オンライン診断の設定

- オンライン診断の設定に関する情報，1 ページ
- オンライン診断の設定方法，2 ページ
- オンライン診断のモニタリングおよびメンテナンス，7 ページ
- オンライン診断テストの設定例，8 ページ
- オンライン診断に関する追加情報，10 ページ
- オンライン診断設定の機能履歴と情報，11 ページ

オンライン診断の設定に関する情報

オンライン診断

オンライン診断では、動作中のネットワークにスイッチが接続されている間に、スイッチのハードウェア機能についてテストし、確認することができます。

オンライン診断には、異なるハードウェア コンポーネントをチェックするパケット スイッチング テストが含まれ、データ パスおよび制御信号が確認されます。

オンライン診断では、次の領域の問題が検出されます。

- ハードウェア コンポーネント
- インターフェイス（イーサネット ポートなど）
- はんだ接合

オンライン診断は、オンデマンド診断、スケジュール診断、ヘルスモニタリング診断に分類できます。 オンデマンド診断は、CLI から実行されます。スケジュールされた診断は、動作中のネットワークにスイッチが接続されているときに、ユーザが指定した間隔または指定した時刻に実行されます。ヘルスモニタリングは、バックグラウンドでユーザが指定した間隔で実行されます。デフォルトでは、30 秒ごとにヘルス モニタリング テストが実行されます。

オンライン診断を設定したあと、手動で診断テストを開始したり、テスト結果を表示したりできます。 また、スイッチまたはスイッチスタックに設定されているテストの種類、およびすでに実行された診断テスト名を確認できます。

オンライン診断の設定方法

オンライン診断テストの開始

スイッチで実行する診断テストを設定したあと、**diagnostic start** 特権 EXEC コマンドを使用して診断テストを開始します。

テストを開始したら、テスト プロセスの停止はできません。

手動でオンライン診断テストを開始するには、次の特権 EXEC コマンドを使用します。

手順の概要

1. **diagnostic start switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**diagnostic start switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**} 例： `Controller# diagnostic start switch 2 test basic`	診断テストを開始します。 **switch** *number* キーワードは、スタック構成スイッチだけでサポートされます。 範囲は 1 ～ 4 です。 次のいずれかのオプションを使用してテストを指定できます。 • *name*：テストの名前を入力します。 • *test-id*：テストの ID 番号を入力します。 • *test-id-range*：カンマとハイフンで区切ってテスト ID の範囲を整数で入力します。 • **all**：すべてのテストを開始します。 • **basic**： 基本テスト スイートを開始します。 • **complete**：完全なテスト スイートを開始します。 • **minimal**：最小限のブートアップ テスト スイートを開始します。 • **non-disruptive**：ノンディスラプティブ テスト スイートを開始します。 • **per-port**：ポート単位のテスト スイートを開始します。

	コマンドまたはアクション	目的

オンライン診断の設定

診断モニタリングをイネーブルにする前に、障害しきい値およびテストの間隔を設定する必要があります。

オンライン診断のスケジューリング

特定のスイッチについて指定した時間、または日、週、月単位でオンライン診断をスケジューリングできます。 スケジューリングを削除するには、コマンドの **no** 形式を入力します。

手順の概要

1. **configure terminal**
2. **diagnostic schedule switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**} {**daily** | **on** *mm dd yyyy hh:mm* | **port** *inter-port-number port-number-list* | **weekly** *day-of-week hh:mm*}

手順の詳細

	コマンドまたはアクション	目的
ステップ **1**	**configure terminal** 例： `Controller# configure terminal`	グローバル コンフィギュレーション モードを開始します
ステップ **2**	**diagnostic schedule switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**} {**daily** | **on** *mm dd yyyy hh:mm* | **port** *inter-port-number port-number-list* | **weekly** *day-of-week hh:mm*} 例： `Controller(config)# diagnostic schedule switch 3 test 1-5 on July 3 2013 23:10`	特定日時のオンデマンド診断テストをスケジュールします。 **switch** *number* キーワードは、スタック構成スイッチだけでサポートされます。 範囲は 1 ～ 4 です。 スケジュールするテストを指定する場合は、次のオプションを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。

	コマンドまたはアクション	目的
		• *test-id-range*：**show diagnostic content** コマンドの出力に表示される複数のテストの ID 番号です。 • **all**：すべてのテスト ID です。 • **basic**：基本的なオンデマンドの診断テストを開始します。 • **complete**：完全なテスト スイートを開始します。 • **minimal**：最小限のブートアップテストスイートを開始します。 • **non-disruptive**：ノンディスラプティブテストスイートを開始します。 • **per-port**：ポート単位のテスト スイートを開始します。 テストは次のようにスケジュールできます。 • 毎日：**daily** *hh:mm* パラメータを使用します。 • 特定日時：**on** *mm dd yyyy hh:mm* パラメータを使用します。 • 毎週：**weekly** *day-of-week hh:mm* パラメータを使用します。

ヘルス モニタリング診断の設定

スイッチが稼働中のネットワークに接続されている間に、スイッチに対しヘルス モニタリング診断テストを設定できます。 ヘルス モニタリング テストの実行間隔を設定したり、テスト失敗時のスイッチの Syslog メッセージ生成をイネーブルにしたり、特定のテストをイネーブルにできます。
テストをディセーブルにするには、コマンドの **no** 形式を入力します。

デフォルトでは、ヘルスモニタリングはディセーブルですが、スイッチはテストの失敗時にSyslogメッセージを生成します。

手順の概要

1. **configure terminal**
2. **diagnostic monitor interval switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all**} *hh:mm:ss milliseconds day*
3. **diagnostic monitor syslog**
4. **diagnostic monitor threshold switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all**} **failure count** *count*
5. **diagnostic monitor switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all**}
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Controller# configure terminal`	グローバル コンフィギュレーション モードを開始します
ステップ 2	**diagnostic monitor interval switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all**} *hh:mm:ss milliseconds day* 例： `Controller(config)# diagnostic monitor interval switch 2 test 1 12:30:00 750 5`	指定のテストに対し、ヘルス モニタリングの実行間隔を設定します。 **switch** *number* キーワードは、スタック構成スイッチだけでサポートされます。 テストを指定する場合は、次のいずれかのパラメータを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示される複数のテストの ID 番号です。 • **all**：すべての診断テスト 間隔を指定する場合は、次のパラメータを設定します。 • *hh:mm:ss*：モニタリング間隔（時間、分、秒）。指定できる範囲は、*hh* が 0 ～ 24、*mm* および *ss* が 0 ～ 60 です。 • *milliseconds*：モニタリング間隔（ミリ秒（ms））。範囲は 0 ～ 999 です。 • *day*：モニタリング間隔（日数）。範囲は 0 ～ 20 です。
ステップ 3	**diagnostic monitor syslog** 例： `Controller(config)# diagnostic monitor syslog`	（任意）ヘルス モニタリング テストの失敗時にスイッチが Syslog メッセージを生成するように設定します。
ステップ 4	**diagnostic monitor threshold switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all**} **failure count** *count* 例： `Controller(config)# diagnostic`	（任意）ヘルスモニタリングテストの失敗しきい値を設定します。 テストを指定する場合は、次のいずれかのパラメータを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。

	コマンドまたはアクション	目的
	`monitor threshold switch 2 test 1 failure count 20`	• *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示される複数のテストの ID 番号です。 • **all**：すべての診断テスト 失敗しきい値 *count* に指定できる範囲は 0 ～ 99 です。
ステップ 5	**diagnostic monitor switch** *number* **test** {*name* | *test-id* | *test-id-range* | **all**} 例： `Controller(config)# diagnostic monitor switch 2 test 1`	指定のヘルス モニタリング テストをイネーブルにします。 **switch** *number* キーワードは、スタック構成スイッチだけでサポートされます。 範囲は 1 ～ 9 です。 テストを指定する場合は、次のいずれかのパラメータを使用します。 • *name*：**show diagnostic content** コマンドの出力に表示されるテストの名前です。 • *test-id*：**show diagnostic content** コマンドの出力に表示されるテストの ID 番号です。 • *test-id-range*：**show diagnostic content** コマンドの出力に表示される複数のテストの ID 番号です。 • **all**：すべての診断テスト
ステップ 6	**end** 例： `Controller(config)# end`	特権 EXEC モードに戻ります。

次の作業

間隔をデフォルト値またはゼロに変更するには、**no diagnostic monitor interval test***test-id* | *test-id-range* } グローバル コンフィギュレーション コマンドを使用します。 ヘルスモニタリング テストに失敗した場合、**no diagnostic monitor syslog** コマンドを使用して、Syslog メッセージの生成をディセーブルに設定します。 失敗しきい値を削除するには、**diagnostic monitor threshold test***test-id* | *test-id-range* }**failure count** コマンドを使用します。

オンライン診断のモニタリングおよびメンテナンス

オンライン診断テストとテスト結果の表示

スイッチまたはスイッチ スタックに設定されているオンライン診断テストを表示し、この表に示す **show** 特権 EXEC コマンドを使用してテスト結果を確認することができます。

表 *1* : 診断テスト設定および結果を表示するコマンド

コマンド	目的
show diagnostic content switch [*number* | **all**]	スイッチに対して設定されたオンライン診断を表示します。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic status	現在実行中の診断テストを表示します。
show diagnostic result switch [*number* | **all**] [**detail** | **test** {*name* | *test-id* | *test-id-range* | **all**} [**detail**]]	オンライン診断テストの結果を表示します。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic switch [*number* | **all**] [**detail**]	オンライン診断テストの結果を表示します。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic schedule switch [*number* | **all**]	オンライン診断テストのスケジュールを表示します。 **switch** [*number* | **all**] パラメータは、スタック構成スイッチだけでサポートされます。
show diagnostic post	POST 結果を表示します （この出力は、**show post** コマンドの出力と同じです）。

オンライン診断テストの設定例

例：診断テストの開始

次に、テスト名を指定して診断テストを開始する例を示します。

```
Controller# diagnostic start switch 2 test TestInlinePwrCtlr
```

次に、すべての基本診断テストを開始する例を示します。

```
Controller# diagnostic start switch 1 test all
```

例：ヘルス モニタリング テストの設定

次に、ヘルス モニタリング テストを設定する例を示します。

```
Controller(config)# diagnostic monitor threshold switch 1 test 1 failure count 50
Controller(config)# diagnostic monitor interval switch 1 test TestPortAsicStackPortLoopback
```

例：スケジュール診断テスト

次に、特定のスイッチに対して、特定の日時にスケジュール診断テストを実行するようにスケジューリングする例を示します。

```
Controller(config)# diagnostic schedule test DiagThermalTest on June 3 2013  22:25
```

次の例では、指定されたスイッチで毎週特定の時間に診断テストを行うようスケジューリングする方法を示します。

```
Controller(config)# diagnostic schedule switch 1 test 1,2,4-6 weekly saturday 10:30
```

例：オンライン診断の表示

次に、オンデマンド診断設定を表示する例を示します。

```
Controller# show diagnostic ondemand settings

Test iterations = 1
Action on test failure = continue
```

次に、障害の診断イベントを表示する例を示します。

```
Controller# show diagnostic events event-type error

Diagnostic events (storage for 500 events, 0 events recorded)
Number of events matching above criteria = 0

No diagnostic log entry exists.
```

次に、診断テストの説明を表示する例を示します。

```
Controller# show diagnostic description switch 1 test all

DiagGoldPktTest :
        The GOLD packet Loopback test verifies the MAC level loopback
        functionality. In this test, a GOLD packet, for which doppler
        provides the support in hardware, is sent. The packet loops back
        at MAC level and is matched against the stored packet. It is a non
        -disruptive test.

DiagThermalTest :
        This test verifies the temperature reading from the sensor is below the yellow
        temperature threshold. It is a non-disruptive test and can be run as a health
monitoring test.

DiagFanTest :
        This test verifies all fan modules have been inserted and working properly on the
board
        It is a non-disruptive test and can be run as a health monitoring test.

DiagPhyLoopbackTest :
        The PHY Loopback test verifies the PHY level loopback
        functionality. In this test, a packet is sent which loops back
        at PHY level and is matched against the stored packet. It is a
        disruptive test and cannot be run as a health monitoring test.

DiagScratchRegisterTest :
        The Scratch Register test monitors the health of application-specific
        integrated circuits (ASICs) by writing values into registers and reading
        back the values from these registers. It is a non-disruptive test and can
        be run as a health monitoring test.

DiagPoETest :
        This test checks the PoE controller functionality. This is a disruptive test
        and should not be performed during normal switch operation.

DiagStackCableTest :
        This test verifies the stack ring loopback functionality
        in the stacking environment. It is a disruptive test and
        cannot be run as a health monitoring test.

DiagMemoryTest :
        This test runs the exhaustive ASIC memory test during normal switch operation
        NG3K utilizes mbist for this test. Memory test is very disruptive
        in nature and requires switch reboot after the test.

Controller#
```

次に、ブートアップ レベルを表示する例を示します。

```
Controller# show diagnostic bootup level

Current bootup diagnostic level: minimal

Controller#
```

オンライン診断に関する追加情報

関連ドキュメント

関連項目	マニュアル タイトル
システム管理コマンド	*System Management Command Reference (Cisco WLC 5700 Series)*
プラットフォームに依存しないコマンド リファレンス	*Configuration Fundamentals Command Reference, Cisco IOS XE Release 3S*（*Catalyst 3850 Switches*）
プラットフォームに依存しない設定情報	*Configuration Fundamentals Configuration Guide, Cisco IOS XE Release 3S*（*Catalyst 3850 Switches*）

標準および **RFC**

標準/**RFC**	タイトル
なし	—

MIB

MIB	**MIB** のリンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

オンライン診断設定の機能履歴と情報

リリース	変更
Cisco IOS XE 3.2SE	この機能が導入されました。