の応答は XML 形式です。ログメッセージに XML 文字が含まれている場合はその文字がエスケープされるため、XML 準拠は解除されません。

ESC ログファイルの表示

次の表に、さまざまな ESC コンポーネントのログを示します。

ファイル	コンポーネント	説明	ローテーションサイズ	バックアップファイルの数	アクティブ/アクティブ展開
/var/log/esc/escmanager.log	ESCManager	これには、ワークフロー、要求、および持続性を含む ESC マネージャのログが含まれています。	150 MB	10	対応可
/var/log/esc/escmanager_tagged.log	ESCManager	これは <code>escmanager.log</code> と同じですが、 <code>netconf</code> ログイン API 用に読みやすい形式が使用されており、必要に応じて他のパーサーで使用できます。	150 MB	10	対応可

ファイル	コンポーネント	説明	ローテーションサイズ	バックアップファイルの数	アクティブ/アクティブ展開
/var/log/esc/yangesc.log	ESCManager	これには、netconf 要求と通知に関連するログが含まれています。	150 MB	10	対応可
/var/log/esc/error_escmanager.log	ESCManager	すべてのエラーログエントリ。	150 MB	10	対応可
/var/log/esc/trace/event_escmanager.log	ESCManager		150 MB	10	対応可
/var/log/esc/trace/escdatabase.log	ESCManager	データベース関連のログエントリ			対応可
/var/log/esc/trace/debug_yangesc.log	ESCManager		51 MB	2	対応可
/var/log/esc/trace/esc_rest.log	ESCManager		150 MB	10	対応可
/var/log/esc/mona/mona.log	MONA		150 MB	10	対応可
/var/log/esc/mona/actions_mona.log	MONA		150 MB	10	対応可
/var/log/esc/mona/rules_mona.log	MONA		150 MB	10	対応可
/var/log/esc/vimmanager/vimmanager.log	VIM マネージャサービス	詳細な VIM マネージャのログ。	150 MB	10	対応可
/var/log/esc/vimmanager/operations_vimmanager.log	VIM マネージャサービス	VIM マネージャの操作が処理されたことのみを示す簡単なログ。	150 MB	10	対応可

ファイル	コンポーネント	説明	ローテーションサイズ	バックアップファイルの数	アクティブ/アクティブ展開
/var/log/esc/trace/vimmanager /vim_vimmanager.log	VIM マネージャサービス	VIM マネージャと VIM 間の Raw HTTP 要求/応答 (ヘッダーを含む)。このログを追跡する OpenStack では、ログレベルが VIM マネージャのデバッグに設定されている必要があることに注意してください。	150 MB	10	応対可
/var/log/esc/<timestamp>-esc-portal-be.log	ESC UI		10 MB	4	応対可
/var/log/esc/confd/audit.log	confd		10 MB	4	応対可
/var/log/esc/confd/browser.log	confd		10 MB	4	応対可
/var/log/esc/confd/confd.log	confd		10 MB	4	応対可
/var/log/esc/confd/devel.log	confd		10 MB	4	応対可
/var/log/esc/confd/netconf.log	confd		10 MB	4	応対可
/var/log/esc/confd/netconf.trace	confd		10 MB	4	応対可
/var/log/esc/confd/global.data			循環なし		応対可
/var/log/esc/esc_monitor.log	ESC インフラまたは HA		10 MB	4	使用不可

ファイル	コンポーネント	説明	ローテーションサイズ	バックアップファイルの数	アクティブ/アクティブ展開
/var/log/esc/esc_monitor_output.log	ESC インフラまたは HA		10 MB	4	使用不可
/var/log/esc/esc_conf.d.log	ESC インフラまたは HA		10 MB	4	使用不可
/var/log/esc/pgstartup.log	ESC インフラまたは HA		10 MB	4	使用不可
/var/log/esc/spy.log	ESC インフラまたは HA		ログなし (サイズ 0)	ESC で生成されたログはありません。	使用不可
/var/log/esc/catalina.out	Tomcat		循環なし	ESC で生成されたログはありません。エラーのみ。	対応可
/var/log/esc/esc_dbtool.log	DB ツール		循環なし		対応可
/var/log/esc/snmp/snmp.log	SNMP エージェント		循環なし		使用不可

ファイル	コンポーネント	説明	ローテーションサイズ	バックアップファイルの数	アクティブ/アクティブ展開
/var/log/esc/etsi-vnfm/etsi-vnfm.log	ETSI サービス	これは、要求、応答、ペイロード、および必要に応じた一般的なロギング情報を含む、ETSI 処理の主要なログファイルです。	150MB	10	対応可
/var/log/esc/etsi-vnfm/events-etsi-vnfm.log	ETSI サービス	API 要求のみ受信と発信の両方についてログに記録します。	150MB	10	対応可
/var/log/esc/etsi-vnfm/event-details-etsi-vnfm.log	ETSI サービス	実際の JSON ペイロードとともに、API 要求（受信と発信）の両方をログに記録します。	150MB	10	対応可

ファイル	コンポーネント	説明	ローテーションサイズ	バックアップファイルの数	アクティブ/アクティブ展開
/var/log/esc/escadm.log	Escadm サービス	escadm.py から手動と自動の両方のメッセージとエラーをキャプチャするためにログに記録します。このログは、ESC のスタートアップおよび設定変更を追跡する場合に役立ちます。	10 MB	4	応対可
/var/log/esc/elector.log	Elector サービス	ログエントリは、リーダーシップの決定を記録します。	150 MB	10	アクティブ/アクティブのみで使用可能
/var/log/esc/consul_agent.log	Consul エージェント	Consul サーバを使用して ESC Consul エージェントを記録するログエントリ。	150MB	10	アクティブ/アクティブのみで使用可能
/var/log/esc/geo.log	GEO サービス	ログエントリは GEO の状態と遷移を記録	150MB	10	GEO アクティブ/アクティブのみで使用可能