



Kiwi CatTools

Backup and manage your your network devices with ease

訳：ジュピターテクノロジー株式会社

平成20年6月9日

Latest version available from: www.kiwisyslog.com, www.jtc-i.co.jp

Support: support@kiwisyslog.com, support@jtc-i.co.jp

Kiwi CatTools

ネットワーク管理・監視ツール

by Kiwi Enterprises

Kiwi CatTools の特長は次のとおりです。

Windows 98/ME/2000/XP/2003/Vista 上で動作する Standard 版

Windows 2000/XP/2003/Vista 上で動作する Service 版

その場で実行可能な[Run Now]オプションの付いたスケジュールベースの操作性

データベースの暗号化機能

プログラム/デバイスデータベースへのアクセスに対する任意のパスワード保護

Telnet、SSH1、SSH1.5、SSH2 による接続

[Connect via]による中間デバイスの指定。Telnet および SSH は Cisco、HP、Enterasys、CatalystCatOS、CatalystIOS、Linux 機器でサポート

タブ区切りのテキストファイルによるデバイスの一括インポート/エクスポート機能

内蔵のレポート表示ウィンドウ

マルチスレッドによる高速バックアップおよびデータ収集

完全にスクリプト可能な柔軟性。VBscript を使用して独自のカスタムアクティビティ・レポートの作成が可能です。

目次

前書き	0
1 Kiwi CatTools	9
1.1 Kiwi CatTools の製品概要	9
1.2 インストール	14
1.3 Kiwi CatTools を試してみる	14
1.4 ご意見およびバグ報告のお願い	17
1.5 ソフトウェアライセンス条項	17
2 実行方法の選択	19
2.1 CatTools をサービスとして実行する	19
2.1.1 サービスの自動起動	20
2.1.2 サービスとは	21
2.1.3 CatTools Manager とは	21
2.1.4 UNC およびマップ済みドライブへのアクセス	22
2.2 CatTools をアプリケーションとして実行する	22
3 デバイス	23
3.1 カスタムデバイスを作成する	23
3.1.1 カスタムデバイスの作成方法	24
3.1.2 カスタムデバイスタイプファイル(.ini)	27
3.1.3 カスタムデバイススクリプトファイル(.txt)	38
3.1.4 カスタムデバイススクリプトの cl 変数および関数	43
3.1.5 カスタムデバイスのテスト	57
3.2 デバイス固有の情報	59
3.2.1 Dell 製デバイス	59
3.2.2 ASA dap.xml ファイル	59
3.2.3 HP2500 シリーズのスイッチ	60
3.2.4 3Com Superstack スイッチ	60
3.2.5 Cisco Terminal Server を経由した接続	63
3.2.6 Cisco および Netscreen でサポート可能な SSH2	65
3.2.7 Cisco VPN	66
3.3 サポート対象外デバイスおよびデバイスアクティビティ	67
4 アクティビティ	67
4.1 概要	67

4.2	スケジュール化されているアクティビティに関する詳細の追加/編集フォーム	68
4.2.1	[Activity]タブ	69
4.2.2	[Time]タブ	70
4.2.3	[Devices]タブ	71
4.2.4	[E-mail]タブ	72
4.2.5	[Options]タブ	72
4.3	アクティビティの一覧	73
4.3.1	DB.UpdateDevice.Password field	73
4.3.1.1	アクティビティの設定	74
4.3.2	DB.UpdateDevice.Text field	75
4.3.2.1	アクティビティの設定	75
4.3.3	Device.Backup.Running Config	76
4.3.3.1	アクティビティの設定	77
4.3.3.2	設定情報のバックアップの必要性について	80
4.3.4	Device.CLI.Modify Config	82
4.3.4.1	アクティビティの設定	82
4.3.4.2	Device.CLI.Modify Config アクティビティの使用タイミング	85
4.3.4.3	デバイスごとに異なるコマンドを送信するには	86
4.3.5	Device.CLI.Send commands	87
4.3.5.1	アクティビティの設定	87
4.3.5.2	Catalyst CatOS スイッチのポートを Enable/Disable にするには	90
4.3.5.3	Cisco IOS イメージを TFTP サーバーにバックアップするには	91
4.3.5.4	Cisco IOS の実行中の設定情報を TFTP サーバーにダウンロードするには	92
4.3.5.5	Cisco IOS の設定情報を TFTP からデバイスにアップロードするには	92
4.3.5.6	TFTP を介して Cisco IOS をアップグレードするには	93
4.3.5.7	デバイスごとに異なるコマンドを送信するには	94
4.3.5.8	複数のデバイスの出力を 1 つのファイルにまとめて保存するには	95
4.3.6	Device.ConnectivityTest.Login	96
4.3.6.1	アクティビティの設定	97
4.3.7	Device.ConnectivityTest.Ping	97
4.3.7.1	アクティビティの設定	98
4.3.8	Device.InterDevice.Ping	99
4.3.8.1	アクティビティの設定	100
4.3.8.2	Device.InterDevice.Ping アクティビティの使用タイミング	101
4.3.9	Device.TFTP.Upload Config	102

4.3.9.1	アクティビティの設定.....	102
4.3.9.2	サポート対象デバイス.....	103
4.3.10	Device.Update.Banner.....	103
4.3.10.1	アクティビティの設定.....	103
4.3.10.2	サポート対象デバイス.....	104
4.3.11	Device.Update.Password.....	105
4.3.11.1	アクティビティの設定.....	105
4.3.11.2	サポート対象デバイス.....	106
4.3.12	Report.ARP table.....	106
4.3.12.1	アクティビティの設定.....	107
4.3.13	Report.CDP Neighbors table.....	108
4.3.13.1	アクティビティの設定.....	108
4.3.13.2	サポート対象デバイス.....	109
4.3.14	Report.Compare.Running Startup.....	109
4.3.14.1	アクティビティの設定.....	110
4.3.14.2	サポート対象デバイス.....	111
4.3.15	Report.Compare.Two files.....	112
4.3.15.1	アクティビティの設定.....	112
4.3.16	Report.Error info table.....	113
4.3.16.1	アクティビティの設定.....	114
4.3.16.2	サポート対象デバイス.....	115
4.3.17	Report.MAC address table.....	115
4.3.17.1	アクティビティの設定.....	115
4.3.18	Report.Port info table.....	116
4.3.18.1	アクティビティの設定.....	117
4.3.19	Report.SNMP.System summary.....	117
4.3.19.1	アクティビティの設定.....	118
4.3.19.2	サポート対象デバイス.....	118
4.3.20	Report.Version table.....	119
4.3.20.1	アクティビティの設定.....	119
4.3.21	Report.X-Ref.Port MAC ARP.....	120
4.3.21.1	アクティビティの設定.....	120
4.4	内部関数.....	121
4.4.1	テキスト無視.....	122
4.4.2	ファイル名変数.....	122

4.4.3	メタデータ	123
4.4.3.1	メタコマンド	124
4.4.3.1.1	%ctDB コマンド	124
4.4.3.1.1.1	デバイステーブル	126
4.4.3.1.2	%ctUM	127
4.4.3.2	メタ変数	128
4.5	カスタムアクティビティの作成	129
4.5.1	カスタムアクティビティの作成方法	131
4.5.2	カスタムアクティビティタイプファイル(.ini)	135
4.5.3	カスタムアクティビティメインスクリプトファイル(.txt)	141
4.5.4	カスタムアクティビティクライアントスクリプトファイル(.txt)	145
4.5.5	カスタムアクティビティデバイススクリプトファイル(.custom)	148
4.5.6	カスタムアクティビティスクリプトに使用でき cl./ct.変数および関数	153
4.5.6.1	クライアントスクリプトファイルで使用する cl.変数および関数	153
4.5.6.2	メインスクリプトファイルで使用する ct.変数および関数	160
4.5.7	カスタムアクティビティのテスト	168
4.6	デバイスでサポートされていないアクティビティ	171
5	メニュー	172
5.1	[File]	172
5.1.1	[Database]	172
5.1.1.1	[Export]	172
5.1.1.1.1	[Export devices to tab delimited file]	172
5.1.1.1.2	[Export activities to tab delimited file]	172
5.1.1.2	[Import]	173
5.1.1.2.1	[Import devices from tab delimited file]	173
5.1.1.2.2	[Import activities from tab delimited file]	177
5.1.1.3	[Change encryption key password]	178
5.1.1.4	[Backup current database]	179
5.1.1.5	[Restore database from backup]	180
5.1.1.6	[Squeeze current database]	180
5.1.1.7	他のデータベースを開く [Open/New]	180
5.1.1.8	新しいデータベースを作成する [Open/New]	181
5.1.2	[Delete]	181
5.1.2.1	[Delete Infolog.txt file]	181
5.1.3	[Enable capture mode]	181

5.1.4	[Debug]	182
5.1.4.1	[Create sanitised database report for Kiwi Technical support]	182
5.1.5	[Exit].....	183
5.2	[View].....	183
5.2.1	[View Reports folder]	183
5.2.2	[View Captured Data folder].....	183
5.2.3	[View Configs folder].....	184
5.2.4	[View Statistics folder].....	184
5.2.5	[View Email log].....	184
5.2.6	[View Activity log]	184
5.2.7	View Info log.....	184
5.3	[Options].....	185
5.3.1	[Setup]	185
5.3.1.1	[Email]	185
5.3.1.1.1	[General Options].....	185
5.3.1.1.2	[Primary SMTP Server Setup].....	187
5.3.1.1.3	[Secondary SMTP Server Setup]	188
5.3.1.2	[Logging].....	189
5.3.1.3	[Misc]	192
5.3.1.4	[TFTP Server].....	193
5.4	[Interface].....	195
5.4.1	[Panels].....	195
5.4.2	[Themes].....	195
5.5	[Help].....	196
5.5.1	[Contents].....	196
5.5.2	[Online FAQ]	196
5.5.3	[Purchase Kiwi CatTools]	196
5.5.4	[Enter registration details].....	196
5.5.5	[About]	197
6	[Panels].....	197
6.1	概要.....	198
6.2	[Devices]	202
6.2.1	[Add]	202
6.2.1.1	[Device info].....	203
6.2.1.2	[Passwords]	204

6.2.1.3	[Prompts].....	205
6.2.1.4	[Contact info].....	206
6.2.1.5	[Extra info].....	206
6.2.2	[Remove].....	207
6.2.3	[Edit].....	207
6.2.4	[Copy].....	207
6.2.5	[Filter]	208
6.2.6	[Show All]	209
6.3	[Activities].....	209
6.3.1	[Add]	209
6.3.2	[Edit].....	210
6.3.3	[Copy].....	210
6.3.4	[Run now]	210
6.3.5	[Start timer]	210
6.3.6	[Remove].....	210
6.4	[Activity log].....	210
6.4.1	[Clear].....	210
6.4.2	[View report].....	211
6.5	[Compare].....	211
6.5.1	[Save].....	211
6.5.2	[Clear].....	211
6.5.3	[Select].....	211
6.6	[Info log]	211
6.6.1	ビューフィルタ(ドロップダウンリスト).....	211
6.6.2	[Clear].....	212
6.7	[Report]	212
6.7.1	[Open].....	212
6.7.2	[Save].....	212
6.7.3	[Clear].....	212
6.7.4	[Delete].....	212
6.7.5	[Refresh].....	212
6.8	[TFTP]	213
6.8.1	[Start]	213
6.8.2	[Stop]	213
6.8.3	[Clear].....	213

6.9	[Display]	213
6.9.1	[Clear].....	213
6.10	[Mail].....	213
7	通知	214
7.1	Eメール.....	214
8	カスタムスクリプティング	215
8.1	スクリプトエディター	216
9	問題が発生したときの対応策.....	217
9.1	デバイス固有の問題について	217
9.2	[Info Log]に表示される"Error: 70 Permission denied".....	217
9.3	障害報告時のお願い	218
9.4	リモートデスクトップシステム.....	219
9.5	リモート認証.....	220
9.6	アンチウイルスツール	220
9.7	XP ファイアウォール.....	221
9.8	サービスは稼動しているがスケジューラーが機能しない	221
10	インストール手順.....	221
10.1	CatTools の自動インストール	222
10.2	インストール/アップグレード後の問題.....	222

1 Kiwi CatTools



Kiwi CatTools - コンフィギュレーション自動化追跡ツール

Program copyright 1996 - 2008 Kiwi Enterprises. All rights reserved.

ホームページ:

<http://www.kiwisyslog.com>

サポート:

<http://www.kiwisyslog.com/support>

はじめてご使用になるとき:

Kiwi CatTools を使ってみる

FAQ・オンラインヘルプ:

<http://www.kiwisyslog.com/support>

評価用ライセンス:

<http://www.jtc-i.co.jp/>

ライセンスのご購入先:

<http://www.jtc-i.co.jp/>

アップグレード情報:

<http://www.kiwisyslog.com/kiwi-cattools-upgrading>

リリースノート履歴:

<http://www.kiwisyslog.com/cattools/history>

1.1 Kiwi CatTools の製品概要

Kiwi CatTools はルーター、スイッチ、ファイアウォールなどの**デバイスのコンフィギュレーションを自動管理**するためのフリーウェアアプリケーションです。

Cisco / 3Com / Dell / Enterasys / Extreme / Foundry / HP / Junpier / Nortel 製デバイス、その他多くの機器に対応しています。

Kiwi CatTools が実行する多くのタスクの中でも次の機能・特長は日々の管理業務の効率化に大いに貢献します。

- コンフィグレーションの自動バックアップと変更が加えられたときにすぐさま電子メールで管理者に連絡が入る E メール通知機能
- 複数のデバイスに対する Telnet や SSH を通じたコマンド同時発行
- コンフィグレーションを自動的に変更するスケジュール機能
- 全てのネットワークデバイスのパスワードの一括変更

本製品はスクリプトに完全対応したコンフィグレーション管理ツールでもあり、Telnet、SSHなどをサポートする TFTP サーバーが内蔵されています。

以下の OS に対応しています。

- Windows 98
- Windows ME
- Windows 2000
- Windows XP
- Windows 2003
- Windows Vista

Kiwi CatTools には Application 版と Service 版の 2 つのライセンスがあります。両バージョンとも 1 つのパッケージに同梱されています。

Application 版は対話形式で実行され、システムにログオンしているユーザーが 1 人のみのときに使用できます。

Service 版は自動 NT サービスとして実行されます。このバージョンでは、ユーザーが CatTools を操作する際にログオンする必要はありません。

Kiwi CatTools の特長

- Windows 98/ME/2000/XP/2003/Vista 上では標準的なアプリケーションとして動作
- Windows 2000/XP/2003/Vista 上ではサービスとして動作
- スケジュールベースのオペレーションもその場で実行可能な[Run Now]オプション
- 重要なデータベースフィールドの暗号化機能
- プログラムへのアクセスに対するパスワード保護機能
- Telnet, SSH1, SSH1.5, SSH2 による接続
- [Connect via]によるデバイス中継接続
- タブ区切りのテキストファイルによるデバイスの一括インポート/エクス

ポート機能

- 内蔵レポート表示ウィンドウ
- マルチスレッドによる高速バックアップとデータ収集
- マルチスレッド TFTP サーバー内蔵
- VBscript を使用して独自のカスタムデバイスタイプの作成を実現する、デバイステンプレートスクリプト

サポートしているアクティビティ: *

- DB.UpdateDevice.Password Field (CatTools データベースの[Device]テーブルにあるパスワードフィールドを更新します。)
- DB.UpdateDevice.Text Field (CatTools データベースの[Device]テーブルにあるテキストフィールドを更新します。)
- Device.Backup.Running Config (使用中のコンフィグファイルのバックアップを取り、ディスクに保存されている最新のコンフィグファイルと比較し、差分レポートを出力します。)
- Device.CLI.Modify Config (使用中のコンフィグファイルにコマンドを送信します。)
- Device.CLI.Send commands (特権モードでコマンドを送り、結果をキャプチャしてファイル化します。)
- Device.ConnectivityTest.Login (各デバイスに有効モードでログインします。)
- Device.ConnectivityTest.Ping (各デバイスに ping コマンドを送り、応答結果を返します。)
- Device.InterDevice.Ping (各デバイスから一連のアドレスまたはリストに記載されているすべてのデバイスに対し ping コマンドを送ります。)
- Device.TFTP.Upload Config (1 つのデバイスに対してテキストのコンフィグファイルアップロードします。)
- Device.Update.Banner (デバイスに対してバナーを適用します。)
- Device.Update.Password (指定のデバイスのパスワードを変更します。)
- Report.ARP table (ARP テーブルのエントリのレポートを生成し、変更を追跡します。)
- Report.CDP Neighbors table (隣接したデバイスのレポートを生成します。)
- Report.Compare.Running Startup (デバイスの現在のコンフィグファイルとスタートアップ時のコンフィグファイルを比較し、差分のレポート

を生成します。)

- Report.Compare.Two files (指定した 2 つのファイルを比較し、差分レポートを生成します。)
- Report.Error info table (多くの Cisco 製デバイスに表示されたエラーカウンタのレポートを生成します。)
- Report.MAC address table (ネットワーク上に存在するすべての MAC アドレスのリストを生成します。)
- Report.Port info table (インターフェイスのコンフィギュレーションおよびデバイスのステータスのレポートを生成します。)
- Report.SNMP.System summary (SNMP を介して収集したデバイス情報のサマリレポートを生成します。)
- Report.Version table (デバイスのシリアル番号、ハードウェアおよびソフトウェアのバージョンのレポートを生成します。)
- Report.X-Ref.Port MAC ARP (ポート/MAC/ARP の各レポートの相互参照を生成します。)

サポートしているデバイス: *

CatTools が現在サポートしているデバイスのメーカーは次のとおりです。

- Cisco (ルーター、IOS & CatOS スイッチ、ファイアウォール等)
- 3Com
- Dell
- Enterasys
- Extreme
- Foundry
- HP
- Juniper
- Nortel
- その他 現在サポートしているデバイスについては、弊社 Web サイトで公開しているデバイスの一覧を参照してください。

特定のデバイスに関する詳細については、「3.2 デバイス固有の情報」を参照してください。

* サポート対象のデバイス、レポート、アクティビティは日々更新され、増えています。時々弊社の [Web サイト](#) をチェックして、最新のバージョンをご使用になっていることを確認してください。

Kiwi CatTools では上記のデバイスに対し、汎用性が高く最新の CLI 構文を使用できます。ご使用のデバイスが Kiwi CatTools 内で正常に動作しない場合、そのデバイスがまだサポートされていない可能性があります(あるいは、実行しようとしているアクティビティのみに対してサポート対象外である可能性があります)。また、そのデバイスが使用している CLI 構文が Kiwi CatTools でサポートしている CLI 構文と異なっているために起きている現象である可能性もあります。どちらにせよ、弊社 [Web サイト](#) にアクセスして問題点をご連絡くだされば、そのデバイスをサポート対象として加えるための調査を行います。

アクティビティとデバイスの制限(ライセンス比較)

無償版

- データベース登録可能な最大デバイス数 - 5
- スケジュール可能なアクティビティ数 - 5
- 同時処理可能な最大 TFTP セッション数 - 2
- マルチスレッド処理 - 不可

Engineer 版

- データベース登録可能な最大デバイス数 - 20
- スケジュール可能な最大アクティビティ数 - 20
- 同時処理可能な最大 TFTP セッション - 10
- マルチスレッド処理 - 最高 5

Enterprise 版

- データベース登録可能な最大デバイス数 - 無制限 ファイルサイズ制限によって異なります。
- スケジュール可能な最大アクティビティ数 - 無制限 ファイルサイズ制限によって異なります。
- 同時処理可能な最大 TFTP セッション - 100
- マルチスレッド処理 - 最高 30

Enterprise 版、Professional 版、Enterprise 版のご購入方法

無償版は無料でお好きな期間使用できます。

登録するデバイスを増やしたい場合は、無償版以外の製品を[販売代理店](#)よりご購入ください。

1.2 インストール

CatTools のインストールを開始すると、アプリケーションとしてインストールするのかサービスとしてインストールするのかを選択する画面が表示されます。

この 2 つのオプションの違いについての詳細は、「2 実行方法の選択」の項を参照してください。

1.3 Kiwi CatTools を使ってみる

概要:

CatTools は大きく分けて次の 4 つの手順で使用を開始できます。

1. [Options]メニューの[Setup]で E メール情報等の共通項目を設定します。
2. [Devices]タブで 1 つ以上のネットワークデバイスの詳細情報を入力します。
3. [Activities]タブでアクティビティを作成し、使用中のデバイスを関連付けます。
4. 作成したアクティビティを実行します。[Run now]ボタンを押してすぐ実行することも、スケジューラを使って指定した日時に実行することもできます。

各手順の詳細について

CatTools の標準設定オプションを定義するには...

- [Options]メニューから[Setup]をクリックします。
- E メールオプションを設定します。
 - エラー通知およびレポートの送信先 E メールアドレスを各フィールドに入力します。
 - [From Adress]フィールドに自分の E メールアドレスを入力します。
 - ご使用のプライマリ SMTP メールサーバーの詳細情報を入力します。
 - サーバー認証が必要な場合、その詳細を入力します(ほとんどのサーバーがこの作業を必要としません。)
- [Test]ボタンを押して E メールがきちんと送受信されることを確認します。
- その他の設定については、使い始めてしばらくの間はデフォルト値のままでも使用しても構いません。

デバイスをデータベースに登録するには...

- [Device]タブをクリックします。
- [Add]ボタンをクリックします。
- 入力が必要な各フィールドに入力します。フィールド内にカーソルを置くと、そのフィールドの説明がステータスバー([Device Information]画面の下部)に表示されます。
- アスタリスク(*)付きのフィールドは入力必須フィールドです。必ず入力してください。

アクティビティをスケジュール化するには...

- [Activities]タブをクリックします。
- [Add]ボタンをクリックします。
- ドロップダウンリストから実行するアクティビティを選択します。
(例:Device.Backup.Running Config)
- そのアクティビティにわかりやすい名前を付け、説明を入力します。名前は他と重複しないよう注意してください。
- そのアクティビティに適用する有効期間を選択します(永久に有効な場合は Permanent、1 回実行して無効にする場合は Run once then deactivate を選択します)。
- レポートファイルの場所および E メールアドレスの項目は通常はデフォルトのままにしておきます。
- [Time]タブをクリックして、そのアクティビティを実行する時刻を設定します。[Reoccurring]で「Never」を選択すると、その都度[Run now]ボタンを押して手動で実行する必要があります。
- 必要であれば、データベースに作成したアクティビティを[Favorite schedule]として登録することができます。こうすると、他のアクティビティに対して設定値を再利用できるようになり、スケジュールの[Time]タブで入力の手間を省くことができます。
- [Device]タブをクリックして、このアクティビティに関連付けるデバイスすべてにチェックを入れます。
- [Options]タブをクリックし、このアクティビティに関連する項目を設定します。大抵はデフォルトのまま使用して構いません。

スケジュール化したアクティビティを実行するには...

- [Activity]タブをクリックします。
- スケジュール化するアクティビティにチェックを入れます。
- [Start timer]ボタンをクリックします。
- CatTools は「Active Timer Mode」になり、スケジュール設定に従って、アクティビティを実行します。

今すぐアクティビティを実行するには...

- [Activity]タブをクリックします。
- 今すぐ実行したいアクティビティにチェックを入れます。
- [Run now]ボタンをクリックします(あるいはアクティビティにカーソルを合わせ右クリックし、表示されるリストの中から[Run now]をクリックします)。
- 選択したアクティビティが実行され、IDOL モードに戻ります。

クライアントアクティビティを監視するには...

- [Info log]タブをクリックします。
- 画面下部にあるドロップダウンリストから、表示するメッセージレベルを選択します(情報メッセージのみ)。
- クライアントやメインプログラムから指定したレベルのメッセージを受信すると、そのメッセージがこの画面に表示されます。
- 最新の受信メッセージがリストの一番上に表示されます。
- 各エントリは列の項目名をクリックすることによって、昇順/降順に並べ替えることができます。

結果を表示するには...

- [Activity log]タブをクリックします。
- アクティビティが実行されると、その結果がこの画面に表示されます。

レポートを表示するには...

- [Report]タブをクリックします。[Open]ボタンをクリックし、この画面に表示するテキスト形式のレポートを選択します。

キャプチャしたデータおよびコンフィグファイルを表示するには…

- [View]メニューから表示したいフォルダを選択します。
- Windows エクスプローラーが開き、選択したフォルダ内が表示されます。
- フォルダ階層をたどり、任意のファイルを開いてください。

1.4 ご意見およびバグ報告のお願い

弊社では皆様からのご意見やプログラムの改善点のご報告をお待ちしております。弊社 Web サイトの[サポートフォーム](#)をご利用ください。

1.5 ソフトウェアライセンス条項

使用許諾契約書

お客様は無償版の Kiwi CatTools を、登録せずに無期限で使用できるものとしません。その場合、Kiwi Software メーリングリストに参加されることをお勧めしません。本メーリングリストではバグレポート、使用方法のヒント、新バージョンの発行などを順次お知らせしております。

無償版の Kiwi CatTools(以下「ソフトウェア製品」と表示)は、登録することによって無償版では使用できない機能や特長を使用できるようにすることが可能です。1 台の PC にインストールされているソフトウェア製品 1 つに対して 1 つの登録キーが必要です。登録キーには重複しない固有のシリアル番号と登録コードが含まれ、フル機能で使用するプログラム 1 つに対して 1 組のシリアル番号と登録コードが必要となります。

本ソフトウェア製品をインストールし、使用するためには以下の禁止条項に同意していただく必要があります。

- ソフトウェア本体ならびに付属文書の全部および一部を元にした逆コンパイル、リバースエンジニアリング、逆アセンブル、改変、派生物の生成などの行為を禁じます。
- 著作権または Kiwi Enterprise の所有権表示を削除する行為を禁じます。
- ソフトウェア製品の登録キーを正式に登録されたエンドユーザー以外の第三者に配布する行為を禁じます。
- ソフトウェア製品を第三者に貸し出す行為を禁じます。
- Kiwi Enterprise から直接発行されたものでない登録キー/シリアル番号の使用

を禁じます。

ライセンス契約の解除

本ライセンス契約書に記載されている使用許諾条項に違反した場合、他のいかなる権利に影響を与えることなく、Kiwi Enterprise は本ライセンス契約を解除することができます。その場合、本ソフトウェアおよび登録キーを含むすべての付属物およびその複製を破棄するものとします。

所有権

本ソフトウェア製品およびKiwi Enterprises社のWebサイトにて公開している全ての情報は著作権によって保護され、その所有権はKiwi Enterprise社に帰属するものとします。よって、Kiwi Enterprises社の書面による事前の承認を得ず、いかなる形式においても複製、再生産、改変、発行、アップロード、投稿、転送、配布などの行為はできません。

承認を得たい場合は、Kiwi EnterpriseのWebサイトの[サポートページ](#)に記載されている連絡先までご連絡ください。

ソフトウェア製品ライセンス

本ソフトウェア製品は著作権法ならびに著作権に関する国際条約、その他知的財産法および関連条約による保護を受けています。

本ソフトウェアはライセンス許諾による使用許可を与えるものであり、製品そのものが販売されるわけではありません。

免責事項

本ソフトウェア製品は、市場性、特定目的との適合性、権利侵害がないことなどの点を含む(ただし、必ずしもこれらに限定されない)明示または暗示の保証をすることなく、現状で提供するものとします。明示保証の例外範囲を認めていない、あるいは適用しない一部の国や地域では、上記の例外事項すべてが適用されないことがあります。

本ソフトウェアには技術上の間違いや誤字・脱字が含まれている可能性があり、これらの修正・変更・更新などは予告なく行うものとします。

また、Kiwi Enterprisesは本ソフトウェアに関わる改良やその他の変更を随時予告なく行います。

危険度の高い活動での使用禁止

本ソフトウェアは障害対策が施されておらず、フェールセーフ機構を要するような危険な環境での使用や再販を想定して設計、生産されていません。核施設での稼働、航空機航行/通信システム、航空管制システム、直接的な人命救助装置、武器システム等がこれに該当します。また、本ソフトウェアが使用できないことによって直接・間接に人命を脅かし、傷害に繋がる危険がある、あるいは重大な肉体的環境的ダメージを与えるおそれのある、いかなる環境およびシステムが含まれます。

Kiwi Enterprise では、危険度の高い活動における本ソフトウェアの使用に関していかなる明示的暗示的保証をしないことを明言します。

派生的損害に対する免責

適用される法律が許容する最大範囲において、いかなる場合においても Kiwi Enterprise 社またはソフトウェアの制作者が、本ソフトウェアの使用または使用できないことによって生じる(商業上の損失、中断、商情報の喪失、その他いかなる金銭的損失を含むが制御できない)損害に対して、たとえ Kiwi Enterprise 社がそのような損害の可能性について言及していた場合であっても、いかなる責任を負うことできません。

2 実行方法の選択

CatTools はアプリケーションとしてもサービスとしてもインストールできます。

Service 版は Kiwi CatTools を Windows のサービスとしてインストールするため、ユーザーが 1 人も Windows にログオンしていない状態でも、プログラムを実行できます。こちらを選択すると、サービスを制御するためのソフトウェア Kiwi CatTools Manager も同時にインストールされます。

Application 版は Kiwi CatTools を通常の Windows アプリケーションとしてインストールするため、プログラムを実行するには最低 1 人のユーザーが Windows にログオンしている必要があります。

2.1 CatTools をサービスとして実行する

Service 版は Kiwi CatTools を Windows のサービスとしてインストールするため、ユーザーが 1 人も Windows にログオンしていない状態でも、プログラムを実

行できます。こちらを選択すると、サービスを制御するためのソフトウェア Kiwi CatTools Manager も同時にインストールされます。

CatTools のインストールタイプを選択するユーザーアカウントに、CatTools の操作に必要なタスクを実行するための十分な権限が設定されていることが大切です。通常はローカルシステムアカウントで十分ですが、サービスとして CatTools を実行する際に支障を感じるようであれば、このユーザーアカウントは使い始めのアカウントとしてのみ使用する、ということで構いません。

CatTools Manager を使えば CatTools のサービスを対話形式で操作することができます。いわばユーザーインターフェイス(GUI)のようなものであると考えてください。

ご使用のライセンスが Enterprise か Professional かを問わず、無人でアクティビティを実行する必要がある、または 24 時間体制でスケジュール化する必要がある場合はサービスとしてインストールすることをお勧めします。CatTools を実行するための専用マシンを用意できる場合は、サービスとしてインストールするほうが良いでしょう。

2.1.1 サービスの自動起動

CatTools をサービスとしてシステムにインストールすると、自動的にスタートアップメニューに登録されます。すなわち、サーバーを再起動すると自動的に CatTools のサービスが開始される状態になります。コンソールにユーザーが 1 人もログインしていなくてもサービスは開始されます。

サービスの依存関係 (Windows 2000 Server および Windows 2003 Server をご使用の場合は特に注意してください)

CatTools のサービスは大半の OS 下で問題なく開始されます。しかし、一部の Windows 2000/2003 Server システムでは、他のいくつかのシステムサービスが開始されるのを待ってからでないと、CatTools のサービスが開始できない場合があります。サービスが開始できないときには、再起動後のコンソール画面上に "One or more system services failed to start" というエラーメッセージが表示されます。

CatTools の起動前に必要なサービスを確実に開始させるには、CatTools をインストールしたフォルダにある、CatTools Manager の設定ファイル (KiwiCatToolsManager-Settings.ini) を変更する必要があります。

サービスの依存関係を追加するには...

- メモ帳で.../KiwiCatToolsManager-Settings.ini ファイルを開きます。
- 次の行を設定ファイルに書き加えます。
[Service Options]
Dependencies="LanmanWorkstation;TCP/IP;WMI"
- ini ファイルを保存します。
- CatTools Manager を実行します。
- [Uninstall]ボタンをクリックします(CatTools のサービスが停止され、CatTools Service がアンインストールされます)。
- [Install]ボタンをクリックします(CatTools がサービスとしてインストールされ、新しい依存関係の情報で設定されます)。
- [Start]ボタンをクリックし、CatTools サービスが正常に開始されることを確認してください。

上記の手順を行うことによって、まず、ワークステーション、WMI(Windows Management Interface)、TCP/IP スタックの各サービスが開始され、次に CatTools サービスの開始されるようになります。

2.1.2 サービスとは

サービスとは、標準的なアプリケーションから取得した様々なルールに応じてマシンと情報をやりとりするプログラムのことです。標準的なアプリケーションは、その時点でログインしているユーザーの制限に応じて実行することができます。サービスは、デフォルトではローカルシステムアカウントとして指定されている特別なアカウントの下で実行され、通常はファイルへのアクセス許可などの点で多くの制約を受けます。

サービスはマシンのスイッチが入られると同時に起動されます。サービスを開始させるために誰かがログインする必要はありません。

2.1.3 CatTools Manager とは

CatTools Manager は CatTools のインターフェイスであり、CatTools サービスとの対話(または管理)を行う際に使用します。

必要であれば、Application 版を次の 2 つの部分に分けてインストールすることができます。

1. データ処理時にバックグラウンドで実行し、各アクティビティを実行す

るサービスとしての部分

2. 画面に表示して対話しながらデバイスやアクティビティを管理する Manager としての部分

Manager のインターフェイスは入力された要求を取得しサービスに渡します。サービスはそれを処理して応答を返します。Manager は返された応答内容を翻訳し画面に表示します。

2.1.4 UNC およびマップ済みドライブへのアクセス

デフォルトでは、Service 版は Local System アカウントを使ってログインするよう設定されています。このアカウントは、CatTools を実行する際にマシンにアクセスするために使用しているユーザーアカウントとは異なるアカウントです。ユーザーアカウントによって定義され、そのユーザーアカウントからでないとはアクセスできないマップ済みドライブに対して、このアカウントからアクセスすることはできません。

可能であれば、CatTools Service によって実行されるアクティビティがネットワーク上のパスや事前に保存されているデータにアクセスしてその情報を読み取る必要のある場合は、UNC のフルパス名を指定して、そのように処理させてください。この作業をしないと、不正確な結果(たとえば、バックアップアクティビティを何度実行しても常に「Configuration is New」というステータスが返されるなど)を示すアクティビティレポートが出力される可能性があります。これは、CatTools Service では、現在の設定ファイルのアクティビティ設定の項目で指定されているネットワークパスにアクセスできないために起こる現象です。

2.2 CatTools をアプリケーションとして実行する

Application 版は Kiwi CatTools を通常の Windows のアプリケーションとしてインストールするため、プログラムを実行するにはユーザーが Windows にログオンしていなければなりません。

単発のアクティビティを実行する必要がある場合やマシンのそばに人がいる状態で実行するときは Application 版をインストールすることをお勧めします。また、Application 版は自身の PC で CatTools をたまたま実行するユーザーにとって最適です。

3 デバイス

CatTools は広範なメーカーの様々な機種をサポートしています。

現在サポートしているデバイスおよびそれぞれのデバイスがサポートしているアクティビティの最新リストについては、弊社 Web サイトの[デバイス対応表](#)を参照してください。予め定義されているデバイスタイプを使って、デバイスを追加する方法については、第 6 章「Panels」の「Devices | Add」の項を参照してください。

- カスタムデバイスを作成する - 独自のカスタムデバイスのスクリプトを作成する方法とそれを CatTools のデバイスタイプリストに追加する方法について説明します。
- デバイス別固有情報 - 特定のデバイスに関する重要事項について記載しています。
- サポート対象外のデバイス/デバイスアクティビティ - 説明とリストへのリンクを記載しています。

3.1 カスタムデバイスを作成する

CatTools には、ご使用のデバイスが予め定義されているデバイスタイプに該当しない場合に独自のカスタムデバイスタイプを作成する機能とそのためのスクリプトファイルが付属しています。

必要な知識・経験等

CatTools にカスタムスクリプトを追加する作業をスムーズに進めるためには、ある程度の Visual Basic によるスクリプト作成に関する知識または経験があったほうが望ましいといえます。

しかし、CatTools のルートディレクトリの下にある¥Templates フォルダにあるサンプルコードのテンプレートファイルに記載されているコメントや同じフォルダ内にあるヘルプファイルを活用すれば、初心者レベルに近い技術しかなくても、ある程度のことはいえます。

概要

CatTools でサポートデバイスを追加するには、次の 2 つのファイルが必要です。

1)デバイスタイプファイル(.ini ファイル) - 次の項目について定義します。

- デバイスタイプ名
- デバイス ID
- デバイスの追加や編集を行う画面のデバイスフォームに表示される、ユーザーインターフェイスのフィールド値およびデフォルト値

2)デバイススクリプトファイル(.txt ファイル) -作成するデバイスタイプに適用するコードを記述します。このファイルがあって初めて、作成したタイプのデバイスに CatTools からログインする、違うモードでログイン/ログアウトする、異なるアクティビティ(たとえば、設定のバックアップを取る、CLI コマンドを送信する、設定を変更する等)を実行する、コマンドを解釈して得られたデータをレポートとして出力する等ができるようになります。

デバイススクリプトファイルには、CatTools の内部プログラムコードに含まれている変数に対する参照や関数の呼び出しも記述することができます。これらには「cl」という接頭辞が付けられています。これら cl が付いた関数や変数のリストは、本章のカスタムデバイススクリプトの開発についての項でも、その活用法について説明しています。

- カスタムデバイスの作成方法 - カスタムデバイスの作成方法について順を追って説明します。
- カスタムデバイスタイプファイル(.ini) - .ini ファイルの詳細とその作成方法について、実際に 1 つのカスタムデバイスタイプに対する .ini ファイルを作成しながら、その手順を詳しく説明します。
- カスタムデバイススクリプトファイル(.txt) - .txt ファイルの詳細とカスタムデバイススクリプトの作成方法について説明します。
- cl.変数および関数 - カスタムデバイススクリプトファイルで使用される CatTools の内部変数および関数について説明します。
- カスタムデバイスのテスト - カスタムデバイスをテストする簡単な方法について説明します。

3.1.1 カスタムデバイスの作成方法

CatTools でカスタムデバイスを作成するための簡易ガイド

ここでは CatTools でカスタムデバイスを作成する手順を説明します。CatTools の GUI にカスタムデバイスタイプが表示されるようになるまでの手順とカスタムデバイススクリプトファイルの作成方法についても、簡単に説明します。

CatTools にはカスタムデバイスのテンプレートファイルが付属しており、本書でも説明文中で参照するよう指示しています。このテンプレートは Cisco 製ルーター用として作成されています。ご使用のデバイスで正常にスクリプトが動作するようにするためには、いったん作成したデバイススクリプトファイルを何度も修正する必要があるかもしれません。

カスタムデバイスタイプファイル(.ini)、カスタムデバイススクリプトファイル(.txt)およびデバイススクリプトで CatTools の内部関数を使用する場合のサンプルコードについての詳細は、本章の関連ページを参照してください。

カスタムデバイスの作成手順

1) カスタムデバイスタイプファイル(.ini)を作成します。

CatTools のルートディレクトリの下の¥Templates フォルダ内にある Custom.Device.Template.ini というファイルをコピーし、¥Devices フォルダの下に保存します。このとき次の形式で新しいファイル名を付けます。

Custom.Manufacturer.Type (例 Custom.Cisco.FirewallPIX)

2) .ini ファイルを編集します。

メモ帳などのテキストファイルエディターを使ってステップ 1 で作成した.ini ファイルを開きます。.ini ファイルに書かれているテンプレートのデフォルト値を以下のように書き換えます。

```
[device]
name=Custom.Manufacturer.Type
id=4000
```

name を重複しない固有のデバイス名(例:Custom.Cisco.Router)に変更します。

注意: スペースを入れず、ピリオド(.)で区切ってください。

id を重複しない固有の番号(他のデバイスの.ini ファイルでも使用していない番号)に変更します。4000 ~ 4999 までの番号を使用してください。

```
[item_Name]
default=Unique device name
```

default を重複しない固有のデバイス名に変更します。

3) 保存し、再起動します。

.ini ファイルの編集が終わったら、¥Devices フォルダに保存してファイルを閉じます。CatTools を再起動すると、[Device Information]画面の[Device type]ドロップ

ブダウンリストに新しいカスタムデバイスタイプが表示されます。

4) カスタムデバイススクリプトファイル(.txt)を作成します。

CatTools のルートディレクトリの下の¥Templates フォルダ内にある Custom.Device.Template.txt というファイルをコピーし、¥Scripts フォルダの下に保存します。このとき、2)で[device]の name に入力した値と同じ名前を付けてファイルを保存してください。ファイルの拡張子は必ず.txt にしてください。

例:

(device .ini ファイルの設定が以下の場合)

```
[device]
name=Custom.Cisco.FirewallPIX
```

スクリプトファイルの名前は *Custom.Cisco.FirewallPIX.txt* にします。

5) .txt ファイルを編集します。

テキストファイルエディター*を使ってステップ 4)で作成した.txt ファイルを開きます。開いたファイルの「SCRIPT NOTES」の項に記載されている手順に従って、ご使用のデバイスに応じてスクリプトをカスタマイズしてください。

(*注意: .txt ファイルはメモ帳を使って開くこともできますが、構文がハイライト表示されるようなスクリプトエディターを使ったほうが .txt ファイルの編集作業が楽になります。)

6) 保存し、テストします。

.txt ファイルの編集が終わったら、¥Script フォルダに保存してファイルを閉じます。作成したカスタムデバイスをテストするには、そのデバイススクリプトでサポートするアクティビティを作成し、[Run now]ボタンをクリックして手動で実行してみます(Device.ConnectivityTest.Login が最初に行うアクティビティとして適しています。どのデバイスも複雑なアクティビティを実行するには、まずスムーズにログインできなければならないためです)。

[Info Log]ペインを表示してエラーが発生していないかどうか確認し、必要であれば再度.txt ファイルを編集します。

詳細については、「カスタムデバイスのテスト」の項を参照してください。

カスタムデバイスの作成方法の確認

- デバイスタイプファイル(.ini)を作成する。
- デバイススクリプトファイル(.txt)を作成する。

3.1.2 カスタムデバイスタイプファイル(.ini)

CatTools で使用されているデバイスタイプ

CatTools では、テキストファイル(.ini ファイル)を使ってデバイスタイプを定義し、その定義ファイルは CatTools のルートディレクトリの下にある¥Devices フォルダに保存します。[Device Information]画面の[Device type]ドロップダウンリストに表示される項目ごとに対応する.ini ファイルが存在します。

CatTools を起動すると、¥Devices フォルダにあるすべての .ini ファイルが検索され、内容が読み込まれます。検索されたデバイスタイプはすべて[Device type]ドロップダウンリストに表示され、選択してデバイスを定義することができるようになります。

新しいカスタムデバイスタイプを追加するには、新しい .ini ファイルを作成し、デバイスとその特徴を定義する必要があります。

.ini ファイルに記述する項目の概要

.ini ファイルの内容は[...]という形で括弧にくくって項目別に分けられています。主な項目について説明します。

[info]項目

これは、その .ini ファイルが CatTools のファイルであることを示す必須項目です。次のように記述します。

```
[info]
cookie=Kiwi CatTools
version=3
author=Kiwi Enterprises
```

[device]項目

CatTools のインターフェイス([Device type]ドロップダウンリストに表示される名称等)やスクリプト内で使用される名前を定義するもので、必須項目です。また、CatTools のデータベース内で使用するデバイスタイプの固有キー(ID)もここで定義します。次のように記述します。

```
[device]
name=Custom.Cisco.Router
id=4000
```

[device]項目には .ini ファイルの名前と似ている(あるいは同じ)名前を定義できますが、ここで定義する名前はユーザーインターフェイス内で使用されるデバイス名に反映され .ini ファイルの名前が変更されるわけではありません。

CatTools では予め定義されているデバイスタイプ用に 0 ~ 3999 までの ID が予約されています。作成するカスタムのデバイスタイプに設定できる番号は 4000 ~ 4999 までです。ID を定義する際は、使用しようとしている番号が重複しない固有の番号であることを確認してください。

[item]項目

[item_xxx]の各項目では、CatTools の[Device Information]画面のユーザーインターフェイスで使用される入力フィールドについて定義します。次のように記述します。

```
[item_group]
name=Group
default=Default
required=1
info="The logical group that this device belongs to."
```

CatTools のユーザーインターフェイスとして表示される入力フィールドごとに [item_xxx]という項目を作成し、個別に定義します。

name には、入力フィールドや選択フィールドの横に表示されるテキスト(またはラベル)を定義します。

default には、新しいデバイスをデータベースに追加する際に使用されるデフォルト値を定義します。

required には、そのフィールドに有効なデータを入力すること(予め定義されているリストから選択するかユーザー自身で値を入力するといった操作)が必須であるか必須でないかを指定します。必須でない場合は 0、必須である場合は 1 を指定します。

info には、このフィールドに関連する説明を定義します。ここで定義したテキストはフィールド内にカーソルを置いたときにユーザーインターフェイスのステータスバーに表示されます。

上記の[item_group]はユーザーインターフェイス内で使用されている標準的なテキストボックス入力フィールドの例です。

入力フィールドがリストボックスの場合は、該当する[item_xxx]項目に *list* を追

加してリストボックスに表示される選択項目を定義する必要があります(以下を参照してください)。

```
[item_model]
name=Model
default=Other
required=0
info="The device model number."
list=501,515,Other
```

list には、設定対象のリストボックス内に表示される各選択項目の文字列をコンマで区切って記述します。*default* には、新しいデバイスを追加する際に選択項目内のどの項目をデフォルト値として表示するかを指定します。

入力フィールドがチェックボックスである場合は、0 または 1 を指定します。1 以外の値はすべて 0 とみなされます。チェックボックスの記述例については以下を参照してください。この例では、チェックボックスのデフォルト値として 1 が指定されています(デフォルトでチェックが入っている状態が表示されます)。

```
[item_require_vty_login]
name=Initial login requires password
default=1
required=0
info="This device requires an initial password for access"
```

各フィールドのサイズ(入力可能なデータの長さ)は CatTools データベースの設計によって異なります。CatTools ユーザーインターフェイスを介して入力されるデータはすべて、設定対象のフィールドに対して長すぎる場合は適宜切り詰められて表示されます。

カスタムデバイスタイプファイル(.ini)のテンプレート

CatTools にはカスタムデバイスタイプを新しく作成する際に、作成元として活用できるテンプレートファイルが付属しています。テンプレートファイルの名前は *Custom.Device.Template.ini* で、¥Templates フォルダの下にあります。

このテンプレートファイルは、CatTools で予め定義されているデバイスタイプの一部として付属しているものです。CatTools のすべての定義済みデバイスのデバイスタイプファイルは、CatTools 製品の新バージョンが発行される際に随時(必

要であれば)更新されます。

そのため、新しくカスタムデバイスタイプを作成する目的でテンプレートファイル自体を変更することは絶対にしないでください。テンプレートを元に新しいカスタムデバイスタイプを作成する必要がある場合は、テンプレートファイルをコピーして¥Devices フォルダに別の名前を付けて保存するようにしてください。

カスタムデバイスタイプファイル(.ini)の作成

CatTools にカスタムデバイスタイプを新しく追加するとき、最初にするのは ¥Templates フォルダの下にあるテンプレートファイル Custom.Device.Template.ini をコピーして ¥Devices フォルダの下に別の名前を付けて保存することです。

このファイルを作成元として新しいデバイスタイプファイルを定義します。ファイル名を付ける際には、なるべく次の命名規則に従ってください。

Custom.Manufacturer.Type

Custom Custom というテキストは CatTools の定義済みデバイスタイプと識別するために使用します。ここで指定した名前は [Device type] ドロップダウンリスト内にグループ化されて表示されません。

Manufacture デバイスのメーカーまたはサプライヤー名 (Cisco, Nortel, 3Com, Juniper, Foundry, Enterasys 等) を指定します。

Type デバイスのタイプ (router, switch, firewall 等) を指定します。*

*ここにデバイスのモデル番号を指定することは避けてください。メーカーと種類が同じ別モデルに再利用できるようにするためです。たとえば、PIX のモデル 501 と Cisco PIX のモデル 515 は *Custom.Cisco.Firewall*、*Custom.Cisco.PIX* または *Custom.Cisco.FirewallPIX* というスクリプトを使用できます。同種のデバイスでモデルが異なる場合はそのデバイスの .ini ファイルの [item_Model] 項目で指定するか、実行時に [Model] ドロップダウンリストに表示される項目として対象モデルを追加できます。

注意: CatTools の [Model] フィールドに表示される定義済みのデバイスタイプにどんな値が設定されていても、関連付けられているデバイススクリプトファイル (.txt) の動作に影響しません。

この命名規則に従うことをお勧めしますが、カスタムデバイスのスクリプトファイルに対する考え方は、アプリケーションの制限内であればユーザーにできる

限り自由に設計してもらうことを念頭に開発されています。したがって、モデル固有の動作をカスタムデバイススクリプト内で指定する必要がある場合は、それに応じてデバイススクリプトファイルを変更してください。

.ini ファイルの編集

次に、カスタムデバイスの.ini ファイルを開き、必要であれば各項目の値を変更します。

下記のコード例は .ini ファイルの例ですが、いくつかの項目の値を別色で表示しています。

最も大事な項目の値で変更しなければならないものは**赤で表示**しています。これらの項目には**重複しない固有の値**を指定する必要があります。固有の値を設定しなかった場合、作成したデバイスタイプが CatTools の[Device Type]ドロップダウンリストに表示されません。

必要時に変更する可能性のある項目の値は**緑で表示**しています。

コメントは**青で表示**しています(注意：実際の .ini ファイルではコメントは削除してください)。

黒文字で表示している部分はすべて元の設定のままにしておくことをお勧めします。ただし、*name* や *info* など一部の項目についてはご使用のデバイスに関する深い値に変更しても構いません。

name に指定するテキスト値はフォーム内に表示されるフィールドラベルとなります。*info* に指定するテキスト値はフォーム内のフィールドを選択したとき(つまり、カーソルを置いたとき)にステータスバーに表示されるテキストとなります。

.ini ファイルの変更が終わったら、必ず¥Devices フォルダの下に保存してください。新しいカスタムデバイスタイプが[Device type]ドロップダウンリストに表示されるようにするためには、CatTools を再起動する必要があります。

```
[info]
```

```
cookie=Kiwi CatTools
```

```
version=3
```

```
author=Kiwi Enterprises 自分の名前を入力するかデフォルトのままにしておいてください。
```

```
[device]
```

```
name=Custom.Manufacturer.Type 重複しない固有の名前に変更してください
```

(例 : Custom.Cisco.Router)。注意:区切りには空白を入れず、ピリオド(.)を入力してください。

id=4000 4000 ~ 49999 までの番号を入力してください。ここで入力する番号は重複しない固有の番号でなければなりません。

Device info

[item_Group]

name=Group

default=Default デバイスのデフォルトのグループ名を入力します。このままでも構いません。

required=1

info="The logical group that this device belongs to."

[item_Name]

name=Name

default=Unique device name 重複しない固有のデバイス名を入力します。

required=1

info="A unique name for this device. e.g. sales-router or head-office-3500."

[item_HostAddress]

name=Host Address

default=127.0.0.1

required=1

info="IP address or host name of the device."

[item_Filename]

name=File Name

default=

required=1

info="The base file name to use for this device (unique)."

[item_Model]

name=Model

default=Custom 下記の list 項目の中からデフォルトで表示するモデルを指定します。このままでも構いません。

required=0

info="The device model number."

list=Custom モデル番号を追加する場合はコンマで区切ってください。例：
Custom,501,515

```
[item_ConnectVia]
name=Connect via
default=Direct connect
required=1
info="The name of another device to connect to first."
```

```
[item_Telnet]
name=Method
default=Telnet 下記の list 項目の中からデフォルトで表示する通信方法を指定し
ます(SSH のみに対応しているデバイスの場合、適宜デフォルト値を変更して
ください)。
required=1
list=Telnet,SSH1,SSH2,SSH2-nopty,SSH1-DES,SSH1-3DES,SSH1-Blowfish,Cis
co SSH リストは適宜変更してください。
info="Connection method to use."
```

```
[item_TelnetPort]
name=Port
default=23 下記の list 項目の中からデフォルトで表示するポートを指定します。
ユーザーインターフェイスの中で[Method]が変更されると、[Port]フィールドの
値は自動的に変更されます。
required=1
list=23,22
info="Port number to use."
```

Passwords

```
[item_VTYPass]
name=VTY Password
default=
required=0
info="VTY password."
```

```
[item_EnablePass]
name=Enable Password
```

default=
required=0
info="Enable or privilege password."

[item_PrivilegeLevel]
name=Privilege Level
default=
required=0
info="Sets the enable mode privilege level. (Not required in most cases)"

[item_ConsolePass]
name=Console Password
default=
required=0
info="The console (com port connection) password."

[item_AAAUsername]
name=Username
default=
required=0
info="AAA/TACACS/RADIUS/Local username."

[item_AAAPassword]
name=Password
default=
required=0
info="AAA/TACACS/RADIUS/Local password."

[item_SNMPRead]
name=SNMP Read
default=
required=0
info="SNMP Read community name."

[item_SNMPWrite]
name=SNMP Write
default=
required=0

info="SNMP Write community name."

[item_RequireVTYLogin]

name=Initial login requires password

default=1

required=0

info="This device requires an initial password for access"

[item_LoginUsesAAA]

name=Initial login requires username/password

default=0

required=0

info="The initial access requires a username and password"

[item_EnableUsesAAA]

name=Enable mode requires username/password

default=0

required=0

info="Enable mode access requires a username and password"

Prompts

[item_VTYPrompt]

name=VTY Prompt

info="Expected VTY prompt from the device. (Only required if non standard prompt is used)"

[item_EnablePrompt]

name=Enable Prompt

info="Expected enable mode prompt from the device. (Only required if non standard prompt is used)"

[item_ConsolePrompt]

name=Console Prompt

info="Expected console prompt from the device. (Only required if non standard prompt is used)"

[item_AAAPrompt]

name=Username prompt

info="Expected Username prompt from the device or AAA server. (Only required if non standard prompt is used)"

[item_AAAPassPrompt]

name=Password prompt

info="Expected AAA Password prompt from the device or AAA server. (Only required if non standard prompt is used)"

Contact info

[item_Address1]

name=Address1

default=

required=0

info="Location of the device"

[item_Address2]

name=Address2

default=

required=0

info="Location of the device"

[item_Address3]

name=Address3

default=

required=0

info="Location of the device"

[item_ContactName]

name=Contact Name

default=

required=0

info="The name of the person responsible for this device."

[item_ContactPhone]

name=Contact Phone

default=

required=0

info="How to contact the person responsible for this device"

[item_ContactEmail]
name=Contact E-mail
default=
required=0
info="How to contact the person responsible for this device"

[item_ContactOther]
name=Contact Other
default=
required=0
info="Any additional contact info"

[item_AlertEmail]
name=Alert e-mail
default=
required=0
info="Who to notify by e-mail of any alarms or alerts for this device"

[item_SerialNumber]
name=Serial number
default=
required=0
info="The serial number of this device"

[item_AssetTag]
name=Asset Tag
default=
required=0
info="Asset tag information"

[item_Identification]
name=Identification
default=
required=0
info="Identification info for this device"

[item_SerialOther]
name=Other info

```
default=
required=0
info="Any other serial number information"

[item_ActivitySpecific1]
name=Activity Specific1
default=
required=0
info="Information specific to a particular activity"

[item_ActivitySpecific2]
name=Activity Specific2
default=
required=0
info="Information specific to a particular activity"
```

3.1.3 カスタムデバイススクリプトファイル(.txt)

CatTools のデバイススクリプト

CatTools のデバイススクリプトはテキストファイル(.txt ファイル)を使って定義し、CatTools のルートディレクトリの下にある`Scripts` フォルダに保存します。

アクティビティが実行される際、CatTools はデバイスタイプ.ini ファイルの [device]項目の name に指定されている値を読み込み、どのデバイススクリプトファイル(.txt)を使用すべきかを判断します。したがって、それぞれの.txt ファイルは、対応する.ini ファイルの name に指定されている名前と同じファイル名が付けられていなければなりません(.ini ファイルと.txt ファイルの名前を、拡張子以外まったく同じにすることで混乱を避けることができます)。

CatTools の定義済みデバイスタイプにはすべてスクリプトファイル(.txt)が関連付けられており、それらのファイルは暗号化されています。暗号化されている理由は2つあります。まず、弊社の知的財産権を保護することです。もう1つは、実行時にスクリプトが動作しないという事態を招くおそれのある無断変更を防ぐためです。

カスタムデバイススクリプトファイル(.txt)のテンプレート

CatTools にはカスタムデバイススクリプトを新しく作成する際に、作成元として活用できるテンプレートファイルが付属しています(このテンプレートは Cisco

製ルーターをサンプルに記述されています)。テンプレートファイルの名前は *Custom.Device.Template.txt* で、CatTools のルートディレクトリの下 の ¥Templates フォルダの下にあります。

このテンプレートファイルは、CatTools で予め定義されているデバイスタイプの一部として付属しているものですが、暗号化はされていません。CatTools のすべての定義済みデバイスのデバイススクリプトファイルは、CatTools 製品の新バージョンが発行される際に随時(必要であれば)更新されます。このため、新しくカスタムデバイススクリプトを作成する目的でテンプレートファイル自体を変更することは絶対にしないでください。テンプレートを元に新しいカスタムデバイススクリプトを作成する必要がある場合は、テンプレートファイルをコピーして ¥Scripts フォルダに別の名前を付けて保存するようにしてください。

カスタムデバイススクリプトファイル(.txt)の作成

CatTools にカスタムデバイススクリプトを新しく追加するとき、最初にするこ とは ¥Templates フォルダの下にあるテンプレートファイル *Custom.Device.Template.txt* をコピーして ¥Scripts フォルダの下に別の名前を付けて保存することです。このとき、デバイスタイプ.ini ファイルの [device] 項目で name に指定したものと 同じ名前で、.txt という接尾辞を付けて保存してください。このファイルを作成元として新しいデバイススクリプトファイルを定義し ます。

例:

Cisco ASA Firewall デバイス用のカスタムデバイススクリプトを新規作成すると します。準備段階として *Custom.Cisco.FirewallASA.ini* という.ini ファイルを作 成し、[device]項目の name に *Custom.Cisco.FirewallASA* と指定します。次に、 *Custom.Cisco.FirewallASA.txt* という名前を付けたカスタムスクリプトファイル を用意します。

.txt ファイルの編集

次に、カスタムデバイス.txt ファイルを開き、必要であればご使用のデバイスに 合わせてスクリプトが動作するよう、各項目の値を変更します。

.txt ファイルには、変更する際の手順やヒントなど多くのコメントが書かれてい ますが、注意しなければならない重要な項目はいくつかしかなく、それらはスクリプトファイルの冒頭に書かれています。

共通事項:

スクリプトの 1 行目に次のように記述されていることがあります。

```
Attribute VB_Name = "Dev_CustomDeviceTemplate"
```

これは Visual Basic(あるいは同等のスクリプト言語)開発環境で表示したときに、そのデバイススクリプトファイルを示す Visual Basic モジュール名となります。

この行は VB 開発環境でファイルの編集を行っているときは表示されませんが、他のエディターでは表示される場合があります。

"Dev_CustomDeviceTemplate" はカスタムテンプレートファイルのモジュールに付けられた名前です。カスタムデバイスファイルがたくさんある場合は、この名前を作成したデバイススクリプトのファイル名を反映した名前に変更するほうが良いこともあります。そのようにすることによって、複数のカスタムデバイススクリプトファイルをご使用の VB 開発環境で同時に(たとえば、1 つのプロジェクト内などで)開くことができるようになります。

スクリプトの編集にテキストベースのエディター(メモ帳など)を使用している場合は、モジュール名の変更は任意で構いません。しかし、混乱を避けるには、この名前を変更しておいたほうが良いでしょう。

```
Option Explicit
```

この文はスクリプティングコードの一定基準を満たすことを宣言しています。したがって、**Dim** 文を使ってすべての変数を明確に宣言する必要があります。宣言を変更するときは **ReDim** 文を使います。未宣言の変数名を使おうとすると、コンパイル時にエラーが発生します。

注意: デバイススクリプトファイルは Visual Basic スクリプト言語(VB スクリプト)を使用しているため、「types」という変数は宣言されていません。

定数宣言:

Option Explicit 文の下にはすべてのスクリプト定数の宣言文を記述します。

```
Private Const SCRIPT_NAME = "Device Template"
```

SCRIPT_NAME 変数には参照するスクリプトファイルを指定します。この参照はスクリプトが呼び出されると必ず[Info Log]ペインおよび Infolog.txt ファイル

内に表示されます。

```
Private Const DEVICE_
```

DEVICE_変数はデバイスの構成やデバイスからの応答を定義します。たとえば、デバイスからユーザー名を入力するよう求めるプロンプトを受け取ると必ず、「Username:」と表示されるようにしたい場合、次のように指定します。

```
DEVICE_USERNAMEPROMPT = "Username:"
```

```
Private Const COMMAND_
```

COMMAND_変数は、アクティビティの実行時に特定のタスクまたはアクションを実行するために CatTools からデバイスに対して送られるコマンドを定義します。たとえば、CatTools からデバイスに対して enable mode にするよう求める enable コマンドを送信させたい場合、次のように指定します。

```
COMMAND_ENTERENABLEMODE = "enable"と指定します。
```

メモ:

```
--- SCRIPT NOTES ---
```

SCRIPT NOTES には、有用な情報とスクリプト作成の初期段階において実行すべき作業について記載されています。ここに記載されている内容は必要に応じて適宜書き換えて構いません。たとえば、スクリプト設定の初期に必要な作業を終え、もう再びその作業をする必要がない場合、SCRIPT NOTES に記載されている、それらに関するテキストを全部削除することができます。複雑なルーチンやスクリプト固有の振る舞いなど、デバイススクリプトの管理担当者にとって役立つ情報はここに記載しておいてください。このような情報は完全なものである必要があり、全文をここに記載するか、完全な情報が記載されている関連項目を参照先として記載してください。

```
--- DEVICE NOTES ---
```

DEVICE NOTES には、デバイス固有の振る舞いや特殊な動作など、デバイススクリプトの管理担当者にとって役立つ情報を記載します。

例 : *"The device you are creating the custom script for only supports SSH2."*

必須関数:

CatTools がデバイスにアクセスしアクティビティを実行するためには、デバイススクリプト内で指定しなければならない必須関数がいくつかあります。以下にそ

の例を挙げます。

Login	デバイスに対する認証要求
SendPostLoginCommands	デバイスにログインできた後に送信されるコマンド
EnterEnableMode	優先モードに変更
SendPostEnterEnableModeCommands	有効モードに変更後の環境を設定するコマンド
SendSessionTerminationCommands	デバイスを終了またはログアウトするために必要なコマンドを送信

SendPost...関数には、必ずしもすべてのコマンドを指定する必要はありませんが、関数が記述されており TRUE の値が返ってこなければなりません。

同様に、ご使用のデバイスに Enable モードという概念がなくても、デバイススクリプトでは EnterEnableMode 関数を指定する必要があり、戻り値として TRUE を返すコードを示す行が 1 行指定されている必要があります。

サポートされていないアクティビティ:

デバイスのテンプレートファイルには、CatTools を搭載したクライアント内部で動作する関数 cl.CatToolsNoSupport を呼び出す関数が多く記述されています。アクティビティがサポートされていないことを示すメッセージを[Info Log]に表示するには、cl.CatToolsNoSupport を呼び出す必要があります。

大抵のアクティビティで cl.CatToolsNoSupport 関数の呼び出しに必要なコードは 1 行で記述できます。

```
Function Activity_UpdatePassword()  
    Call cl.CatToolsNoSupport  
End Function
```

CatTools のメインアクティビティのスクリプトに戻り値を返すために 2 行目を記述する必要がある場合もあります。

```
Function Activity_TFTPUpload(iActionType, sTFTPServer,  
sTFTPFile, sResponseFile, bDoSaveToNVRAM)  
    Call cl.CatToolsNoSupport  
    Activity_TFTPUpload = "Not supported"
```

End Function

カスタムデバイススクリプトでこれらのアクティビティのいずれかに対するサポートを追加したい場合は、`cl.CatToolsNoSupport` 関数の呼び出しを新しいコード行で置き換え、必要な場合は戻り値も変更する必要があります。

スクリプトファイルの保存

作成したスクリプトファイルに CatTools がアクセスできるようにするには、スクリプトファイルを¥Scripts フォルダに保存する必要があります。通常、スクリプトファイルのコードを変更した際に、その変更部分を読み込むために CatTools を停止させ再起動する必要はないのですが、テスト時に変更部分が認識されていないため余儀なく再起動しなければならないことも少なくありません。

カスタムデバイススクリプトのテスト

スクリプトの設定が終わったら、CatTools で新しいデバイスに対するアクティビティを設定し正常に動作するかどうかをテストしてください。

詳細については、「カスタムデバイスのテスト」の項を参照してください。

有効なスクリプトのご提供のお願い

ご使用のカスタムデバイスを CatTools でサポート可能にできた場合、作成したカスタムスクリプトファイル(.txt)とカスタムデバイスタイプファイル(.ini)を弊社 Web サイトの[製品サポートページにあるフォーム](#)を使用して弊社まで添付ファイルで送信していただくと、今後、製品付属の定義済みデバイスタイプとして扱わせていただくことを検討いたします。

提供していただいたデバイスに対する需要、スクリプトの複雑性、利用可能な技術情報源など検討項目は多岐にわたるため、そのデバイスを CatTools の定義済みデバイスタイプとして追加するかどうか、追加する場合はその時期に関しては何の保証もいたしかねます。しかし、送信していただいたスクリプトはすべて将来の参考用として分類・保管させていただきます。

3.1.4 カスタムデバイススクリプトの cl.変数および関数

カスタムデバイステンプレートのスクリプトには、CatTools クライアントの内部変数に対する参照やクライアント関数の呼び出しが多く記述されています。これらの変数および関数には「cl.」という接頭辞が付いていますので、識別できます。

関数および変数はテンプレートスクリプト自体には(主にスクリプトファイルのサイズを抑制するために)記録されていませんが、それぞれの関数や変数、サブルーチンをどのような目的で使用するのか、パラメータを必要とする関数は何か、戻り値はどのようなものがあるのかについては、カスタムデバイステンプレートに記述されている各関数および変数の詳細を以下にまとめましたので参照してください。

変数

デバイスのコンフィグレーション変数([Device Information]画面に表示される各タブから読み込まれます。)

[Device Info]タブ

cl.CurDevName	デバイスを参照するためにユーザーが割り当てた固有の名前([Name]フィールド)
cl.CurDevTelnet	Telnet、SSH などデバイスへの通信手段([Method]フィールド)

[Passwords]タブ

cl.CurDevVTYPass	デバイスの VTY パスワード ([VTY Password]フィールド)
cl.CurDevEnablePass	デバイスを有効化するためのパスワード ([Enable Password]フィールド)
cl.CurDevPrivilegeLevel	デバイスの有効モード優先レベル ([Privilege Level]フィールド)
cl.CurDevAAAUsername	デバイスのユーザー名 ([Username]フィールド)
cl.CurDevAAAPassword	デバイスのパスワード ([Password]フィールド)
cl.CurDevRequireVTYLogin	ログイン時にパスワードのみ入力するデバイス ([Initial login requires password]チェックボックス)
cl.CurDevLoginUsesAAA	ログイン時にユーザー名とパスワードの両方を入力するデバイス ([Initial login requires username/password]チェックボックス)
cl.CurDevEnableUsesAAA	有効モードにするのにユーザー名とパスワードを入力するデバイス ([Enable mode requires username/password]チェックボックス)

[Prompts]タブ

cl.CurDevVTYPrompt	デバイスに対してユーザーが割り当てた VTY カスタムプロンプト([VTY Prompt]フィールド)
cl.CurDevEnablePrompt	デバイスに対してユーザーが割り当てた有効モードのカスタムプロンプト([Enable Prompt]フィールド)
cl.CurDevAAAUserPrompt	デバイスに対してユーザーが割り当てたデバイス AAA のユーザー名のカスタムプロンプト([Username prompt]フィールド)
cl.CurDevAAAPassPrompt	デバイスに対してユーザーが割り当てたデバイス AAA のパスワードのカスタムプロンプト([Password prompt]フィールド)

その他の変数

cl.RxBuffer	デバイスから送られてきた応答データの文字列
cl.ScheduleNumber	現在のスケジュール番号
cl.DeviceHostnameID	ログイン成功後に取得されたデバイスのホスト名
cl.DeviceVTYPrompt	ホスト名。DEVICE_STANDARDPROMPT で終結する。
cl.DeviceEnablePrompt	ホスト名。DEVICE_PRIVILEGEDPROMPT で終結する。
cl.DeviceConfigPrompt	ホスト名。DEVICE_CONFIGPROMPT で終結する。

代表的な関数

一般およびファイル関連

cl.Initialise	cl.変数を初期化(またはデフォルト値に設定)します。
cl.Delay	CatTools を一定時間強制的に停止させます。
cl.GetUniqueDeviceFileName	デバイスに基づいて¥ClientTemp フォルダ内に固有の名前のファイルを生成します。
cl.DBMetaCmd	コマンドがデータベースのメタコマンドであるかどうかチェックし、処理します。
cl.UtilityMetaCmd	コマンドがユーティリティのメタコマンドであるかどうかチェックし、処理します。

cl.Log	Infolog.txt ファイルにテキスト行を送信します。 この情報は[Info Log]タブに表示されます。
cl.LogToFile	データをファイルに書き出します。
アクティビティ関連	
cl.CatToolsNoSupport	デバイススクリプトでサポートされていないアクティビティを呼び出します。
cl.DBCheckScheduleOption	指定したアクティビティ内で特定のオプションが選択されているかどうかを確認します。
デバイス関連	
cl.FlushRxBuffer	cl.RxBuffer 文字列を消去します。次のデータを受信する前に前の応答データをクリアしたいときに使用します。
cl.DetermineHostname	デバイスのホスト名を設定し、cl.変数のプロンプトを表示します。
cl.SendData	テキストをデバイスに送信します。
cl.SendAndWaitForEcho	テキストをデバイスに送信し、応答を待ちます。
cl.SendAndWaitForPrompt	テキストをデバイスに送信し、応答を待ち、デバイスにプロンプトを表示します。
cl.WaitForData	デバイスから特定のデータ文字列が返されるのを待ちます。
cl.WaitForMultData	デバイスから特定のデータ文字列に含まれる文字を1つでも含むものが返されるのを待ちます。
テキスト操作関連	
cl.ReplaceText	指定したデータの文字列内のサブストリングを新しいサブストリングで置き換えます。
cl.TextRemoveBlankHeaderLines	文字列の冒頭にある空白のヘッダー行(デバイスの出力バッファ等)を削除します。
cl.TextRemoveLinesContainingText	特定のテキストのサブストリングを含む行を削除します。
cl.TextInText	ある文字列のサブストリングの開始点を返します。
cl.TextRemoveTextUpTo	ある文字列または指定したサブストリング

などの文字数を超えるテキストを切り詰めます。

関数およびサブルーチンの詳細およびコード例

ここでは、上記の関数について入力パラメータやコード例なども織り交ぜながらさらに詳しく説明します。

`cl.Initialise()`

この関数は `cl` 変数を次のように初期化(またはデフォルト値に設定)します。

```
cl.DeviceHostnameID = ""
cl.DeviceVTYPrompt = ""
cl.DeviceEnablePrompt = ""
cl.DeviceConfigPrompt = ""
```

```
cl.WaitForEcho = True
```

```
cl.WaitForTime = 0
```

`cl.Delay(IDuration)`

この関数は CatTools を一定時間強制的に停止させます。

入力パラメータは 1 つのみです。

`IDuration` 停止させる時間を秒単位で指定します。

`cl.GetUniqueDeviceFileName(sFolderName, sPrefixName) As String`

この関数はデバイスに基づいて¥ClientTemp フォルダ内に固有の名前のファイルを作成します。パスまたは固有のファイル名の文字列が戻されます。

値で受け渡される入力パラメータを 2 つ、オプションのパラメータを 2 つ指定できます。

`sFolderName` ファイル名の冒頭に付け加える文字列値を指定します。
空白の場合""と指定します。

`sPrefixName` ファイル名の `sFolderName` の後に追加する文字列値を指定します。

例: ¥ClientTemp フォルダ内の一時ファイルに出力結果を書き込む。

```
'create the unique filename using only the prefix 'CLI'
sClientResultsFile = cl.GetUniqueDeviceFileName("", "CLI")
```

'write the data in the RxBuffer to a file in %ClientTemp folder using the filename just created.

```
cl.LogToFile sClientResultsFile, cl.RxBuffer
```

cl.DBMetaCmd(sCmd) As Long

この関数は、コマンドがデータベースメタコマンドであるかどうかを確認します。データベースメタコマンドである場合は、CatTools がそのコマンドを実行します。データベースメタコマンドを使うと、直接 CatTools データベースのテーブル内のフィールドをアクティビティの一部として更新することができます。デバイス上では実行されません。データベースメタコマンドの構文は次のとおりです。

%ctDB:tablename:fieldname:value

入力パラメータは 1 つです。

sCmd 評価されるコマンドの文字列値を指定します。

戻り値:

- 0 コマンドがデータベースメタコマンドでないことを示しています。
- 1 データベースメタコマンドであると評価され、実行されたことを示しています(同時にデバッグメッセージが[Info Log]タブ内に表示されます)。
- 1 データベースメタコマンドであると評価されたが、実行されなかったことを示しています(同時にエラーメッセージが[Info Log]タブ内に表示されます)。

例: 送信するコマンドがデータベースメタコマンドであるかどうか確認し、(デバイスコマンドなど)データベースメタコマンドでない場合、デバイスにそのコマンドを転送する。

```
lRetVal = cl.DBMetaCmd(sCmd)
If lRetVal = 0 Then 'not a database meta command so send
    it cl.FlushRxBuffer
    ' send command to device and wait for echo
    bRetVal = cl.SendAndWaitForEcho(sCmd)
    If bRetVal then
        ' command has been echoed, so lets execute it
        cl.SendData vbCr
    End if
End If
```

cl.UtilityMetaCmd(sCmd) As Long

この関数は、コマンドがユーティリティメタコマンドであるかどうかを確認します。ユーティリティメタコマンドである場合は、CatTools がそのコマンドを実行します。ユーティリティメタコマンドを使うと、Device.CLI.Send コマンドや Device.CLI.Modify Config アクティビティ内で使用する様々なオプションを設定することができます。デバイス上では実行されません。ユーティリティメタコマンドは CatTools の一部の内部機能のデフォルトの振る舞いを変更します。

入力パラメータは1つです。

sCmd 評価されるコマンドの文字列値を指定します。

戻り値:

- 0 コマンドがユーティリティメタコマンドでないことを示しています。
- 1 ユーティリティメタコマンドであると評価され、実行されたことを示しています(同時にデバッグメッセージが[Info Log]タブ内に表示されます)。
- 1 ユーティリティメタコマンドであると評価されたが、実行されなかったことを示しています(同時にエラーメッセージが[Info Log]タブ内に表示されます)。

例: 送信するコマンドがユーティリティメタコマンドであるかどうか確認し、ユーティリティメタコマンドでない場合、デバイスにそのコマンドを転送する。

```
lRetVal = cl.UtilityMetaCmd(sCmd)
If lRetVal = 0 Then 'not a utility meta command so send it
    cl.FlushRxBuffer
    ' send command to device and wait for echo
    bRetVal = cl.SendAndWaitForEcho(sCmd)
    If bRetVal then
        ' command has been echoed, so lets execute it
        cl.SendData vbCr
    End if
End If
```

cl.Log(iPriority, sMessage)

この関数は Infolog.txt ファイルにテキスト行を送信します。この情報は[Info Log] ペインに表示されます。

入力パラメータは2つです。

iPriority	1~4の整数で送信される行のロギングレベルを指定します。1=エラー、2=警告、3=情報、4=デバッグ
sMessage	送信されるテキストメッセージを指定します。

例: レベル4(デバッグ)のメッセージ"Login Device Template: Custom Device 1"というテキストを出力し、送信する (デバイステンプレートスクリプトの Const SCRIPT_NAME および[Device Information]画面の[Device Info]タブに表示されている[Name]フィールドに「Custom Device 1」と入力してデバイスの名前を付けたものとする)

```
cl.Log 4, "Login " & SCRIPT_NAME & ": " & cl.CurDevName
```

cl.LogToFile(sFilename, sData, bAppend)

この関数はデータをファイルに書き込みます。

入力パラメータは3つです。

sFilename	書き込むファイルの名前とパスを文字列で指定します。
sData	書き込むデータを文字列で指定します。
bAppend	ブール値で指定します(デフォルト値または指定されていない場合は False です)。True の場合ファイルに文字列が追記され、False の場合ファイルが上書きされます。

例: 現在受信したデバイスデバイスからの応答をCドライブのtemp.txtファイルに書き込む。

```
cl.LogToFile "c:¥temp.txt", cl.RxBuffer, 1
```

cl.CatToolsNoSupport()

この関数は、現在デバイススクリプトでサポートされていないアクティビティを呼び出します。

cl.DBCheckScheduleOption(IScheduleNumber, IOptionNumber) As Long

この関数は、指定したアクティビティで特定のオプションが選択されているかどうかを確認します。

入力パラメータは2つです。

IScheduleNumber 現在のスケジュール番号を Long で指定します。
IOptionNumber チェックするアクティビティ内のオプション項目を Long で指定します。

例: アクティビティ(ここでは Device.CLI.Send command アクティビティ)によりコマンドの出力結果が E メール送信されるかどうか確認する。

```
lRetVal = cl.DBCheckScheduleOption(cl.ScheduleNumber, 8)
If lRetVal = 1 then
    'code to email commands goes here...
End if
```

cl.FlushRxBuffer()

この関数は cl.RxBuffer 文字列を削除します。通常は、次の応答データを受信する前に受信済みのデバイス応答データをすべて削除する際に使用します。

cl.DetermineHostname(Optional ByVal vStandardPrompt As Variant, Optional ByVal vPrivilegedPrompt As Variant, Optional ByVal vConfigPrompt As Variant) As Boolean

この関数は下記の cl.変数を設定し、確立します。成功した場合には True の値が返されます。

cl.DeviceHostnameID 以下のプロンプトで「シード」として使われるデバイスのホスト名です。例: DevHost123
cl.DeviceVTYPrompt ホスト名を指定し、末尾に DEVICE_STANDARDPROMPT を付けます。
例: DevHost123>
cl.DeviceEnablePrompt ホスト名を指定し、末尾に DEVICE_PRIVILEGEDPROMPT を付けます。
例: DevHost123#
cl.DeviceConfigPrompt ホスト名を指定し、末尾に DEVICE_CONFIGPROMPT を付けます。
例: DevHost123(

値によって渡されるオプションの入力パラメータは 3 つあります。

vStandardPrompt デバイスの標準モードのプロンプト(または VTY プロンプト)を指定します。指定されていない場合はデフォルトで「>」が使用されます。

<i>vPrivilegedPrompt</i>	デバイスの優先モードのプロンプト(または Enable プロンプト)を指定します。指定されていない場合はデフォルトで「#」が使用されます。
<i>vConfigPrompt</i>	デバイスのコンフィグレーションモードのプロンプト(または VTY プロンプト)を指定します。指定されていない場合はデフォルトで「(」が使用されます。

`cl.SendData(sDataToSend)`

この関数は指定したテキストをデバイスに送信します。

入力パラメータは 1 つです。

<i>sDataToSend</i>	デバイスに送信されるデータの文字列を指定します。
--------------------	--------------------------

例: デバイスにキャリッジリターン(CR)を送信する。

```
cl.SendData vbCr
```

`cl.SendAndWaitForEcho(sDataToSend) As Boolean`

この関数はデバイスにテキストを送信し、エコーを待ちます。テキストのエコーを待つことによって送信したい文字列が実際に送信されたことを確認します。(文字列を送信してもデバイスからそっくりそのままエコーバックされてこない場合、すなわち置換されたり壊れたりしている場合に備えることが目的です)。戻り値はブール値です。期待通りの文字列がエコーされてきた場合は True、されなかった場合は False が戻されます。

入力パラメータは 1 つです。

<i>sDataToSend</i>	デバイスに送信し、エコーバックされることを期待するデータ文字列を指定します。
--------------------	--

この関数は末尾に「`cl.SendData vbCr`」を付けることができます。これを付けることによってエコーされた文字列がデバイス上で実行されます。

例: デバイスに実行中のコンフィグレーションを表示するコマンドを送信し、エコーを待つ。エコーされてきたらコマンドを実行する。

```
bRetVal = cl.SendAndWaitForEcho("show running-config")  
If bRetVal then  
    cl.SendData vbCr  
End if
```

cl.SendAndWaitForPrompt(sDataToSend) As Boolean

この関数はテキストをデバイスに送信し、エコーを待ちます。エコーされてきた場合、その文字列が実行され、関数は有効なデバイスプロンプト(標準プロンプトまたは優先プロンプト)が戻されるまで待ちます。

通常、この関数が呼び出されるのは、出力ページングを止める'term len 0'などの既知の有効コマンドがデバイス上で実行されたが、出力結果が得られないときです。戻り値はブール値で返されます。文字列がエコーバックされ、コマンド実行後に得られたプロンプトが期待通りのものであった場合は True、そうでない場合は False が返されます。

入力パラメータは1つです。

sDataToSend 送信、エコー、デバイス上で実行されるデータの文字列を指定します。

cl.WaitForData(sData, ITimeout) As Boolean

この関数は、特定の文字列データがデバイスから戻ってくるのを、指定した時間の間待ちます。戻り値はブール値で返されます。指定したタイムアウト時間までに文字列が見つかった場合は True、見つからなかった場合は False が返されます。

入力パラメータは2つです。

sData デバイスから戻ってくるのを待つデータ文字列を指定します。

ITimeout 待つ時間を秒単位で指定します。

例: 30 秒以内にコンフィグモードを終了して Enable モードのプロンプトに戻ったかどうかを確認する。

```
cl.SendData Chr(26) ' i.e. CTRL-Z
If cl.WaitForData('HostName#', 30) = False Then
    cl.Log 4, "Failed to exit Configure Terminal mode"
End if
```

cl.WaitForMultData(rgMult, Optional iChoices = 0, Optional ITimeout = 0) As Long

この関数は、指定した範囲に含まれる文字列データアイテムが、デバイスに指定した時間内に戻ってくるのを待ちます。戻り値は最初に見つかった文字列データ

のアイテムを表すロング値で返されます。指定したタイムアウト時間までにアイテムが見つからなかったときは0が返されます。

入力パラメータは3つです。

<i>rgMult</i>	デバイスから戻されるのを期待している文字列データアイテムの範囲を指定します。
<i>iChoices</i>	(任意指定)検索するアイテムの数を指定します。 rgMult(0)の場合は指定しません。指定されていない場合は、関数自体が算出するため指定しなくても構いません。
<i>lTimeOut</i>	(任意指定)待ち時間を秒単位で指定します。このパラメータの指定は任意です。指定されていない場合は CatTools に内蔵されているデフォルトのタイムアウト値が適用されます。

例: 30 秒以内にデバイスプロンプトの1つが戻されるのを待つ。

```
rgMult(1) = "HostName1>"
rgMult(2) = "HostName1#"
rgMult(3) = "HostName1(Config)"
iChoices = 3
lTimeOut = 30

iRetVal = cl.WaitForMultData(rgMult, iChoices, lTimeOut)
If iRetVal = 0 Then
    cl.Log 4, "Failed to receive device prompt"
End if
```

送信のみの場合:

```
cl.WaitForMultData(rgMult)
```

CatTools のデフォルトのタイムアウト値を使う場合:

```
cl.WaitForMultData(rgMult, , lTimeOut)
```

指定したタイムアウト値を使う場合、*iChoices* の値を適用します。

定数 `COMMAND_TIMEOUT` を使ってタイムアウト値を指定している場合、次のように記述してタイムアウト値を増やすことができます。

```
Const COMMAND_TIMEOUT = 30 '(ie. in seconds)

iTimeoutMultiplier = 2 '(increase by a factor of 2)
cl.WaitForMultData(rgMult, , COMMAND_TIMEOUT *
iTimeoutMultiplier)
```

cl.ReplaceText(sData, sFind, sReplace) As String

この関数は指定したデータストリングに含まれているサブストリングを新しいサブストリングに置換します。すべての対象サブストリングが置換されます。戻り値は置換された新しいデータストリングで返されます。置換されなかった場合は元のストリングで返されます。入力パラメータである *sData* の長さが 0 の場合、ヌルが返されます。

入力パラメータは 3 つです。

<i>sData</i>	検索対象のストリングを指定します。
<i>sFind</i>	置換前のサブストリングを指定します。
<i>sReplace</i>	置換後のサブストリングを指定します。

例 1: *sData* オブジェクトに含まれている "old text" という文字列をすべて "new data" に置換する。

```
sData = cl.ReplaceText(sData, "old text", "new text")
```

例 2: *sData* オブジェクトに含まれているヌルを削除する。

```
sData = cl.ReplaceText(sData, Chr(0), "")
```

cl.TextRemoveBlankHeaderLines(ByVal sData) As String

この関数は、デバイスの出力バッファなどストリングの冒頭にある空白のヘッダーラインを削除するために使います。主に、デバイスから送信されてきたコンフィグレーションデータの出力を送信してバックアップファイルを作成する目的で使用します。出力バッファストリングの先頭から処理を開始し、キャリッジリターン<CR>や改行<LF>のみを含む行を削除します。戻り値はバッファストリングが切り詰められた文字列値で返されます。

値によって渡される入力パラメータは 1 つです。

<i>sData</i>	処理対象のストリングを指定します。
--------------	-------------------

例: バッファストリングオブジェクトである *sConfigData* の先頭に含まれている

空白行をすべて削除する。

```
sConfigData = cl.TextRemoveBlankHeaderLines(sConfigData)
```

`cl.TextRemoveLinesContainingText(ByVal sData, ByVal sFind) As String`

この関数は、デバイス出力バッファから特定のテキストサブストリングを含む行を削除するために使います。主に、デバイスから送信されてきた出力データを送信してテキストファイルやレポートを作成する目的で使用します。戻り値は、指定のサブストリングを含むために削除された行で構成された、新しいバッファとなる文字列値で返されます。

値によって渡される入力パラメータは2つです。

<code>sData</code>	検索対象のストリングを指定します。
<code>sFind</code>	検索中のサブストリングを指定します。

例: バッファストリングオブジェクト `sConfigData` から、「--More--」等デバイスページングプロンプトを含むすべてのコンフィグ行を削除する。

```
Const DEVICE_MORETEXT = "--More--"

sConfigData = cl.TextRemoveLinesContainingText(sConfigData,
DEVICE_MORETEXT)
```

`cl.TextInText(ByVal sData, ByVal sFind) As Integer`

この関数はストリングに含まれているサブストリングの開始点を返します。何も検索されなかった場合、あるいはすべての入力ストリングの長さが0の場合は0が返されます。

値によって渡される入力パラメータは2つです。

<code>sData</code>	検索対象のストリングを指定します。
<code>sFind</code>	検索中のサブストリングを指定します。

例: デバイスヘッダーテキストの開始点を検索する。値が0の場合、見つからなかったことを示すテキストを[Info Log]に送る。

```
Const DEVICE_CONFIGHEADERTEXT = "Generating configuration:"
iRetVal = cl.TextInText(sConfigData, DEVICE_CONFIGHEADERTEXT)
If iRetVal = 0 then
    cl.Log 4, "Failed to find device configuration header text"
```

```
End if
```

`cl.TextRemoveTextUpTo(ByVal sData, ByVal sFind, Optional bAndIncluding As Boolean = False, Optional bForwards As Boolean = True) As String`

この関数は、指定したサブストリングまでの長さでテキストを切り詰めたいときに使用します。切り詰め処理は冒頭からでも末尾からでも、どちらの方向にでも実行できます。戻り値は、該当するテキストに対して切り詰め処理がなされた後の文字列で返されます。sFind で指定したサブストリングが見つからなかった場合は、元の sData で指定したストリングが返されます。

値によって渡される入力パラメータおよび任意パラメータが 2 つずつあります。

<i>sData</i>	検索するストリングを指定します。
<i>sFind</i>	切り詰めるサブストリングを指定します。
<i>bAndIncluding</i> (任意指定)	ブール値で指定します。True に設定すると、sFind で指定したサブストリングまでの長さで切り詰められます。False に設定すると、sFind で指定したサブストリングは切り詰められません(デフォルト値は False です)。
<i>bForwards</i> (任意指定)	ブール値で指定します。True に設定すると検索中のストリングの始め(冒頭)から切り詰め処理を実行します(デフォルト値は True です)。False に設定するとストリングの最後(末尾)から逆方向に切り詰め処理を実行します。

例: ヘッダーテキストの出力をチェックし、見つかった場合は指定したサブストリングまでの長さで切り詰める。

```
Const DEVICE_CONFIGHEADERTEXT = "Generating configuration:"  
  
If cl.TextInText(sConfigData, DEVICE_CONFIGHEADERTEXT) > 0 Then  
    sConfigData = cl.TextRemoveTextUpTo(sConfigData,  
    DEVICE_CONFIGHEADERTEXT, True, True)  
End If
```

3.1.5 カスタムデバイスのテスト

作成したカスタムデバイススクリプトのテストは比較的簡単に実行できます。

デバイスタイプ .ini ファイルとデバイススクリプト .txt ファイルの設定が終わ

れば、CatTools で新しいデバイスタイプを使ったアクティビティを設定することで新しいデバイスがきちんと認識され機能するかどうかを確かめることができます。

テストに最も適した基本的なアクティビティは Device.ConnectivityTest.Login です。このアクティビティを実行することによって、デバイスにアクセス可能かどうかを確認できます。デバイス上で実行される大半のアクティビティはデバイスにログインしなければ実行できないため、Device.ConnectivityTest.Login アクティビティが失敗すると他のアクティビティも失敗する可能性が高いといえます。

CatTools には、この他にもカスタムデバイススクリプトが動作するかどうかを確認するための方法がいくつかあります。

1) [Info Log]に表示されるメッセージ

[Info Log]ペインでは、アクティビティが実行中に表示されるメッセージを確認できます。表示されるメッセージのレベルは[Info Log]ウィンドウ下部にあるドロップダウンリストでフィルタして表示できます。「4) Show Debug events and above」を選択すると、アクティビティの実行中に表示された全ての cl.log メッセージが表示されます。

[Info Log]に表示されるメッセージは全て CatTools のルートフォルダの下にある InfoLog.txt ファイルに書き込まれます。このファイルは短期間のうちに肥大しやすく、そのようなときは[File]メニューの[Delete]から[Delete Infolog.txt file]を選択して内容をクリアできます。

デバイス固有の問題解決に役立つレベル 4 のメッセージを表示するためのクライアントスクリプト関数 cl.Log 4 の"message"がデバイススクリプト内のどこどこに記述されています。

2) デバッグログ

[File]メニューの下の[Enable Capture Mode]を選択してオンにすると、CatTools とエンドデバイス間の通信のロギングが開始され、¥Debug フォルダ内にその通信を記録するためのファイルが作成されます。この機能は、コマンドを入力してもスクリプトが正常にデバイスと通信を行っていないように思えるときに非常に便利です。スクリプトによって、デバイスにコマンドを送信すると既知のプロンプトが表示されるまで待つが、期待したプロンプトが表示されない場合は最終的にタイムアウトになるように設定されています。デバッグログキャプチャモ-

ドのファイルには実際にデバイスから何が CatTools に送信されたかが記録されているため、その内容に応じてスクリプトを修正することができます。

3.2 デバイス固有の情報

ここでは、CatTools で様々なデバイスを設定するときや、それらの動作を理解するのに役立つ詳細情報をデバイスごとに記載します。

3.2.1 Dell 製デバイス

一部の Dell 製デバイス(例:PowerConnect スイッチ 5224, 5324, 6024 等)では、バックアップやレポートの設定を行うためにデバイスの出力をキャプチャしようとする、システムまたはポートからのメッセージが表示されて処理が中断されることがあります。

これらのメッセージはターミナルウィンドウに送出されるため、他の出力データと同時にキャプチャされる可能性があります。その結果、レポートの問題やバックアップアクティビティの compare 関数の設定にまつわる問題が発生することがあります。

このようなメッセージが表示されないようにするには、configuration モードで次のコマンドを送信してデバイスのグローバル設定を変更します。

```
no logging console
```

このコマンドにより、コンソールやターミナルのウィンドウ上にシステムメッセージがまったく表示されなくなります。システムメッセージを表示されるよう元に戻すには、次のコマンドを送信します。

```
logging console level
```

level - コンソール上に表示されるメッセージのロギングを指定したレベル emergency(緊急)、alert(警告)、critical(危険)、error(エラー)、warning(注意)、notification(通知)、informational(情報)、debugging(デバッグ) までに制限するとき指定します。

3.2.2 ASA dap.xml ファイル

ASA(v8.0)の DAP(Dynamic Access Policies)設定は、実行中のコンフィグファイルやスタートアップコンフィグファイルとは別に、フラッシュメモリ上に保存さ

れている dap.xml というファイルに記録されています。CatTools の通常の Backup アクティビティではこのファイルのバックアップは行いません。

このファイルを CatTools でバックアップするには、ファイルを TFTP でデバイスに転送する 2 つ目のアクティビティを作成する必要があります。

次のコマンドを使って Device.CLI.Send commands アクティビティを作成してください。

```
copy disk0:/dap.xml tftp://tftpserver-name/dap.xml
dap.xml
tftpserver-name
%ctDeviceName.%ctDateISO.dap.xml
```

3.2.3 HP2500 シリーズのスイッチ

Kiwi CatTools では 24xx、25xx および 26xx シリーズのスイッチをコマンドラインモードでサポートしています。

これらのスイッチが自動的にコマンドラインモードになるよう設定するには、次の作業を行う必要があります。

- コマンドライン画面を呼び出します。
- 「setup」とタイプします。
- Logon Default を CLI に設定します。
- 変更を保存します。

3.2.4 3Com Superstack スイッチ

CatTools で 3Com スイッチを使用する場合の問題点

サポート対象スイッチ

3Com SuperStack スイッチ:

610 / 630 / 1100 / 3300 / 3300 FX / 3300XM / 3300TM / 3300SM / 3300MM

背景

3Com スイッチの設定を取得するのに Telnet が最適というわけでも、Telnet で十分だというわけでもありませんが、他に実利的な選択肢は多くありません。CatTools では、様々な display コマンドの出力をキャプチャすることによって、スイッチが故障した場合でも判読可能な設定値リストを得ることができるよう

になっています。3Com 製品では show config のような便利なコマンドが使えないため、このバックアップ方法が次善の策なのです。完全な設定値リストを得るために必要なコマンドをすべて実行するため、バックアップには 1 台のスイッチにつき最大 5 分かかってしまいます。今後発行する CatTools のバージョンでは SNMP か HTTP を使って設定を取得できるようにする予定で開発を進めています。

警告

3COM スwitchの CLI にアクセスしている最中に、時々、理由もなく電源が入ったままスイッチが再起動することがあります。この現象が起きると Telnet セッションが切断され、すべてのネットワークトラフィックがスイッチの起動シーケンスが終わるまで止まってしまいます。起動シーケンスの完了にはおよそ 15 秒かかります。

スイッチの再起動は Telnet セッションが高速ポート(100Mbps)から発せられている場合に起こりやすくなるようです。スイッチの電源が入ったまま再起動する現象が 1 回でも起きると、再びクラッシュする可能性が高くなります。このようなときは電源コードを抜いて電源を落としてから電源を入れなおすと、スイッチが安定した状態に戻り、クラッシュが起きにくくなります。

クラッシュは特定のコマンドとは関係なく起きるようです。

このようなクラッシュは、バージョン 2.62 以前のすべてのバージョンのスイッチエージェントソフトウェアで発生することが確認されています。

ユーザー名に「Security」ではなく「Admin」とタイプしてログインするとクラッシュしにくくなります。

この問題については 3Com 社製スイッチの次期リリースで解決されていることを期待しています。ご意見、フィードバック等を是非お寄せください。

推奨スイッチソフトウェア

最新の 3Com スwitchソフトウェアをご使用ください。バージョン 2.62 は以下からダウンロードできます。

ftp://ftp.3com.com/pub/superstack-ii/superstack-ii-1100/s2s02_62.exe

バックアップのお勧め

3Com 製スイッチのクラッシュの問題に対しては、トラフィックの使用率が低い時間帯にデバイスのバックアップを取るのみが、現段階で取り得る唯一の対

処方法です。バックアップの頻度をなるべく減らすためにも、設定が変更されたときにのみバックアップを取るのが効率的で有効であるといえます。バックアップ作業を行うときは、「Security」ではなく「Admin」でログインしてください。

この段階では、設定は頻繁に変更されているように見えます。取得された設定には様々なパケットカウンタやステータスが含まれているためです。これらの値はバックアップのたびに変更されます。今後発行する CatTools のバージョンでは、取得したデータをより詳細に解析し、より洗練された分かりやすいリストを生成できるようにする予定で開発を進めています。

発行可能なコマンドの一覧

CatTools で発行し、その出力をファイルに取得することができるコマンドは以下のとおりです。

- 1 "bridge display"
- 2 "bridge multicastFiltering routerPort list"
- 3 "bridge port address list all"
- 4 "bridge port summary all"
- 5 "ethernet summary all"
- 6 "feature resilience detail"
- 7 "feature trunk summary"
- 8 "ip interface display"
- 9 "snmp trap display"
- 10 "system display"
- 11 "system inventory"
- 12 "system module display"
- 13 "system security access display"
- 14 "system security user display"
- 15 "bridge vlan summary all"
- 16 "bridge vlan detail" (定義済みの VLAN 番号)
- 17 "bridge port detail" (1 ~ スイッチポート番号)
- 18 "feature trunk detail" (1 ~ スイッチポート番号)

[Device Information]画面での設定

[Username]および[Password]フィールドにデバイスにログインするためのユーザー名とパスワードを入力します。[VTY Password]と[Enable Password]フィールドはパスワードを使用しないため空白のままにしておいて構いません。

3Com 製スイッチで標準的に使用されているユーザー名は「Monitor」、「Manager」および「Security」です。パスワード等の変更を行う場合は、「Security」でログインする必要があります。

スイッチスタックへのアクセス

1つのスイッチスタックはケーブルで接続されている最大4台のスイッチで構成されます。各スイッチにはユニット番号(1~4)が振られて識別されます。IPアドレスはスタック内の各デバイスに振られている番号と同じになります。

スタック内のスイッチごとに個別のデバイスを作成します。CatToolsで「LANE card configuration」グループの下の「Session to slots:」というフィールドにユニット番号を入力します。有効な値は1~4です。

デバイスの名前には、「Sales-Switch-3」のようにユニット番号を含める必要があります。末尾の「3」はスタック内のユニット3であることを示しています。

CatTools 経由でコマンドを CLI に送信する

enable プロンプトメニューで Device.CLI.Send コマンドを使うことによって、コマンドを 3Com スwitchの CLI に送出することができます。代表的なパターンの例を下に挙げます。CatTools の正規バージョンでは出力結果をキャプチャしてファイルにエクスポートできる機能が使用できます。

「admin」ユーザーのパスワードを変更するためのコマンド

```
system security user modify admin  
newpass  
newpass  
private
```

タグ付けを行ってポート 3 を VLAN 2 に追加するためのコマンド

```
bridge vlan addport 2 3 802.1Q
```

メニュー項目を元に戻す場合は「q」(quit)コマンドを入力してください。

3.2.5 Cisco Terminal Server を経由した接続

CatTools V3 ではデバイスタイプとして Cisco Terminal Server をサポートしています。

このデバイスタイプは、コンソールポートを介してターミナルサーバータイプの

デバイスに物理的に接続されているデバイスに対して CatTools で接続できるように設計されています。ターミナルサーバーデバイスとそれに接続されているデバイスの両方または片方に対する認証もサポートしています。認証はローカルでもリモートでも可能です。

ターミナルサーバーを設定するには・・・

1. [Device Type]で「Cisco Terminal Server」を選択して新しいデバイスを作成します。
2. [Host Address]にデバイスの正しい IP アドレスを設定します。
3. [Connect via]で「Direct connect」を選択します。
4. このデバイスの認証を、ターミナルサーバーのポート使用時はパスワードを入力する必要があるように設定します。

ターミナルサーバーからデバイスに接続するには、サーバーに接続されているデバイスを次のように設定します。

1. CatTools の[Device Type]で実際のデバイスタイプと同じものを選択します。
2. [Connect via]でターミナルサーバーデバイスを経由して接続するよう指定します。
3. [Method]で「Telnet」を選択します。
4. [Port]にデバイスが接続されているターミナルサーバー上のポートを指定します。例：非同期ポート 1 は TCP ポート 2001 に該当します。
5. デバイスにアクセスする際、認証を行うよう設定します。

ターミナルサーバーに接続しているデバイス上でアクティビティを実行するために、そのデバイスを選択します。

CatTools からデバイスに接続するときは、先にサーバーに接続しなければなりません。CatTools は、ターミナルサーバーを介して接続されているデバイスに定義されているポートを使って、ターミナルサーバーに接続します。ポートを使用するための認証が要求された場合は、ターミナルサーバーから認証情報を取得して使用します。CatTools でターミナルサーバーへの認証が成功すると、CR が送信されてポートが有効になります。次に、ターミナルサーバーに接続しているデバイスに対する認証がそのデバイスに割り当てられている証明書を使って行われます。

CatTools でターミナルサーバーを利用する上で、注意しなければならないことがいくつかあります。

1. 通常のコネクションは comm ポートを経由すること。CatTools からデバイスへアクセスすると、かなり時間がかかることが予測されます。
2. ターミナルサーバーへは CatTools から直接接続されていなければなりません。CatTools では、別のデバイスを経由して接続する必要のあるターミナルサーバーの使用をサポートしていません。
3. ターミナルサーバーに接続してコマンドを発行させる、またはバックアップを取る必要がある場合は、CatTools で標準のデバイスタイプのデバイスをもうひとつ作成します(例えば、Cisco router general というデバイスタイプの Cisco 2509 ルータを作成します)。作成したデバイスのホストアドレス、接続方法などの詳細は正しく設定します。ポートは標準的な接続方式で使用するポートを選択します(例えば、Telnet の場合はポート 23 を選択します)。
4. ターミナルサーバーとデバイスとの接続が切れた後でも、ポートコネクションはタイムアウトまでは開いたままになっている可能性があります。タイムアウトまでの間隔はデバイス上で設定されている値によって異なります。認証が必要なデバイスで、ポートコネクションがタイムアウトする前に再接続要求を出すと、CatTools から "Did not receive VTY entry prompt ..." というメッセージが表示され、接続できない場合があります。このエラーは、CatTools でデバイスに認証を送信したときに、送信されたデバイス側ではポートがまだ使用可能になっているため認証を必要としないために発生します。CatTools でデバイスの認証オプションの設定を解除すれば、接続できるようになります。

このことが問題である場合、(Cisco ルーター側の)実行タイムアウトの値を低く設定してコンソールとのコネクションが長時間残らないようにするという方法も考えられます。

3.2.6 Cisco および Netscreen でサポート可能な SSH2

多くの Cisco IOS デバイスでは、バージョン 12 以降の SSH2 をサポートしています。Netscreen IOS デバイス V5+でも SSH2 をサポートしています。CatTools で Netscreen デバイスへ接続するには、SSH2 プロトコルの改良型である SSH2-nopty というプロトコルを使えば接続可能になる場合があります。

これは、コマンドプロセッサへの接続に SSH を使用する際に*nix で使用される偽のターミナルとの関連なのですが、ネットワークデバイスによっては SSH2-nopty を使用しなくても接続できるものもあります。

3.2.7 Cisco VPN

CatTools に Cisco VPN デバイスを追加する場合、まずは[Device Type]で「Cisco.VPN」を選択して設定してみてください。

Cisco.VPN デバイスタイプは、Cisco VPN デバイスに採用されているメニュードリブンシステムを操作するために送信する必要のあるコマンドを決めやすいように設計されています。

このデバイスタイプを使用すれば、ほとんどの場合、個々のメニュー、サブメニューあるいはメニュー項目にアクセスする際に使用される番号システム内での差分を処理できます。Cisco VPN 設定のバックアップに関し、まだ何らかの問題が残っている場合は、[製品サポートフォーム](#)を使って弊社までご連絡ください。

<参考> 通常、Cisco VPN デバイスに付属のメニューインターフェイスに CatTools からアクセスするには、設定画面を呼び出す一連のメニュー項目番号を送信します。

V4.0 以前の VPN IOS では 2.6.3 をメニューシーケンスとして使用します。Cisco.Other.VPN3000 ではこれがデフォルト値として設定されています。V4.1 以降の IOS ではメニューシーケンスとして 2.8.3 を使用します。Cisco.V4.1.VPN3000 ではこれがデフォルト値として設定されています。中間の V4.01 IOS では 2.7.3 を使用します。

Cisco 3002 以前では 2.5.1 を使用します。Cisco.Older.VPN3002 ではこれがデフォルト値として設定されています。

デバイスタイプとして Cisco.VPN を選択しており、デバイスのバックアップに失敗する場合は、上記の VPN タイプのどれかを選択して試みてください。VPN3000 のデバイスタイプのどれを選択してもデバイスのバックアップが成功しないときは、バックアップアクティビティの[Use alternate command]フィールドに「2.7.3」と入力してみてください。

注意: [Use alternate command]フィールドに入力すると、アクティビティを実行するデバイスとして選択されたすべてのデバイスに設定されているデフォルトのコマンドが上書きされます。

3.3 サポート対象外デバイスおよびデバイスアクティビティ

対応するデバイスタイプは日々増えています。

CatTools で現在使用可能なデバイスタイプおよび各デバイスタイプで対応しているアクティビティの最新リストについては、弊社 Web サイトの[デバイス対応表](#)を参照してください。

ご使用のデバイスをサポートするために必要なデバイスタイプが記載されていない場合は [Knowledge Base](#) の記事を参照してください。ここにはご使用のデバイスをサポートするための調査に必要なことが詳しく記載されています。[製品サポートフォーム](#)を使ってこの情報を弊社までご連絡ください。

ご使用のデバイスが一覧に記載されているが、必要なアクティビティが現在そのデバイスタイプではサポート対象外となっている場合は、[製品サポートフォーム](#)を使ってできるだけ詳しい情報と可能ならば PuTTY で関連コマンドをキャプチャして送ってください。

4 アクティビティ

CatTools のアクティビティ機能を使用すると、ネットワークデバイス上で実行するアクションを自動化できます。

- [概要](#)
- スケジュール化されているアクティビティの詳細編集フォーム
- アクティビティの一覧
- 内部関数
- カスタムアクティビティの作成 CatTools で使用できる独自のカスタムアクティビティの作成方法
- デバイスでサポートされていないアクティビティ 情報およびリンク

4.1 概要

アクティビティとは、CatTools で 1 台以上のネットワークデバイスに対して実行する内部タスクのことをいいます。対象となるデバイスは CatTools のデータベースに登録しておく必要があります。

タイマー機能を使って特定の時刻に実行するようスケジュール化したり、[Run

now]ボタンをクリックしてすぐにも実行することもできます。

アクティビティを実行すると、クライアントが起動し、デバイスに対してクエリーを実行してタスクを完了するために必要な情報を取得します。

返された情報をすべて収集し、集めた情報を元に後で利用できるよう保存するかその場で解析するのかを決定します。

結果はEメールで通知されます。

ネットワーク上に存在するすべてのデバイスに対して、使用中の設定をバックアップするアクティビティを実行する、という基本的なケースの手順は次のとおりです。

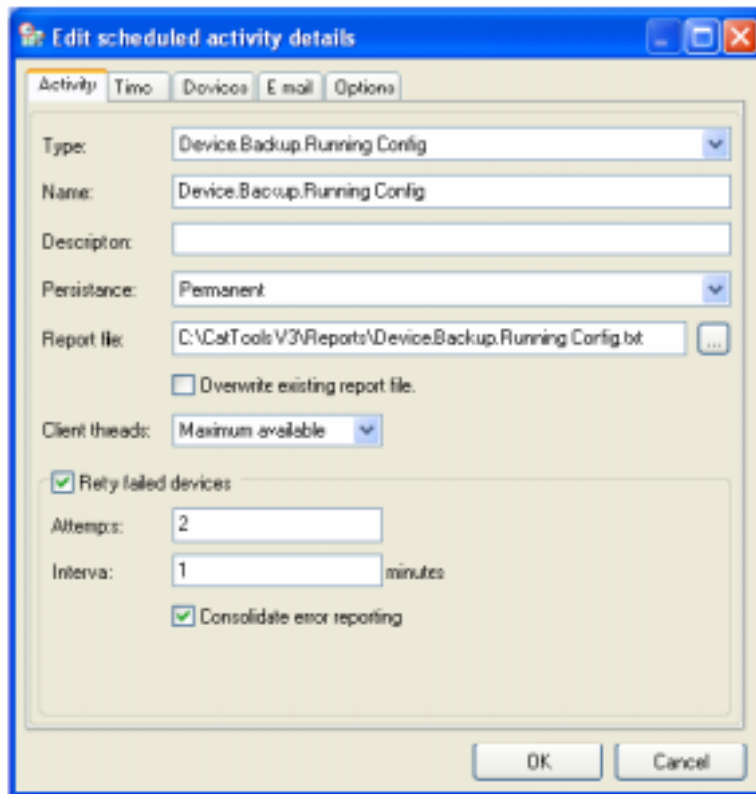
- [Type]で「Device.Backup.Running Config」を選択して新しいアクティビティを作成します。
- [Devices]タブの[Select All]ボタンをクリックして全てのネットワークデバイスに関連付けます。
- [OK]をクリックしてアクティビティを保存します。
- [Run Now]をクリックして、アクティビティを起動します。

アクティビティの進捗状況は[Info Log]または[Display]タブで監視できます。

4.2 スケジュール化されているアクティビティに関する詳細の追加/編集フォーム

この項ではアクティビティの追加および編集時に使用するインターフェイスの概要を説明します。

下図は[Edit scheduled activity details]フォームの例です。



4.2.1 [Activity]タブ

[Activity]タブではどのような種類のアクティビティを作成しているのかを定義し、重要な項目の設定を行います。

- [Type] 作成したいアクティビティのタイプをドロップダウンリストから選択します。
- [Name] アクティビティを識別するための名前を入力します。重複しない固有の名前を付ける必要があります。
- [Description] アクティビティの目的など、より詳細な情報を入力できるテキストフィールドです。
- [Persistence] ドロップダウンリストから選択して、アクティビティの有効期間を永久か1回のみかを指定します。
- [Report file] 通常は、アクティビティの実行履歴を記録しているテキストファイルへのパスを入力します。
- [Overwrite existing report file] このチェックボックスによりレポートファイルを上書きするか情報を追記するかを指定します。上書きする場合はチェックを入れます。

- [Client threads] 負荷を分散させるため、同時に実行するアクティビティの数をドロップダウンリストから選択します。これはご使用のライセンスの種類によって制限がありますが、要するに CatTools で同時に通信するデバイスの最大数だと考えて構いません。この設定はクライアントがデバイスに通信要求を発するアクティビティのみに対して有効です。当然、実行するスレッドの数が多いほど CatTools で割り当てられたタスクを完了するまでの時間が早くなります。
- [Retry failed devices] このチェックボックスをオンにすると、アクティビティの実行時に失敗したデバイスに対する再試行オプションを設定できます。再試行は指定した値(分単位)に応じた間隔で実行されます。再試行が行われている間は、そのアクティビティは実行中またはビジー (running/busy) であると認識されます。
- [Attempts] このアクティビティが失敗したデバイスに対する再試行回数を指定します。1~10 までの値を指定できます。
- [Interval] 再試行と次の再試行の間隔を分単位で指定します。1~60 までの値を指定できます。
- [Consolidate error reporting] このチェックボックスをオンにすると、アクティビティによるレベル 1 エラーのレポート方式を指定できます。このオプションは特定のエラーの内容にこだわらず、エラーが発生したかしなかったかのみ注目する場合に有効です。例えば、デバイスのバックアップを実行している際に、あるデバイスに最初は接続できなかったが、2 回目には接続できたというようなとき、最初の失敗にはあまりこだわらなくてもよい場合があります。むしろ、「バックアップが成功したか否か」が問題です。このようなときにこのオプションにチェックを入れておくと便利です。再試行のたびに発生したエラーを全て表示したいときは、チェックを外すようにしてください。

4.2.2 [Time]タブ

[Time]タブではアクティビティのスケジュール設定を行います。

Times

[Reoccurring]にチェックを入れ、ドロップダウンリストからスケジュールを選択すると、その指定したスケジュールでアクティビティが実行されます。

Days of the week

チェックボックスをオンにすることによって、アクティビティを実行する曜日を

指定できます。

Dates

アクティビティの開始日と終了日を選択できます。[Add]、[Remove]、[Move up]、[Move down]ボタンをクリックすることによって、カスタムのスケジュール日時を追加、削除、順番の入れ替えを行うことができます。このオプションを設定するときは次のことに気を付けてください。

- 設定したにも関わらずアクティビティが開始されない場合は、現在の日付が[Start date]に設定した日付よりも前になっていないか確認してください。
- 何らかの理由でアクティビティが停止する場合は、現在の日付が[End date]に設定した日付を過ぎていないか確認してください。

Favorite schedules

複数の異なるアクティビティに対して、1つのカスタマイズしたスケジュールを割り当てることができます。スケジュールを作成したら、[Save As]ボタンをクリックして保存しておけば、必要に応じて保存したスケジュールをロードし、複数のアクティビティに適用することができます。

4.2.3 [Devices]タブ

[Devices]タブではデバイスとアクティビティの関連付けを行います。

デバイスごとに設けられているチェックボックスをオン/オフすることによって、このアクティビティとの関連付けの有無を切り替えることができます。

次の4つのボタンが表示されています。

- [Select all] 表示されている全てのデバイスを選択します。
- [Deselect all] 表示されている全てのデバイスの選択を解除します。
- [Filter(F3)] [Database filter]ウィンドウが開き、表示するデバイスのフィルタリングを設定できます。[Contains]フィールドに入力するテキストは大文字/小文字が区別されます。このウィンドウ内での操作の詳細については別項で記載します。
- [Show all] フィルタを解除し、データベースに登録されている全てのデバイスを表示します。

デバイスを選択する方法は次に挙げるように複数の方法があります。

- 各デバイスのチェックボックスをクリックする。
- デバイス名の書かれている行をダブルクリックする。Windows を操作する際に標準的に用いられているリスト選択の方法を使って複数のデバイスを選択し、右クリックするとコンテキストメニューが表示されます。[Enable selected]オプションを選択すると、選択されている全てのデバイスのチェックボックスがオンになります。
- フィルタ機能を使うことによって、例えば「Head Office」という文字を含むデバイスをすべて 1 つのグループとして表示することができます。フィルタ後のデバイスが表示されている状態で[Select all]ボタンをクリックすると、そのフィルタ後のデバイスすべてにチェックが入ります。

4.2.4 [E-mail]タブ

[E-mail]タブでは現在のアクティビティに対する通知メッセージを送信するか否かを選択します。

Eメール通知には、エラー通知とレポートの2種類があります。両方とも CatTools の初期設定で設定したデフォルトの受信者に電子メールで送信されます。受信者はこの画面の[Alternative e-mail]フィールドに入力することによって変更できます。

アクティビティが実行されているデバイスのどれかでレベル 1 の緊急エラーが発生すると、アクティビティのどの段階であってもすぐにエラー通知メールが送信されます。この種の通知メールを受信したい場合は、[Email errors from this activity]にチェックを入れてください。

レポートとはアクティビティの結果の概要を記述した電子メールで、アクティビティの最後に送信されます。[Attach additional report with sortable columns]チェックボックスをオンにすると、HTML 形式のレポートが添付されて送信されます。この種の通知メールを受信したい場合は、[Email statistics/reports from this activity]にチェックを入れてください。

4.2.5 [Options]タブ

[Options]タブの内容はアクティビティの種類によって異なります。

それぞれのアクティビティについては該当する項を参照してください。

DB.UpdateDevice.Password field

DB.UpdateDevice.Text field

Device.Backup.Running Config

Device.CLI.Modify Config

Device.CLI.Send commands

Device.ConnectivityTest.Login

Device.ConnectivityTest.Ping

Device.InterDevice.Ping

Device.TFTP.Upload Config

Device.Update.Banner

Device.Update.Password

Report.ARP table

Report.CDP Neighbors table

Report.Compare.Running Startup

Report.Compare.Two files

Report.Error info table

Report.MAC address table

Report.Port info table

Report.SNMP.System summary

Report.Version table

Report.X-Ref.Port MAC ARP

4.3 アクティビティの一覧

この項では、CatTools の全アクティビティについて記載します。

4.3.1 DB.UpdateDevice.Password field

DB.UpdateDevice.Password field アクティビティを利用すれば、CatTools データベースのデバイスパスワードフィールドに入力されるパスワードを管理することができます。

他のアクティビティと違い、このアクティビティはネットワークデバイスに対して物理的な操作を行うわけではなく、CatTools データベースの[Device]テーブルに登録されている複数のデバイスに対してバッチ処理を実行します。

このアクティビティの使用例を説明します。

ネットワークデバイスが RADIUS サーバーをリモート認証用として使用しており、RADIUS サーバー上でのみパスワードを変更した、とします。今度はこの新しい情報で CatTools 内のすべてのデバイスの設定を更新しなければなりません。このような状況で、DB.UpdateDevice.Password field アクティビティを作成すれば、デバイスごとに個別に設定を変更するのではなく、全てのデバイスに一度に新しいパスワードを適用できます。

詳細については、次の「アクティビティの設定」の項を参照してください。

4.3.1.1 アクティビティの設定

[Activity]タブ:

新しいアクティビティを作成し、[Type]で「DB.UpdateDevice.Password field」を選択します。

[Time]タブ:

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ:

表内のデバイス名の隣にあるチェックボックスをオンにして、このアクティビティを適用したいデバイスを選択します。

[E-mail]タブ:

このアクティビティのレポートに関する設定を行います。

[Options]タブ:

- [VTY Password]チェックボックス CatTools データベースに登録されている[VTY Password]フィールドの値を変更できます。
- [AAA Password]チェックボックス CatTools データベースに登録されている[AAA Password]フィールドの値を変更できます。
- [Enable Password]チェックボックス CatTools データベースに登録さ

れている[Enable Password]フィールドの値を変更できます。

- [Console Password]チェックボックス CatTools データベースに登録されている[Console Password]フィールドの値を変更できます。
- [SNMP Read]チェックボックス CatTools データベースに登録されている[SNMP Read]フィールドの値を変更できます。
- [SNMP Write]チェックボックス CatTools データベースに登録されている[SNMP Write]フィールドの値を変更できます。
- [Stop on error]チェックボックス アクティビティ処理の途中でエラーが発生したときはいつでも、ジョブを中止し、CatTools に制御を戻します。

4.3.2 DB.UpdateDevice.Text field

DB.UpdateDevice.Text field アクティビティを実行すると、CatTools データベースの[Device]テーブルに設定されている任意のテキストフィールドが更新されます。

他のアクティビティと違い、このアクティビティはネットワークデバイスに対して物理的な操作を行うわけではなく、CatTools データベースの[Device]テーブルに登録されている複数のデバイスに対してバッチ処理を実行します。

このアクティビティの使用例を説明します。

ネットワークデバイスが RADIUS サーバーをリモート認証用として使用しており、RADIUS サーバー上でのみユーザー名を変更した、とします。今度はこの新しい情報で CatTools 内のすべてのデバイスの設定を更新しなければなりません。このような状況で、DB.UpdateDevice.Text field アクティビティを作成すれば、デバイスごとに個別に設定を変更するのではなく全てのデバイスに一度に新しいユーザー名を適用できます。

詳細については、次の「アクティビティの設定」の項を参照してください。

4.3.2.1 アクティビティの設定

[Activity]タブ:

新しいアクティビティを作成し、[Type]で「DB.UpdateDevice.Text field」を選択します。

[Time]タブ:

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ:

表内のデバイス名の隣にあるチェックボックスをオンにして、このアクティビティを適用したいデバイスを選択します。

[E-mail]タブ:

このアクティビティのレポートに関する設定を行います。

[Options]タブ:

- [Database Field](x4)チェックボックス ドロップダウンリストからフィールドを選択し、該当するデータベースのテキストフィールドの値を入力します。データベース内で登録されているデバイスの詳細項目から更新したい項目を選択できます(例:Info¥Port は[Devices]の[Device info]タブにある[Port]フィールドに対応します)。
- [New database value](上のチェックボックスをオンにすると入力可能になります。) 選択したデータベース項目の変更後の値を入力します。
- [Stop on error]チェックボックス アクティビティ処理の途中でエラーが発生したときはいつでも、ジョブを中止し、CatTools に制御を戻します。

4.3.3 Device.Backup.Running Config

Device.Backup.Running Config アクティビティを実行すると、実行中のデバイスの設定のバックアップを取り、ディスクに保存されている現在のコンフィグファイルと比較します。差分がある場合、現在のコンフィグファイルが¥Dated Configs フォルダに移動され、ファイル名に現在の日付が付け加えられます。次に、新しくダウンロードされたコンフィグファイルが現在のコンフィグファイルとして置き換えられます。¥Reports フォルダの下に差分の HTML 形式のレポートファイルが生成され、指定されたユーザー宛にそのコピーが電子メールで送信されます。

処理手順:

1. SSHまたはTelnet 経由でデバイスに対するコネクションが確立されます。
2. Show running configuration コマンド(あるいは同等のコマンド)が発行されます。
3. 設定値が収集され一時ファイルとしてディスク上に保存されます。
4. ディスク上にこのデバイスに対する既存のコンフィグファイルが存在するかどうかを確認されます。
5. 存在しない場合、新しくダウンロードされたコンフィグファイルが¥Configs

- フォルダの下に現在のデバイスのバックアップファイルとして保存されます。
6. 既存のファイルが存在する場合、新しくダウンロードされたコンフィグファイルと比較されます。
 7. 差分があれば、それについてのテキスト形式と HTML 形式のレポートファイルが作成され、¥Reports フォルダの下に保存されます。
 8. 差分レポートのコピーが電子メールで送信されます。
 9. 既存のコンフィグファイルが¥Dated configs フォルダに移動され、現在の日付がファイル名に付け加えられます。
 10. 新しくダウンロードされたコンフィグファイルが¥Configs フォルダに置かれ、現在のデバイスのバックアップファイルとして保存されます。

詳細については、次の「アクティビティの設定」の項を参照してください。

4.3.3.1 アクティビティの設定

[Activity]タブ:

新しいアクティビティを作成し、[Type]で「Device.Backup.Running Config」を選択します。

[Time]タブ:

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ:

表内のデバイス名の隣にあるチェックボックスをオンにして、このアクティビティを適用したいデバイスを選択します。

[E-mail]タブ:

このアクティビティのレポートに関する設定を行います。

[Options]タブ:

[Use alternate command]

実行中のコンフィグファイルのバックアップを取るために発行されたコマンドを入力します。デバイスタイプによって異なるコマンドを入力する必要があります。例えば、Cisco 製ルーターの場合は Show running-config を使用します。違うコマンドを送信したい場合は、このフィールドに入力してください。必ず、このアクティビティに割り当てた全てのデバイスで実行可能なコマンドを入力するようにしてください。Cisco VPN にのみ有効なコマンドを入力した場合は、[Devices]タブで Cisco VPN デバイスのみが選択されていることを確認してください。

[Current config file]

現在のデバイスコンフィグレーションを保存するフォルダとファイル名を指定します。

デフォルト: ...¥Configs¥%GroupName%¥Config.Current.Running.%BaseFile%.txt

%GroupName%および%BaseFile%変数は、実際のオペレーション時にフォルダ名とファイル名に置き換えられ、同名のフォルダとファイルが生成されます。

例:

C:¥Program files¥Cattools¥Configs¥Default¥Config.Current.Running.Sales_Router.txt

この値は新しいコンフィグファイルの保存先となります。既存のファイルが存在する場合は、新しくダウンロードされたファイルと比較されます。変更点がない場合はどちらのファイルも変更されません。

[Dated config file]

古いコンフィグレーションファイルを保存するフォルダとファイル名を指定します。変更が検出されると、現在のファイルはここで指定したフォルダに移動されます。

デフォルト:...¥Dated Configs¥%GroupName%¥

Config.Dated.Running.%BaseFile%.%DateISO%-%TimeHHMM%.txt

%GroupName%、%BaseFile%、%DateISO%および%TimeHHMM%変数は、実際のオペレーション時にフォルダとファイル名に置き換えられ、同名のフォルダとファイルが生成されます。

例: C:¥Program files¥Cattools¥Dated Configs¥

Default¥Config.Dated.Running.Cisco805.20031006-1914.txt

使用可能な変数の一覧については「ファイル名変数」の項を参照してください。

[HTML compare report]

HTML形式の差分レポートを保存するフォルダとファイル名を指定します。この差分レポートには変更前と変更後の設定情報が並べて記載されています。変更点は色を変えてマークされています。

デフォルト:

...¥Reports¥ConfigChanges¥%GroupName%¥Config.Changes.Running.%Base File%.%DateISO%-%TimeHHMM%.htm

%GroupName%、%BaseFile%、%DateISO%および%TimeHHMM%変数は、実際のオペレーション時にフォルダとファイル名に置き換えられ、同名のフォルダとファイルが生成されます。

例: C:\Program files\Cattools\Reports\ConfigChanges\
Default\Config.Changes.Running.Cisco805.20031006-1914.htm

[Text summary diff report]

テキスト形式の差分レポートを保存するフォルダとファイル名を指定します。この差分レポートファイルには新旧の設定情報の変更部分が一覧で記載されています。このレポートの方がサイズも小さく、シンプルに変更部分を確認することができます。HTML形式のレポートのように新旧の両方の設定情報がフルで表示されず、差分行のみが表示されます。

デフォルト:

...\Reports\ConfigChanges%\%GroupName%\Config.Changes.Running.%BaseFile%.%DateISO%-%TimeHHMM%.txt

%GroupName%、%BaseFile%、%DateISO%および%TimeHHMM%変数は、実際のオペレーション時にフォルダとファイル名に置き換えられ、同名のフォルダとファイルが生成されます。

例: C:\Program files\Cattools\Reports\ConfigChanges\
Default\Config.Changes.Running.Cisco805.20031006-1914.txt

[Only notify by e-mail if configs have changed]

このチェックボックスをオンにすると、何回でも設定のバックアップを実行できますが、変更が検出されたときのみ電子メールで通知されるようになります。

[Attach reports to e-mail]

サブオプション:

- HTML compare report only (デフォルト)
- Text diff report only
- Both reports

電子メールで受信する差分レポートのレベルを選択します。

完全な比較レポートが必要な場合はHTML比較レポートを受信するよう設定し、変更点のみのレポートが良い場合はテキストレポートを受信するよう設定します。

レポートを安全でないメディア(企業ネットワークやインターネット等)を介して送信する場合は、HTML レポートにはご使用のデバイスの全設定情報が記載されていますので注意してください。安全でないメディアを介して HTML レポートの送信はしないようにするか、レポートを zip で圧縮してパスワード保護しておくことをお勧めします。パスワードは長いほうが破られにくくなります。

[Zip attachments]

ネットワークトラフィックの負荷を減らし、パスワード保護を可能にするには、このチェックボックスをオンにして添付ファイルを zip で圧縮するようにします。これはお勧めの方策であるためデフォルトでオンになっています。この機能を使うことによって、電子メールのサイズを抑制し、複数のファイルを 1 つの添付ファイルにまとめることができます(400 台のデバイスがあるときなどに非常に便利です)。

[Password protect zip file]

zip ファイルにパスワードを追加することによって設定情報やレポートを保護できます。パスワードは長いほうが破られにくくなります。パスワード保護をしておけば、ご使用の電子メールの受信箱に転送されるまでの間ネットワークを覗き見しているユーザーから大切な設定情報などが盗まれないように守ることができます。

4.3.3.2 設定情報のバックアップの必要性について

どうして設定情報のバックアップを取ることが重要なのでしょうか。

Device.Backup.Running Config アクティビティは Kiwi CatTools に付属している数多くのデフォルトアクティビティの一つです。このアクティビティは、選択したデバイスから使用中の設定情報をコピーし、ローカルディスク上にテキストファイルとして保存するよう設計されています。

注意しなければならないのは、このときコピーされるのは現在デバイス上で実行されている設定パラメータのみがコピーされるということです。多くのデバイスには設定情報のローカルの保存用ディスクが装備されています。ここに 1 つ以上のコンフィグファイルが存在している場合があります。CatTools で管理されているネットワークデバイスの大半に、電源が投入されると同時にロードされ、実行される起動用コンフィグファイルが存在します。

この実行中の設定情報は Telnet セッションなどを介してデバイスにログオンすれば誰でもリアルタイムで変更できるため、使用中の設定情報は必ずしも起動コ

ンフィグファイルの内容と一致するとは限りません。これらの異なるバージョンのデバイスの設定情報を取得することは KiwiCatTools の最大の特長であるといえます。

例えば、次のような状況を考えてみましょう。夜間勤務のエンジニアがネットワークトラブルに遭遇し、大急ぎでルーターに Telnet で接続して問題を起こしているデバイスにいくつかの変更を加え、内部の顧客に対するサービスを復旧させました。しかし、そのエンジニアはパニック状態にあったため、これらの変更を記録することも保存することも忘れてしまいました。後日、そのデバイスの電源を切ってすぐに入れなおす操作を行うと、このエンジニアが行った変更は失われてしまうため、おそらく同じ問題が発生します。それが一日のうちで最も忙しい時間帯である可能性があることは否定できません。同じ問題に悩まされることになりませんが、対処しようにも修正した設定はなくなってしまい、担当したエンジニアは家で寝ています。初めから問題解決に奔走しなければなりません。このことは、コストの無駄使いでもあり、個人（あるいは担当部署）の信用失墜にも繋がりがねません。

このような起こりがちな状況を、アクティビティを利用することによって回避する方法がいくつかあります。

まず、常に偶発的な出来事が発生したときに元の状態に戻すような変更管理を実践することがベストです。当然、既存の設定をすべて元通りに戻し、再設定することが実際には不可能な場合、このようなプランは無意味です。変更を加える前に必ず現在の設定情報をすべて取得して復帰可能な形式でコピーしておくことを習慣付ける必要があります。KiwiCatTools は、この一連の作業に非常に有効な機能を備えています。

次に、CatTools にはアクティビティの一部として動作する差分レポート(Diff Report)機能があります。この機能はファイルに保存されている旧バージョンの設定情報と、ダウンロードした設定情報の比較を行います。変更点が検出された場合、差分部分が色分けされている HTML 形式のレポートが生成されます(Diff Report というのは「差」を表す *differnt* から付けられた名称です)。デフォルトでは、このレポートが生成されたら `Reports` フォルダの下に保存され、同時に電子メールで送信されるよう設定されています。

したがって、上記の夜勤エンジニアの場合、ルーターの変更を行う前後に `Device.Backup.Running Config` を実行していれば、どのような変更が行われたのかをレポートで確認できた上、変更前後の設定情報が記載された 2 つのテキスト

ファイルも得られていたことになります。

このテキストファイルのどちらかを使ってルーターの設定を任意の状態に戻したり、あるいは、Diff Report を元に変更点を確認しながら手動で再設定できたはずです。

4.3.4 Device.CLI.Modify Config

Device.CLI.Modify Config アクティビティを実行すると、1 台または指定した複数のデバイスの設定を一括で変更されます。

CatTools は選択したデバイスに順番にログインし、指定した設定変更を実行し、エラーが起きた場合は次のファイルにレポートとして書き出します。

```
...¥Reports¥Device.CLI.Modify Config.txt
```

Device.CLI.Send commands アクティビティでは通常のコマンドラインプロンプト(あるいは Enable モード)で同じことが実行されますが、この Device.CLI.Modify Config アクティビティでは Config モード(Config ターミナル)でコマンドが実行されます。

注意:このアクティビティの動作を簡単にテストし、体験するには、下記のコマンドを使ってインターフェイスの記述を変更してください。

```
int e0
description this is a test
```

上記のコマンドをコピーして description を変えて交互に実行し、そのたびに内容が切り替わることを確認してください。

Device.Backup.Running Config を併用すれば、コンフィグファイル内の変更を確認し、電子メールでレポートを受信する等ができるようになります。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.4.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.CLI.Modify Config」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを毎晩 0 時に実行させたい場合は[Recurring]で「At Midnight」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからこのアクティビティを実行したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Config commands to be entered into the device (Max 10,000 chrs)]

このフィールドには選択したデバイスの設定を変更するためのコマンドを入力します。例えば、Cisco IOS ルーターの場合、通常は Config モードで入力するコマンドを入力します(Config モードにするには、Enable プロンプトで「config term」と入力します)。Cisco IOS ルーターの VTY ログインパスワードを変更したいときは次のコマンドをこのフィールドに入力してください。

```
Line vty 0 4
password mynewpassword
```

CatTools は Config モードに切り替わり、このフィールドに入力されたとおり正確にコマンドを実行します。したがって、デバイスの設定を Telnet 経由により手動で行っているときと同様に正確に入力してください。

複数のコマンドを入力するときは、コマンドごとに改行してください。 コマンドとコマンドの間に空白の行を入れると、Telnet セッションで Enter キーを押したと解釈されます。

このフィールドに入力できる文字数は 10,000 文字までという制限があります。複数のコマンドを組み合わせ、極めて込み入った設定変更を行う場合でも十分なスペースです。

コマンド内では多くの変数を使用でき、これらの変数はコマンドがデバイスに届いた時点で実際の値に置換されます。使用可能な変数については該当する項を参照してください。

コマンドの一覧に記載されているメタコマンドも入力できます。メタコマンドの詳細については「メタコマンド」を参照してください。

[Or, read commands from file]

このオプションは前述のオプションのチェックボックスと通常のパス/ファイル名の入力フィールドで構成されています(このオプションと前記のオプションは同時に指定することはできません。チェックボックスをオフにすると、前記の設定コマンドを入力するためのフィールドが無効になり、コマンドはパス/ファイル名の入力フィールドで指定したテキストファイルからのみ読み込まれます)。

テキストファイル内でコマンドを記述する際の規則は、上記の設定コマンドフィールドに入力するときと同じです。各行は1つのコマンドとして読み込まれ、デバイスの Config モードに渡されます。コマンド間に空白行が含まれる場合は Enter キーを押したものとして認識されます。

この2つのオプションの違いは主に、テキストファイルにはフィールドにコマンドを入力するときのような文字数の制限がないということです。

[Save device output to file]

これもチェックボックスとパス/ファイル名のフィールドで構成されています。デフォルトではチェックボックスがオンになっており、デバイスの名前と日付に基づいて付けられたデフォルトのファイル名が入力されています。

これは基本的にアクティビティが実行されるたびに記録される、デバイスの固有情報となります。デバイスからの出力は全てこのテキストファイルに Telnet セッションで表示されたとおり正確に記録されます。

選択されたデバイスごとに固有の出力ファイルが作成されます。

以下のオプションはすべてチェックボックスで、デフォルトでは全てがオンになっています。

[Overwrite existing capture file] アクティビティの実行後の出力から作成されるファイルが上書きされます。チェックボックスをオフにすると、ファイルに追記され日付が変わると新しいファイルが作成されます。

[Answer [yes] to any confirmation prompts] コマンドによっては確認プロンプト("are you sure? Y or N")が表示される場合があります。その場合に、コマンドを入力した時点で実際にそれが実行されることを望んでいるという前提で、自動的に Yes で答えるように設定します。チェックボックスをオフにすると、確認要求に対しアクティビティから拒否を示す "n" が応答として返されます。

[Stop on error] エラーが発生したらすぐさまアクティビティを中断します。

[Save running config to start-up config when complete] 実行中のデバイスの設定情報をメモリにコピーします。変更された設定情報は起動時に使用されます。

4.3.4.2 Device.CLI.Modify Config アクティビティの使用タイミング

このアクティビティは次のような特長を備えているために、有用なネットワーク管理ツールとして活用できます。

- ネットワークに対し非常に細かいタイミングでコントロールの実行が可能である。
- 複数のコントロールを自動化でき、手動で実行したタスクについては入力されたとおりのコマンドを正確に再実行できる。

いずれにせよ、同じコマンドライン命令を習得すればよく、これらのコマンドはユーザーの代わりに Kiwi CatTools のプログラムによって Telnet 経由で実行されます。Kiwi CatTools 自体は非常に使用方法が簡潔で覚えやすいプログラムであるため、ユーザーはこの機能を利用することで大きな恩恵を得ることができます。

このアクティビティの最も有用と思われる特長は、時間のコントロールにあります。スケジュール化しているアクティビティの組み合わせが違っていても、1日単位あるいは週単位の通常の業務の流れの中で日々変化するニーズに合わせてネットワークを調整できるのです。

CatTools はユーザーにとって使いやすいようにスケジュール化された反復的なコンフィグレーションタスクを管理できるよう設計されています。そのため、このようなタスクを手動で実行すると起こりがちなミスを予防することができるのです。また、現在の設定情報のバックアッププロセスを確実に遂行できるだけでなく、バックアップの履歴を使って前の状態に戻すこともできます。指定した時間にネットワークを通じてコマンドライン命令をリモートから実行することによって、ネットワークとその状態をアクティブに管理できます。

例:

日中のみ使用される共有の会議室に有線で繋がっているスイッチポートが複数ある場合、特定のインターフェイスを午後 6 時にシャットダウンし、別のアクティビティによって午前 7 時にそのインターフェイスが有効になるように設定変更するようスケジュール化します。このようにしておけば、認証されていないユーザーが終業時刻後に共有の会議室からネットワークに進入するのを確実に

防ぐことができます。

設定変更をスケジュール化することによって、指定した時刻でのアクセスリストのエントリを追加したり、削除することも可能になります。給与システムへのアクセスを日中の終業時間内に限るといった使い方が考えられます。

使用方法は他にもたくさん考えられるでしょう。

4.3.4.3 デバイスごとに異なるコマンドを送信するには

選択したデバイスごとに異なる方法で設定する方法は2つあります。

1. デバイスごとに個別に Device.CLI.Modify Config アクティビティを作成し、[Options]タブでコマンドを入力する。

この方法では、設定を変更したいデバイスごとに個別にアクティビティを作成し、スケジュール化するという作業を行う必要があります。

2. Device.CLI.Modify Config アクティビティを1つだけ作成し、設定を変更したいデバイス全てにチェックを入れる。[Or, read commands form file]に%BaseFile%の値を含んだファイル名を指定する。次に、デバイスごとに個別にテキストファイルを作成し、設定に含めたいコマンドを記述する。

例:

ファイル名は C:¥CatTools¥%BaseFile%.txt

設定対象のデバイスは以下のものとします。

デバイス名	ベースファイル名	コンフィグコマンドファイル名
CiscoRouter1	CiscoRouter1	C:¥CatTools¥CiscoRouter1.txt
CiscoRouter2	CiscoRouter2	C:¥CatTools¥CiscoRouter1.txt
Sales_Switch	Sales Switch	C:¥CatTools¥Sales_Switch.txt

空白や無効な文字はベースファイル名ではアンダーバー(_)に置換されます。

それぞれのテキストファイル内に実行したいコマンドリストを記述してください。

ファイル C:¥CatTools¥CiscoRouter1.txt へのコマンド記述例を次に示します。

```
Interface s 0/0
Description New WAN link to head office
No shutdown
```

テキストファイルごとに異なるコマンドを入力したら、アクティビティを実行します。Kiwi CatTools は自動的に%BaseFile%の値に基づいてデバイスとそのデバイスに対応するテキストファイルとのマッチングを行います。

4.3.5 Device.CLI.Send commands

Device.CLI.Send commands アクティビティを実行すると、選択した複数のデバイスに対して、あたかも通常のコマンドラインプロンプト(あるいは Enable モード)からコマンドを入力しているかのように一連の CLI コマンドを発行します。

CatTools では自動的に以下のことが実行されます。

1. 対象デバイスにログインします。
2. 端末ディスプレイのウィンドウ長の問題の対応をとります。
3. 必用であれば、Enable モードまたは優先(Privileged)モードに切り替えます。
4. 指定したコマンドを順に入力し、有効な応答が返ってくるまでしばらく待ちます。
5. 必要であれば、入力したコマンドとデバイスからの応答をテキストファイルにログとして記録します。
6. エラーが発生したときに処理を続行するか中止するかはユーザーが決めます。
7. デバイスから切断します。

詳細については次の「アクティビティの設定」の項を参照してください。

4.3.5.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.CLI.Send commands」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを毎晩 0 時に実行させたい場合は[Recurring]で「At Midnight」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからこのアクティビティを実行したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[List of commands to be entered into the device (Max 10,000 chrs)]

このフィールドには選択したデバイスに送信するコマンドを入力します。

入力可能な CLI コマンドの例を挙げます。

(Cisco ルーターの場合)

Clear counters

Clear interface

(Cisco Catalyst CatOS スイッチの場合)

Set Port 1/1 disable

Set Port 2/12-24 enable

コマンドごとに改行して入力する必要があります。コマンドとコマンドの間に空白の行を入れると、Telnet セッションで Enter キーを押したと解釈されます。

このフィールドに入力できる文字数は 10,000 文字までという制限があります。複数のコマンドを組み合わせ、極めて込み入った設定変更を行う場合でも十分なスペースです。

[Or, read commands from file]

このオプションは前述のオプションのチェックボックスと通常のパス/ファイル名の入力フィールドで構成されています(このオプションと前記のオプションは同時に指定することはできません。チェックボックスをオフにすると、前記の設定コマンドを入力するためのフィールドが無効になり、コマンドはパス/ファイル名の入力フィールドで指定したテキストファイルからのみ読み込まれます)。

テキストファイル内でコマンドを記述する際の規則は、上記の設定コマンドフィールドに入力するときと同じです。各行は 1 つのコマンドとして読み込まれ、デバイスの Config モードに渡されます。コマンド間に空白行が含まれる場合は

Enter キーを押したものとして認識されます。

この2つのオプションの違いは主に、テキストファイルにはフィールドにコマンドを入力するときのような文字数の制限がないということです。

コマンド内では多くの変数を使用でき、これらの変数はコマンドがデバイスに届いた時点で実際の値に置換されます。使用可能な変数の一覧については該当する項を参照してください。

[Enter commands in enable mode]

このチェックボックスをオンにすると、コマンドがデバイスに届く前に Enable モードに切り替わります。

[Save device output to file]

このチェックボックスをオンにすると、デバイスから返された全出力がファイル名フィールドに指定したファイルに書き込まれます。デフォルトではチェックボックスがオンになっており、デバイスの名前と日付に基づいて付けられたデフォルトのファイル名が入力されています。

これは基本的にアクティビティが実行されるたびに記録される、デバイスの固有情報となります。デバイスからの出力は全てこのテキストファイルに Telnet セッションで表示されたとおり正確に記録されます。

選択されたデバイスごとに、%DeviceName%および%DateISO%変数に基づいて固有の出力ファイルが作成されます。

使用可能な変数の一覧については該当する項を参照してください。

以下のオプションはすべてチェックボックスで、デフォルトでは全てがオンになっています。

[Overwrite existing capture file]

アクティビティの実行後の出力から作成されるファイルが上書きされます。チェックボックスをオフにすると、ファイルに情報が追記され、日付が変わると新しいファイルが作成されるようになります。

[Answer [yes] to any confirmation prompts]

コマンドによっては確認プロンプト("are you sure? Y or N")が表示される場合があります。その場合に、コマンドを入力した時点で実際にそれが実行されることを望んでいるという前提で、自動的に Yes で答えるように設定します。チェック

ボックスをオフにすると、確認要求に対しアクティビティから拒否を示す"n"が応答として返されます。

[Stop on error]

コマンドがデバイスに届いたときにエラーが発生したら、すぐさまアクティビティを中断します。

4.3.5.2 Catalyst CatOS スイッチのポートを Enable/Disable するには

Cisco Catalyst CatOS スイッチ(4000, 5000, 6000)は Set タイプのコマンドを使って設定を変更します。

Device.CLI.Send commands アクティビティをスケジュール化することにより、特定の時刻にスイッチのポートを Enable または Disable にすることが簡単にできます。

朝、ポートを Enable にするアクティビティを作成し、スケジュール化します。

[Time]タブでの設定

[Custom Schedule] 07:00 Monday to Friday

[Option]タブでの設定

[X] List of commands to be entered on device:

Set Port 3/1-24 enable (スロット 3 の 1 ~ 24 までのポートを Enable にする)

Set Port 5/1-12 enable (スロット 5 の 1 ~ 12 までのポートを Enable にする)

[X] Enter commands in enable mode

[X] Save device output to file [ログファイル名を指定]

[X] Overwrite existing capture file

[X] Answer yes to any confirmation prompts

[X] Stop on error

次に、夕方以降ポートを Disable にするアクティビティを別に作成し、スケジュール化します。

[Time]タブでの設定

[Custom Schedule] 19:00 Monday to Friday

[Option]タブでの設定

[X] List of commands to be entered on device:

Set Port 3/1-24 disable (スロット 3 の 1 ~ 24 までのポートを Disable にする)

Set Port 5/1-12 disable (スロット 5 の 1 ~ 12 までのポートを Disable にする)

[X] Enter commands in enable mode

[X] Save device output to file [ログファイル名を指定]

[X] Overwrite existing capture file

[X] Answer yes to any confirmation prompts

[X] Stop on error

4.3.5.3 Cisco IOS イメージを TFTP サーバーにバックアップするには

Device.CLI.Send commands アクティビティを使って、Enable プロンプトで次のコマンドを送信します。

[Option]タブでの設定

[X] List of commands to be entered on device:

copy flash: tftp:

c2500-ik8s-l.122-5.bin (バックアップするイメージファイルの名前)

192.168.1.200 (リモート TFTP サーバーの IP アドレス)

c2500-ik8s-l.122-5.bin (TFTP サーバー上で使用するイメージファイルの名前)

[X] Enter commands in enable mode

[X] Save device output to file [ログファイル名を指定]

[X] Overwrite existing capture file

[X] Answer yes to any confirmation prompts

[X] Stop on error

4.3.5.4 Cisco IOS の実行中の設定情報を TFTP サーバーにダウンロードするには

Device.CLI.Send commands アクティビティを使って、Enable プロンプトで次のコマンドを送信します。

[Option]タブでの設定

[X] List of commands to be entered on device:

copy running tftp

192.168.1.200 (リモート TFTP サーバーの IP アドレス)

%ctGroupName/%ctDeviceName-Config.txt (サーバー上で設定情報を記録するファイル名)

(空白行 - 確認ダイアログへの回答)

[X] Enter commands in enable mode

[X] Save device output to file [Specify the log file name here]

[X] Overwrite existing capture file

[X] Answer yes to any confirmation prompts

[X] Stop on error

変数の対応は以下のとおりとなります。

%ctGroupName = デバイスのグループ名

%ctDeviceName = デバイス名

注意: 旧バージョンの Cisco IOS の一部には、実行中の TFTP コマンドが確認クエリーに対し実行中のコンフィグファイルとなるソースファイル名を応答させるためにコマンド後に空白行を挿入する必要があるものがあります。

注意: TFTP の「Download」フォルダ以下でサブフォルダを指定するにはスラッシュで区切ってファイル名を指定してください。

4.3.5.5 Cisco IOS の設定情報を TFTP からデバイスにアップロードするには

Device.CLI.Send commands アクティビティを使って IOS をデバイスにアップロードする方法については、次の例を参照してください。

[Option]タブでの設定

[X] List of commands to be entered on device:

copy tftp running

192.168.1.200(リモート TFTP サーバーの IP アドレス)

%ctGroupName¥%ctDeviceName-Config.txt(サーバー上で設定情報を記録する
ファイル名)

%ctCR (TFTP デバイスを起動するための必須コマンド)

[X] Enter commands in enable mode

[X] Save device output to file [Specify the log file name here]

[X] Overwrite existing capture file

[X] Answer yes to any confirmation prompts

[X] Stop on error

注記 :

- 1) メタ変数の対応は以下のとおりとなります。
%ctGroupName = デバイスのグループ名
%ctDeviceName = デバイス名
%ctCR = キャリッジリターン
- 2) TFTP の「Upload」フォルダ以下のサブフォルダを指定するにはバック
スラッシュで区切ってファイル名を指定してください。
- 3) %ctCR 変数を指定すると、デバイスに<CR>を送信し TFTP セッション
をアクティブにします。
- 4) Device.TFTP.Upload Config アクティビティでも IOS をデバイスに
アップロードできます。

4.3.5.6 TFTP を介して Cisco IOS をアップグレードするには

Device.CLI.Send commands アクティビティを使って、Enable プロンプトで次
のコマンドを送信します。

[Option]タブでの設定

[X] List of commands to be entered on device:

erase flash:(既存のフラッシュファイルを削除し新しいファイルのためのディス

ク領域を確保します)

copy tftp: flash:

192.168.1.200 (リモート TFTP サーバーの IP アドレス)

c2500-ik8s-l.122-5.bin(送信元ファイル名)

c2500-ik8s-l.122-5.bin(受信先ファイル名)

[X] Enter commands in enable mode

[X] Save device output to file [Specify the log file name here]

[X] Overwrite existing capture file

[X] Answer yes to any confirmation prompts

[X] Stop on error

4.3.5.7 デバイスごとに異なるコマンドを送信するには

選択したデバイスごとに異なる方法で設定する方法は2つあります。

1. デバイスごとに個別に Device.CLI.Send commands アクティビティを作成し、[Options]タブでコマンドを入力する。

この方法では、設定を変更したいデバイスごとに個別にアクティビティを作成し、スケジュール化するという作業を行う必要があります。

2. Device.CLI.Send commands アクティビティを1つだけ作成し、設定を変更したいデバイス全てにチェックを入れる。[Or, read commands form file] に%BaseFile%の値を含んだファイル名を指定する。次に、デバイスごとに個別にテキストファイルを作成し、設定に含めたいコマンドを記述する。

例:

ファイル名は C:¥CatTools¥%BaseFile%.txt

設定対象のデバイスは以下のものとします。

デバイス名	ベースファイル名	コンフィグコマンドファイル名
CiscoRouter1	CiscoRouter1	C:¥CatTools¥CiscoRouter1.txt
CiscoRouter2	CiscoRouter2	C:¥CatTools¥CiscoRouter1.txt
Sales_Switch	Sales Switch	C:¥CatTools¥Sales_Switch.txt

空白や無効な文字はベースファイル名ではアンダーバー(_)に置換されます。

それぞれのテキストファイル内に実行したいコマンドリストを記述してください。

ファイル C:\%CatTools%\CiscoRouter1.txt へのコマンド記述例を次に示します。

```
Interface s 0/0
Description New WAN link to head office
No shutdown
```

テキストファイルごとに異なるコマンドを入力したら、アクティビティを実行します。Kiwi CatTools は自動的に%BaseFile%の値に基づいてデバイスとそのデバイスに対応するテキストファイルとのマッチングを行います。

4.3.5.8 複数のデバイスの出力を 1 つのファイルにまとめて保存するには

通常、Device.CLI.Send commands アクティビティはデバイスから出力されたデータをデバイスごとに個別にファイルを作成し、そこに記録します。

デフォルトのキャプチャファイル名は以下のとおりです。

```
..\%Captured Data%\%GroupName%\%DeviceName%.DeviceOutput.%DateISO%.txt
```

複数のデバイスの出力を 1 つのファイルにまとめるにはキャプチャファイル名を次のように変更してください。

```
..\%Captured Data%\MyActivityData.txt
```

[Options]タブでは[Overwrite existing capture file]のチェックを外してください。こうすると、各デバイスの出力データは共通のキャプチャファイルに追記されます。

[Activity]タブでは、[Client threads]を「1 thread」に設定します。こうすると、出力ファイルに 1 台のデバイスからのみ書き込みが可能になります。

各デバイスからの出力間に開始タグと終了タグを挿入するには、デバイスに送信するコマンドとコマンドの間にコメントを数行挿入する必要があります。

例えば、入力しているコマンドが Show ARP ならば次のように指定します。

```
! %ctDeviceName - Start
Show ARP
! %ctDeviceName - Finish
```

コマンドがデバイスに届くと、%ctDeviceName が現在のデバイス名

(SalesRouter-1 等)に置換されます。

最初と最後のコマンドが「!」で始まっているため、Cisco ルーターではこれらは無視されます。

Catalyst スイッチ(CatOS)の場合は「!で」ではなく「#」を使ってください。

キャプチャされたデータファイルは次のように表示されます。

```
! Cisco2950Switch1 - Start
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.1.1 3 0008.02b9.1111 ARPA Vlan1
Internet 192.168.1.2 1 000d.9d89.2222 ARPA Vlan1
! Cisco2950Switch1 - Finish
! Cisco2950Switch2 - Start
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.1.3 6 0008.a118.3333 ARPA Vlan1
Internet 192.168.1.4 0 0008.02cf.4444 ARPA Vlan1
! Cisco2950Switch2 - Finish
```

各デバイスからキャプチャされたデータは Start と Finish というタグで挟まれています。

4.3.6 Device.ConnectivityTest.Login

Device.ConnectivityTest.Login アクティビティを使って、選択したデバイスにログインできます(Device.ConnectivityTest.Ping アクティビティと同様、デバイス上の設定は何も変更しませんが、接続可能か否かとログイン可能か否かのみを確認します)。

CatTools は選択したデバイスに順番にログインし、エラーを次のレポートファイルに記録します。

```
...¥Reports¥Device.ConnectivityTest.Login.txt
```

一般的に、ログインテストを行う理由は、多少の違いはあっても ping 検査を行うときとほぼ同様です。

ログインテストは単純な ping よりやや深く検査を行って、プロセスにおいてデバイスとターミナル間の双方向アクセスが有効であること、およびユーザーとデバイス間のネットワークオペレーションが正常に稼動することを確認します。また、パスワードが変更されていないこともチェックします。指定したデバイスに

まだログイン可能なスタッフがいることをチェックしたり、ログインパスワードが勝手に変更されていないことを確認したい場合もあります。

レポートファイルにはテストを行ったデバイスのステータスの履歴が記録されており、法廷で争わねばならないような事態が生じたときに活用することが期待できます。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.6.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.ConnectivityTest.Login」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを 10 分おきに実行させたい場合は[Recurring]で「Every 10 minutes」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

ログインテストを実行したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Enter enable mode if possible]チェックボックス 対象デバイスに Enable モードがあり、適切なパスワードがデバイスファイルで設定されている場合、CatTools はデバイスにログインするたびに Enable モードへ切り替えます。

[Only record login failures]チェックボックス アクティビティが失敗した場合のみ記録します。これをオンにしておくことによって、レポートファイルのサイズを最小限に抑え、ハードディスクの領域を節約できます。

4.3.7 Device.ConnectivityTest.Ping

Device.ConnectivityTest.Ping アクティビティを実行すると、選択した複数のデバイスに対して ping が実行されます。

CatTools は選択したデバイスに順番に ping を実行し、エラーを次のレポートファイルに記録します。

...¥Reports¥Device.ConnectivityTest.Ping.txt

ping 検査は単純な ICMP Echo 要求であり、CatTools を実行しているマシンから選択したデバイスに対して送信されます。

ping は CatTools と選択したデバイス間とのネットワークオペレーションが正常に稼働しているかどうかを IP レベルでのみテストします。これはネットワークやユーザーのネットワーク利用能力の検査手段として役立つ場合も役立つ場合もあり、ネットワークがどのように設計されているかによって、その評価は大きく異なります。

レポートファイルにはテストを行ったデバイスのステータスの履歴が記録されており、法廷で争わねばならないような事態が生じたときに活用することが期待できます。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.7.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.ConnectivityTest.Ping」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを 15 分おきに実行させたい場合は[Recurring]で「Every 15 minutes」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストから ping 検査を実行したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Response timeout (seconds)] ping 要求が失敗したと認識するまでの待ち時間を秒単位で設定します。デフォルト値は 2 秒です。

[Ping packet send count] アクティビティが実行されるたびに各デバイスに送信される ping 要求の数を指定します。デフォルト値は 5 パケットです。

[Only record ping failures]チェックボックス アクティビティが失敗した場合の

み記録します。これをオンにしておくことによって、レポートファイルのサイズを最小限に抑え、ハードディスクの領域を節約できます。

[Log Error if 100% not reported] 全ての ping が 100%成功しなかった場合にエラーとなります。

4.3.8 Device.InterDevice.Ping

Device.InterDevice.Ping アクティビティを実行すると、選択したデバイスから複数のホストに対して ping が実行されます。

CatTools は選択したデバイスに順番にログインし、選択したデバイスごとに、あるいは選択されているデバイス以外のすべてのデバイスに対して ping を実行します。

どちらのデバイスを ping するのかを指定する方法は 2 つあります。

1. [Options]タブの[Ping a list of Hosts]フィールドにホスト名と IP アドレスを入力する。CatTools は選択したデバイスにログインし、ここに記載されたホストに対して順番に ping を実行します。

2. ([Options]タブの)[Or, ping all other selected devices]チェックボックスをオンにする。CatTools は最初に選択されたデバイスから他の全てのデバイスに対して ping を実行し、次に 2 番目のデバイスに移動してそこから他の全デバイスに対して ping を実行し・・・というように順番にすべての選択したデバイス間での ping を実行し終わるまで ping を送信し続けます。この検査を実行することによって、デバイス間の接続を完全にチェックすることができます。

全ての ping が終わると CatTools によって表形式のレポートが作成されます。その表にはデバイスごとに、そのデバイスから発せられた ping の結果が記載されています。

[Options]タブには CatTools がレポートを発行するのは ping が失敗したときのみにするオプションもあります。このチェックボックスをオンにしておけば、レポートファイルのサイズを最小限に抑えることができ、ネットワーク障害の発生時にのみ注目することができます。

注意:

ホスト名と IP アドレスの指定方法は 2 通りあります。

- ホスト名がわからない場合は IP アドレスのみ入力します。複数の IP ア

ドレスを指定する場合は改行(キャリッジリターンか Enter)で区切る必要があります。

- ホスト名がわかる場合は、ホスト名と IP アドレスの両方をコンマと空白で区切って指定します。ここでもホスト名と IP アドレスの組み合わせごとに改行する必要があります。

ホスト名は、レポート内に記載されている表の列名として使用されます。

例

Main Server, 192.168.1.1

Sales router, 192.168.2.1

ISP DNS, 202.30.45.21

詳細については、次項の「アクティビティの設定」を参照してください。

4.3.8.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.InterDevice.Ping」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを 1 分おきに実行させたい場合は[Recurring]で「Every minute」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

ping を送信する送信元デバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Ping a list of hosts (Display name, IP address)]フィールドにホスト名と IP アドレスのリストを入力するか、[Or, ping all other selected devices]チェックボックスをオンにします(このオプションをオンにすると、ホスト名と IP アドレスを入力するフィールドが無効になります。2つのオプションを同時に指定することはできません)。

[Only record ping failures]チェックボックス オンにすると、ping が失敗したと

きのみ記録します。

4.3.8.2 Device.InterDevice.Ping アクティビティの使用タイミング

Device.InterDevice.Ping アクティビティはあらゆる角度からネットワーク全体の検査を実行しそのレポートを作成できるため、有用なネットワーク管理ツールとして活用できます。

ルーター接続によるネットワークにおいては多くの場合、ルーターはネットワークを実際に使用する必要のあるデバイスよりも近い場所に設置されているため、ルーターに対して ping を実行するだけではネットワークが正しく動作していることを高いレベルで証明するには不十分です。

例えば、リモートルーターに ping を実行する際に WAN インターフェイスの IP アドレスを使用することは合理的であり一般的によく使われる方法です。したがって、Wan とルーターが正常に動作している限り、ping 要求は成功します。が、これはそのルーターから遠く離れた LAN が意図したとおりに動作していることを示しているわけではありません。また、ネットワーク上の特定のポイントからルーターに対して ping が可能であることを確認するだけで、他のポイントからの ping を実行しているわけでも遠く離れた場所にあるデバイスからネットワークを使用するのに支障がないことをテストできるわけでもありません。

リモートにある LAN の各セグメント上(あるいは各リモートルーターの後ろ)に Linux デバイスが 1 台ずつ設置されている場合、これらのデバイスに対して順番に CatTools からアクセスし、そこからネットワーク上の主要デバイスに対して ping を実行します。この方が、中央のあるポイントから WAN インターフェイスとなっているルーターに対してのみ ping を実行するよりも、ネットワークが正しく動作していることを高く証明できます。

デフォルトでは、このアクティビティによって生成されるレポートは次のファイルに情報を追記するよう設定されています。

```
...¥Reports¥Device.InterDevice.Ping.txt
```

このファイルには、サービスプロバイダが停止した場合やネットワークに影響を及ぼすような事態が生じたときに、法的な根拠となりうるデータが含まれており、大変有用であるといえます。このアクティビティを 5 分おきなど定期的に行うようスケジュール化しておけば(ただし、ネットワーク上に設置されているデバイスの数が膨大でなく、ロードに非常に時間がかかる場合を除きます)、2 つのイベントに接続するための ping が失敗したことを示すタイムスタンプ付きの記

録を利用できます。これでは ping 失敗の原因は特定できませんが、有用な証拠として利用できます。

中央デバイスの有効性チェック

このテストでは、支社から本社サーバーへの接続性を検査することができます。支社のルーター(または支社内に設置されている Linux マシン)を全て選択し、本社のサーバーに対して ping を実行します。これにより、顧客すべてが本社サーバーにアクセス可能であることを確認します。

4.3.9 Device.TFTP.Upload Config

Device.TFTP.Upload Config アクティビティを実行すると、テキスト形式のコンフィグファイルを指定のデバイスにアップロードします。

このアクティビティが現在サポートしているデバイスについては「サポート対象デバイス」の項を参照してください。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.9.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.TFTP.Upload Config」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

アクティビティを実行する時間を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからレポートしたいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[TFTP Server] TFTP サーバーのアドレスを指定します。

[TFTP FileName] 設定情報が含まれているファイルのファイル名を指定するか、デフォルトのままにしておきます。

[Save device output to file]チェックボックス デバイスから実行されたセッションの出力をファイルに保存します。

[Overwrite existing capture file]チェックボックス キャプチャファイルを上書きするか追記するかを切り替えます。

[Save to NVRAM]チェックボックス 新しい設定情報を非 CatOS デバイスのNVRAM に書き込みます。

4.3.9.2 サポート対象デバイス

Device.TFTP.Upload Config アクティビティを現在サポートしているデバイスは以下のとおりです。

- Cisco Route
- Cisco Switch IOS
- Cisco Switch CatOS

4.3.10 Device.Update.Banner

Device.Update.Banner アクティビティを利用することで、簡単にデバイスにバナーを適用できます。

このアクティビティが現在サポートしているデバイスについては「サポート対象デバイス」の項を参照してください。

CatTools はバナーコマンドを選択したデバイスに順番に送信します。コマンドの成否結果は次のレポートファイルに記録されます。

```
...¥Reports¥Device.Update.Banner.txt
```

このアクティビティではコマンド変数の使用をサポートしています。

画面上に表示されるテキストボックスから入力することもファイルから読み込むこともできますが、どちらもプレーンテキストで記述する必要があります。区切り記号や構文コマンドの語句を入力する必要はありません。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.10.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.Update.Banner」を選択しま

す。

[Time]タブ

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ

表内のデバイス名の隣にあるチェックボックスをオンにして、このアクティビティを適用したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Banner text]チェックボックス バナーテキストをテキストボックスから読み込むかファイルから読み込むかを指定します。チェックボックスをオンにすると、テキストフィールドが入力可能になります。オフにすると、適用するバナーテキストが記述されたテキストファイルを選択できるようになります。

[Which banner is to be supported] 適用するバナーを選択します。すべてのデバイスですべてのバナーがサポートされているわけではないので、注意してください。

[Save device output to file]チェックボックス 使用済みのバナーコマンドを保存するファイルを指定します。

[Overwrite existing capture file]チェックボックス キャプチャファイルを上書きするのか追記するのかを指定します。

[Save running config to start-up config when complete] バナーを NVRAM に書き込むか否かを指定します。

デバイスによってはこのオプションをサポートしていないものもあります。

4.3.10.2 サポート対象デバイス

Device.Update.Banner アクティビティを現在サポートしているデバイスは以下のとおりです。

Cisco.Switch.CatOS

Cisco.Switch.IOS

Cisco.Router.General

Cisco.Firewall.PIX

4.3.11 Device.Update.Password

Device.Update.Password アクティビティを実行すると、デバイスの一部のパスワードが変更されます。

このアクティビティが現在サポートしているデバイスについては「サポート対象デバイス」の項を参照してください。

変更可能なのは Enable モードパスワード、Enable Secret パスワード、VTY パスワードおよび Console パスワードです。

[Options]タブで、どのパスワードを変更するのかが選択し、それに対する新しいパスワードを入力してください。

Enable モードパスワードも Enable Secret パスワードもフィールドを空白にしておくことができます。空白にすることによってパスワードが何も設定されていない状態にリセットすることができます。VTY パスワードおよび Console パスワードのフィールドは空白にしておくことはできません。

アクティビティが完了すると、選択したデバイスごとにレポートが生成されます。レポートには新旧両方のパスワードが記載されています。これらのパスワードはデフォルトではプレーンテキストとして表示されます。[Options]メニューの[Setup]を選択すると表示される[Kiwi CatTools Setup]画面の[Misc]タブで[Hide password change report passwords]をオンにすればレポート内のパスワードを隠すことができます。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.11.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Device.Update.Password」を選択します。

[Time]タブ

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ

表内のデバイス名の隣にあるチェックボックスをオンにして、このアクティビ

ティを適用したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[New enable]チェックボックス オンにすると新しい Enable パスワードを入力するためのフィールドが有効になります。

[New enable secret]チェックボックス オンにすると新しい Enable Secret パスワードのためのフィールドが入力可能になります。

[Choose password for CatTools]ドロップダウンリスト CatTools に記憶させたいデバイスのパスワードの種類を指定します。

[New VTY]チェックボックス オンにすると新しい VTY パスワードを入力するためのフィールドが有効になります。

[Console password]チェックボックス オンにすると新しい Console パスワードを入力するためのフィールドが有効になります。

[Save running config to start-up config when complete]チェックボックス デバイスで実行されている設定情報をメモリに書き込んで、変更後の設定が起動時の設定となるようにするか否かを指定します。

4.3.11.2 サポート対象デバイス

Device.Update.Password アクティビティを現在サポートしているデバイスは以下のとおりです。

- Cisco Router
- Cisco Switch IOS
- Cisco Switch CatOS

4.3.12 Report.ARP table

Report.ARP table アクティビティを実行すると、選択したデバイスの ARP テーブルを元データとして使用したレポートが生成されます。

Cisco IOS デバイス上では Show IP ARP コマンドのみが実行され、結果テーブル

ルが自動的に生成されます。

生成されたデータはテキストファイルで保存されます。デフォルトでは次のファイルに保存されます。

...¥Reports¥Master ARP table.txt

アクティビティにより MAC アドレスと IP アドレスおよびデバイスのインターフェイスが自動的にインデックス付けされ、要求があれば DNS 経由でホスト名が解決されます。デフォルトでは、テーブルはアクティビティが実行されるたびに更新されるようになっており、時が経つにつれネットワークに接続されているデバイスの履歴データとして蓄積されていきます。レポートには「First Seen」列と「Last Seen」列が設けられ、これらの列のエントリにはタイムスタンプが記録されています。

Excel 等のスプレッドシート形式のプログラムにファイルをインポートすれば、MAC アドレスやホスト名を指定して検索したり、特定のポートを使用しているデバイスのリストを作成することもできます。

[Options]タブでは、代替コマンドを指定したり、レポートに記載されている MAC エントリの有効期限を設定したり、IP アドレスの解決をするか否かを設定することができます。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.12.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.ARP table」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを 1 時間おきに実行させたい場合は[Recurring]で「Every hour」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからレポートを生成したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Use alternate command]チェックボックス オンにするとコマンドの入力フィールドが有効になり、CatTools によって発行されるデフォルトコマンドで正しい結果が得られない場合の代替コマンドを設定できます。ご使用のデバイスの中に、CatTools のデフォルトのデバイスリストに記載されておらず、未だサポートされていないデバイスがある場合に有用です。デフォルトではこのチェックボックスはオフになっています。

[Days until entries expire]チェックボックス オンにすると関連する数値フィールドが有効になります(デフォルト値は 30 です)。エントリをテーブル内に表示する期間を日数で設定します。このアクティビティが実行されるたびに MAC アドレステーブルから期限を越えた古いエントリが削除されます。テーブルのサイズを適度な大きさに抑えることができます。

[Resolve IP addresses to host names]チェックボックス オンにするとレポートに使用するホスト名と IP アドレスの解決が行われます。アクティビティが実行されるたびに、プログラムがテーブルをスキャンしてホスト名フィールドが空白になっているエントリを検索します。IP アドレスのリストが生成され、まとめてホスト名との解決が実行されます。リゾルバは一度に 100 件の DNS ルックアップを試行します。あるホスト名に対して IP アドレスが返ってきたら、テーブルは更新されます。

4.3.13 Report.CDP Neighbors table

Report.CDP.Neighbors table アクティビティを実行すると、CDP neighbor コマンドを使って指定したデバイスから見えるネットワークデバイスすべてのレポートが生成されます。このアクティビティが現在サポートしているデバイスについては「サポート対象デバイス」の項を参照してください。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.13.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.CDP Neighbors table」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

アクティビティを実行する時間を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからレポートしたいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Use alternate command]チェックボックス オンにするとコマンドの入力フィールドが有効になり、CatTools によって発行されるデフォルトコマンドで正しい結果が得られない場合の代替コマンドを設定できます。ご使用のデバイスの中に、CatTools のデフォルトのデバイスリストに記載されておらず、未だサポートされていないデバイスがある場合に有用です。デフォルトではこのチェックボックスはオフになっています

[Hide duplicate IP address entries]チェックボックス オンにすると、重複するIP アドレスのエントリが隠されます。

4.3.13.2 サポート対象デバイス

Report.CDP Neighbors table アクティビティを現在サポートしているデバイスは以下のとおりです。

- Cisco Router
- Cisco Switch IOS
- Cisco Switch CatOS

4.3.14 Report.Compare.Running Startup

Report.Compare.Running Startup アクティビティを実行すると、デバイスの実行中の設定と起動設定を比較し、違いがある場合はその差分レポートが生成されます。

処理手順

まず、選択したデバイスに接続し、実行中のコンフィグファイルと起動時のコンフィグファイルを表示するためのコマンドを送信します。これらのファイルはアクティビティの実行過程で「ClientTemp」フォルダに保存されます。デバイスからこの2つのコンフィグファイルを取得すると2つのファイル間の比較が実行されレポートが生成されます。比較のプロセスは Report.Compare.Two files アクティビティに非常によく似ています。

レポートの生成

見つけた差分のレポート方法は次に挙げる 3 種類のレポートを任意に組み合わせることで生成できます。

- 1) 他のアクティビティと同様に[Activity]タブの[Report file]で指定したファイルに実行済みのジョブとその結果を順次記録したレポート。
- 2) 表の中に差分情報が並べて表示され、色分けされた HTML 形式のレポート
- 3) 変更部分のみを上から表示したテキスト形式のレポート

2つのファイルのパスにはファイル名変数のパラメータを含めることができます。

注意:すべてのデバイスで現在このアクティビティがサポートされているわけではありません。詳しくは「サポート対象デバイス」の項を参照してください。

このアクティビティの詳細については、次項「アクティビティの設定」を参照してください。

4.3.14.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.Compare.Running Startup」を選択します。

[Time]タブ

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ

表内のデバイス名の隣にあるチェックボックスをオンにして、このアクティビティを適用したいデバイスを選択します。すべてのデバイスがこのアクティビティをサポートしているわけではないので注意してください。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Alternate Startup command]チェックボックス オンにすると、デバイスに対して発行すると起動設定情報を返すコマンドを入力するテキストフィールドが有効になります。

[Alternate Running command]チェックボックス オンにすると、デバイスに対

して発行すると実行中の設定情報を返すコマンドを入力するテキストフィールドが有効になります。

[Ignore Text] 比較が実行されるたびに新しいファイルに含まれる、元のファイルと違いが生じやすい情報(タイムスタンプ等)を無視したい場合にこのフィールドに入力します。詳細については「Ignore Text」の項を参照してください。

[HTML compare report] HTML 形式の差分レポートを保存するフォルダとファイル名を指定します。この差分レポートには使用中の設定情報と起動時の設定情報が並べて記載されています。差分部分は色を変えてマークされています。

[Text summary diff report] テキスト形式の差分レポートを保存するフォルダとファイル名を指定します。この差分レポートファイルには使用中と起動時の設定情報の差分が一覧で記載されています。このレポートの方がサイズも小さく、サンプルに変更部分を確認することができます。HTML 形式のレポートのように両方の設定情報がフルで表示されず、差分行のみが表示されます。

[Only notify by e-mail if differences found] このチェックボックスをオンにすると、起動時と使用中の設定情報に違いが見つかった場合にそのことを報告するメールを受信できます。

[Attach reports to email] 上記の電子メールは差分が検出されたことを通知するのみです。差分内容を確認したい場合は、このチェックボックスをオンにして、添付するレポートの形式を選択します。

[Zip Attachments] レポートを圧縮する場合は zip ファイルの名前を入力します。zip ファイルは電子メールに添付するのみで保存するわけではないので、パスを入力する必要はありません。

[Password protect Zip file] 上記のオプションが指定されており、zip ファイルを暗号化したい場合、ここでパスワードを設定します。

4.3.14.2 サポート対象デバイス

Report.Compare.Running Startup アクティビティを現在サポートしているデバイスは以下のとおりです。

Cisco.Switch.IOS
Cisco.Router.General
Cisco.Firewall.PIX

4.3.15 Report.Compare.Two files

Report.Compare.Two files アクティビティを実行すると、指定した2つのファイルの比較を実行し、見つかった差分がレポートされます。

レポートの生成

見つかった差分のレポート方法は次に挙げる3種類のレポートを任意に組み合わせることで生成できます。

- 1) 他のアクティビティと同様に[Activity]タブの[Report file]で指定したファイルに実行済みのジョブとその結果を順次記録したレポート
- 2) 表の中に差分情報が並べて表示され、色分けされたHTML形式のレポート
- 3) 変更部分のみを上から表示したテキスト形式のレポート

2つのファイルのパスにはファイル名変数のパラメータを含めることができます。

注意

このアクティビティはデバイスを選択する必要がありません。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.15.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.Compare.Two files」を選択します。

[Time]タブ

アクティビティの実行をスケジュール化したい場合はここで設定します。

[Devices]タブ

このタイプのアクティビティはデバイスに対して実行するものではありません。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[File 1] 比較が実行されるときに最初に使用されるファイルを指定します。新しいファイルと考えて構いません。

[File 2] 比較が実行されるときに次に使用されるファイルです。元ファイルと考

えて構いません。

[Ignore Text] 比較が実行されるたびに新しいファイルに含まれる、元のファイルと違いが生じやすい情報(タイムスタンプ等)を無視したい場合にこのフィールドに入力します。詳細については「Ignore Text」の項を参照してください。

[HTML compare report] 結果を HTML 形式で得たい場合にチェックボックスをオンにして、そのレポートファイルの保存先を指定します。

[Text summary diff report] 結果をテキスト形式で得たい場合にチェックボックスをオンにして、そのレポートファイルの保存先を指定します。

[Only notify by e-mail if differences found] このチェックボックスをオンにすると、新旧のファイル間に違いが見つかった場合にそのことを報告するメールを受信できます。

[Attach reports to email] 上記の電子メールは差分が検出されたことを通知するのみです。差分内容を確認したい場合は、このチェックボックスをオンにして、添付するレポートの形式を選択します。

[Zip Attachments] レポートを圧縮する場合は zip ファイルの名前を入力します。zip ファイルは電子メールに添付するのみで保存するわけではないので、パスを入力する必要はありません。

[Password protect Zip file] 上記のオプションが指定されており、zip ファイルを暗号化したい場合、ここでパスワードを設定します。

4.3.16 Report.Error info table

Report.Error info table アクティビティを実行すると、選択したデバイスごとに全てのインターフェイスのエラーカウンタ情報を収集し、レポートされます。

このアクティビティが現在サポートしているデバイスについては「サポート対象デバイス」の項を参照してください。

レポートのデフォルトの保存先は以下のとおりです。

...¥Reports¥Master Error info table.txt

このアクティビティには 4 つのオプションがあります。

最初のオプションでレポートのヘッダーの順番、表示する列、デフォルト名を使いたくない場合の表示列のヘッダー名を指定します。

2つ目のオプションでは列の総数を計算します。アクティビティを実行するデバイスごとに概要を記した行が追加されます。

3つ目のオプションではデータ収集終了後にインターフェイスのカウンタをクリアできます。

4つ目のオプションをオンにすると、VLANからのエラーカウンタを収集できます。この最後のオプションをサポートしているのはCisco IOSデバイスのみです。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.16.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.Error info table」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを4時間おきに実行させたい場合は[Recurring]で「Every 4 hours」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatToolsのホストからレポートを生成したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Specify Report Headers] レポートのヘッダーの順番、表示する列、表示列のヘッダー名を指定します。この表の最初の行はレポートされる可能性のあるすべての情報が表示されています。任意に各列のチェックボックスをオン/オフしてレポートする列としない列を指定できます。列をドラッグして任意の位置に移動させたり、幅を変更するなどのレイアウト調整もここで行います。

[Calculate column totals]チェックボックス オンにすると、列の総数が計算されます。エラー列ごとに概要を記した行が追加されます。アクティビティを実行するデバイスごとの総数が示されます。

[Clear the interface counters]チェックボックス オンにすると、現在の全データの収集が終わったらデバイスのエラーカウンタをクリアします。

[Collect VLAN information]チェックボックス オンにすると、VLAN からのエラーカウンタを収集します。この最後のオプションをサポートしているのは Cisco IOS デバイスのみです。

4.3.16.2 サポート対象デバイス

Report.Error info table アクティビティを現在サポートしているデバイスは以下のとおりです。

- Cisco Router
- Cisco Switch IOS
- Cisco Switch CatOS
- Cisco PIX firewall

4.3.17 Report.MAC address table

Report.MAC address table アクティビティを実行すると、スイッチおよびブリッジとして機能しているルータから MSC アドレスエントリ(CAM テーブルエントリ)を収集し、そのデータを蓄積するためのテーブルが生成されます。

データはテキストファイルで保存されます。デフォルトの保存先は以下のとおりです。

...¥Reports¥Master MAC address table.txt

[Options]タブでは代替コマンドを指定したり、レポートに表示する MAC エントリの期限を設定することができます。

このアクティビティはデフォルトでデバイスタイプに応じて 1 つ以上のコマンドを発行する場合があります。このような動作が実行されたからといって、選択した全てのデバイスにおいて発行されたコマンドが必ずしもすべて有効であるとは限りません。CatTools では成功したコマンドから情報を収集し、コマンドを実行しても有効な結果が得られない場合にのみ処理を継続するため、このことが問題やエラーを引き起こすことはありません。

代替コマンドを指定するとデフォルトのコマンドがすべて上書きされます。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.17.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.MAC address table」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを1時間おきに実行させたい場合は[Recurring]で「Every hour」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからレポートを生成したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Use alternate command]チェックボックス オンにするとコマンドの入力フィールドが有効になり、CatTools によって発行されるデフォルトコマンドで正しい結果が得られない場合の代替コマンドを設定できます。ご使用のデバイスの中に、CatTools のデフォルトのデバイスリストに記載されておらず、未だサポートされていないデバイスがある場合に有用です。デフォルトではこのチェックボックスはオフになっています。

[Days until entries expire]チェックボックス オンにすると関連する数値フィールドが有効になります(デフォルト値は 30 です)。エントリをテーブル内に表示する期間を日数で設定します。このアクティビティが実行されるたびに MAC アドレステーブルから期限を越えた古いエントリが削除されます。テーブルのサイズを適度な大きさに抑えることができます。

4.3.18 Report.Port info table

Report.Port info table アクティビティを実行すると、インターフェイスの設定情報や選択したデバイスのステータスのスナップショットを作成し、そのデータをタブ区切りのレポートファイル内に蓄積します。

生成されたデータはテキストファイルで保存されます。デフォルトの保存先は以下のとおりです。

...¥Reports¥Master Port info table.txt

収集される情報は名前、VLAN、ポート種別(Ethernet、Fast Ethernet、Full Duplex

等)、ステータス、速度などです。

[Option]タブでは、デフォルトのコマンドが正しい出力結果を出さない場合の代替コマンドを入力できるオプション項目があります。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.18.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.Port info table」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを正午に実行させたい場合は[Recurring]で「At Midday」を選択します。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからレポートを生成したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Use alternate command]チェックボックス オンにするとコマンドの入力フィールドが有効になり、CatTools によって発行されるデフォルトコマンドで正しい結果が得られない場合の代替コマンドを設定できます。ご使用のデバイスの中に、CatTools のデフォルトのデバイスリストに記載されておらず、未だサポートされていないデバイスがある場合に有用です。デフォルトではこのチェックボックスはオフになっています。

4.3.19 Report.SNMP.System summary

Report.SNMP.System summary アクティビティを実行すると、標準 SNMP をサポートしている任意のデバイスから取得したデータのサマリレポートが生成されます。

レポートに表示するすべての列は選択可能で列名を変更することもできます。表示可能な列は最大 5 つで独自の OID を入力して列のデータを作成できます。

詳しくは「サポート対象デバイス」の項を参照してください。

このアクティビティの詳細については、次項「アクティビティの設定」を参照してください。

4.3.19.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.SNMP.System summary」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

アクティビティの実行をスケジュール化したい場合はここで設定します。

その他の項目はデフォルトのままにしておきます。

[Devices]タブ

SNMP 情報を収集したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Specify Report Headers] レポートのヘッダーの順番、表示する列、表示列のヘッダー名を指定します。

[Show Uptime in Ticks]チェックボックス オンにすると、デバイスのアップタイムを日、時、分などではなくマークで表示します。

[Collect Custom SNMP Date] 5個のチェックボックスと入力ボックスが並んでいます。チェックボックスをオンにしてデータを収集するための OID を入力してください。該当するレポートヘッダーにもチェックを入れれば、レポート内にチェックした項目が表示されます。

4.3.19.2 サポート対象デバイス

Report.SNMP.System summary アクティビティは SNMP MIB2 のシステム情報を出力できるデバイスであればどんなデバイスでも実行可能です。

ただし、CatTools のサポート対象となっているデバイスである必要があります。新しいデバイスを追加し、このアクティビティに対するデバイス特有のロジックを CatTools が解析できないデバイスタイプとしてそれを呼び出すことができます。デバイスおよび SNMP リードコミュニティストリングを検索してデバイス

にアクセスするために CatTools が使用するのは IP アドレスのみです。

4.3.20 Report.Version table

Report.Version table アクティビティを実行すると、選択したデバイスのソフトウェアおよびハードウェアのバージョンをタブ区切りで記述したレポートが生成されます。

生成されたデータはテキストファイルで保存されます。デフォルトの保存先は以下のとおりです。

...¥Reports¥Master Version table.txt

レポートには以下の情報が記載されます。

- Group
- Device Name
- Serial Number
- Processor
- IOS (version number)
- ROM (revision)
- Boot
- Uptime
- Flash
- NVRAM (size)
- Memory
- Image (filename)

[Report]タブでレポートを表示したとき、列名をクリックするとソートできます。再度クリックすると元の順番に戻ります。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.20.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.Version table」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、アクティビティを毎晩 0 時に実行させたい場合は[Recurring]で「At

Midnight」を選択します。その他の項目はデフォルトのままにしておきます。

その他の項目はデフォルトのままにしておきます。

[Devices]タブ

CatTools のホストからこのアクティビティを実行したいデバイスを選択します。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

4.3.21 Report.X-Ref.Port MAC ARP

Report.X-Ref.Port MAC ARP アクティビティを実行すると、ネットワークポートに対して MAC アドレスと IP アドレスの相互参照レポートを生成します。ARP レポートで既に IP アドレスとホスト名が解決している場合は、ホスト名もレポートに記載されます。

このレポートは情報のベースとして Device.Port info table、Report.MAC address table および Report.ARP table アクティビティによって生成されたレポートを利用しています。Port レポートからすべてのポート情報を収集し、ポートに対応する MAC アドレスと ARP 情報のマッチングを行います。

したがって、この3つのレポート関連アクティビティは相互参照レポートを実行する前に実行する必要があります。

このレポートでは Port レポートで選択済みのデバイスを起点として使用するため、デバイスを選択する必要はありません。

[Options]タブではソースレポートを選択し、表示する MAC アドレス数や無視するインターフェイスを指定できます。

詳細については、次項「アクティビティの設定」を参照してください。

4.3.21.1 アクティビティの設定

[Activity]タブ

新しいアクティビティを作成し、[Type]で「Report.X-Ref.Port MAC ARP」を選択します。その他の項目はデフォルトのままにしておきます。

[Time]タブ

例えば、[Times]の[Custom schedule]チェックボックスをオンにし、時間を「14:00」と入力し、[Days of the week]の「Monday」のみ選択します。このように設定すると、アクティビティは毎週月曜の午後2時に実行され、相互参照レポートが生成されます。その他の項目はデフォルトのままにしておきます。

[Devices]タブ

既存のレポートをデータとして使用するため、ここでデバイスを選択する必要はありません。

[E-mail]タブ

このアクティビティのレポートに関する設定を行います。

[Options]タブ

[Current Port info table] 相互参照に使用するポート情報テーブルのパスを指定します。

[Current MAC address table] 相互参照に使用するMACアドレステーブルのパスを指定します。

[Current ARP table] 相互参照に使用するARPテーブルのパスを指定します。

[Max MAC entries per port] レポートに表示する、1つのポートに対応するMACエントリの数を指定します。中継ポートや同様のポートがある場合に指定しておくくと便利です。

5つ以上のMACエントリがある場合にここで「5」と入力すると、新しい順に5番目までのMACエントリが表示されます。

[Interfaces to exclude] 例えば、デバイスとデバイスを繋ぐ中継ポートや大量のMACエントリがあるポートなどを指定します。

インターフェイスを除外するにはポート名を正確に入力するか、同様の名前のポートをすべて除外するための正規表現式を入力します。例えば、「Giga|Serial」と入力すると、「Giga」または「Serial」という語を含む名前のポートすべてを除外します。

4.4 内部関数

この項では、CatToolsでアクティビティをカスタマイズするために使用すると便利な内部関数とメソッドについて説明します。

4.4.1 テキスト無視

Report.Compare.Two files または Report.Compare.Running Startup アクティビティのどちらかを実行してファイルの比較を行おうとするとき、オプションで無視するテキストを指定することができます。

この指定に合致する行は無視行(ignored lines)として分類されます。無視行は差分レポートではハイライトで表示されますが、変更や差分としてのフラグは付けられません。したがって警告の電子メールも送信されません。

テキスト無視の構文は次の 2 つの部分で構成されます。

1. まず、無視するテキストをどのように処理するのかを定義します。

「^」は指定したテキストが含まれるすべての行を無視することを意味しています。

「<」は指定したテキストで始まる行を無視することを意味しています。

「>」は指定したテキストで終わる行を無視することを意味しています。

2. 次に、行の比較が行われる際に検索されるテキストを記述ます。

この機能の使用例を以下に挙げます。

^Time 「Time」というテキストを含む行すべてを無視します。

<Time 「Time」というテキストで始まる行すべてを無視します。

>Time 「Time」というテキストで終わる行すべてを無視します。

「|」記号で区切ることによって複数のテキスト無視文を連結させることができます。

例：

^! Last configuration change at

「Last configuration change at」というテキストを含む行のみを無視します。

<Current configuration

「Current configuration」というテキストで始まる行のみを無視します。

^! Last configuration change at|<Current configuration

「Last configuration change at」というテキストを含む行と「Current configuration」というテキストで始まる行を無視します。

4.4.2 ファイル名変数

CatTools のいたるところにあるファイル名フィールド内で使用可能な変数は数多くあり、これらを利用することで容易にバックアップやレポートファイルに固有の名前を付けることができます。

これらの変数はアクティビティが起動すると実際の値に置き換えられます。

例:

```
C:¥CatTools V3¥Captured Data¥%GroupName%¥  
%DeviceName%.DeviceOutput.%DateISO%.txt
```

ファイル名変数はメタデータ変数とは異なります。混同しないよう注意してください。

変数	値
%GroupName%	フォルダ名として使用されるデバイスグループ名
%BaseFile%	デバイス固有のファイル名
%DateISO%	ISO 形式の日付
%DateYYYY%	yyyy 形式の日付(年のみ)
%DateMM%	mm 形式の日付(月のみ)
%DateDD%	dd 形式の日付(日のみ)
%DateWW%	週の何番目の日(1~7 までの値)
%DateWDN%	曜日(例: Monday)
%TimeHHMM%	hhmm 形式の時間
%DeviceName%	ファイルシステムに適したデバイス名
%AppPath%	末尾にバックスラッシュを含む CatTools 実行ファイルのパス

%GroupName%および%DeviceName%はデバイス固有の変数です。したがって、アクティビティは複数のデバイスに関連付けられているため、これらがアクティビティ(サマリレポートファイル)に対して解決されることはありません。

4.4.3 メタデータ

メタデータとは、ユーザーが CatTools の内部変数や関数にアクティビティ内からアクセスするときに使用するメソッドを記述するために使用する項です。

メタデータはファイル名変数とは異なります。混同しないよう注意してください。

メタデータコマンド/変数はアクティビティの中でユーザーが定義した一連のコ

マンド内で使用します。1つのアクティビティに対して使用するメタデータコマンドは1つである場合もありますが、実際には複数のデバイスコマンド間に複数のメタデータコマンドが散りばめられて使用されているのが普通です。

コマンドはアクティビティリストに記載されているデバイスごとに CatTools によって評価され、次のいずれかの処理が実行されます。

- 1) 時間どおりに CatTools の内部アクションを実行する。メタコマンド
- 2) 実際のデータ値に変数を置き換える。メタ変数

メタデータを使用できるアクティビティは以下のとおりです。

```
Device.CLI.Send commands
Device.CLI.Modify Config
Device.Update.Banner
```

4.4.3.1 メタコマンド

%メタコマンドは CatTools メタデータのサブセットです。

これを使用することによって、アクティビティのある時点で CatTools の内部機能を実行させることができます。

以下は現在利用可能なメタコマンドの一覧です。

メタコマンド	説明
%ctDB	各デバイスのプロパティ情報が記載されている CatTools 付属のデバイステーブルを更新します。更新できるデバイスフィールドの一覧については「デバイステーブル」の項を参照してください。
%ctUM: EchoOff	コマンドを送信し、そのエコーバックを待たないためのフラグを設定します。
%ctUM: EchoOn	コマンドを送信し、そのエコーバックを待つためのフラグを設定します。
%ctUM: Timeout	コマンドに対する応答を待つ最大時間を設定します。
%ctUM: PauseTime	コマンドを送るまでの待ち期間を設定します。

4.4.3.1.1 %ctDB コマンド

%ctDB メタコマンドを使用すると、アクティビティの一部として CatTools データベースのテーブル内のあるフィールドを直接更新できます。

構文

```
%ctDB:tablename:fieldname:new field value
```

tablename には更新したいフィールドの含まれているデータベースのテーブル名を入力します。フィールドの一覧については「デバイステーブル」の項を参照してください。

new field value にはデータベースのフィールドに表示したい内容を入力します。この値にはコロン(:)を挿入することができます。コマンド文法により区切り文字として最初の3つのコロンのみ検索されます。

例えば、リモートの認証サーバーのパスワードを変更したい場合、パスワードの変更自体はサーバー側で可能ですが、CatTools にはそのことが通知されません。選択したデバイスに対する CatTools データベースのパスワードを変更するメタコマンドを使って Device.CLI.Send commands アクティビティを設定すれば、デバイス側と CatTools データベースに登録されているパスワードの変更が同時にできます。コマンドは次のように使用します。

```
%ctDB:Device:AAAPassword:thenewpassword
```

この場合、Device.CLI.Send commands アクティビティのコマンドとしてメタコマンドを1つだけ入力する必要があります。アクティビティが実行されると、選択したすべてのデバイスに対する Device テーブルの AAAPassword フィールドに「thenewpassword」と入力されます。CatTools はデバイスの1つにログインするたびにこの新しいパスワードを使用するようになります。

デバイスのローカルユーザー名およびパスワードのプロパティを設定している場合にもこの種のコマンドを使用する場合があります。デバイスのプロパティを設定するデバイスコマンドを実行してから CatTools のデータベースフィールドを更新するにはメタコマンドを使用します。このような状況で、デバイスコマンドが成功しなかった場合に CatTools がフィールドを更新しないようにするためには、[Stop on error]をオンにしてください。

Device.CLI.Modify Config アクティビティでローカルユーザー名とパスワードを設定するには次のコマンドを入力します。

```
username joe password fred ...
```

これらの変更を CatTools データベースに反映するには次のコマンドを使用します。

```
%ctDB:Device:AAAUsername:joe
%ctDB:Device:AAAPassword:fred
```

次回デバイスにログインするとこれらの値が CatTools によって使用されます。

4.4.3.1.1.1 デバイステーブル

%ctDB メタコマンドを使って更新できるデバイステーブルのフィールドを以下に挙げます。

フィールド名	最大文字数
HostAddress	255
Filename	255
Model	255
Telnet	255
TelnetPort	255
Session	255
VTYPass	255
ConsolePass	255
EnablePass	255
PrivilegeLevel	255
AAAUsername	255
AAAPassword	255
SNMPRead	255
SNMPWrite	255
RequireVTYLogin	255
LoginUsesAAA	255
EnableUsesAAA	255
VTYPrompt	255
ConsolePrompt	255
EnablePrompt	255
AAAUserPrompt	255
AAAPassPrompt	255
Address1	>255
Address2	>255
Address3	>255
ContactName	>255

ContactPhone	>255
ContactEmail	>255
ContactOther	>255
AlertEmail	>255

上記のフィールドはすべて文字入力フィールドですが、RequireVTYLogin、LoginUsesAAA、EnableUsesAAA など一部のフィールドにはオン/オフを示す数値(0=オフ、1=オン)が含まれています。

4.4.3.1.2 %ctUM

%ctUM メタコマンドを使用すると、Device.CLI.Send commands および Device.CLI.Modify Config アクティビティで使用する様々なオプションを設定できます。

コマンド

%ctUM: EchoOff

通常、コマンドが送出されると CatTools はそのコマンドのエコーを待ってから次の処理を開始します。しかし、エコーを返さないコマンドもあり、そのようなコマンドが送信された場合はこの時点でアクティビティは失敗に終わります(パスワードを変更するようなアクティビティの場合、特にこの問題が顕著です)。

%ctUM: EchoOff コマンドは CatTools に対象となるアクティビティの実行中にエコーを待たずに次のコマンドを送信させます。

%ctUM: EchoOn

コマンドのエコーを切り替える必要のある場合がありますが、できるだけ速やかにエコーバックを待つ設定に切り替えるほうが良いでしょう。CatTools では、コマンドのエコーを待つことによってデバイスからの送信データと受信データの同期を正確に取り、エラー発生の可能性を最小限に抑えることができるようになっています。

%ctUM: EchoOn コマンドは CatTools に対象となるアクティビティの実行中にエコーを待ってから次のコマンドを送信させます。

%ctUM: Timeout xx

CatTools がコマンドに対する応答の待ち時間はデフォルトで 30 秒に設定されて

います。デバイスによっては一部のコマンドに対する応答に 30 秒以上かかる場合があります。%ctUM: Timeout XX コマンドを使用すると、対象となるアクティビティで使用するすべてのコマンドに対する応答の待ち時間を設定できます。

例えば、show version コマンドのタイムアウトを 50 秒に設定したい場合は次のように記述します。

```
%ctUM: Timeout 50
show version
%ctUM: Timeout 0
```

注意：このコマンドに 0 を指定すると(%ctUM: Timeout 0)タイムアウトがデフォルト値の 30 秒に戻ります。

```
%ctUM:PauseTime xx
```

これは、CatTools が指定した時間(秒)一時停止してから次のコマンドを発行するようにするためのコマンドです。指定可能な最大値は 3600 です。

例えば、次のようにコマンドを指定すると、ping を発行し、50 秒間待ってから再度 ping を発行します。

```
ping 127.0.0.1
%ctUM: PauseTime 50
ping 127.0.0.1
```

4.4.3.2 メタ変数

一部のアクティビティは選択したデバイスに対して発行される一連のコマンドを指定できるものがあります。

Device.CLI.Send commands、Device.CLI.Modify Config および Device.Update.Banner の 3 つはこのようなものの代表的なアクティビティです。

送信するコマンド内にアクティビティの実行時に置き換えられる複数の CatTools メタデータ変数を含めることができます。

例えば、Device.Update.Banner アクティビティで次のようなテキストを設定し、すべてのデバイスに適用します。

```
Hello, I am %ctDeviceName. Contact %ctContactName on %ctContactPhone to gain access to me.
```

下記は使用可能な変数の一覧です。

変数	値
%ctDeviceName	デバイスの[Name]フィールド
%ctGroupName	デバイスの[Group]フィールド
%ctHostName	デバイスの[Host Adress]フィールド
%ctModel	デバイスの[Model]フィールド
%ctAddress1	デバイスの[Address1]フィールド
%ctAddress2	デバイスの[Address2]フィールド
%ctAddress3	デバイスの[Address3]フィールド
%ctContactName	デバイスの[Contact Name]フィールド
%ctContactPhone	デバイスの[Contact Phone]フィールド
%ctContactEmail	デバイスの[Contact Email]フィールド
%ctTimeHHMM	現在の時刻(HHMM 形式)
%ctDateISO	現在の日付(YYYYMMDD 形式)
%ctBaseFile	デバイスのベースファイル名
%ctAppPath	CatTools の実行ファイルのパスをバックスラッシュで区切って指定
%ctCR	キャリッジリターン
%ctLF	改行
%ctTAB	タブ
%ctCTRL-Z	Ctrl キーを押しながら Z。通常は config モードを終了するときを使用

注意：変数名は大文字小文字が区別され、表示されるとおりに入力する必要があります。

4.5 カスタムアクティビティの作成

CatTools には、内蔵のアクティビティでは要件を満たせないときのために、独自のカスタムアクティビティおよびアクティビティのスクリプトファイルを作成する機能があります。

必要な知識・経験等

CatTools にカスタムのスクリプトを追加する作業をスムーズに進めるためには、

ある程度の Visual Basic によるスクリプト作成に関する知識または経験があったほうが望ましいといえます。しかし、CatTools のルートディレクトリの下にある `¥Templates` フォルダにあるサンプルコードのテンプレートファイルに記載されているコメントや同じフォルダ内にあるヘルプファイルを活用すれば、初心者レベルに近い技術しかなくても、ある程度のことはできます。

概要

CatTools でカスタムアクティビティのサポートを追加するには、4 つのファイル (3 つのアクティビティファイルと 1 つのカスタムデバイスファイル) が必要です。

アクティビティファイル

- 1) アクティビティタイプファイル(.ini file) - 以下を定義します。
 - アクティビティ名
 - アクティビティ ID
 - アクティビティのメインスクリプトの(アクティビティに関連付けられている)ファイル名
 - アクティビティのクライアントスクリプトの(アクティビティに関連付けられている)ファイル名
 - ユーザーインターフェイスフィールドの値およびアクティビティの追加または編集時にアクティビティフォームの[Options]タブに表示されるデフォルト値
- 2) アクティビティのメインスクリプトファイル(.txt ファイル)-CatTools データベースからアクティビティオプションの読み込み、出力データを記録するファイルおよびフォルダの作成、変数設定し、CatTools クライアントのスレッド整列、レポートを作成したり CatTools のメインプログラムにメッセージを送信するための後処理などを実行するためのコードを記述します。
- 3) アクティビティのクライアントスクリプトファイル(.txt ファイル) - デバイススクリプトに対し一般的な関数を呼び出すための複数のコード(例えば、デバイスにログインするためのデバイス独自のコマンドを送信し、アクティビティ実行に必要なコマンドを発行し、最後にデバイスからログアウトするスクリプト)を記述します。

デバイスファイル

- 4) デバイススクリプトファイル(.custom ファイル) - カスタムアクティビ

ティ用のデバイスタイプ独自のコード(例えば、デバイスに送信するコマンドやクライアントのアクティビティスクリプトに結果を戻す前にデータの構文解析等)を記述します。

アクティビティのクライアントスクリプトファイルにもメインスクリプトファイルにも関数の呼び出しと CatTools の内部プログラムコードに書かれている変数に対する参照が含まれています。これらにはクライアントスクリプト内では「cl.」、メインスクリプトでは「ct.」という接頭辞が付けられています。cl.および ct.が付いた関数および変数の一覧とその活用法については本章のカスタムアクティビティスクリプトの開発についての項を参照してください。

- カスタムアクティビティの作成方法 - カスタムアクティビティの作成方法についての手順を簡単に説明します。
- カスタムアクティビティタイプファイル(.ini)- カスタムアクティビティタイプファイルに関する情報とその作成方法について説明します。
- カスタムアクティビティメインスクリプトファイル(.txt) - カスタムアクティビティメインスクリプトファイルに関する情報とその作成方法について説明します。
- カスタムアクティビティクライアントスクリプトファイル(.txt) - カスタムアクティビティクライアントスクリプトファイルに関する情報とその作成方法について説明します。・カスタムアクティビティデバイススクリプトファイル(.custom) - カスタムアクティビティデバイススクリプトファイルに関する情報とその作成方法について説明します。
- cl. / ct.変数および関数 - カスタムアクティビティスクリプトファイルで使用される CatTools の内部変数および関数について説明します。
- カスタムアクティビティのテスト - カスタムアクティビティをテストする方法について記載しています。

4.5.1 カスタムアクティビティの作成方法

CatTools でカスタムアクティビティを作成するための簡易ガイド

ここでは CatTools でカスタムアクティビティを作成する手順を説明します。CatTools の GUI にカスタムアクティビティタイプが表示されるようになるまでの手順とカスタムメインスクリプトファイルおよびカスタムクライアントスクリプトとアクティビティファイルの作成方法およびカスタムアクティビティをデバイスタイプに適用可能にする方法についても、その概要を説明します。

CatTools にはカスタムアクティビティのテンプレートファイルが付属しており、本書でも使用されています。このテンプレートは Cisco 製ルーターのシンプルなバージョンレポートを作成するためのものです。スクリプトファイル(.txt および.custom)を作成したら、特定のアクティビティやレポートに合わせてこれらを修正する必要があります。

カスタムアクティビティタイプファイル(.ini)、カスタムアクティビティメインスクリプトファイル(.txt)、カスタムアクティビティクライアントスクリプトファイル(.txt)、カスタムアクティビティデバイススクリプトファイル(.custom)、CatTools の内部関数やサブプロシージャを作成したアクティビティスクリプトで使用するための ct.コードおよび cl.コード例等についての詳細は、本章に記載されている関連ページを参照してください。

カスタムアクティビティの作成手順

1) カスタムアクティビティタイプファイル(.ini)を作成します。

CatTools のルートディレクトリの下に¥Templates フォルダ内にある Custom.Activity.Template.ini というファイルをコピーし、¥Activities フォルダの下に保存します。このとき次の形式で新しいファイル名を付けます。

Custom.Type.Name (例 Custom.Report.Version)

2) .ini ファイルを編集します。

メモ帳などのテキストファイルエディターで先ほど作成した.ini ファイルを開きます。.ini ファイルに書かれているテンプレートのデフォルト値を以下のように書き換えます。

```
[Activity]
name=Custom.Activity.Template
script_main=Main.Custom.Activity.Template.txt
script_client=Client.Custom.Activity.Template.txt
report_file=%AppPath%Reports¥Master Custom Activity Template.txt
id=3999
```

name の項目を重複しない固有のアクティビティ名に変更します。例:Custom.Report.PortStatus. 注意：スペースを入れず、ピリオド(.)で区切ってください。

script_main 項目を関連付けられているメインスクリプトファイル名(「Main.」で始まる重複しない固有のアクティビティ名)に変更します。

`script_client` の項目を関連付けられているクライアントスクリプトファイル名 (「Client.」で始まる重複しない固有のアクティビティ名)に変更します。

`report_file` の項目を重複しない固有のレポート名に変更します。

`id`の項目を重複しない固有の番号(他のアクティビティの.ini ファイルでも使用していない番号)に変更します。4000~4999 までの番号を使用してください。

3) 保存し、再起動します。

.ini ファイルの編集が終わったら、¥Activities フォルダに保存してファイルを閉じます。CatTools を再起動すると、アクティビティの設定画面の[Type]ドロップダウンリストに新しいカスタムアクティビティタイプが表示されます。

4) カスタムアクティビティメインスクリプトファイル(.txt)を作成します。

CatTools のルートディレクトリの下に¥Templates フォルダ内にある Main.Custom.Activity.Template.txt というファイルをコピーし、¥Scripts フォルダの下に保存します。このとき、2)でアクティビティタイプ.ini ファイルの[Activity]の `script_main` に入力した値と同じ名前を付けてファイルを保存してください。ファイルの拡張子は必ず.txt にしてください。

5) メインスクリプト.txt ファイルを編集します。

テキストファイルエディター*を使って4)で作成した.txt ファイルを開きます。開いたファイルの「SCRIPT NOTES」の項に記載されている手順に従って、作成したアクティビティに応じてスクリプトをカスタマイズしてください。

6) メインスクリプト.txt ファイルを保存します。

メインスクリプト.txt ファイルの編集が終わったら、¥Script フォルダに保存してファイルを閉じます。

7) カスタムアクティビティクライアントスクリプトファイル(.txt)を作成します。

CatTools のルートディレクトリの下に¥Templates フォルダ内にある Client.Custom.Activity.Template.txt というファイルをコピーし、¥Scripts フォルダの下に保存します。このとき、2)でアクティビティタイプ.ini ファイルの[Activity]の `script_client` に入力した値と同じ名前を付けてファイルを保存してください。ファイルの拡張子は必ず.txt にしてください。

8) クライアントスクリプト.txt ファイルを編集します。

テキストファイルエディター*を使って7)で作成した.txt ファイルを開きます。開いたファイルの「SCRIPT NOTES」の項に記載されている手順に従って、作成したアクティビティに応じてスクリプトをカスタマイズしてください。

9) クライアントスクリプト.txt ファイルを保存します。

クライアントスクリプト.txt ファイルの編集が終わったら、¥Script フォルダに保存してファイルを閉じます。

10) カスタムアクティビティデバイススクリプトファイル(.custom)を作成します。

CatTools のルートディレクトリの下に¥Templates フォルダ内にある Custom.Device.Template.txt.custom というファイルをコピーし、¥Scripts フォルダの下に保存します。このとき、関連付けられている暗号化済みデバイススクリプトファイル(またはカスタムデバイススクリプトファイル)と同じ名前を付け、ファイル拡張子.txt とさらに.custom という拡張子を付けてファイルを保存する必要があります。

例: (Cisco 製ルーター用のカスタムアクティビティデバイススクリプトファイルを作成する場合)

CatTools 付属の Cisco 製ルーター用暗号化済みスクリプトファイル:

Cisco.Router.General.txt

作成するカスタムアクティビティデバイススクリプトファイルの名前:

Custom.Cisco.General.txt.custom

11) デバイススクリプト.custom ファイルを編集します。

テキストファイルエディター*を使って 10) で作成した .custom ファイルを開きます。開いたファイルの「SCRIPT NOTES」の項に記載されている手順に従って、ご使用のデバイスに応じてスクリプトをカスタマイズしてください。カスタムアクティビティを .custom デバイスファイルに追加する方法については「カスタムアクティビティデバイススクリプトファイル(.custom)」の項を参照してください。

12) デバイススクリプト.custom ファイルを保存します。

デバイススクリプト.custom ファイルの編集が終わったら、¥Script フォルダに保存してファイルを閉じます。

13) 作成したカスタムアクティビティスクリプトをテストします。

全てのカスタムアクティビティスクリプトおよびデバイス.custom スクリプトファイルの設定が終わったら、CatTools で作成した .custom ファイルに関連付けられているデバイスを使って新しいアクティビティを実行し、正常に動作するかどうかを確認してください。カスタムアクティビティのテストは開始タイマーを使って行うのではなく、[Run Now] ボタンを使って手動で行ってください。

[Info Log]ペインでエラーを確認し、必要であれば作成した.txt ファイルと.custom ファイルを編集します。

詳細については「カスタムアクティビティのテスト」の項を参照してください。

カスタムアクティビティの作成方法の確認

- アクティビティタイプファイル(.ini)を作成する。
- アクティビティメインスクリプトファイル(.txt)を作成する。
- アクティビティクライアントスクリプトファイル(.txt)を作成する。
- アクティビティデバイススクリプトファイル(.custom)を作成する。

(*注記 : .txt ファイルはメモ帳を使って開くこともできますが、構文がハイライト表示されるようなスクリプトエディターを使ったほうが .txt ファイルの編集作業が楽になります。)

4.5.2 カスタムアクティビティタイプファイル(.ini)

CatTools で使用されているアクティビティタイプ

CatTools では、テキストファイル(.ini ファイル)を使ってアクティビティタイプを定義し、その定義ファイルは CatTools のルートディレクトリの下にある¥Activities フォルダに保存します。[Add scheduled activity details]および[Edit scheduled activity details]画面の[Type]ドロップダウンリストに表示される項目ごとに対応する.ini ファイルが存在します。

CatTools を起動すると、¥Devices フォルダにあるすべての .ini ファイルが検索され、内容が読み込まれます。検索されたアクティビティタイプはすべてアクティビティの[Type]ドロップダウンリストに表示され、選択してアクティビティを定義することができるようになります。

新しいカスタムアクティビティタイプを追加するには、新しい .ini ファイルを作成し、アクティビティとその特徴を定義する必要があります。

.ini ファイルに書かれている各項目の概要

.ini ファイルの内容は[...]という形で括弧にくくって項目別に分けられています。主な項目について説明します。

[info]項目

これは、その .ini ファイルが CatTools のファイルであることを示す必須項目です。次のように記述します。

```
[info]
cookie=Kiwi CatTools
version=3
author=Kiwi Enterprises
```

[Activity]項目

CatTools のインターフェイス(アクティビティの[Type]ドロップダウンリストに表示される名称等)やスクリプト内で使用される名前を定義するもので、必須項目です。また、CatTools のデータベース内で使用するデバイスタイプの固有キー(ID)もここで定義します。次のように記述します。

```
[Activity]
name=Custom.Activity.Template
script_main=Main.Custom.Activity.Template.txt
script_client=Client.Custom.Activity.Template.txt
description=Collects information from devices and creates a custom report
report_file=%AppPath%Reports¥Master Custom Activity Template.txt
report_overwrite=1
client_threads=0
id=3999
```

[Activity]項目には .ini ファイルの名前と似ている(あるいは同じ)名前を定義できますが、ここで定義する名前はユーザーインターフェイス内で使用されるデバイス名に反映され .ini ファイルの名前が変更されるわけではありません。

CatTools では予め定義しているアクティビティタイプ用に 0 ~ 3999 までの ID を予約しています。作成するカスタムのアクティビティタイプに設定できる番号は 4000 ~ 4999 までです。ID を定義する際は、使用しようとしている番号が重複しない固有の番号であることを確認してください。

[item]項目(アクティビティの[Option]タブに表示される項目)

[item_xxx]の各項目では、CatTools のアクティビティの設定画面のユーザーインターフェイスで使用される入力フィールドについて定義します。次のように記述します。

例:

```
[item_value1]
name=Use alternate command
type=string
checkbox_show=1
checkbox_default=0
value_default=
required=0
info="Alternative command to capture data from device."
```

アクティビティの[Options]タブにユーザーインターフェイスとして表示される入力フィールドごとに[item_xxx]という項目を作成し、個別に定義します。

name では、入力フィールドや選択フィールドの横に表示されるテキスト(またはラベル)を定義します。

type では、*name* で指定したテキストまたはラベルの横に表示される入力フィールドや選択フィールドのタイプを定義します。

checkbox_show および *checkbox_default* では、チェックボックスを表示するかどうか、およびデフォルト値としてチェック有りにするか無しにするかを定義します。

value_default では、新しいアクティビティをデータベースに追加する際に使用されるデフォルト値を定義します。

required では、そのフィールドに有効なデータを入力すること(予め定義されているリストから選択するかユーザー自身で値を入力するといった操作)が必須であるか必須でないかを定義します。必須でない場合は 0、必須である場合は 1 を指定します。

info では、このフィールドに関連する説明を定義します。ここで定義したテキストはフィールド内にカーソルを置いたときにユーザーインターフェイスのステータスバーに表示されます。

上記の[item_value1]はユーザーインターフェイス内で使用されているチェックボックス付きの標準的なテキストボックス入力フィールドの例です。

オプションフィールドに入力可能な数値の範囲を制限する必要がある場合は、*type* または *range* を追加してください。

以下の *range* オプションの例は、チェックボックス付きの入力フィールドで入力可能な値として 1~2000 までの数値に限定する場合の定義方法を記載しています。

```
[item_value2]
name=Days until entries expire
type=range
checkbox_show=1
checkbox_default=0
value_default=30
range_min=1
range_max=2000
required=1
info="Remove entries older than X days from the Master report"
```

入力フィールドがチェックボックスである場合は、0 または 1 を指定します。1 以外の値はすべて 0 とみなされます。チェックボックスの記述例については以下を参照してください。この例では、チェックボックスのデフォルト値として 1 が指定されています(デフォルトでチェックが入っている状態が表示されます)。

```
[item_value3]
name=Resolve IP addresses to hostnames
type=checkbox
checkbox_show=1
checkbox_default=1
info="Resolve all IP addresses found to hostnames via reverse DNS lookup"
```

各フィールドのサイズ(入力可能なデータの長さ)は CatTools データベースの設計によって異なります。CatTools ユーザーインターフェイスを介して入力されるデータはすべて、設定対象のフィールドに対して長すぎる場合は適宜切り詰められて表示されます。

カスタムアクティビティタイプファイル(.ini)のテンプレート

CatTools にはカスタムアクティビティタイプを新しく作成する際に、作成元として活用できるテンプレートファイルが付属しています。

テンプレートファイルの名前は Custom.Activity.Template.ini で、¥Templates というフォルダの中にあります。

このテンプレートファイルは、CatTools で予め定義されているアクティビティタイプの一部として含まれています。CatTools のすべての定義済みアクティビティのアクティビティタイプファイルは、CatTools 製品の最新バージョンが発行される

際に随時(必要であれば)更新されます。

そのため、新しくカスタムアクティビティタイプを作成する目的でテンプレートファイル自体を変更することは絶対にしないでください。テンプレートを元に新しいカスタムデバイスタイプを作成する必要がある場合は、テンプレートファイルをコピーして¥Activities フォルダに別の名前を付けて保存するようにしてください。

カスタムアクティビティタイプファイル(.ini)の作成

CatTools にカスタムアクティビティタイプを新しく追加するとき、最初にすることは ¥Templates フォルダの下にあるテンプレートファイル Custom.Activity.Template.ini をコピーして¥Activities フォルダの下に別の名前を付けて保存することです。

このファイルを作成元として新しいアクティビティタイプファイルを定義します。ファイル名を付ける際には、なるべく次の命名規則に従ってください。

Custom.Type.Name

<i>Custom</i>	"Custom"というテキストは CatTools の定義済みデバイスタイプと識別するために使用します。ここで指定した名前はアクティビティの[Type]ドロップダウンリスト内にグループ化されて表示されます。
<i>Type</i>	アクティビティのタイプ(Report, CLI, Connectivity, Update 等)を指定します。
<i>Name</i>	アクティビティの名前(ARPTable, SendCommands, PingTest, BannerText 等)を指定します。

.ini ファイルの編集

次に、カスタムアクティビティの.ini ファイルを開き、必要であれば各項目の値を変更します。

下記のコード例は .ini ファイルの例ですが、いくつかの項目の値を別色で表示しています。

最も大事な項目の値で変更しなければならないものは赤で表示しています。これらの項目には重複しない固有の値を指定する必要があります。固有の値を設定しなかった場合、作成したアクティビティタイプが CatTools のアクティビティの

[Type]ドロップダウンリストに表示されません。

必要時に変更される可能性のある項目の値は緑で表示しています。コメントは青で表示しています(注意:実際の .ini ファイルではコメントは削除してください)。

黒文字で表示している部分はすべて元の設定のままにしておくことをお勧めします。ただし、name や info など一部の項目については作成したアクティビティに関係の深い値に変更しても構いません。

name に指定するテキスト値はフォーム内に表示されるフィールドラベルとなります。info に指定するテキスト値はフォーム内のフィールドを選択したとき(つまり、カーソルを置いたとき)にステータスバーに表示されるテキストとなります。

.ini ファイルの変更が終わったら、必ず¥Activities フォルダの下に保存してください。新しいカスタムアクティビティタイプがアクティビティの[Type]ドロップダウンリストに表示されるようにするためには、CatTools を再起動する必要があります。

[info]

cookie=Kiwi CatTools

version=3

author=Kiwi Enterprisses 自分の名前を入力するかデフォルトのままにしておいてください。

[Activity]

name=Custom.Activity.Template 重複しない固有の名前(例: Custom.Report.PortStatus)に変更してください。注記:各項目の区切りには空白を入れず、ピリオド(.)を入力してください。

script_main=Main.Custom.Activity.Template.txt 関連付けられているメインスクリプトの名前(Main.アクティビティ名の形式で固有の名前)に変更してください。

script_client=Client.Custom.Activity.Template.txt 関連付けられているクライアントスクリプトの名前(Client.アクティビティ名の形式で固有の名前)に変更してください。

description=Collects information from devices and creates a custom report 必要であれば変更してください。

report_file=%AppPath%Reports¥Master Custom Activity Template.txt 固有のレポート名に変更してください。

report_overwrite=1

client_threads=0

id=3999 4000 ~ 49999 までの番号を入力してください。ここで入力する番号は重複しない固有の番号でなければなりません。

Activity options

[item_value1]

name=Use alternate command

type=string

checkbox_show=1

checkbox_default=0

edit_button=0

value_default=

required=0

info="Alternative command to capture data from device."

[item_value2]

必要に応じてオプションを追加してください(最大 10 個まで指定できます)

4.5.3 カスタムアクティビティメインスクリプトファイル(.txt)

CatTools のアクティビティスクリプト

CatTools のアクティビティスクリプトはテキストファイル(.txt ファイル)を使って定義し、CatTools のルートディレクトリの下にある¥Scripts フォルダに保存します。

アクティビティが実行される際、CatTools はアクティビティタイプ.ini ファイルの[Activity]項目の *script_main* に指定されている値を読み込み、どのアクティビティメインスクリプトファイルを使用すべきかを判断します。

(.ini ファイルと.txt ファイルの名前を、.txt ファイルには Main という接頭辞を付ける以外まったく同じにすることで混乱を避けることができます)。

CatTools の定義済みアクティビティタイプにはすべてアクティビティメインスクリプトファイル(.txt)が関連付けられており、それらのファイルは暗号化されています。暗号化されている理由は 2 つあります。まず、弊社の知的財産権を保護することです。もう 1 つは、実行時にスクリプトが動作しないという事態を招くおそれのある無許可の変更を防ぐためです。

カスタムアクティビティメインスクリプトファイル(.txt)のテンプレート

CatTools にはカスタムアクティビティメインスクリプトを新しく作成する際に、作成元として活用できるテンプレートファイルが付属しています。

テンプレートファイルの名前は Main.Custom.Activity.Template.txt で、CatTools のルートディレクトリの下に¥Templates というフォルダの中にあります。

このテンプレートファイルは、CatTools で予め定義されているアクティビティスクリプトの一部として付属しているものですが、暗号化はされていません。CatTools のすべての定義済みアクティビティスクリプトファイルは、CatTools 製品の新しいバージョンが発行される際に随時(必要であれば)更新されます。このため、新しくカスタムアクティビティメインスクリプトを作成する目的でテンプレートファイル自体を変更することは絶対にしないでください。テンプレートを元に新しいカスタムアクティビティメインスクリプトを作成する必要がある場合は、テンプレートファイルをコピーして¥Scripts フォルダに別の名前を付けて保存するようにしてください。

カスタムアクティビティメインスクリプトファイル(.txt)の作成

CatTools にカスタムアクティビティメインスクリプトを新しく追加するとき、最初にするのは¥Templates フォルダの下にあるテンプレートファイル Main.Custom.Activity.Template.txt をコピーして¥Scripts フォルダの下に別の名前を付けて保存することです。このとき、アクティビティタイプ.ini ファイルの [Activity]項目で script_main に指定したものと同一名前で保存してください。

このファイルを作成元として新しいデバイススクリプトファイルを定義します。

メインスクリプトファイル(.txt)の編集

次に、カスタムメインスクリプト.txt ファイルを開き、必要であればご使用のデバイスに合わせてスクリプトが動作するように、各項目の値を変更します。

.txt ファイルには、変更する際の手順やヒントなど多くのコメントが書かれていますが、注意しなければならない重要な項目はいくつかしかなく、それらはスクリプトファイルの冒頭に書かれています。

スクリプトの 1 行目に次のように記述されていることがあります。

```
Attribute VB_Name = "Custom_Activity_Template"
```

これは Visual Basic(あるいは同等のスクリプト言語)開発環境で表示したときに、そのカスタムアクティビティメインスクリプトファイルを示す Visual Basic モ

ジュール名となります。この行は VB 開発環境でファイルの編集を行っているときは表示されませんが、他のエディターでは表示される場合があります。

"Custom_Activity_Template"はカスタムテンプレートファイルのモジュールに付けられた名前です。カスタムアクティビティメインスクリプトファイルがたくさんある場合は、この名前を作成したカスタムアクティビティメインスクリプトのファイル名を反映した名前に変更するほうが良いこともあります。そのようにすることによって、複数のカスタムアクティビティメインスクリプトファイルをご使用の VB 開発環境で同時に(たとえば、1つのプロジェクト内などで)開くことができるようになります。

スクリプトの編集にテキストベースのエディター(メモ帳など)を使用している場合は、モジュール名の変更は任意で構いません。しかし、混乱を避けるには、この名前を変更しておいたほうが良いでしょう。

メモ:

--- SCRIPT NOTES ---

SCRIPT NOTES には、有用な情報とスクリプト作成の初期段階において実行すべき作業について記載されています。ここに書かれている内容は必要に応じて適宜書き換えて構いません。たとえば、スクリプト設定の初期に必要な作業を終え、もう再びその作業をする必要がない場合、SCRIPT NOTES に記載されている、それらに関するテキストを全部削除することができます。複雑なルーチンやスクリプト固有の振る舞いなど、デバイススクリプトの管理担当者にとって役立つ情報はここに記載しておいてください。このような情報は完全なものである必要があり、全文をここに記載するか、完全な情報が記載されている関連項目を参照先として記載してください。

--- ACTIVITY NOTES ---

ACTIVITY NOTES には、デバイス固有の振る舞いや特殊な動作など、アクティビティスクリプトの管理担当者にとって役立つ情報を記載します。

例: "The activity only works for Cisco IOS devices".

必須関数:

CatTools がアクティビティを実行するために、カスタムアクティビティメインスクリプト内で指定しなければならない必須関数は次の関数のみです。

```
Function Main()
```

アクティビティメインスクリプトに独自の関数やサブルーチンを定義することもできますが、それらは Function Main()内から呼び出す必要があります。独自の関数やサブルーチンにはコメントを必ず付けておき、後で自分自身や他の開発者が見たときにその関数が何であるのか理解できるようにしておくことは大切です。

スクリプトファイルの保存

作成したスクリプトファイルに CatTools がアクセスできるようにするには、スクリプトファイルを¥Scripts フォルダに保存する必要があります。

通常、スクリプトファイルのコードを変更した際にその変更部分を読み込むために CatTools を停止させ再起動する必要はないのですが、テスト時に変更部分が認識されていないため余儀なく再起動しなければならないことも少なくありません。

カスタムアクティビティスクリプトのテスト

スクリプトの設定が終わったら、CatTools で新しいアクティビティを既存のデバイスを使って実行するよう設定し、正常に動作するかどうかをテストしてください。

選択したデバイスには対応する.custom ファイルが作成されている必要があります、アクティビティの実行に関係する必要なコードが記述されていなければなりません。必要なコードが含まれていないと、[Info Log]タブに次のようなエラーが表示されます。

```
"Client script error: Type Mismatch: [activity name] on line:  
[line number]"
```

詳細については、「カスタムアクティビティのテスト」の項を参照してください。

有効なスクリプトのご提供のお願い

カスタムアクティビティを CatTools でサポート可能にできた場合、作成したカスタムクライアントスクリプトファイル(.txt)とメインスクリプトファイル(.txt)、カスタムアクティビティタイプファイル(.ini)、カスタムアクティビティを参照するデバイスの.custom ファイルとともに弊社まで送ってください。今後、製品付属の定義済みアクティビティタイプとして扱わせていただくことを検討いたします。

ます。弊社 Web サイトの[製品サポートページ](#)にあるフォームを使用して弊社まで添付ファイルにて送信してください。

提供していただいたアクティビティに対する需要、スクリプトの複雑性、利用可能な技術情報源など検討項目は多岐にわたるため、そのアクティビティを CatTools の定義済みアクティビティタイプとして追加するかどうか、追加する場合はその時期に関しては何の保証もいたしかねます。しかし、送信していただいたスクリプトはすべて将来の参考用として分類・保管させていただきます。

4.5.4 カスタムアクティビティクライアントスクリプトファイル(.txt)

CatTools のアクティビティスクリプト

CatTools のアクティビティスクリプトはテキストファイル(.txt ファイル)を使って定義し、CatTools のルートディレクトリの下にある¥Scripts フォルダに保存します。

アクティビティが実行される際、CatTools はアクティビティタイプ.ini ファイルの[Activity]項目の *script_client* に指定されている値を読み込み、どのアクティビティクライアントスクリプトファイルを使用すべきかを判断します。

(.ini ファイルと.txt ファイルの名前を、.txt ファイルには Client という接頭辞を付ける以外まったく同じにすることで混乱を避けることができます)

CatTools の定義済みアクティビティタイプにはすべてアクティビティクライアントスクリプトファイル(.txt)が関連付けられており、それらのファイルは暗号化されています。暗号化されている理由は2つあります。まず、弊社の知的財産権を保護することです。もう1つは、実行時にスクリプトが動作しないという事態を招くおそれのある無許可の変更を防ぐためです。

カスタムアクティビティクライアントスクリプトファイル(.txt)のテンプレート

CatTools にはカスタムアクティビティクライアントスクリプトを新しく作成する際に、作成元として活用できるテンプレートファイルが付属しています。

テンプレートファイルの名前は Client.Custom.Activity.Template.txt で、CatTools のルートディレクトリの下に¥Templates というフォルダの中にあります。

このテンプレートファイルは、CatTools で予め定義されているアクティビティスクリプトの一部として付属しているものですが、暗号化はされていません。CatTools のすべての定義済みアクティビティスクリプトファイルは、CatTools

製品の新しいバージョンが発行される際に随時(必要であれば)更新されます。このため、新しくカスタムアクティビティクライアントスクリプトを作成する目的でテンプレートファイル自体を変更することは絶対にしないでください。テンプレートを元に新しいカスタムアクティビティクライアントスクリプトを作成する必要がある場合は、テンプレートファイルをコピーして¥Scripts フォルダに別の名前を付けて保存するようにしてください。

カスタムアクティビティクライアントスクリプトファイル(.txt)の作成

CatTools にカスタムアクティビティクライアントスクリプトを新しく追加するとき、最初にするのは¥Templates フォルダの下にあるテンプレートファイル Client.Custom.Activity.Template.txt をコピーして¥Scripts フォルダの下に別の名前を付けて保存することです。このとき、アクティビティタイプ.ini ファイルの [Activity]項目で *script_client* に指定したものと同一名前で保存してください。

このファイルを作成元として新しいデバイススクリプトファイルを定義します。

クライアントスクリプトファイル(.txt)の編集

次に、カスタムクライアントスクリプト.txt ファイルを開き、必要であればご使用のデバイスに合わせてスクリプトが動作するよう、各項目の値を変更します。

.txt ファイルには、変更する際の手順やヒントなど多くのコメントが書かれていますが、注意しなければならない重要な項目はいくつかしかなく、それらはスクリプトファイルの冒頭に書かれています。

スクリプトの1行目に次のように記述されていることがあります。

```
Attribute VB_Name = "Act_Custom_Template"
```

これは Visual Basic(あるいは同等のスクリプト言語)開発環境で表示したときに、そのカスタムアクティビティクライアントスクリプトファイルを示す Visual Basic モジュール名となります。この行は VB 開発環境でファイルの編集を行っているときは表示されませんが、他のエディターでは表示される場合があります。"Act_Custom_Template" はカスタムテンプレートファイルのモジュールに付けられた名前です。カスタムアクティビティクライアントスクリプトファイルがたくさんある場合は、この名前を作成したカスタムアクティビティクライアントスクリプトのファイル名を反映した名前に変更するほうが良いこともあります。そのようにすることによって、複数のカスタムアクティビティクライアントスクリプトファイルをご使用の VB 開発環境で同時に(たとえば、1つのプロジェクト

内などで)開くことができるようになります。

スクリプトの編集にテキストベースのエディター(メモ帳など)を使用している場合は、モジュール名の変更は任意で構いません。しかし、混乱を避けるには、この名前を変更しておいたほうが良いでしょう。

メモ:

--- SCRIPT NOTES ---

SCRIPT NOTES には、有用な情報とスクリプト作成の初期段階において実行すべき作業について記載されています。ここに書かれている内容は必要に応じて適宜書き換えて構いません。たとえば、スクリプト設定の初期に必要な作業を終え、もう再びその作業をする必要がない場合、SCRIPT NOTES に記載されている、それらに関するテキストを全部削除することができます。複雑なルーチンやスクリプト固有の振る舞いなど、デバイススクリプトの管理担当者にとって役立つ情報はここに記載しておいてください。このような情報は完全なものである必要があり、全文をここに記載するか、完全な情報が記載されている関連項目を参照先として記載してください。

--- ACTIVITY NOTES ---

ACTIVITY NOTES には、デバイス固有の振る舞いや特殊な動作など、アクティビティスクリプトの管理担当者にとって役立つ情報を記載します。

例: "The activity allows 3 Options:..." (続けて各オプションの意味と入力値の一覧を記載)

必須関数:

CatTools がアクティビティを実行するために、カスタムアクティビティクライアントスクリプト内で指定しなければならない必須関数は次の関数のみです。

```
Function Client()
```

アクティビティクライアントスクリプトに独自の関数やサブルーチンを定義することもできますが、それらは Function Main()内から呼び出す必要があります。独自の関数やサブルーチンにはコメントを必ず付けておき、後で自分自身や他の開発者が見たときにその関数が何であるのか理解できるようにしておくことは大切です。

スクリプトファイルの保存

作成したスクリプトファイルに CatTools がアクセスできるようにするには、スクリプトファイルを¥Scripts フォルダに保存する必要があります。通常、スクリプトファイルのコードを変更した際にその変更部分を読み込むために CatTools を停止させ再起動する必要はないのですが、テスト時に変更部分が認識されていないため余儀なく再起動しなければならないことも少なくありません。

カスタムアクティビティスクリプトのテスト

スクリプトの設定が終わったら、CatTools で新しいアクティビティを既存のデバイスを使って実行するよう設定し、正常に動作するかどうかをテストしてください。

選択したデバイスには対応する.custom ファイルが作成されている必要があります。アクティビティの実行に関係する必要なコードが記述されていなければなりません。必要なコードが含まれていないと、[Info Log]タブに次のようなエラーが表示されます。

```
"Client script error: Type Mismatch: [activity name] on line: [line number]"
```

詳細については、「カスタムアクティビティのテスト」の項を参照してください。

有効なスクリプトのご提供のお願い

カスタムアクティビティを CatTools でサポート可能にできた場合、作成したカスタムクライアントスクリプトファイル(.txt)とメインスクリプトファイル(.txt)、カスタムアクティビティタイプファイル(.ini)、カスタムアクティビティを参照するデバイスの.custom ファイルとともに弊社まで送ってください。今後、製品付属の定義済みアクティビティタイプとして扱わせていただくことを検討いたします。弊社 Web サイトの[製品サポートページ](#)にあるフォームを使用して弊社まで添付ファイルにて送信してください。

提供していただいたアクティビティに対する需要、スクリプトの複雑性、利用可能な技術情報源など検討項目は多岐にわたるため、そのアクティビティを CatTools の定義済みアクティビティタイプとして追加するかどうか、追加する場合はその時期に関しては何の保証もいたしかねます。しかし、送信していただいたスクリプトはすべて将来の参考用として分類・保管させていただきます。

4.5.5 カスタムアクティビティデバイススクリプトファイル(.custom)

CatTools のデバイススクリプト

CatTools のデバイススクリプトはテキストファイル(.txt ファイル)を使って定義し、CatTools のルートディレクトリの下にある¥Scripts フォルダに保存します。

アクティビティが実行される際、CatTools はデバイスタイプ.ini ファイルの [device]項目の name に指定されている値を読み込み、どのデバイススクリプトファイル(.txt)を使用すべきかを判断します。したがって、それぞれの.txt ファイルは、対応する.ini ファイルの name に指定されている名前と同じファイル名が付けられていなければなりません。

CatTools の定義済みデバイスタイプにはすべてスクリプトファイル(.txt)が関連付けられており、それらのファイルは暗号化されています。暗号化されている理由は2つあります。まず、弊社の知的財産権を保護することです。もう1つは、実行時にスクリプトが動作しないという事態を招くおそれのある無許可の変更を防ぐためです。

デバイス.custom スクリプト

CatTools へのカスタムアクティビティスクリプトの導入に伴い、カスタムアクティビティを定義済みデバイスタイプに追加する作業を簡略化するために、現バージョンの CatTools では暗号化されているデバイススクリプトファイル(.txt)をロードした後で関連付けられているデバイス.custom ファイルが¥Scripts フォルダの下にあるかどうかをチェックします。

アクティビティの実行時にデバイスタイプに関連付けられているデバイス.custom ファイルが見つかると、.custom ファイルの内容が読み込まれ定義済みデバイスタイプの暗号化済みスクリプトの内容に追加されます。このファイルは拡張デバイスタイプスクリプトの一時ファイルとして保存され、アクティビティに適用されます。デバイス.custom ファイルと暗号化されているデバイスファイルはアクティビティが実行されるたびにこのようにマージされます。

デバイス.custom ファイルはその暗号化スクリプトにファイル名で関連付けられているため、例えば、Cisco 製ルーター(Cisco.Router.General.txt)の.custom ファイルを追加するには、¥Scripts フォルダの下に Cisco.Router.General.txt.custom というファイルを作成する必要があります。

CatTools をアップグレードしたりシステムに再インストールするときは、必ず定義済みデバイスタイプとアクティビティタイプのスクリプトファイルを上書き

する必要があります。デバイス上でカスタムアクティビティを呼び出すことを別の.custom ファイルで定義しておけば、作成したカスタム関数が不注意で壊れたり失われるのを避けることができます。

デバイス.custom テンプレート

CatTools にはカスタムアクティビティデバイスファイル(.custom)を新しく作成する際に、作成元として活用できるテンプレートファイルが付属しています。

テンプレートファイルの名前は Custom.Device.Template.txt.custom で、CatTools のルートディレクトリの下に¥Templates というフォルダの中にあります。

このテンプレートファイルは、CatTools で予め定義されているスクリプトの一部として付属しているものですが、暗号化はされていません。CatTools のすべての定義済みスクリプトは、CatTools 製品の新バージョンが発行される際に随時(必要であれば)更新されます。このため、新しくデバイス.custom スクリプトを作成する目的でテンプレートファイル自体を変更することは絶対にしないでください。テンプレートを元に新しいデバイス.custom スクリプトを作成する必要がある場合は、テンプレートファイルをコピーして¥Scripts フォルダに別の名前を付けて保存するようにしてください。

デバイス.custom スクリプトファイルの作成

CatTools に.custom スクリプトを新しく追加するとき、最初にするのは ¥Templates フォルダの下にあるテンプレートファイル Custom.Device.Template.txt.custom をコピーして¥Scripts フォルダの下に別の名前を付けて保存することです。このとき、関連付けられている暗号化済みデバイススクリプトファイルと同じファイル名を付け、ファイル拡張子.txt とさらに.custom という拡張子を付けてファイルを保存する必要があります。

このファイルを作成元として新しいデバイススクリプトファイルを定義します。

例: 「Custom_Report_PortStatus」というカスタムアクティビティを作成し、このアクティビティを Cisco IOS スイッチで実行したい場合

カスタムアクティビティクライアントスクリプトファイルである Client.Custom.Report.PortStatus.txt は、コマンドを送信してポートデータをキャプチャし、結果を解析してアクティビティクライアントスクリプトに戻すという一連の処理を実行するために、関数 Custom_Report_GetPortStatus()を呼び出す必要があります。

このカスタムアクティビティ(デバイスタイプスクリプトファイル Cisco.Switch.IOS.txt)を Cisco IOS スイッチに追加するには、次の作業を行う必要があります。

- 1) Cisco IOS スイッチデバイスタイプに対する .custom ファイルを作成します。¥Templates フォルダの下にある *Custom.Device.Template.txt.custom* をコピーして¥Scripts フォルダの下に別の名前を付けて保存します。このとき、実行時に *Cisco.Switch.IOS.txt* スクリプトとマージさせるために *Cisco.Switch.IOS.txt.custom* という名前で保存する必要があります。
- 2) .custom ファイルを開き、サンプル関数を削除します。
- 3) カスタムアクティビティ関数 *Custom_Report_GetPortStatus()* とコードを記述します。

最初は関数が正常に呼び出されるかどうかだけをテストするために、次のような単純なコードを関数内に追加して[Info Log]タブに情報メッセージを表示するよう定義します。

```
Function Custom_Report_GetPortStatus()  
    cl.Log 3, "Testing custom activity: Custom_Report_PortStatus"  
End Function
```

- 4) ファイルを保存します。

.custom ファイルの編集

.custom スクリプトファイルの編集に関して、基本的に気をつけなければならない点は3つしかありません。¥Templates フォルダの下にある .custom テンプレートファイルをコピーし¥Scripts フォルダの下に保存したら次のことを確認してください。

- 1) テンプレートファイルの1行目を削除しないでください。

```
Attribute VB_Name = "Dev_CustomDeviceTemplate_custom"
```

" "内の値は変更できますが、必ず「Attribute VB_Name」として定義してください。この部分を変更すると、実行時にエラーが発生する可能性が高くなってしまいます。

- 2) ヘッダー部分にはコメントなしのコード(プログラムコード)を入力しな

いでください。また、上記の Attribute VB_Name 行を重複して記述することは絶対にしないでください。重複して記述すると、コードがストリップアウトしたり実行時にエラーが発生することがあります。.custom ファイルのヘッダー部分にある SCRIPT NOTES には自分自身にとって役に立つ情報などを記載しておきます。後で見たときや他の開発者にとって有用であると思われることはコメントとしてここに自由に記載してください。

3) 記述するプログラムコードはすべて関数またはサブルーチンの中に含まれるようにする必要があります。実行時にエラーが発生したり、定義済み暗号化デバイススクリプトに使用されている変数やオブジェクトを再定義するのを避けるためには、変数およびオブジェクトの宣言を関数またはサブルーチンの範囲内に限ると良いでしょう。独自の関数やサブルーチンにはコメントを必ず付けておき、後で自分自身や他の開発者が見たときにその関数が何であるのか理解できるようにしておくことは大切です。

.custom スクリプトファイルの保存

作成した.custom スクリプトファイルに CatTools がアクセスできるようにするには、スクリプトファイルを¥Scripts フォルダに保存する必要があります。

通常、スクリプトファイルのコードを変更した際にその変更部分を読み込むために CatTools を停止させ再起動する必要はないのですが、テスト時に変更部分が認識されていないため余儀なく再起動しなければならないことも少なくありません。

カスタムデバイススクリプト(.custom)のテスト

スクリプトの設定が終わったら、CatTools で.custom ファイルに関連付けられているデバイスを使ってカスタムアクティビティを実行するよう設定し、正常に動作するかどうかをテストしてください。

詳細については、「カスタムデバイスのテスト」の項を参照してください。

有効なスクリプトのご提供のお願い

カスタムデバイスを CatTools でサポート可能にできた場合、作成したカスタムクライアントスクリプトファイル(.txt)とカスタムデバイスタイプファイル(.ini)を弊社まで送ってください。今後、製品付属の定義済みアクティビティタイプとして扱わせていただくことを検討いたします。弊社 Web サイトの[製品サポートページ](#)にあるフォームを使用して弊社まで添付ファイルにて送信してください。

提供していただいたデバイスに対する需要、スクリプトの複雑性、利用可能な技術情報源など検討項目は多岐にわたるため、そのデバイスを CatTools の定義済みアクティビティタイプとして追加するかどうか、追加する場合はその時期に関しては何の保証もいたしかねます。しかし、送信していただいたスクリプトはすべて将来の参考用として分類・保管させていただきます。

4.5.6 カスタムアクティビティスクリプトに使用でき cl./ct.変数および関数

カスタムアクティビティクライアントスクリプトおよびメインスクリプトのファイル内には CatTools の内部変数、関数およびサブルーチンに対する参照が記述されています。これらの変数や関数、サブルーチンには「cl.」(CatTools クライアントプログラム)あるいは「ct.」(CatTools メインプログラム)という接頭辞が付いていますので、識別できます。

これらの関数、変数、サブルーチンはテンプレートスクリプト自体には(主にスクリプトファイルのサイズを抑制するために)記録されていませんが、それぞれの関数や変数をどのような目的で使用するのか、パラメータを必要とする関数は何か、戻り値はどのようなものがあるのかについては、カスタムアクティビティテンプレート内で使用されている変数、関数、サブルーチンの詳細を次の項にまとめましたので参照してください。

- クライアントスクリプトファイルで使用する cl.変数および関数
- メインスクリプトファイルで使用する ct.変数および関数

4.5.6.1 クライアントスクリプトファイルで使用する cl.変数および関数

CatTools のクライアントスクリプトで使用する「cl.」変数、関数、サブルーチンの一覧とカスタムアクティビティクライアントスクリプトファイルでの実装例を以下に挙げます。この一覧に記載した以外の CatTools 内で使用される内部クライアント変数/関数もありますが、スクリプトのテンプレートファイル内で使用されているものとカスタムアクティビティクライアントスクリプトを作成する際に役立つ変数/関数のいくつかについて説明してあります。

変数

cl.AppPath	CatTools のアプリケーションパス(インストール先ディレクトリ)です。
cl.ScheduleNumber	現在のスケジュール番号です。
cl.DeviceName	現在の接続先デバイス名です。

cl.TelnetConnectionStatus (プロトコルエンジンコントロール)クライアントスレッドのインスタンスに対する接続ステータスです。

例: 現在のクライアント接続ステータスをチェックし、ステータスが「OK」でない場合は関数を終了させる場合

```
If StrComp(cl.TelnetConnectionStatus, "OK", vbTextCompare) <> 0 Then Exit Function
```

cl.QuitNow アクティビティを停止するために複数の箇所でチェックされる変数です。(例えば)アクティビティの実行中に[Stop]ボタンが押されると、1に設定されます。この変数はアクティビティクライアントスクリプト内でのみ参照可能です。メインスクリプトを参照する場合はct.QuitNow変数を使用してください。

代表的な関数およびサブルーチン

汎用およびファイル関連

cl.Log	Infolog.txt ファイルにテキスト行を送信します。この情報は[Info Log]タブに表示されます。
cl.LogToFile	データをファイルに書き出します。
cl.FileExists	ファイルが存在するかどうかをチェックします。
cl.CreateFile	指定したファイル名とパスで新しいファイルを作成します。
cl.DeleteFile	指定したファイル名とパスのファイルを削除します。
cl.ReadFromFile	指定したファイルの内容を読み込みます。
cl.ReplaceClientFilenameVariables	クライアントレポートのファイル名に書かれているファイル名変数またはメタ変数を実際のテキスト値に置換します。
cl.SaveResults	結果を一時ファイル(.txt)に書き出し¥ClientTemp フォルダの下に保存します。

アクティビティ関連

cl.DBCheckScheduleOption	指定したアクティビティ内で特定のオプションが選択されているかどうかを確認します。
cl.DBScheduleGetField	CatTools データベースからアクティビティフィールド値を取得します。

デバイス関連

cl.DisconnectHost	(プロトコルエンジンコントロール)クライアントスレッドのインスタンスをエンドデバイスから切断します。
-------------------	--

関数およびサブルーチンの詳細およびコード例

ここでは、上記の関数およびサブルーチンについて入力パラメータやコード例なども織り交ぜながらさらに詳しく説明します。

cl.Log (iPriority、 sMessage)

このサブルーチンはテキスト行を Infolog.txt ファイルに送信します。この情報は [Info Log] ペインに表示されます。入力パラメータは2つです。

<i>iPriority</i>	1~4 の整数で送信される行のロギングレベルを指定します。1=エラー、2=警告、3=情報、4=デバッグ
<i>sMessage</i>	送信されるテキストメッセージを指定します。

例:アクティビティの[Options]タブにある[Use alternate command]フィールドに指定がなく、レベル3(情報)のメッセージを出力する

```
Dim sAltCommand

sAltCommand = cl.DBScheduleGetField(cl.ScheduleNumber,
"OptionsString1")
If Len(Trim(sAltCommand)) = 0 Then
    cl.Log 3, "No alternate command specified"
End if
```

cl.LogToFile (sFilename, sData, bAppend)

このサブルーチンは指定したファイルにデータを書き込みます。入力パラメータは3つです。

<i>sFileName</i>	書き込むファイルの名前とパスを文字列で指定します。
<i>sData</i>	書き込むデータを文字列で指定します。
<i>bAppend</i>	ブール値で指定します。この値が 0 以外(すなわち False 以外)に指定されているときファイルに情報が追記されます。

例:アクティビティの[Options]タブにある[Use alternate command]フィールドに指定がない場合にレベル 2(警告)のメッセージを[Info Log]に出力し、¥Reports フォルダにあるファイルにそのテキストを追記する

```
Dim sLogResultsFile
Dim sAltCommand
sLogResultsFile = cl.AppPath & "Reports¥" & cl.UniqueFileID
& ".txt"
sAltCommand = cl.DBScheduleGetField(cl.ScheduleNumber,
"OptionsString1")
If Len(Trim(sAltCommand)) = 0 Then
    cl.Log 2, "No alternative command specified"
    Call cl.LogToFile(sLogResultsFile, "No alternative
command specified in activity Options tab", 1)
End if
```

cl.FileExists(sFileName) As Boolean

この関数は指定したファイルが存在するかどうかをチェックします。ファイルが見つかった場合は True、見つからなかった場合は False が返されます。

入力パラメータは 1 つです。

<i>sFileName</i>	存在するかどうか確認するファイルのファイル名とパスを文字列で指定します。
------------------	--------------------------------------

cl.CreateFile(sFileName) As Boolean

この関数は指定したファイル名とパスで新しいファイルを作成します。ファイルが既に存在している場合またはファイル作成が成功した場合は True、ファイルが作成できず既存のファイルも見つからない場合は False が返されます。

入力パラメータは 1 つです。

sFileName 作成するファイルのファイル名とパスを文字列で指定します。

`cl.DeleteFile(sFileName) As Boolean`

この関数は指定したファイル名とパスのファイルを削除します。ファイル削除が成功した場合は True、ファイルが削除できなかった場合は False が返されます。

入力パラメータは 1 つです。

sFileName 削除するファイルのファイル名とパスを文字列で指定します。

`cl.ReadFromFile(sFileName) As String`

この関数は指定したファイルの内容を読み込みます。ファイルデータが文字列で返されます。CatTools 関数の `cl.ReadFromFile` と同様の処理を実行しますが、アクティビティクライアントスクリプト内からのみ呼び出すことができます。

入力パラメータは 1 つです。

sFileName 内容を読み込むファイルのファイル名とパスを文字列で指定します。

例: 指定したファイル (この場合は CLI コマンドリスト)の内容を読み込む

```
Dim sFileName
Dim sCommandList
If cl.DBCheckScheduleOption(cl.ScheduleNumber, 1) = 0 Then
    ' Option selected to load CLI commands to send to device
    from a file
    sFileName = cl.DBScheduleGetField(cl.ScheduleNumber,
    "OptionsString2")
    sCommandList = cl.ReadFromFile(sFileName)
End If
```

`cl.ReplaceClientFilenameVariables(sData) As String`

この関数はクライアントレポートファイルに記述されているファイル名変数またはデバイスに送信されるコマンド内に記述されているメタ変数を実際のテキスト値に置換します。置換後の変更された文字列が戻されます。

入力パラメータは1つです。

sData ファイル名とパス、あるいはチェック対象のコマンドリストを文字列で指定します。

例:指定したファイルのファイル名変数を置換し、その内容（この場合は CLI コマンドリスト)の内容を読み込む。次に、コマンドリストに含まれているメタ変数を置換する

```
Dim sFileName
Dim sCommandList
If cl.DBCheckScheduleOption(cl.ScheduleNumber, 1) = 0 Then
    ' Option selected to load CLI commands to send to device from a
    file
    sFileName = cl.DBScheduleGetField(cl.ScheduleNumber,
    "OptionsString2")
    ' Replace any filename variables
    sFileName = cl.ReplaceClientFilenameVariables(sFileName)
    ' Read in the file contents
    sCommandList = cl.ReadFromFile(sFileName)
    ' Replace meta (command) variables with values
    sCommandList = cl.ReplaceClientFilenameVariables(sCommandList)
End If
```

`cl.SaveResults(sData, bAppend)`

このサブルーチンは結果を一時ファイル(.txt)に書き出し¥ClientTemp フォルダの下に保存します。

入力パラメータは2つです。

sData 書き出すデータを文字列で指定します。
bAppend ブール値で指定します。0(False)を指定するとファイルは上書きされ、1(True)を指定するとファイルに文字列が追記されます。

例:現在のデバイスの応答(sResults)を一時ファイル(.txt)に書き出し、その際既存のデータを上書きしてファイルを保存する

```
Call cl.SaveResults(sResults, 0)
```

cl.DBCheckScheduleOption(IScheduleNumber, IOptionNumber) As Long

この関数は指定したアクティビティ内で特定のオプションが選択されているかどうかを確認します。

入力パラメータは2つです。

<i>IScheduleNumber</i>	現在のスケジュール番号を Long で指定します。
<i>IOptionNumber</i>	チェック対象のアクティビティ内のオプションを Long で指定します。IOptionNumber は 1~10 までの値で指定する必要があります。

例:アクティビティ(この場合は Device.CLI.Send commands アクティビティ)をチェックし、コマンドの出力結果を電子メール送信するかどうかを確認する

```
lRetVal = cl.DBCheckScheduleOption(cl.ScheduleNumber, 8)
If lRetVal = 1 then
    ' code to email commands goes here...
End if
```

cl.DBScheduleGetField(IScheduleNumber, sFieldName)

この関数は CatTools データベースから指定したアクティビティのフィールド値を取得します。CatTools 関数の ct.DBScheduleGetField と同様の処理を実行しますが、アクティビティクライアントスクリプト内からのみ呼び出すことができます。

入力パラメータは2つです。

<i>IScheduleNumber</i>	現在のスケジュール番号を Long で指定します。
<i>sFieldName</i>	データベースに登録されている、値を取得したいフィールドの名前を文字列で指定します。

例: (上記の cl.DBCheckScheduleOption 関数と併用)アクティビティの[Options]タブにある[Use alternate command]のチェックがオンになっているかどうかチェックする。オンになっているとき、データベースから送信したい代替コマンドを取得する

```
'Check the option 1 check-box to see if selected - i.e. set to 1
If cl.DBCheckScheduleOption(cl.ScheduleNumber, 1) = 1 Then
    ' Get value from database for associated string field (being field
    "OptionsString1" in the database) to set a variable value
```

```
sAltCommand = cl.DBScheduleGetField(cl.ScheduleNumber,  
"OptionsString1")  
End If
```

cl.DisconnectHost

このサブルーチンは(プロトコルエンジンコントロール)クライアントスレッドのインスタンスを伝度デバイスから接続します。

クライアントスレッドから完全に切り離すためにデバイスをログアウトしてから呼び出す必要があります。

4.5.6.2 メインスクリプトファイルで使用する ct.変数および関数

CatTools のメインスクリプトで使用する「ct.」変数、関数、サブルーチンの一覧とカスタムアクティビティメインスクリプトファイルでの実装例を以下に挙げます。この一覧に記載した以外の CatTools 内で使用される内部 ct.変数/関数もありますが、メインスクリプトのテンプレートファイル内で使用されているものとカスタムアクティビティメインスクリプトを作成する際に役立つ変数/関数のいくつかについて説明してあります。

変数

ct.ScheduleNumber	現在のスケジュール番号です。
ct.Devices	接続先デバイスを文字列で指定します。複数あるときはパイプ文字()で区切ります(現在のスケジュールに関連付けられている複数のデバイスを指定するときなどに使用します)
ct.ScriptFileClient	アクティビティによって使用されるクライアントスクリプトファイルのファイル名を保存するための文字列変数です。
ct.QuitNow	アクティビティを停止するために複数の箇所でチェックされる変数です。(例えば)アクティビティの実行中に[Stop]ボタンが押されると、1 に設定されません。この変数はアクティビティメインスクリプト内でのみ参照可能です。クライアントスクリプトを参照する場合は cl.QuitNow 変数を使用してください。

代表的な関数およびサブルーチン

汎用およびファイル関連

ct.Log Infolog.txt	ファイルにテキスト行を送信します。この情報は[Info Log]タブに表示されます。
ct.ReadFromFile	指定したファイルの内容を読み込みます。
ct.ReplaceFilenameVariables	レポートのファイル名に書かれているファイル名変数を実際のテキスト値に置換します。
ct.CreateMasterTable	マスターテーブルファイルを作成します。
ct.CreateAndSendReport	レポートファイル(.txt、.html 等)を作成し、電子メールで送信します。
ct.RemoveClientResults	¥ClientTemp フォルダにある既存のクライアント一時ファイル(.txt)を削除します。

アクティビティ

ct.MarshalThreads	アクティビティスケジュール内で使用されているクライアントスレッドを管理します。
ct.DBScheduleGetField	CatTools データベースからアクティビティのフィールド値を取得します。

デバイス

ct.RemoveHeader	データブロックから最初の 1 行を削除します。
-----------------	-------------------------

関数およびサブルーチンの詳細およびコード例

ここでは、上記の関数およびサブルーチンについて入力パラメータやコード例なども織り交ぜながらさらに詳しく説明します。

ct.Log(sDevice, iPriority, sMessage)

このサブルーチンはテキスト行を Infolog.txt ファイルに送信します。この情報は [Info Log] タブに表示されます。

入力パラメータは 3 つです。

<i>sDevice</i>	デバイス名を文字列で指定します。空の値("")が指定されている(あるいは VBNulString が指定されている)場合、「Kiwi CatTools」となりデバイス固有のメッセージではなく内部アプリケーションが関連付けられて
----------------	---

	いるログメッセージを参照します。
<i>iPriority</i>	1~4 の整数で送信される行のロギングレベルを指定します。1=エラー、2=警告、3=情報、4=デバッグ
<i>sMessage</i>	送信されるテキストメッセージを指定します。

例:特定のデバイスではなく Kiwi CatTools に対して(sDevice = "")レベル 1(エラー)のメッセージ"No devices have been selected"を出力する

```
ct.Log "", 1, "No devices have been selected"
```

ct.ReadFile(sFileName) As String

この関数は指定したファイルの内容を読み込みます。ファイルデータが文字列で返されます。CatTools 関数の *cl.ReadFile* と同様の処理を実行しますが、アクティビティメインスクリプト内からのみ呼び出すことができます。

入力パラメータは1つです。

<i>sFileName</i>	内容を読み込むファイルのファイル名とパスを文字列で指定します。
------------------	---------------------------------

例:「ReportData」というファイルの内容を読み込み、ヘッダーの1行目を削除する場合

```
Dim sReportFile
Dim sReportData

' Get the file name and path from the activity settings and replace any filename variables.
sReportFile = ct.DBScheduleGetField(ct.ScheduleNumber,
  "ReportFile")
sReportFile = ct.ReplaceFilenameVariables("", ReportFile)

' Read the contents of the report file, remove the first line of data (header row) and save
the results to sReportData variable.
sReportData = ct.RemoveHeader(ct.ReadFile(sReportFile))
```

ct.ReplaceFilenameVariables(sDeviceName, sFileName) As String

この関数はレポートファイル名に記述されているファイル名変数を実際のテキスト値に置換します。置換後の変更された文字列が戻されます。

入力パラメータは2つです。

<i>sDeviceName</i>	デバイス名を文字列で指定します。sDeviceName に空("")の値を指定すると、アクティビティのレポートファイル名にデバイス固有のファイル名変数が含まれていないと解釈されます。
<i>sFileName</i>	チェック対象のファイル名およびパスを文字列で指定します。

例:アクティビティのレポートファイル名を取得し、ファイル名変数の置換を実行し、その値を文字列で保存する場合

```
sReportFile = ct.DBScheduleGetField(ct.ScheduleNumber,
"ReportFile")
sReportFile = ct.ReplaceFilenameVariables("", sReportFile)
```

ct.CreateMasterTable(sReportFile, sHeader, bOverwrite, sDeviceList, sClientFilePrefix, sTableFormat) As Boolean

この関数はマスターテーブルファイルを作成します。ファイルが作成されると True、作成されないと False が戻されます。

入力パラメータは6つです。

<i>sReportFile</i>	作成するレポート/マスターテーブルファイルの名前とパスを文字列で指定します。
<i>sHeader</i>	アクティビティメインスクリプトで定義されているとおりにレポートのヘッダーに表示する文字列をタブで区切って指定します。
<i>bOverwrite</i>	ブール値で指定します。0(False)を指定するとファイルは上書きされ、1(True)を指定するとファイルに文字列が追記されます。
<i>sDeviceList</i>	現在のスケジュールに関連付けられているデバイスを文字列で指定します。(ct.Devices 変数を使用して)複数のデバイスが関連付けられているときはパイプ文字()で区切ります。
<i>sClientFilePrefix</i>	あとでマスターテーブルファイルに統合するために¥ClientTemp フォルダの下にアクティビティによって作成される一時クライアント(デバイス)ファイルのファイル名接頭辞を文字列で指定します。
<i>sTableFormat</i>	テーブル定義の出力を変更できるアクティビティ(例

例えば、Report.SNMP.System Summary アクティビティの[Options]タブで列名の変更、列の追加/削除、列の並べ替えなどが行えるようなアクティビティ)で使用する文字列をパイプ文字(|)で区切って指定します。

例: まず、バージョンレポートに対するマスターテーブルを作成する。次に、レポートファイルの名前を取得してデータベースに登録されているアクティビティ設定の値を上書きする。レポートヘッダーを定義して一時クライアント(デバイス)ファイルに「CustomReportTemplate」という接頭辞をつける

```
Dim sReportFile
Dim bOverwrite
Dim sHeader

sReportFile = ct.DBScheduleGetField(ct.ScheduleNumber,
    "ReportFile")
sReportFile = ct.ReplaceFilenameVariables("", ReportFile)

bOverwrite = ct.DBScheduleGetField(ct.ScheduleNumber,
    "ReportOverwrite")

sHeader = "Group" & vbTab & "Device Name" & vbTab & "IP
Address" & vbTab & "Serial #" & vbTab & _ "Processor" & vbTab
& "IOS" & vbTab & "Uptime"

' Create the master table
bRetVal = ct.CreateMasterTable(sReportFile, sHeader,
    bOverwrite, ct.Devices, "CustomReportTemplate", "")

ct.CreateAndSendReport(sScheduleName, sHeader, sData, sReportFile)
```

この関数はレポートファイル(.txt、.html 等)を作成し、オプションで設定されている場合は電子メールで送信します。

入力パラメータは4つです。

<i>sScheduleName</i>	レポートのタイトルおよび電子メールの件名に使用するスケジュール名を指定します。
<i>sHeader</i>	アクティビティメインスクリプト内で定義されている

るとおりにレポートのヘッダーに表示する文字列をタブで区切って指定します。

sData

レポートデータを文字列で指定します。

sReportFile

作成するレポートのファイル名とパスを文字列で指定します。

例: シンプルなバージョンレポートを作成し、データベースに登録されているアクティビティ設定値からアクティビティ名とレポートファイル名を取得してレポートヘッダーを定義する

```
Dim sClientFile
Dim sScheduleName
Dim bOverwrite
Dim sReportFile
Dim sHeader

' Prefix for temporary client file filenames
sClientFile = "CustomReportTemplate"

' Get the schedule name from the activity settings
sScheduleName = ct.DBScheduleGetField(ct.ScheduleNumber,
"Name")

' Get overwrite option value
bOverwrite = ct.DBScheduleGetField(ct.ScheduleNumber,
"ReportOverwrite")

' Get report file name
sReportFile = ct.DBScheduleGetField(ct.ScheduleNumber,
"ReportFile")
sReportFile = ct.ReplaceFilenameVariables("", ReportFile)

' Define the report header
sHeader = "Group" & vbTab & "Device Name" & vbTab & "IP
Address" & vbTab & "Serial #" & vbTab & _ "Processor" & vbTab
& "IOS" & vbTab & "Uptime"

' Create the master table
Call ct.CreateMasterTable(sReportFile, sHeader, bOverwrite,
```

```
ct.Devices, sClientFile, "")

' Call CatTools function to create and send the report
Call ct.CreateAndSendReport(sScheduleName & " (Custom
Template Report)", sHeader,
ct.RemoveHeader(ct.ReadFromFile(sReportFile))
, sReportFile)

' Finally, clean up by deleting any temp client files
Call ct.RemoveClientResults(sClientFile)
```

`ct.RemoveClientResults(sClientFile)`

このサブルーチンは¥ClientTemp フォルダの下にある、sClientFile で指定した文字列で始まるクライアント一時ファイル(.txt)をすべて削除します。

入力パラメータは 1 つです。

sClientFile 検索対象のファイル名を文字列で指定します。

`ct.MarshalThreads(ByVal sDeviceList As String, ByVal sScriptFile As String, ByVal sFunctionName As String, ByVal IMaxProcessCount As Long, ByVal sUniqueFileID As String)`

このサブルーチンはアクティビティスケジュール内で使用されているクライアントスレッドを管理します。

入力パラメータは 5 つです。

sDeviceList 現在のスケジュールに関連付けられているデバイス (このパラメータは ct.Devices 変数で指定します) を文字列で指定します。複数のデバイスが関連付けられているときはパイプ文字(|)で区切ります。

sScriptFile 使用するクライアントスクリプトファイル(このパラメータは ct.ScriptFileClient 変数で指定します)の名前を文字列で指定します。

sFunctionName クライアントスクリプトファイル内で使用する関数名を文字列で指定します(通常は「Client」で始まる名前です)。

IMaxProcessCount アクティビティで使用するクライアントスレッドの

最大数を指定します(この値は CatTools のバージョンによって自動的に制御されます)

sUniqueFileID

アクティビティにより¥ClientTemp フォルダの下に作成されるクライアント(デバイス)一時ファイル(.txt)のファイル名に付ける接頭辞を文字列で指定します。

例: シンプルなバージョンレポートのクライアントスレッドを起動し、データベースに登録されているアクティビティ設定値からクライアントスレッドの最大値を取得する

```
Dim lMaxClientThreads
```

```
' Get the number of client threads from the activity settings
```

```
lMaxClientThreads = ct.DBScheduleGetField(ct.ScheduleNumber,  
"ClientThreads")
```

```
' Start the threads
```

```
Call ct.MarshalThreads(ct.Devices, ct.ScriptFileClient,  
"Client", lMaxClientThreads, "CustomReportTemplate")
```

```
ct.DBScheduleGetField(IScheduleNumber, sFieldName)
```

この関数は CatTools データベースから指定したアクティビティのフィールド値を取得します。クライアント関数 `cl.DBScheduleGetField` と同様の処理を実行しますが、アクティビティメインスクリプト内からのみ呼び出すことができます。

入力パラメータは 2 つです。

IScheduleNumber 現在のスケジュール番号を Long で指定します。

sFieldName データベースに登録されている、値を取得したいフィールドの名前を文字列で指定します。

例: アクティビティのレポートファイル名を取得し、`sReportFile` 変数に指定された文字列と同じ値を保存する場合

```
sReportFile = ct.DBScheduleGetField(ct.ScheduleNumber,  
"ReportFile")
```

```
ct.RemoveHeader(sData) As String
```

この関数は指定したデータブロックの 1 行目を削除します。この処理は `vbCrLf`

の最初のインスタンスを検索し、その行以前のデータをすべて削除することによって実現されます。削除されるデータの1行目の文字列が戻されます。

入力パラメータは1つです。

`sData` 処理するデータを文字列で指定します。

4.5.7 カスタムアクティビティのテスト

カスタムアクティビティのテストは比較的簡単な方法で実行できます。

アクティビティタイプファイル(.ini)、アクティビティメインスクリプト(.txt)、アクティビティクライアントスクリプト(.txt)を準備し、作成したカスタムアクティビティを適用するデバイス固有のコードを記述したデバイス.custom ファイルを作成したら、CatTools でアクティビティを新規作成し[Type]からテストしたいカスタムアクティビティを選択します。[Devices]タブでは.custom ファイルを作成したデバイスを選択します。

CatTools では数種類のテスト方法で試すことができます。

1) Info Log メッセージ

[Info Log]ペインではアクティビティの実行中に表示されるメッセージを確認できます。表示されるメッセージのレベルはウィンドウ下部にあるドロップダウンリストでフィルタして表示できます。「4) Show Debug events and above」を選択すると、アクティビティの実行中に表示された全ての cl.log メッセージを表示できます。

[Info Log]に表示されるメッセージは全て CatTools のルートフォルダの下にある InfoLog.txt ファイルに書き込まれます。このファイルは短期間のうちに肥大しやすく、そんなときは[File]メニューの[Delete]、[Delete Infolog.txt file]を選択して内容をクリアできます。

デバイス固有の問題のトラブルシューティングに役立つレベル 4 のメッセージを表示するためのクライアントスクリプト関数 cl.Log 4 の"message"がデバイススクリプト内のところどころに記述されています。

新しいカスタムアクティビティスクリプトのテストを実行しているときに最も頻繁に表示される可能性のある Info Log エラー(レベル 1-エラー)は次のようなエラーです。

- "Client script error: Type Mismatch: [activity name] on

```
line: [line number]"
```

このエラーは以下の条件のいずれかに当てはまるときに表示されます。

- カスタムアクティビティを実行するデバイスとして選択したデバイス用として定義された.custom ファイルが見つからないとき
- デバイスの.custom ファイルにカスタムアクティビティクライアントスクリプト内で定義された関数が記述されていないとき
- .custom ファイルに記述されている関数のコード内に誤りがあるとき

CatTools によってデバイスに関連付けられている.custom ファイルが検索されると、[Info Log]に次のようなメッセージ(レベル 4-デバッグ)が表示されます。

```
"Loading custom activities for device"
```

ご使用のデバイスに対してこのメッセージが表示されていない場合(注: [Info Log]タブで「4) Show Debug events and above」が選択されている必要があります)、ファイルが存在しないか開けない(ユーザーアカウント権限をチェックしてください)、あるいは.custom ファイルの名前を付けるときにタイプミスをした可能性があり、そのために CatTools が.custom ファイルを検索できなかったことを示しています。

上記のデバッグメッセージが表示されている場合、.custom ファイルの内容をチェックしてカスタムアクティビティ関数が記述されていること、ファイル名にタイプミスがないこと、コードに誤りがないことを確認してください(スクリプトエラーの発生点を見つけるためにコード内に計画的に cl.log 4,"..."Info Log デバッグメッセージを追加することもできます)。

選択したデバイスがカスタムアクティビティをサポートしていない場合、アクティビティの[Devices]タブで選択を外すか、そのデバイスの.custom ファイル内に記述されているカスタムアクティビティ関数に以下のコードを追加します。

```
Function [YourCustomActivityFunctionName]()  
    Call cl.CatToolsNoSupport  
End Function
```

このようにしてからアクティビティを実行すると、関数が呼び出されますが CatTools の内部関数 cl.CatToolsNoSupport によって下記のようなメッセージ(レベル 2 - 警告)が[Info Log]タブに表示されます。

```
"Device type: [SCRIPT_NAME constant] has not had this
```

```
functionality added yet. Skipping this device"
```

表示される可能性のあるエラー

- "Client script error: Variable is undefined: '[variable name]' on line: [line number]"

このエラーは以下の条件のいずれかに当てはまるときに表示されます。

- 宣言されていない変数がコード内に記述されているとき。Dim 文または Redim 文を使って変数を正しく宣言してください(変数の宣言時にタイプミスをしないよう注意)。
- 作成したコードに CatTools の内部変数/関数(cl.または ct.) に対する呼び出しや参照が記述されている、あるいは定義済みのクラス/タイプに対する呼び出しや参照が記述されているが「cl.」や「ct.」、クラス名などをタイプミスしているとき
- "Client script error: Object doesn't support this property or method: '[property or method]' on line: [line number]"

このエラーは以下の条件に当てはまるときに表示されます。

- .custom ファイル内に関数またはサブルーチンの呼び出しが記述されているが、その関数またはサブルーチンが存在しないとき(コード内にタイプミスがないかどうか確認してください)。
- "Client script error: Wrong number of arguments or invalid property assignment: '[Function or Sub name]' on line: [line number]"

このエラーは以下の条件に当てはまるときに表示されます。

- .custom ファイル内に関数またはサブルーチンの呼び出しが記述されているが、その関数またはサブルーチンの引数の数が間違っていて入力されているとき

2) デバッグログ

[File]メニューの下の[Enable Capture Mode]を選択してオンにすると、CatTools とエンドデバイス間の通信のロギングが開始され、¥Debug フォルダ内にその通信を記録するためのファイルが作成されます。この機能は、初期コネクションを

確立後にスクリプトが正常にデバイスと通信を行っていないように思えるときに非常に便利です。スクリプトによって、デバイスにコマンドを送信すると既知のプロンプトが表示されるまで待つが、期待したプロンプトが表示されない場合は最終的にタイムアウトになるように設定されています。デバッグログキャプチャファイルには実際にデバイスから何が CatTools に送信されたかが記録されているため、その内容に応じてスクリプトを修正することができます。

4.6 デバイスでサポートされていないアクティビティ

対応するデバイスタイプは日々増えています。

CatTools で現在使用可能なデバイスタイプおよび各デバイスタイプで対応しているアクティビティの最新リストについては、弊社 Web サイトの[デバイス対応表](#)を参照してください。

ご使用のデバイスが一覧に記載されているが、必要なアクティビティが現在そのデバイスタイプではサポート対象外となっている場合は、[製品サポートフォーム](#)を使ってできるだけ詳しい情報と可能ならば PuTTY で関連コマンドをキャプチャして送ってください。

5 メニュー

CatTools のメインメニューに表示されているメニュー項目は次のとおりです。

- File
- View
- Options
- Interface
- Help

5.1 [File]

CatTools の[File]メニューから表示されるメニュー項目は次のとおりです。

- Database
- Delete
- Enable capture mode
- Debug
- Exit

5.1.1 [Database]

CatTools データベース上では数多くのオペレーションを実行できます。

5.1.1.1 [Export]

アクティビティやデバイスの設定内容はフラットファイルにエクスポートできます。

5.1.1.1.1 [Export devices to tab delimited file]

このオプションをクリックすると、デバイス情報がタブ区切りのファイルでエクスポートされます(MS-Excel 等でレポートを作成するときに適しています)。

通常のファイル選択画面が表示されますので、作成するファイルの保存先フォルダを探しファイル名を入力してください。

5.1.1.1.2 [Export activities to tab delimited file]

このオプションをクリックするとアクティビティデータがタブ区切りのファイルでエクスポートされます(MS-Excel 等でレポートを作成するときに適しています)。

通常のファイル選択画面が表示されますので、作成するファイルの保存先フォルダを探し、ファイル名を入力してください。

5.1.1.2 [Import]

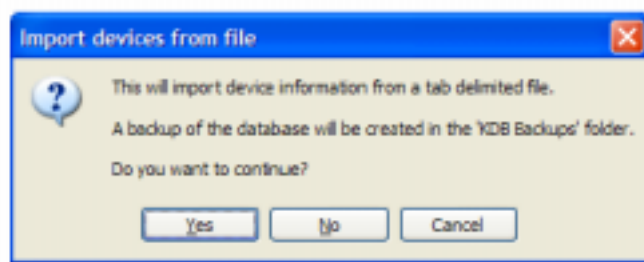
アクティビティやデバイスの設定内容はフラットファイルからインポートできます。

5.1.1.2.1 [Import devices from tab delimited file]

このメニューを使用すると、タブ区切りのファイルからデバイス情報をインポートできます。

現在のデータベースはインポートが実行される前にバックアップされます。

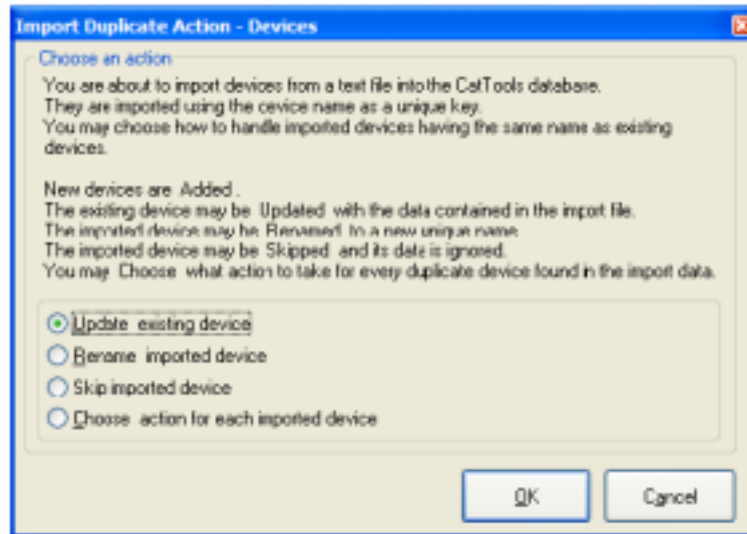
このオプションをクリックするとインポート操作を継続するかどうかを確認する画面が表示されます。



[はい]をクリックすると続きます。[いいえ]または[キャンセル]をクリックするとインポート操作を中断できます。

[はい]をクリックしたら、インポートするファイルを選択します。ファイル名に「Export_Device.txt」というデフォルト値が入力されています。インポートするファイルを探し、[開く]をクリックします。

インポートするデバイスが既存のデバイスと同じ名前の場合にどのように処理するかを指定するオプションが表示されます。

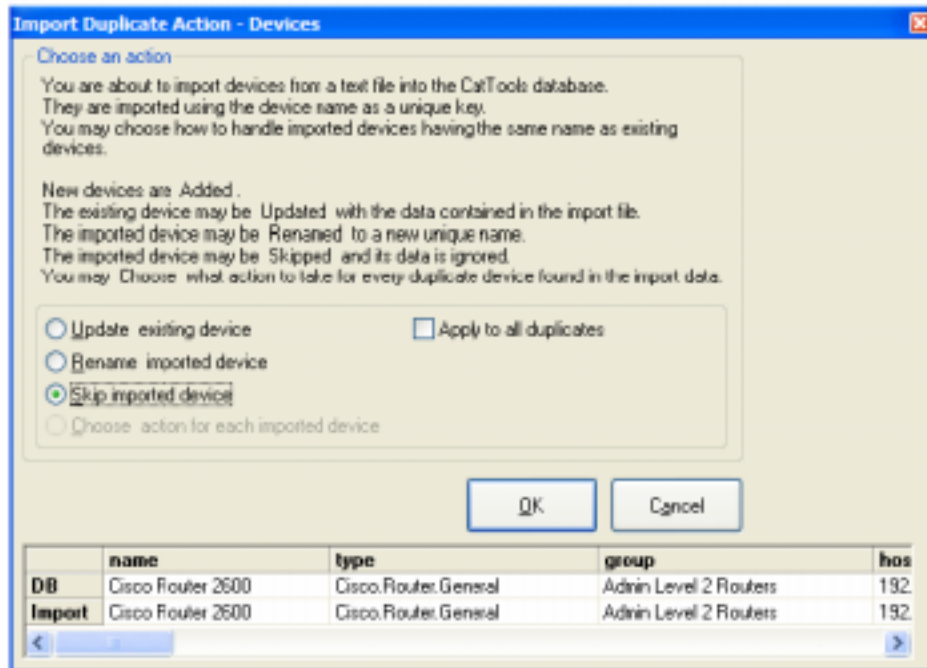


インポート処理を実行するとその度に新しいデバイスが追加されます。

重複したデバイスが見つかったときに適用する処理を選択します。

- [Update existing device with the imported device's data]現在のデバイスデータのすべてのフィールドがインポートするデータで上書きされます。インポートするデータ内に見つからなかったフィールドはそのままの状態に変更されません。
- [Rename imported device]現在のデバイス名の末尾に「_imported_0」という文字が追加された新しいデバイスが作成されます。同じ名前のデバイスが複数あるときは最後の数字が1ずつ増していきます。
- [Skip imported device]インポートされるデバイスデータが無視されます。

インポートするデバイスごとにアクションを選択します。重複する名称のデバイスが検索されたときは常に次の画面が表示されます。



2つのデバイスの差を表示してどのアクションを取るか判断してください。
以降表示される全ての重複デバイスすべてに同じアクションを適用したいときは[Apply to all duplicates]も選択します。
[Cancel]ボタンをクリックするとインポート処理が中止されます。

インポート操作が完了すると次のようなメッセージが表示されます。



ファイルからインポートされたデバイスの総数が表示されます。

警告やエラーが表示されているときは[Info Log]タブで詳細を確認してください。

デバイスインポートファイルの形式

ファイルの冒頭はヘッダー行になり、各フィールドについて定義します。ヘッダー行以降はデータとして処理されます。

例:

Name	Type	Group	HostAddress	Filename	Model	ConnectVia	Telnet	TelnetPort
Cisco Router 2800	Cisco Router General	Test Group	127.0.0.1	Cisco_Router_2800	Cisco 2800	Direct connect	Telnet	23

フィールドの並び順は意識しないで構いませんが、Export devices to tab delimited file を実行したときと同じ命名規則に従う必要があります。

最低限ファイルで定義されていなければならないフィールドは次のとおりです。

- Type (デバイスのタイプ)
- Name (デバイスの重複しない固有名)
- HostAddress (デバイスの IP アドレスまたはホスト名)
- Filename (重複しない固有のデバイスファイル名。パスや拡張子は必要ありません。)

以下はデバイスフィールドとして指定可能なフィールドの一覧です。

- Type
- Group
- Name
- HostAddress
- Filename
- Model
- ConnectVia
- Telnet
- Session
- VTYPass
- ConsolePass
- EnablePass
- PrivilegeLevel
- AAAUsername
- AAAPassword
- SNMPRead
- SNMPWrite
- RequireVTYLogin
- LoginUsesAAA
- EnableUsesAAA

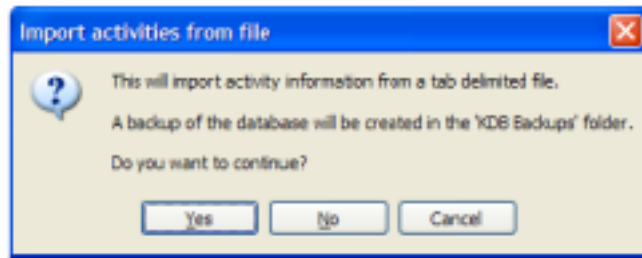
- VTYPrompt
- ConsolePrompt
- EnablePrompt
- AAAUserPrompt
- AAAPassPrompt
- Address1
- Address2
- Address3
- ContactName
- ContactPhone
- ContactEmail
- ContactOther
- AlertEmail
- SerialNumber
- AssetTag
- Identification
- SerialOther
- ActivitySpecific1
- ActivitySpecific2

インポートファイルの作成元となるようなテンプレートを作る最も簡単な方法は、CatTools でいくつかのサンプルデバイスを追加し、[File]メニューから[Export]、[Export devices to tab delimited text]と選択してファイルにエクスポートしておくことです。このファイルには必要なすべてのフィールドが記述されています。

フィールドに指定されているプレーンテキスト値 (HostAddress, VTYPass, EnablePass SNMPRead 等)は通常はデータベース内で暗号化されているため、インポート処理の段階で自動的に暗号化されます。

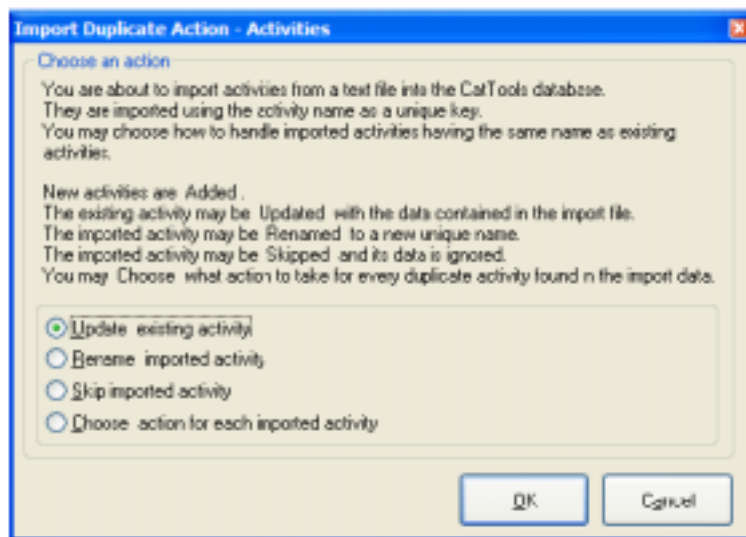
5.1.1.2.2 [Import activities from tab delimited file]

このオプションを使用すると、CatTools から以前タブ区切り形式のファイルでエクスポートされたアクティビティをインポートすることができます。



最初に表示される、操作を続けるかどうかを問う画面で[はい]を選択すると、通常のファイル選択画面が表示されます。インポートするファイルを選択してください。

次に、インポートするファイル内に既存のアクティビティと同じ名前のアクティビティが含まれていたときに、どのように処理するかを選択する画面が表示されます。



[Import Duplicate Action]画面ではデバイスをインポートするときと同じように指定します。

インポートが完了するとインポートの成否とインポートされたアクティビティ数を示すプロンプトが表示されます。

5.1.1.3 [Change encryption key password]

Kiwi CatTools ではデバイスフィールドに書かれている情報の漏洩を予防するための保護策として業界標準の RC4 暗号方式を採用しています。

セキュリティ効果を高めるためにプログラムの実行時にパスワードを入力しなければならないようにすることが可能です。このようにすることによって、プログラムにアクセスできるユーザーをパスワードを知っている信頼のおける人物に限ることができます。

パスワードはデバイスフィールドを暗号化する際に使用します。パスワードを忘れてしまうと、プログラムを使用できずデバイスデータにアクセスできなくなります。万が一パスワードを忘れてしまったときは、弊社 Web サイトの製品サポートページにあるフォームを使用してデータベースを送信してください。パスワードを復元できるよう試みます。

標準モードではプログラムはパスワード保護されており、起動時にパスワードを入力するよう要求されます。

Service 版の場合、コンソール上のユーザーインターフェイスを表示しようとするたびにパスワードの入力を要求する画面が表示されます。

パスワード設定を削除して空白にすればパスワードを要求する画面は表示されなくなります。パスワードが設定されていなくても、デバイスデータベースの暗号化が解除されるわけではありませんので安全です。

パスワードを変更するには

このメニューオプションをクリックすると、次の 3 つのフィールドが表示されたパスワード変更画面が表示されます。

1. Existing password:
2. New password:
3. Confirm new password:

旧いパスワード(初期値は空白)を入力し、新しいパスワードを 2 回入力します。

パスワードをリセットして何も指定されていない状態に戻すには、新しいパスワードのフィールドとそのパスワードの確認フィールドの両方に空白を入力します。

設定したパスワードは忘れないようにしてください！

5.1.1.4 [Backup current database]

このオプションをクリックするとデータベースファイルのバックアップファイ

ルが¥¥KDB Backups フォルダの下に作成されます。

このフォルダはCatToolsをインストールすると標準で¥CatToolsの下に作成されます。

.kdb ファイルは上書きや情報の追加ができませんが、保存するたびに新しいファイルが作成され次のようにファイル名の末尾に連番で番号が振られます。

Backup-1, Backup-2, Backup-3... ..

最大9つのバックアップファイルが作成されます。バックアップが実行されるたびに Backup 1 が Backup 2 に、Backup 2 が Backup 3 になり Backup 9 になると最後で、以降は一番古いバックアップファイルから順次削除されていきます。

5.1.1.5 [Restore database from backup]

このオプションを使用すると、データベースのバックアップファイルを CatTools のメインフォルダにコピーし、現在のデータベースとして適用します。

CatTools のメインフォルダにコピーされたデータベースには Restored1_KiwiDBCatTools_kdb というファイル名が付けられます。「Restored1」の末尾に振られた番号は復元したデータベースに固有の名前を付けるための連番です。

5.1.1.6 [Squeeze current database]

CatTools データベースには通常のデバイスやアクティビティの保守トラフィックによる未使用の領域が含まれている場合があります。この未使用の領域は一定期間を経たら再利用することが非常に望ましい運用であるといえます。

データベースの圧縮機能を使用すれば、データベースファイル内の未使用領域を再利用することが可能になります。また、同時にデータに付けられている索引をすべてリフレッシュされます。

データベースの圧縮を実行するには、圧縮しようとしているデータベースに対して他のユーザーが同じ操作を実行していないことが条件です。

5.1.1.7 他のデータベースを開く [Open/New]

CatTools では互換性のあるデータベースであれば、どのフォルダにありとそれに対するフルアクセス権限があれば開くことができます。[CatTools Database Selection]画面では必要なデータベースを探し、選択することができます。

ただし、CatTools の前バージョンでデータベースを開くときは注意が必要です。互換性のないファイルは開けません。互換性のあるデータベースでも最新バージョンでない場合は、そのファイルを開こうとしている CatTools のバージョンで必要であればアップグレードしなければなりません。

5.1.1.8 新しいデータベースを作成する [Open/New]

このオプションを使用すると、空のデータベースが作成されます。新しいデータベースのデフォルト名は New_KiwiDB-CatTools.kdb です。

5.1.2 [Delete]

次のオプションを使用して情報ログファイルを削除することによってディスク領域を広げることができます。

- Delete Infolog.txt file

5.1.2.1 [Delete Infolog.txt file]

情報ログファイルは CatTools のメインディレクトリの下(通常は C:\Program files\CatTools\ "n"はバージョン番号)に「Infolog.txt」という名前で保存されています。

ログファイルを削除するには[File]メニューから[Delete]、[Delete Infolog.txt file]をクリックしてください。

注：[Options]の下にある[Setup]メニューに表示される[Logging]タブでこのファイルの最大サイズを指定できます。

5.1.3 [Enable capture mode]

キャプチャモードはデバイスへのアクセス時に生じた問題を診断するときや、弊社の技術サポートから要求されたときに使用します。

アクティビティが原因もわからずに失敗した場合、弊社技術サポートにより [Enable capture mode]をクリックしてからアクティビティを実行するよう指示されることがあります。この操作によって CatTools とデバイス間の通信がすべてログファイルとして記録されます。次に、このファイルを弊社技術サポートまで送っていただき問題解決の糸口として活用させていただきます。

[Enable capture mode]メニューオプションはチェックボックスのような動作をします。デフォルトはオフです。

[File]メニューから[Enable capture mode]をクリックすると、即キャプチャモードが有効になります。有効になっていることは[File]メニューをクリックすると[Enable capture mode]の隣にチェックマークが表示されていることによって確認できます。

このオプションを再度クリックするとキャプチャモードは即無効になり、チェックマークが消えます。

取得したデータはDebug フォルダの下に、キャプチャ対象のデバイス名に応じた名前の付けられたファイルに保存されます。たとえば、スレッド 1 上で実行中の 2950 というデバイスの場合、DebugLog-1-2950.txt という名前のファイルが作成されます。

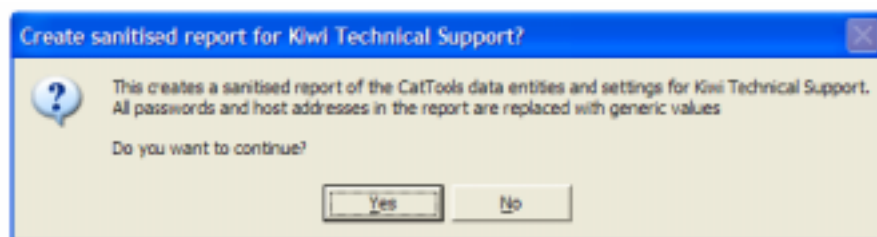
5.1.4 [Debug]

このオプションを使用すると、デバッグ関連の問題が生じたときのサポート用として送信していただくデバッグ情報を作成できます。

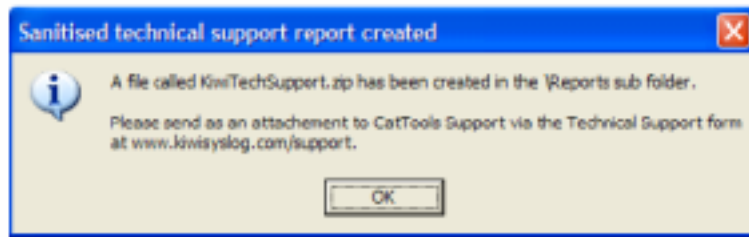
5.1.4.1 [Create sanitised database report for Kiwi Technical support]

このメニューオプションを選択すると、弊社技術サポートに送っていただくことによって問題解決に非常に役立ついくつかのファイルの含まれた zip ファイルが作成されます。潜在的に安全な情報が取り交わされます。

このメニューオプションをクリックすると次のようなプロンプトが表示されます。



[はい]をクリックすると次のようなプロンプトが表示されます。



5.1.5 [Exit]

Kiwi CatTools のプログラムを終了し、ウィンドウを閉じます。

重要なお知らせ:

Service 版をご使用の場合、Kiwi CatTools はバググラウンドでサービスとして実行を続けます。Kiwi CatTools のサービスを停止するにはコントロールパネルの中にある[サービス]管理ツールを使って停止してください。

5.2 [View]

[View]メニューでは様々な CatTools データファイルをメモ帳を使って表示します。

5.2.1 [View Reports folder]

[View]から[View Reports folder]をクリックすると、Windows Explorer が起動し %Report フォルダが開きます。

スケジュール化されているアクティビティのレポートは%Reports フォルダに保存されています。このメニューオプションをクリックすると、Windows Explorer で直接%Reports フォルダが開き、内容を確認できます。

レポートファイルは MS-Excel、Kiwi Logfile Viewer などの外部プログラムや [Reports]タブからレポートファイルにアクセスし CatTools 付属のビューワーを使って内容を表示できます。

5.2.2 [View Captured Data folder]

[View]から[View Captured Data folder]をクリックすると、Windows Explorer が起動し取得したデータが表示されます。

スケジュール化されているアクティビティの取得済みデータは%Captured Data フォルダに保存されています。このメニューオプションをクリックすると、

Windows Explorer で直接¥Captured Data フォルダが開き、内容を確認できます。

レポートファイルは MS-Excel、Kiwi Logfile Viewer などの外部プログラムや [Reports] タブからレポートファイルにアクセスし CatTools 付属のビューワーを使って内容を表示できます。

5.2.3 [View Configs folder]

[View] から [View Configs folder] をクリックすると、Windows Explorer が起動し ¥Configs フォルダが開きます。

Device.Backup.Running Config アクティビティによって作成されたデバイスコンフィグファイルは ¥Configs フォルダに保存されています。このメニューオプションをクリックすると、Windows Explorer で直接 ¥Configs フォルダが開き、内容を確認できます。

コンフィグファイルはメモ帳かワードパッドで開くことができます。

5.2.4 [View Statistics folder]

[View] から [View Statistics folder] をクリックすると、Windows Explorer が起動し ¥Statistics フォルダが開きます。

スケジュール化されているアクティビティの統計データは ¥Statistics フォルダに保存されています。このメニューオプションをクリックすると、Windows Explorer で直接 ¥Statistics フォルダが開き、内容を確認できます。

レポートファイルは MS-Excel、Kiwi Logfile Viewer などの外部プログラムや [Reports] タブからレポートファイルにアクセスし CatTools 付属のビューワーを使って内容を表示できます。

5.2.5 [View Email log]

[View] から [View Email log] をクリックすると、メモ帳が起動し CatTools の E メールアクティビティのログファイルが開きます。

5.2.6 [View Activity log]

[View] から [View Activity log] をクリックすると、メモ帳が起動し CatTools のアクティビティのサマリログファイルが開きます。

5.2.7 View Info log

[View]から[View Info log]をクリックすると、メモ帳が起動し CatTools の情報ログファイルが開きます。

5.3 [Options]

このメニューからは各種設定オプションを変更できます。

5.3.1 [Setup]

このメニューオプションをクリックすると、CatTools の一般設定項目を設定するためのフォームが表示されます。

[Email]タブでは、すべての E メール関連のプロパティを設定できます。

[Logging]タブでは、CatTools でシスログメッセージをシスログサーバーに送信し、情報およびアクティビティのログの管理を行うよう設定できます。

[Misc]タブでは、その他の設定を行います。

[TFTP Server]タブでは、CatTools の TFTP サーバーを管理するための設定を行います。

5.3.1.1 [Email]

[Email]タブではすべての E メール関連プロパティを設定できます。

さらに次のタブに分かれています。

- General Options
- Primary SMTP Server Setup
- Secondary SMTP Server Setup

5.3.1.1.1 [General Options]

[Send To:] (E メール通知)

[Errors:] CatTools からエラー通知メールを送信したい E メールアドレスを指定します。スケジュール化されているアクティビティの実行時に発生したすべてのエラーは 1 つのエラー警告として集約されます。エラー通知メールには発生したエラーの種類とそのエラーが発生したデバイスが記載されます。

コンマ(,)またはセミコロン(;)で区切れば複数の E メールアドレスを指定できます。

例: joe@company.com,mary@company.com,fred@company.com

このフィールドの下にあるチェックボックスをオンにすると、タイマーモードのときのみエラー通知が送信されます。

[Reports/Stats:] CatTools からレポートと統計を送信したい E メールアドレスを指定します。スケジュール化されているアクティビティの多くがレポートまたはある種の統計表を出力します。

コンマ(,)またはセミコロン(;)で区切れば複数の E メールアドレスを指定できます。

例: joe@company.com,mary@company.com,fred@company.com

このフィールドの下にあるチェックボックスをオンにすると、タイマーモードのときのみレポートが送信されます。

[From Address:]

CatTools から E メール通知を送信する際に「送信者」に指定するアドレスを入力します。このフィールドを指定することによって、受信 E メール通知をご使用の E メールクライアントで管理しやすくなります。特に SPAM をフィルタするとき便利です。

実際にしようされている E メールアドレスを省略せずに指定することをお勧めします。一部の SMTP サーバーはドメインのアドレスを代表する電子メールのみを送信します。例えばメールアドレスが mycompany.com だとすると、送信者の E メールアドレスには単に「CatTools」と表示するのではなく「CatTools@mycompany.com」と表示します。

[Logging Options]

1つ目のチェックボックスでは電子メールアクティビティのログを取るか否かを指定します。通常、このログには情報レベルとエラーレベルのメッセージが記録されます。

2つ目のチェックボックスではメッセージのロギングレベルを詳細に設定するか否かを指定します。E メールサーバーを通過する E メールメッセージトラフィックの完全な情報をチェックする必要があるような問題が発生したときにオンにしておく便利です。記録されるデータが多いため、必要時以外はこのオプションはオフにしておいてください。

E メールログを表示するためと削除するためのボタンが2つあります。

[Send Interval:] Eメールの送信間隔を秒単位で指定します。メーラーが E メール

メッセージのバッチを送信するまでの待ち時間となります。20 と指定されている場合、20 秒ごとにメーラーが起動し送信するメッセージのチェックを行います

[Messages / Interval:] 1 つの送信セッションの中でメールサーバーにメーラーから送信される E メールメッセージの数を指定します。E メールサーバーに対する E メールトラフィックの総数を管理できます。

[Send Retries:] メーラーが 1 つのメッセージの送信を何度試行するかを設定します。エラーのためメッセージが送信できないようなとき、ここで指定した送信試行回数を超えるとそのメッセージに送信不能フラグが付けられます。

[Test Email]

ボタンをクリックすると、送受信両方のメールサーバーに対して現在のメール設定が有効であるかどうかのテストが実行されます。テストメッセージは[Errors] および[Reports/Stats]に指定したアドレスに送信されます。

5.3.1.1.2 [Primary SMTP Server Setup]

[Use this server]

このチェックボックスにより CatTools で E メールを送信する際に使用するサーバー指定の有効/無効を切り替えます。

[Server Name / Address:]

CatTools で E メール通知を送信したい SMTP サーバーの IP アドレスまたは名前を入力します。

[Port:]

SMTP サーバーが使用するポートを入力します。デフォルトで 25 に設定されています。

[Timeout:]

CatTools が SMTP サーバーに接続を試みる際の待ち時間のタイムアウト値を秒単位で指定します。デフォルトで 30 に設定されています。メールサーバーへの接続がうまくいかないときはローカル SMTP リレーアプリケーションを使用することをお勧めします。<http://www.ocloudsoft.com> の MailDirect を推奨します。

[Authentication:]

一部の SMTP サーバーは(特にリモートの ISP やリレーサーバーの場合)E メール

の送信前に認証情報を要求します。しかし、大部分の SMTP サーバーは認証を行わないため、この設定は空白のままでも構いません。SMTP で送信し POP3 で受信する場合と混同しないよう注意してください(この場合は必ずユーザー名とパスワードの認証を行います)。

[Type:] 送信前の認証を(none), SMTP AUTH, POP の中から選択します。

[Username:] 認証に使用するユーザー名を入力します。

[Password:] 認証に使用するパスワードを入力します。

[POP Server Name:] 認証に使用する POP サーバーの名前またはアドレスを入力します。

[POP Port:] POP サーバーのポート番号を入力します。通常は 110 です。

[Test Email]

ボタンをクリックすると、プライマリメールサーバーに対して現在のメール設定が有効であるかどうかのテストが実行されます。テストメッセージは[Errors]および[Reports/Stats]に指定したアドレスに送信されます。

5.3.1.1.3 [Secondary SMTP Server Setup]

CatTools では、プライマリサーバーが何らかの原因で接続できないときに使用する代替メールサーバーを設定しておくことができます。

メッセージを送信しようとしてメーラーがプライマリメールサーバーに接続を試みたが接続できない場合、すぐにセカンダリサーバーに対する接続を試みます。

メーラーがプライマリサーバーに対してメール送信を試みている間に接続が切れてしまった場合、メーラーはすぐに同じコネクションへの再接続を試みるのではなく、送信間隔として指定した時間待ってから再びプライマリサーバーへの接続を試みます。

プライマリサーバーを使用することが指定されていない場合は常にセカンダリサーバーが使用されます。

[Use this server]

このチェックボックスにより CatTools で E メールを送信する際に使用するサーバー指定の有効/無効を切り替えます。

[Server Name / Address:]

CatTools で E メール通知を送信したい SMTP サーバーの IP アドレスまたは名前を入力します。

[Port:]

SMTP サーバーが使用するポートを入力します。デフォルトで 25 に設定されています。

[Timeout:]

CatTools が SMTP サーバーに接続を試みる際の待ち時間のタイムアウト値を秒単位で指定します。デフォルトで 30 に設定されています。メールサーバーへの接続がうまくいかないときはローカル SMTP リレーアプリケーションを使用することをお勧めします。http://www.oclousoft.com. の MailDirect を推奨します。

[Authentication:]

一部の SMTP サーバーは(特にリモートの ISP やリレーサーバーの場合)Eメールの送信前に認証情報を要求します。しかし、大部分の SMTP サーバーは認証を行わないため、この設定は空白のままでも構いません。SMTP で送信し POP3 で受信する場合と混同しないよう注意してください(この場合は必ずユーザー名とパスワードの認証を行います)。

[Type:] 送信前の認証を(none), SMTP AUTH, POP の中から選択します。

[Username:] 認証に使用するユーザー名を入力します。

[Password:] 認証に使用するパスワードを入力します。

[POP Server Name:] 認証に使用する POP サーバーの名前またはアドレスを入力します。

[POP Port:] POP サーバーのポート番号を入力します。通常は 110 です。

[Test Email]

ボタンをクリックすると、セカンダリメールサーバーに対して現在のメール設定が有効であるかどうかのテストが実行されます。テストメッセージは[Errors]および[Reports/Stats]に指定したアドレスに送信されます。

5.3.1.2 [Logging]

[Syslog]

[Send logs to Syslog Server]

オンにすると、シスログサーバーに対するログメッセージ送信が有効になります。デフォルトではオフになっています。

[Syslog Server:]

ログを送信したいシスログサーバーの名前または IP アドレスを入力します。

[Protocol:]

ドロップダウンリストから UDP または TCP を選択します。デフォルトは UDP です。

[Port:]

有効なポート番号を指定します。UDP のデフォルトポートは 514 です。

[Facility:]

ドロップダウンリストから送出するシスログメッセージの種類を選択します。デフォルトは Local0 です。

[Logging Level:]

ドロップダウンリストからシスログサーバーに転送するログメッセージのレベルを選択します。デフォルトは Info and above です。

[Test Syslog]

ボタンをクリックすると、上記のオプションで指定した設定を使ってシスログサーバーにテストメッセージが送信されます。

コンフィギュレーションとネットワークコネクションが正しく設定されているかを確認できます。

[Info log]

[Log info log messages to local log file]

オンにすると、ローカルログファイルに対する情報ログメッセージの記録が有効になります。デフォルトではオンになっています。作成されるログファイルは CatTools のインストール先フォルダの下に InfoLog.txt という名前で保存されます。

[Logging level:]

ドロップダウンリストからローカルファイルに記録するメッセージのレベルを選択します。デフォルトは Debug and above です。

[Limit File size:]

デバッグメッセージのロギングが有効になっているとログファイルが肥大化することがあります。そのような状況になったら[File]メニューの[Delete]、[Delete log file]を選択してログファイルを削除するか、予めこのオプションのチェックボックスをオンにしてフィールドに情報ログの最大サイズをMb単位で指定することによって制限をかけておく方が良いでしょう。ログファイルが指定したサイズに達すると、それ以上書き込みができない状態となりファイル名が `infolog_prev.txt` に変更されます。新しいログファイルが作成され、以前のログファイルは現在のログの記録用としてではなく将来の参照用として保存されます。新しく作成されたログファイルが最大サイズになり名前が変更されると同時に古いログは削除されます。

このオプションを指定することによって現在のログと過去のログの 2 つのログファイルが作成されることとなります。

オフにすればログファイルは無限に肥大化します。

[Activity log]

[Log Activity summary to local log file]

オンにすると、ローカルログファイルに対する CatTools アクティビティの概要の記録が有効になります。デフォルトではオンになっています。作成されるログファイルは CatTools のインストール先フォルダの下に `Activitylog.txt` という名前で作成されます。

[Limit File size:]

このチェックボックスをオンにすると、アクティビティログのサイズをこのフィールドに指定した値までに制限することができます。サイズはMB単位で指定してください。ログファイルが指定したサイズに達すると、それ以上書き込みができない状態となりファイル名が `activitylog_prev.txt` に変更されます。新しいログファイルが作成され、以前のログファイルは現在のログの記録用としてではなく将来の参照用として保存されます。新しく作成されたログファイルが最大サイズになり名前が変更されると同時に古いログは削除されます。

このオプションを指定することによって現在のログと過去のログの 2 つのログファイルが作成されることとなります。

オフにすればログファイルは無限に肥大化します。

5.3.1.3 [Misc]

[Telnet client:]

ローカルの Telnet クライアントのパスとファイル名を入力します。デフォルトは Telnet.exe です。

外部の Telnet クライアントとして SecureCRT をご使用の場合、コマンドラインに /TELNET スイッチを追加する必要があります。

例: C:\Program Files\SecureCRT\SecureCRT.EXE /TELNET

[Clear log window before starting each activity]

このチェックボックスをオンにすると、アクティビティの起動時に [Info log] 画面がクリアされます ([Info log] タブの項を参照してください)。デフォルトはオンです。

[Clear error count before starting each activity]

このチェックボックスをオンにすると、アクティビティの起動時にエラー数の表示がクリアされます (メイン画面の Status Line を参照してください)。デフォルトはオンです。

[Automatically enable timer mode]

このチェックボックスをオンにすると、サービス起動後に指定した時間(秒)が経過すると自動的にタイマーが起動します。

[Hide password change report passwords]

このチェックボックスをオンにすると、Device Change password アクティビティによってレポートされるパスワードを隠すことができます。

[Show current database name in status bar]

このチェックボックスをオンにすると、現在の CatTools データベースのフルパスが表示されます。

[Terminate orphaned activity running longer than n minutes]

アクティビティを強制的に終了させるまでの待ち時間を分単位で指定します。デフォルトでは 0 分に設定されており、これは制限がないことを表しています。指定可能な最大値は 2880 分です。この設定はアクティビティが終了しないなど特殊な状況でのみ使用してください。アクティビティが終了しなくなる現象の原因は、論理ループが発生したり、何らかの理由で通信速度が低下するといったことが考えられます。例えば 1 つのデバイスを除いたすべてのセッションが完

了し、最後のセッションがハングアップしている様子のおきのみタイミングが起動します。

この機能が起動し、時間制限値が検出されると CatTools は通常の方法でアクティビティを終了させようとしてます。通常の方法でアクティビティを終了させられない場合は、アクティビティを実行している CatTools_Client.exe アプリケーションを強制終了します。終了後 30 秒経ったら、システムを有効にし、強制終了後のクリーンアップを実行してスケジューラを再起動します。

[HTML report maximum row limit]

HTML レポートの出力行数を指定します。デフォルトは 5000 行です。最小値は 1、絶対最大値は 50000 行です。最大行が 50000 に制限されているのは、HTML ファイルでそれ以上の行数になるとスクロールするのが大変だからです。レポートの全データが必要なときは、テキスト形式で出力したほうが良いでしょう。

[When exiting the Manager should:]

このオプションは CatTools がサービスとしてインストールされているときに表示されます。変更の大半はタイマーがオフラインのときに行う必要があるため、この機能を設定しておくで Manager が閉じている状態でもタイマーを再びスタートさせるリマインダとなります。タイマーを自分で再起動するのを忘れても、この機能を利用すればアクティビティを確実に実行できます。

オプション

[Ignore the timer state] 何のアクションも起こしません。

[Set the timer to ON] Manager を終了させると自動的にタイマーが ON になります。

[Set the timer to OFF] Manager を終了させると自動的にタイマーが OFF になります。

[PROMPT if the timer is off] タイマーが実行されていない場合、Manager を閉じたとき何をするかを問うメッセージボックスを表示します。

5.3.1.4 [TFTP Server]

[Automatically start TFTP server on program startup]

CatTools の起動時に自動的にサーバーを起動します。

[TFTP Port:]

サーバーの受信待機用ポート番号を入力します。通常は 69 です。

[Bindto:]

バインド先のネットワークインターフェイスカードのIPアドレスを入力します。インターフェイスカードが 1 つのみの場合やデフォルトのカードを使用する場合は空白のままにしておいて構いません。

[Upload folder:]

CatTools からデバイスにアップロードするファイルの保存先フォルダ名を入力します。

[Download folder:]

CatTools でデバイスからダウンロードするファイルの保存先フォルダ名を入力します。

[Allow overwrite of existing files]

チェックボックスをオンにすると、フォルダ内の現在のファイルが同じ名前のファイルで上書きされます。

[Allow automatic creation of sub folders]

チェックボックスをオンにすると、ダウンロード中にサブフォルダが自動的に作成されます。

[Remove completed sessions from display after:]

TFTP セッションによって作成されたメッセージを指定した秒数が経過したらウィンドウから削除します。

[Maximum file size that can be created:]

サーバーが作成可能なファイルの最大値を MB で指定します。デフォルトは 20MB です。TFTP クライアントは送信可能なファイルサイズの制限がこれとは別に設定されている場合もあります。周知のとおり、一部の TFTP クライアントには約 32MB というファイルサイズの制限があります(32MB のファイルサイズは通常のプロトコルのブロックサイズで 512 バイトを使用することに相当します)。

ただし、ネットワークおよびデバイスに指定したサイズ以上のデータブロックを処理できる能力がある場合は、この設定を超えた大きさのファイルを作成できる場合があります。CatTools には使用するブロックサイズを指定するオプションはありません。ブロックサイズは TFTP クライアントとサーバー間でネゴシエートされます。

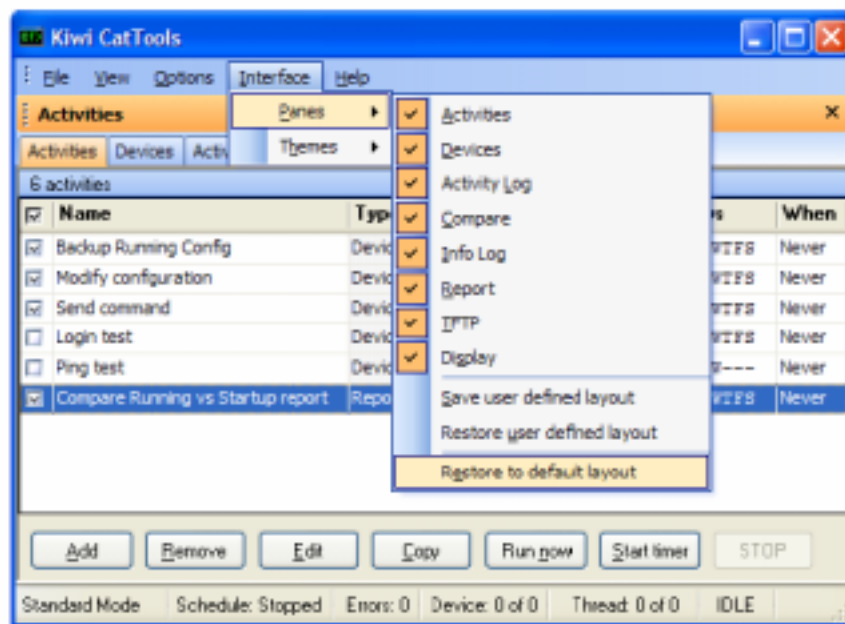
5.4 [Interface]

このメニューからは[Panes]および[Themes]インターフェイスオプションを変更できます。

5.4.1 [Panes]

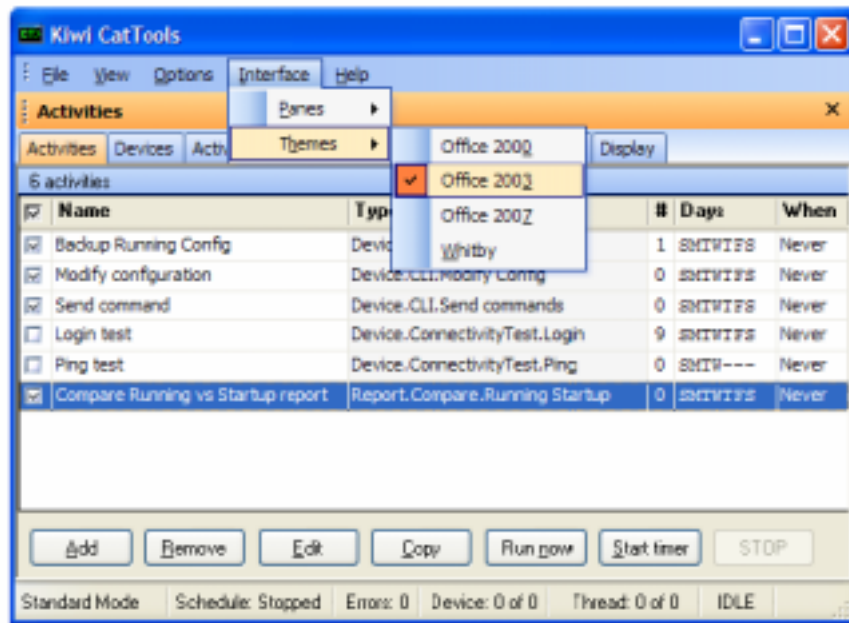
[Interface]の[Panes]の下には複数のオプションがあります。

- [Panes]の下に表示されるペイン一覧のチェックをオン/オフして、そのペインの表示/非表示を切り替えます。
- [Save user defined layout]をクリックすれば、現在のペインレイアウトを保存できます(注:CatTools ではアプリケーションを終了すると自動的に現在のペインレイアウトを保存します)。
- [Restore user defined layout]をクリックすれば、前回保存したペインレイアウトに戻すことができます。
- [Restore to default layout]をクリックすれば、CatTools のデフォルトのペインレイアウトに戻すことができます。



5.4.2 [Themes]

[Interface]、[Themes]の下にはテーマの一覧が表示されます。チェックが付いているもの以外を選択すると、CatTools インターフェイスに適用されている配色等が変更されます。



5.5 [Help]

[Help]メニューからはヘルプを表示し、登録や設定に関する詳細情報を参照したり、CatToolsのWebサイトを閲覧することができます。

5.5.1 [Contents]

ヘルプファイルの目次が表示されます。

5.5.2 [Online FAQ]

ブラウザが起動し、Kiwi CatTools Online Frequently Asked Questions ページが開きます。

5.5.3 [Purchase Kiwi CatTools]

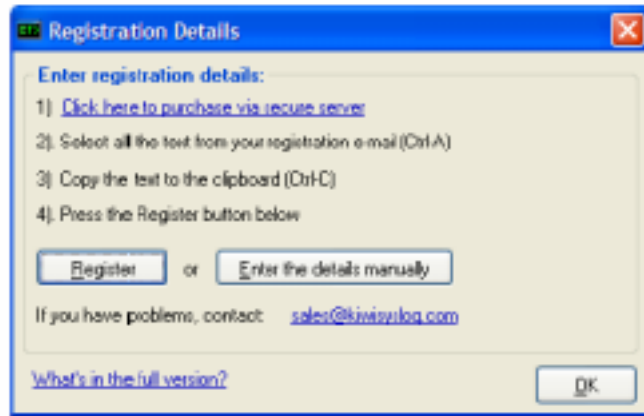
ブラウザが起動し、弊社のオンライン購入ページが開きます。オンライン購入以外の購入手段や支払い方法についても紹介しています。

5.5.4 [Enter registration details]

[Registration Details]ウィンドウが表示されます。

既に Kiwi CatTools のライセンス登録を行ってある場合は、ご使用の登録情報が表示されています。

まだ登録を行っていない場合、基本的な製品購入手続きとライセンス登録方法が記載されている次のウィンドウが表示されます。



登録用 Eメールの本文は[Register]ボタンをクリックすると CatTools によって解析されて必要な情報が自動的に登録されるよう設計されています。

[Registration]ボタンが機能しないときは[Enter the details manually]ボタンをクリックして手動で登録情報を入力するか、sales@kiwisyslog.com にご連絡ください。

5.5.5 [About]

[About]ウィンドウが開き、プログラムのバージョンナンバー、登録内容、サポート情報などが表示されます。

弊社サポートにご質問の際は、ご使用のプログラムのバージョンとビルドをお知らせください。より迅速に対応いたします。

6 [Panels]

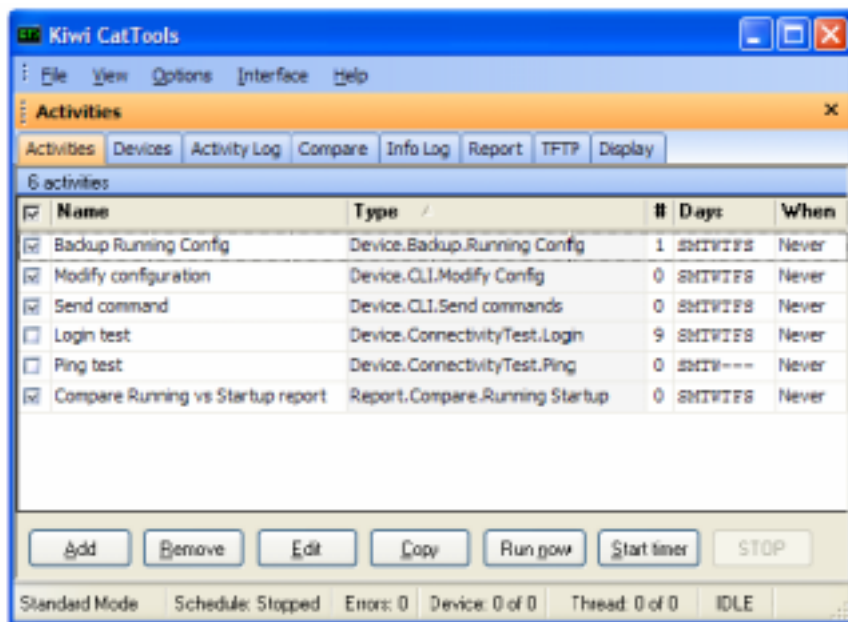
CatTools 3.1.1 でペインドッキングインターフェイスが採用され始めました。3.1.1 以前のバージョンではタブレイアウトのみでした。ペインインターフェイスは基本的にタブレイアウトと同じですが、ユーザーが必要に応じて表示/非表示を切り替えたり、同時に複数のペインを表示させるといったカスタマイズが可能であるという特長があります。

- 概要
- [Devices]ペイン

- [Activities]ペイン
- [Activity Log]ペイン
- [Compare]ペイン
- [Info Log]ペイン
- [Report]ペイン
- [TFTP]ペイン
- [Display]ペイン
- [Mail]ペイン

6.1 概要

CatTools に採用されているドッキングペインインターフェイスは、インストール直後は普通のタブリストのように表示されます(下図参照)が、各ペインはウィンドウ内の任意の場所へドラッグアンドドロップで移動させることができ、完全にユーザー好みのレイアウトを作成することが可能です。



ペインのドッキング

ドッキングの位置は数多くの選択肢があります。

例えば、ウィンドウの上下左右、他のペインの上下左右など自由にドック位置を決めることができます。

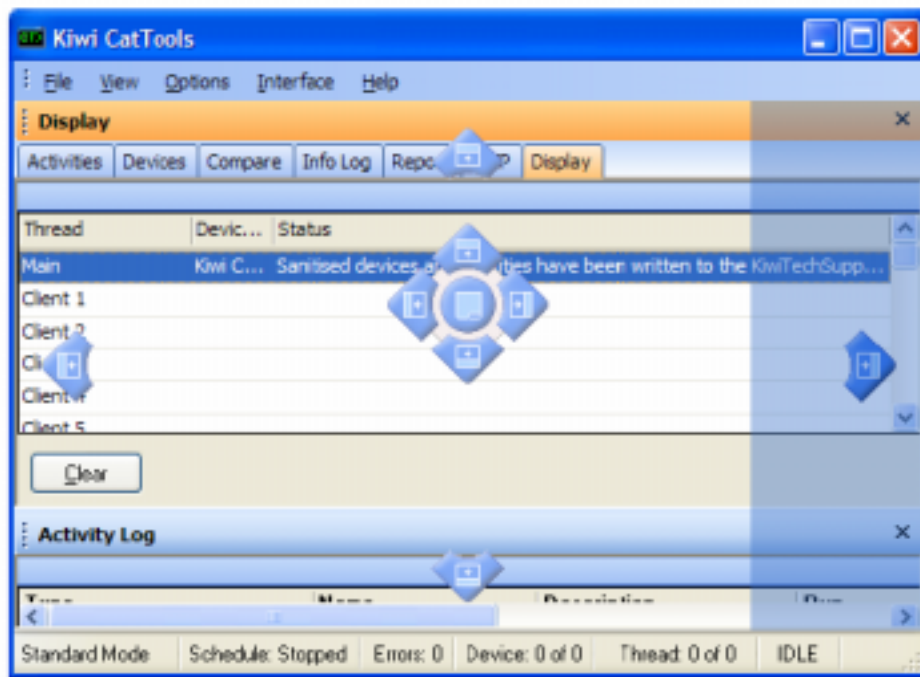
ペインのドック位置を変更するには、次の手順に従ってください。

1. ドック位置を変更したいペインにマウスポインタを合わせて左ボタンを押したままウィンドウ内までドラッグします。ドッキングペインロケーターがウィンドウ内に表示されます。
2. マウスボタンを押し下げたまま、ドッキングペインロケーターの任意のマークの上までカーソルを移動させます。ペインがドックされる位置がハイライト表示で示されます(下図参照)。
3. マウスボタンを放すと、その場所にペインが移動します。

下図は、[Display]ペインをロケーターの右端マークを使ってウィンドウの右端に移動させた場合の例です。

他の全てのペインの右側にくるように配置されます。

[Display]ペインを([Activity Log]ペインをウィンドウ幅で表示させたまま)上端ペインのすぐ右側に配置させたい場合は、真ん中にある菱形の右側マークの上でドロップしてください。



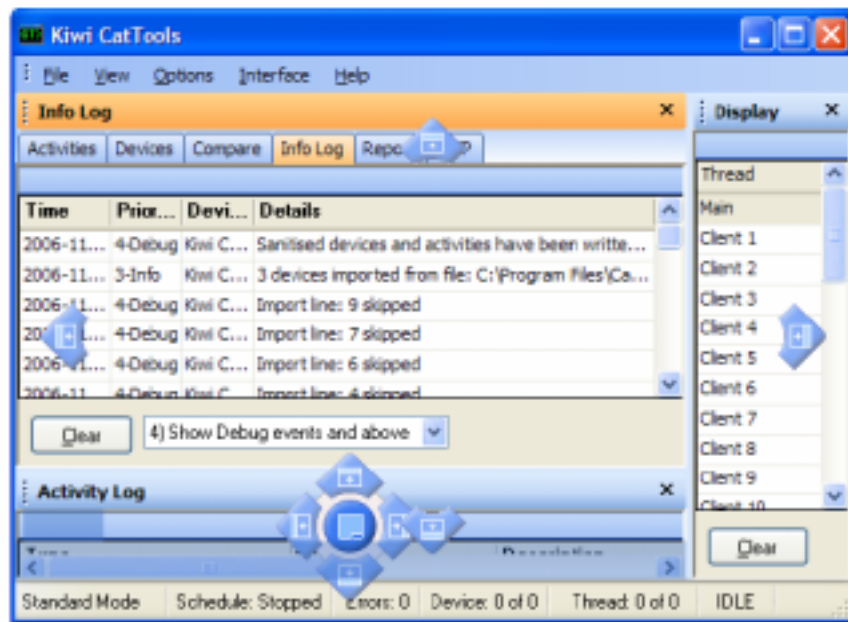
ペインの移動/グループ化

ペインをグループ間で移動するには、次の手順に従ってください。

1. 移動させたいペインにマウスボタンを合わせて左ボタンを押したままグループ化したいペインのウィンドウ内までドラッグします。ドッキングペインロ

- ケーターがウィンドウ内に表示されます。
2. マウスボタンを押し下げたまま、グループ化するペインに表示されるロケータの中央マークの上までカーソルを移動させます。ペイン全体がハイライト表示されます(下図参照)。
 3. マウスボタンを放すと、ペインが新しくグループに追加されます。

下図は[Info Log]ペインを上端グループからウィンドウ下部に表示されている[Activity Log]に移動しグループ化した場合の例です。



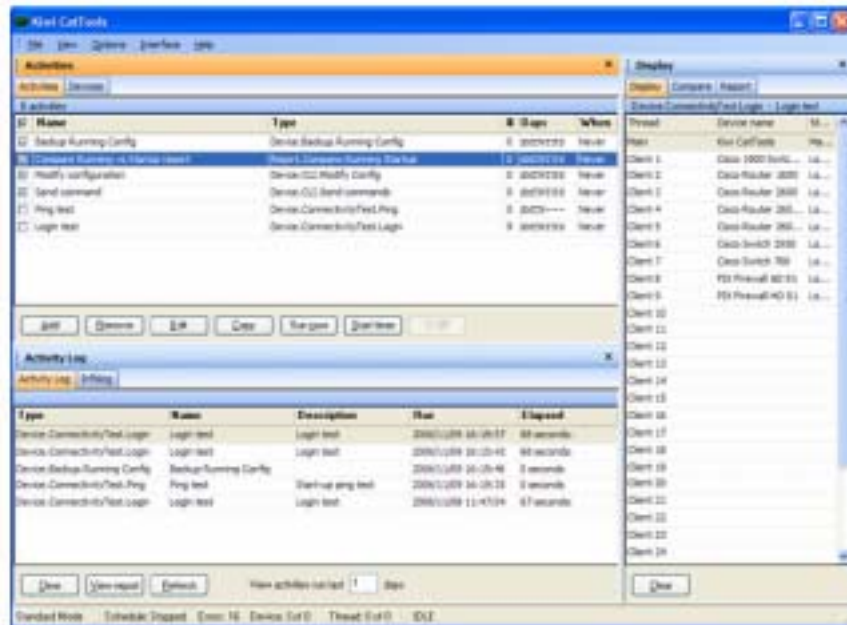
レイアウトの再調整および保存

ペインレイアウトは、マウスカーソルをペインの隅に合わせ左ボタンを押し下げたままロケータが表示されるところまで動かし、さらに移動させたい位置のマーク上のところで放すことによって簡単に再調整できます。

レイアウトが決まったら、[Interface]メニューの[Panes]の下の一覧から[Save user defined layout]を選択して作成したレイアウトを保存することができます。詳細については[Interface]の項を参照してください。

CatTools の終了時にも変更が自動で保存されます。

下図は複数のペイングループごとにウィンドウサイズの変更を行った例です。



ペインをデフォルト状態に戻す

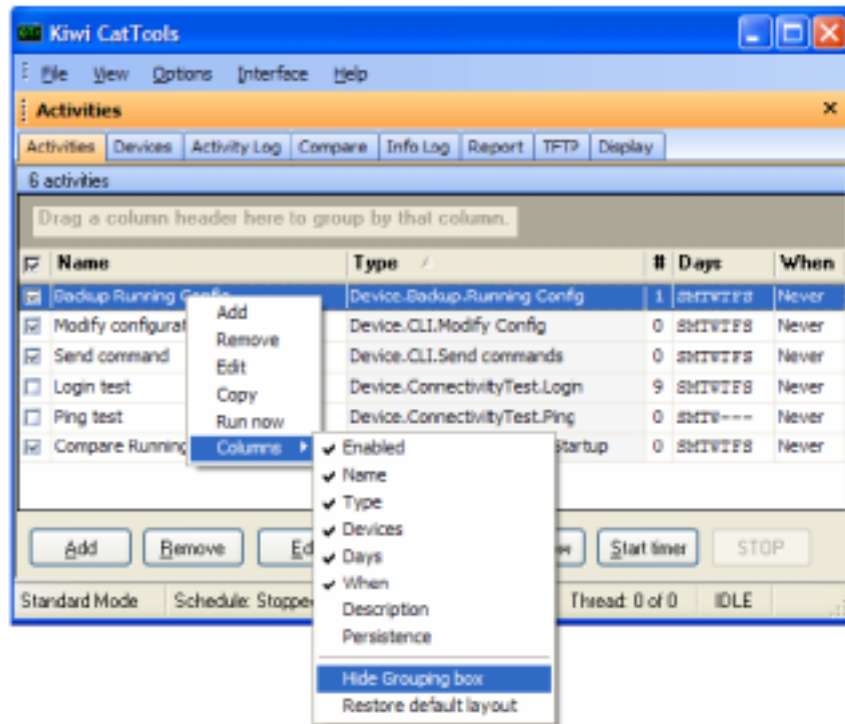
ペインをデフォルト状態に戻すには、[Interface]メニューの[Panes]の下の一覧から[Restore to default layout]を選択します。

詳細については[Interface]の項を参照してください。

右クリックによるショートカットメニュー

各タブの下部にあるボタンの機能は、ペインに表示されている行を選択して右ボタンをクリックしても実行できます。

このショートカットメニューには、現在表示されているペイン独自のアイテムへのショートカットもあります。例えば[Activity]ペインの場合、アクティビティに関するデータ列の表示/非表示を切り替えるためのサブアイテムを含む[Colum]というアイテムがあります。[Show/Hide the grouping box]をクリックすると、データ列ごとにアクティビティがグループ化されて表示されます。[Restore default layout]をクリックすれば、デフォルトのレイアウトに戻すことができます。



6.2 [Devices]

このペインには定義済みデバイスが表示されます。

6.2.1 [Add]

新しいデバイスを追加するには[Add]をクリックします。

[Device Information]ウィンドウは次の5つのタブで構成されています。

- [Device info] - デバイスのメーカーやモデル等について記述します。
- [Passwords] - デバイスのログイン時や有効モードにするとき等のパスワードを設定します。
- [Prompts] - デフォルト設定と異なるコマンドラインプロンプトをデバイスに表示させたいとき記述します。
- [Contact info] - デバイスの担当者の連絡先等を記述します。
- [Extra info] - デバイス特有の追加情報を入力するためのフィールドがあります。

ボタン: (ボタンは上記5種のタブの下部に表示されます。)

[Ping device] - ハイライト表示されているデバイスに ICMP Ping パケットが送

信されません。

[Telnet] - 指定した Telnet プログラムを使ってハイライト表示されているデバイスで Telnet セッションが開始されます。使用する Telnet プログラムは[Setup]画面で指定します。

[OK] - 現在の変更を適用し、メイン画面に戻ります。

[Cancel] - 変更を破棄し、メイン画面に戻ります。

6.2.1.1 [Device info]

[Device Type:] ドロップダウンリストから CatTools のデバイスタイプを選択します。

[Group:] このデバイスを所属させたいグループの名前を入力するかドロップダウンリストから選択します。このフィールドはフリーフォームなので、テキストを入力するだけで新しいグループを作成できます。論理カテゴリーまたは物理カテゴリーでデバイスを分類し、それぞれに適したグループ名を付けてください。

[Name:] デバイ스에重複しない固有の名前を付けてください。

[Host Address:] 新しいデバイスの IP アドレス(標準的な aaa.bbb.ccc.ddd 形式またはホスト名)を入力します。

[File Name:] このデバイスのデータおよびレポートを保存するために CatTools で使用する基本ファイルの名前を定義します。このフィールドにはデバイス名が反映されますが、ファイル名として使用できない文字は変換されます。

[Model:] ドロップダウンリストからデバイスのモデル名を選択します。このフィールドはフリーフォームなので、自由にテキストを入力することができます。また、文書上でのみ使用されるフィールドなので、どんなテキストを入力しても CatTools のオペレーションには影響しません。デバイスに対するセッションを CatTools でどのように処理するかは[Device Type]によって決められます。

[Connect via:] ドロップダウンリストから接続時に経由するデバイスを選択します。デフォルトは「Direct connect」です。ここで別のデバイスを指定する必要があるには、対象デバイスへ直接アクセスできない場合のみです。CatTools では Telnet または SSH を使用してデバイスからデバイスへホッピングさせて最終的な接続先に繋ぐことができます。例えば、アクセスリストの後にデバイスがあるが Linux ボックスがこのデバイスに対しアクセスしている場合、最初に Linux

ボックスを経由して接続し、そこから接続先デバイスへの Telnet あるいは SSH セッションを開始させることができます。

注意: ジャンプポイントとして Cisco ルーターを指定したときは、config ファイルの Line VTY セクションで logging synchronous を無効にしてください。これが有効になっていると、ルーターから Telnet セッションを開始しようとするとエラーが発生することがあります。

[Method:] ドロップダウンリストから接続方式を選択します。デフォルトは「Telnet」です。SSH を使用する場合、特定のデバイスと接続する際に SSH の改良型が必要な場合もあるため注意してください。例えば、Netscreen デバイスは SSH2 をサポートしていますが、CatTools で接続する場合は改良型の SSH2-nopty が必用です。

[Port:] 選択した接続方式のセッションが使用するポート番号を入力します。Telnet のデフォルトポートは 23 であり、通常はデフォルト値を変更する必要はありません。SSH はポート 22 を使用します。

6.2.1.2 [Passwords]

[VTY Password:] 仮想ターミナルまたは初期ログインパスワードを入力します。

[Enable Password:] 有効モードまたは優先モードのパスワードを入力します。

[Privilege Level:] 必用であれば有効モードの優先レベルを指定します。通常は設定する必要はありません。空白のままにしておいてください。Cisco 製ルーターには 0 から 15 までの 16 の優先レベルを指定できます。標準の有効モードはレベル 15 です。

この値は有効コマンドがデバイスに送信される際に追加して送られるため、必用もないのに指定されていると、デバイスが有効モードへの認証試行を実行する際にエラーとなる可能性があります。

[Console Password:] COM ポートからの直接接続を使用する場合はコンソールのパスワードを入力します。

[Username:] ユーザー名(例:AAA, TACACS, RADIUS またはローカルユーザー名/パスワード等)を入力します。

[Password:] パスワードを入力します。

[SNMP Read:] デバイスの SNMP Read コミュニティ名を入力します。デフォルトの SNMP Read コミュニティ名は通常「public」ですが、別の名前に変更することをお勧めします。SNMP.System.Summary アクティビティを実行する場合はこのフィールドは必ず入力する必要があります。指定しない場合、SNMP タイムアウトやその他のエラーが発生します。

[SNMP Write:] デバイスの SNMP Write コミュニティ名を入力します。デフォルトの SNMP Write コミュニティ名は通常「private」ですが、別の名前に変更することをお勧めします。

[Initial login requires password:]チェックボックス オンにすると、このデバイスに初めてログインするときにパスワード(通常はVTYパスワードかコンソールの初期パスワード)の入力が必要になります。

[Initial login requires username/password:]チェックボックス オンにすると、このデバイスに初めてログインするときにユーザー名とパスワードの入力が必要になります。

[Enable mode requires username/password:]チェックボックス オンにすると、このデバイスに有効モードでログインするときにユーザー名とパスワードの入力が必要になります。[Username]フィールドで指定したユーザー名と[Enable Password]に指定したパスワードを使用します。

6.2.1.3 [Prompts]

デバイスに対して標準外のプロンプトを設定する場合のみ各フィールドに入力する必要があります。

[VTY Prompt:] このデバイスに適用する VTY ログインプロンプトを入力します。初めてログインする際のパスワードを聞くときにデバイスに表示されるテキストとなります(入力例 : 「Password: 」)

[Enable Prompt:] このデバイスに適用する Enable プロンプトを入力します。Enable パスワードを聞くときにデバイスに表示されるテキストとなります(入力例 : 「Password: 」)

[Console Prompt:] このデバイスに適用するユーザー名プロンプトを入力します。コンソールへ初めてログインする際のパスワードを聞くときにデバイスに表示されるテキストとなります(入力例 : 「Password: 」)

[Username Prompt:] このデバイスに適用するユーザー名プロンプトを入力します。コンソールへ初めてログインする際のユーザー名を聞くときにデバイスに表示されるテキストとなります(入力例 : 「 Username: 」)

[Password Prompt:] このデバイスに適用するパスワードプロンプトを入力します。コンソールへ初めてログインする際のパスワードを聞くときにデバイスに表示されるテキストとなります(入力例 : 「 Password: 」)

6.2.1.4 [Contact info]

[Address 1:] デバイスの設置場所を入力します(Location 列が有効になっていれば[Devices]リストに表示されます)。

[Address 2:] デバイスの設置場所を入力します。

[Address 3:] デバイスの設置場所を入力します。

[Contact Name:] デバイスの担当者名を入力します。

[Contact Phone:] デバイス担当者の連絡先電話番号を入力します。

[Contact E-mail:] デバイス担当者の連絡先 E メールアドレスを入力します。

[Contact Other:] その他の連絡先に関する情報を入力します。

[Alert e-mail:] このデバイスに設定されているアラームや警告を E メールで通知するユーザーのアドレスを入力します。

6.2.1.5 [Extra info]

[Serial number:] デバイスのシリアル番号を入力します。

[Asset Tag:] アセットタグ情報を入力します。

[Identification:] デバイスの ID 情報を入力します。

[Other info:] その他の情報を入力します。

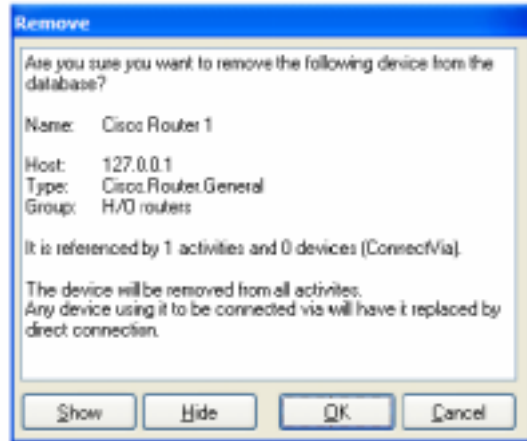
[Activity Specific1:] 特定のアクティビティのみに適用される情報を入力します。

[Activity Specific2:] 特定のアクティビティのみに適用される情報を入力します。

上記のフィールドはすべてフリーフォームなので、自由にテキストを入力できます。

6.2.2 [Remove]

選択したデバイスを削除するには[Remove]ボタンをクリックします。次のダイアログボックスが表示されます。



このウィンドウには選択したデバイスがアクティビティまたは他のデバイスで使用されているかどうか記載されています。[Show]ボタンをクリックするとデバイスが接続している媒体のリストが表示されます。

デバイスの削除を実行することを選択すると、そのデバイスとアクティビティや他のデバイスとの関連付けも削除されます。

6.2.3 [Edit]

選択したデバイスを編集するには[Edit]ボタンをクリックします。

[Device Information]画面が開きます。表示される項目やタブは[Add]ボタンを押したときと同じですが、ハイライト表示で選択されているデバイスの現在の設定内容が表示されています。

6.2.4 [Copy]

選択したデバイスをコピーするには[Copy]ボタンをクリックします。

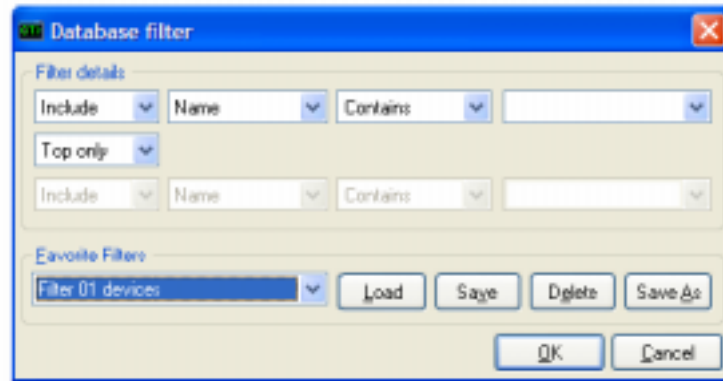
[Device Information]画面が開きます。表示される項目やタブは[Add]ボタンを押したときと同じですが、ハイライト表示で選択されているデバイスの現在の設定内容が表示されています。

最低でもデバイスの名前を変更してこの設定を新しいデバイスとして保存してください。

6.2.5 [Filter]

[Devices]タブに表示されるデバイスをフィルタするには[Filter]ボタンをクリックします。

ボタンをクリックすると、[Database Filter]画面が開きます。



[Filter details]には複数のドロップダウンリストが表示されています。1行目の一番左にあるフィールドには「Include」と「Exclude」の2つのオプションがあります。これに続くフィールドで指定する特定のデバイスタイプ、名前、その他のプロパティに該当するデバイスを含めるか除外するかを指定します。

左から2番目のフィールドには4つのオプションがあります。

1. Group
2. Name
3. Host name
4. Type

これらのオプションを選択することによって、様々な条件でデバイスを絞り込み、該当するデバイスを表示ウィンドウに含めたり除外したりすることができます。

3番目のフィールドではグループや名前等のどの部分をフィルタするかを指定します。3つのオプションがあります。

1. Contains - 指定した文字列がどこかに含まれているという条件でフィルタします。
2. Starts With - 指定した文字列で始まるという条件でフィルタします。
3. Ends With - 指定した文字列で終わるという条件でフィルタします。

一番右にあるフィールドはデバイス名、グループ等に対してフィルタする際に使用する文字列を指定します。

2行目には1つのフィールドのみが表示されています。このフィールドはドロップダウンリストで3つのオプションがあります。

1つ目と2つ目のオプションは単純なブール演算子で、フィルタ条件を AND と OR の論理演算によって接合します。

3つ目の[Top Only]は、1行目のフィルタオプションに入力した条件に該当するデバイスを上から順に表示します。

3行目に表示されるフィールドは1行目と全く同じで、1行目の条件と組み合わせるフィルタ条件を指定します。

[Favourite filters]では、作成したフィルタ条を後で再利用できるように保存することができます。ここに並んでいる各オプションについては一目瞭然なので特に詳しく説明する必要はないでしょう。

[Save As]は新しいフィルタとして保存します。

[Save]は変更したフィルタを保存します。

[Load]は指定したフィルタをロードします。

[Delete]は指定したフィルタを削除します。

6.2.6 [Show All]

[Show All]ボタンをクリックすると、フィルタが外されデータベースに登録されている全てのデバイスが表示されます。

6.3 [Activities]

このペインには定義済みの全てのアクティビティが表示されています。

右クリックするとカスタマイズしたり、主要機能にアクセスするためのメニューが表示されます。

6.3.1 [Add]

新しいアクティビティを追加するには[Add]をクリックします。

[Add a new scheduled activity]ウィンドウが表示されます。

6.3.2 [Edit]

選択したアクティビティを編集するには[Edit]ボタンをクリックします。

[Edit scheduled activity details]ウィンドウが表示されます。

6.3.3 [Copy]

選択したデバイスをコピーするには[Copy]ボタンをクリックします。

[Edit scheduled activity details]ウィンドウが表示されます。

6.3.4 [Run now]

選択されているハイライト表示のアクティビティを今すぐ実行するには[Run now]ボタンをクリックします。

6.3.5 [Start timer]

タイマーを起動し、スケジュール化されているアクティビティを実行するには[Start Timer]をクリックします。

タイマーが実行されている間はチェックされたアクティビティはすべて、各アクティビティに設定されている間隔でスケジュールどおりに実行されます。

6.3.6 [Remove]

現在選択されているアクティビティを削除するには[Remove]ボタンをクリックします。

6.4 [Activity log]

このペインには CatTools を最後に起動したとき以降に実行された全アクティビティのログが表示されています。

これらのログは CatTools のメインフォルダの下に activitylog.txt ファイルとしても保存されています。

6.4.1 [Clear]

表示されている全てのアクティビティ情報をクリアします。

これを実行してもアクティビティログファイルは削除されません。表示がクリアされるだけです。

6.4.2 [View report]

ハイライト表示されているアクティビティと関連付けられているレポートを表示します。このボタンをクリックすると自動的に[Report]ペインに移動し、レポートが表示されます。

6.5 [Compare]

このペインでは選択した2つのファイルの比較レポートを作成できます。

レポートの形式は、変更が検出された際に Device.Backup.Running Config アクティビティによって作成されるレポートと同じです。

6.5.1 [Save]

[Save]ボタンをクリックすると、表示されている比較データをファイルに保存することができます。

6.5.2 [Clear]

[Clear]ボタンをクリックすると、表示されている比較データがすべてクリアされます。T

6.5.3 [Select]

[Select]ボタンをクリックすると、比較する2つのファイルを選択する画面が表示されます。

6.6 [Info log]

このペインには CatTools の実行中に表示された全てのメッセージが発生順に表示されています。

ビューフィルタを使って表示をフィルタすることができます。

ここに表示されているメッセージは CatTools のメインフォルダの下にある infolog.txt ファイルにも記録されています。

6.6.1 ビューフィルタ(ドロップダウンリスト)

[Clear]ボタンの隣のドロップダウンリストから選択することによって、[Info log]に表示されるメッセージを指定するフィルタオプションを変更できます。このドロップダウンリストによって表示されるメッセージのレベルが指定されます。

ドロップダウンリストには次の4つのオプションがあります。

1. Show Error events only
2. Show Warning events and above
3. Show Info events and above
4. Show Debug events and above

デフォルト設定は「Show Info events and above」です。

6.6.2 [Clear]

[Clear]ボタンをクリックすると、[Info Log]画面の表示がクリアされます。

6.7 [Report]

[Report]ペインでは CatTools で作成されたテキストレポートを表示できます。

タブ区切りのテキストファイルも表示できます。

表示するファイル内で指定されているフィールドは変更可能で、必用であれば保存することもできます。

6.7.1 [Open]

レポートファイルを開くには[Open]ボタンをクリックします。

Explorer ウィンドウが表示されます。開きたいファイルを選択し、[Open]ボタンをクリックします。

内蔵のレポートビューワーによってタブ区切りファイルが開かれ[Report]ペイン内に表示されます。表示されたレポートはソート、編集、保存が可能です。

6.7.2 [Save]

現在、表示されているレポートをファイルに保存します。

6.7.3 [Clear]

[Report]画面をクリアします。

6.7.4 [Delete]

[Delete]ボタンをクリックすると、現在表示されているレポートを削除できます。

6.7.5 [Refresh]

[Refresh]ボタンをクリックすると、現在表示されているレポートファイルのビューがリフレッシュされます。この機能を使うと、ビューが開いている間にレポートファイルが更新されたときに、すぐさま変更を確認できます。

6.8 [TFTP]

このペインには CatTools の TFTP サーバーから送信されたメッセージが表示されています。TFTP サーバーを停止したり、始動することもできます。

6.8.1 [Start]

[Start]ボタンをクリックすると、まだ始動していない TFTP サーバーが始動します。

6.8.2 [Stop]

[Stop]ボタンをクリックすると実行中の TFTP サーバーが停止します。

6.8.3 [Clear]

[Clear]ボタンをクリックすると、TFTP サーバーから送信されたメッセージの表示がクリアされます。

6.9 [Display]

[Display]ペインには、CatTools に関連付けられている様々なタスクおよびクライアントごとに現在のメッセージが表示されています。

[Thread]列は発信元のタスクを示し、CatTools のメインタスクなのかクライアントのどれかなのかが表示されています。

[Device]列には現在そのタスクにリンクされているデバイスが表示されています。

[Status]列にはそのタスクから発信された最新のメッセージが表示されています。

6.9.1 [Clear]

[Clear]ボタンをクリックすると[Display]ペインに表示されている全メッセージがクリアされます。

6.10 [Mail]

[Mail]ペインには現在メールキュー内において送信待ち状態のメッセージが表示されています。

ウィンドウ上部のタイトルバーには、現在キューにあるメッセージの総数およびメーラーの現在のステータスが表示されています。

Kiwi Mailer はウィンドウ下部のボタンや右クリックで表示されるメニューから停止させたり開始させたりすることができます。送信前に E メールメッセージを変更する必要があるときやキューからメッセージを削除したいときはメーラーを停止させてください。

この一覧に表示されているメッセージに対し、必用であれば[To:]、[From:]、[Subject:]などのフィールドを個別にインラインで編集することができます。右クリックメニューから[Reset Retry flag]をクリックすれば、ステータスが「retry」になっている送信できなかったメッセージのステータスをリセットできます。

右クリックメニューには、各列の表示を切り替えることによって好みの表示方法にカスタマイズするためのオプションもあります。

メッセージを削除するには、削除する行を選択して右クリックし[Delete]を選択します。メッセージの[State]列にエラーのため送信不能を示す内容が記されている場合、そのメッセージを選択して[Reset State]を選択してステータスをリセットすることができます。こうすればメーラーはそのメッセージを再び送信できるようになります。

7 通知

CatTools にはイベント発生時にそのことを通知するための手段がいくつか用意されています。

- E メール

7.1 E メール

E メールを送ることは CatTools が選択したイベントの発生を知らせる際に使う手段の 1 つです。

CatTools 内蔵のメーラーエンジンは、アクティビティによって作成されたレポートやエラー通知を E メールで送信するための柔軟な機能を備えています。

このエンジンは通常、CatTools のメインタスクとは非同期で実行されます。CatTools によって E メール通知が作成されると、メーラーに渡されます。メー

ラーは通知に基づいてメッセージを作成しますが、すぐにメッセージを送信せず、CatTools で設定されている待ち時間中はそれをキューに入れておきます。CatTools ではメッセージがキューに入ると処理を続行します。設定された間隔でメーラーはメールサーバーに接続し、キューに入っているメッセージをメールサーバーに転送します。その間、CatTools のメインタスクは中断されずに実行状態を保ちます。

キューに入れることによって、メッセージは送信されるまで安全な状態を保つことができるのです。メールサーバーが何らかの理由によって使用できないときは、メッセージはメールサーバーが復旧するまでキューに保存されます。

メーラーはプライマリサーバーが使用できないときは、セカンダリメールサーバー経由でもメッセージを送信できます。

メールサーバーでの認証もサポートされています。ユーザー名とパスワードを使って、標準的に使用されている SMTP AUTH タイプの認証を行います。メーラーはメッセージを SMTP サーバーに送信する前にユーザー名とパスワードをリンク先の POP サーバーに送信することもできます。

詳細については[Mail]ペインの項を参照してください。

8 カスタムスクリプティング

v3.3.1 以降のバージョンの CatTools には、ユーザー独自のカスタムデバイスやアクティビティを作成してユーザーインターフェイスに追加するためのカスタムスクリプティング機能が付属しており、ご使用のデバイスが定義済みデバイスタイプの一覧にない場合や付属のアクティビティではニーズを満たせない場合でも自分でスクリプトを記述することによって対応可能になってります。

カスタムスクリプティングに関する詳細やカスタムファイルの作成手順、カスタムスクリプトのテンプレートの保存先などについては、それぞれ以下のページを参照してください。

- カスタムデバイススクリプトについては、「第 3 章 デバイス」の「3.1 カスタムデバイスを作成する」を参照してください。
- カスタムアクティビティとレポートのスクリプトについては、「第 4 章 アクティビティ」の「4.5 カスタムアクティビティの作成」を参照してください。

カスタムデバイスやアクティビティ、レポートなどを CatTools に追加するときは、スクリプトエディターを使ってスクリプトファイルを編集する必要があります。

8.1 スクリプトエディター

カスタムスクリプトファイルを変更するには、テキストファイルエディターかスクリプトエディターに読み込む必要があります。

Windows に付属のメモ帳(Notepad.exe)でもテキストファイルの編集は可能ですが、構文がハイライト表示(コードが別の色で表示)されるようなスクリプトエディターを使ったほうがスクリプトファイルの編集には適しています。

Visual Basic または Visual Studio がご使用のマシンにインストールされている場合は、どちらかのプログラムを起動してスクリプトファイルを開けば構文がハイライト表示されるので便利です。

このように構文がハイライト表示されるようなエディターを持っていない場合は、インターネットからダウンロードすることもできます。多種多様なエディターが公開されていますが、弊社では PSPad エディターを推奨します(このプログラムは <http://www.pspad.com/> からダウンロードできます)。

インストールが完了したら、次の手順で設定を変更します。カスタムスクリプト用の.txt ファイルや.custom ファイルを開くと構文がハイライト表示されるはずです。

- 1) メニューの[Settings]から[Highlighters settings]を選択します。
すべての言語チェックボックスをオフにし([None]ボタンをクリックします)、MS VBScript のみを選択します。
[Specifications]タブの[File Masks]ボックスに「*.txt」と「*.custom」を入力し、VBScript のファイルフィルタリストにこの2つのファイルタイプを追加します。
- 2) スクリプトファイルを開きメニューの[View]から[Change Syntax]を選択します。使用可能な構文の一覧から「MS VBScript」を選択し、[OK]をクリックすると構文の色分け表示が適用されます。コメント付きコード(カスタムスクリプト作成に役立つ情報として追加されたコード)は緑色で表示されます。

(* 上記の設定方法は 2008 年 1 月 14 日現在の PSPad バージョン 4.5.3 (Build 2298)の場合の手順です。詳細についてはダウンロードした PSPad のヘルプや Web サイトを参照してください。)

9 問題が発生したときの対応策

何か問題が発生したときは、本章を参照して解決策があるかどうかチェックしてください。弊社 Web サイト [CatTools Knowledge Base](#) にも様々な対応策が掲載されており、情報は日々更新されています。

それでも対応策がわからない場合は、CatTools 技術サポートチームまで弊社 Web サイトの[製品サポートフォーム](#)を使ってご連絡ください。

9.1 デバイス固有の問題について

デバイスの中には、CatTools がそのデバイスの動作を処理するために、特殊な設定や構成を必須とするものがあります。このようなデバイスの詳細については、「デバイス固有の情報」の項を参照してください。

9.2 [Info Log]に表示される "Error: 70 Permission denied"

Error: 70 Permission denied (starting client threads)

[Info Log]に"Problem starting CatTools_Client instance #1. Error:70: Permission denied"というエラーが表示されているときは、CatTools_Client DCOM コンポーネントを起動することを許可されていないユーザーが CatTools を実行していることを意味しています。

次の操作をすると、このエラーは表示されなくなります。

- [スタート]メニューの[ファイル名を指定して実行]から dcomcnfg.exe を実行します。
- [コンポーネントサービス]、[コンピュータ]、[マイコンピュータ]、[DCOM の構成]の順にクリックしていきます。
- [CatTools_Client.clsMain]または[CatTools_Client.KiwiSNMP]を選択します。どちらが表示されているかはご使用のシステムにインストールされ

ている CatTools のバージョンによって異なります。

- 右クリックして[プロパティ]を選択します。
- [セキュリティ]タブをクリックします。
- [起動とアクティブ化のアクセス許可]の[カスタマイズ]をオンにします。
- [編集]ボタンをクリックします。
- CatTools を実行しているユーザーの「ローカルからの起動」および「ローカルのアクティブ化」の許可をオンにします。

注：サービスは LocalSystem アカウントで実行されます。

設定を変更したらシステムを再起動し、CatTools がクライアントスレッドを正常に開始できるかどうかチェックしてください。

([Event Log]にはこのことが Event ID: 10016 の DCOM エラーとして表示されません。)

9.3 障害報告時のお願い

CatTools 技術サポートチームに障害報告をされる場合は、できるだけ多くの情報を提供してください。

例えば、デバイスにログオンする際にエラーが生じた場合、表示されたエラーの全文を知らせてください。エラーメッセージは[Info Log]に表示されており、その意味も記載されています。

その問題が発生したデバイスがどのタイプであるかといったことは非常に重要な情報です。CatTools では多種多様なデバイスをサポートしており、それぞれが独特なエラーを引き起こします。

infolog.txt ファイルを送信していただければ、そのエラーが発生したおおよその時間を知ることができます。1つのエラーメッセージに対するテキストファイルが 50MB 以上になりますと、そのファイルを確認する際に何かを見落とす危険が発生しやすくなります。

メッセージやセットアップエラーの生スクリーンショットを送信するのは、他に手段がない場合を除いてできるだけ避けてください。このようなデータはメールサーバーによってブロックされる確立が高く、メールサーバーが圧迫されます。CatTools の[File]メニューにある[Debug]を使ってサポートに送信するテキスト情報を作成し、これを送信してください。

infolog.txt 内のデータを送信する場合は、そのファイル自体を添付して送ってください。サイズが大きい場合は圧縮してください。infolog.txt からエントリを直接コピーアンドペーストで E メール本文に貼り付けると、文字化けする可能性があります。多くの場合、E メールに貼り付けられた情報よりも、その前後を含む情報を確認する必要があります。

infolog.txt を送信していただければ、最も早くご回答できます。

9.4 リモートデスクトップシステム

CatTools が、リモートデスクトップソフトウェアを使用しているマシン上で Client スレッドを実行しようとする、エラーが発生する可能性があります。

RealVNC および TightVNC

CatTools は (Windows Vista 以外の Windows システムで実行されている) TightVNC 1.3.9 および RealVNC 4.3.2 上でテストを実施した結果、問題がないことが確認されました。

これ以前の VNC ではクライアントスレッドの起動時にエラーが発生することがあります。古いバージョンの VNC には、クライアントスレッド起動時にエラーが発生するものがあります。これらのアプリケーションの一部の旧バージョンでは Windows システムを停止させ、クライアントが 1 つのスレッドしか実行できないようにしているようです。

そのような場合、まず VNC を現バージョンにアップグレードしてエラーが発生しなくなるかどうかチェックすることをお勧めします。

VNC ソフトウェアをアップグレードできない場合は、レジストリの変更を試みてください。RegEdit を起動し、use_Deferral の設定を 0 にします。

Key:

```
HKEY_CURRENT_USER\Software\ORL\VNCHooks\Application_Prefs\CatTools_Client.exe
```

```
Set Dword: use_Deferral to 0
```

Key:

```
HKEY_CURRENT_USER\Software\ORL\VNCHooks\Application_Prefs\CatTools.exe
```

```
Set Dword: use_Deferral to 0
```

PC Anywhere

CatTools は (Windows Vista 以外の Windows システムで実行されている) PCAnywhere 12.1 の評価版でテストを実施した結果、問題がないことが確認されました。

9.5 リモート認証

リモート認証を行うネットワーク上で CatTools を実行していると、時々不具合が生じることがあることが確認されています。スケジュール化されているアクティビティの実行時にデバイスにログオンしようとする、時折 CatTools が落ちるという現象です。次の指定時間にアクティビティが実行されるときには、同じデバイスに問題なくログオンできます。

この現象はネットワークの負荷が重いときやリモート認証システムのロード中に顕著に現れます。

デバイス側でタイムアウトまでのリモート認証サーバーの応答待ち時間を長く設置することによって、この現象を回避することができる場合があります。例えば、一部の Cisco デバイスの場合は次のコマンドを発行してタイムアウトを 20 秒間に設定できます (デフォルト値は 5 秒前後の値が設定されています)。

```
set tacacs timeout 20
```

9.6 アンチウイルスツール

一部のアンチウイルスツールには出入りどちらかのネットワークアクセスを遮断する機能が付属しています。

突然 CatTools がどのデバイスにも接続できなくなったり警告メールを送信するメールサーバーに接続できなくなったときは、アンチウイルスプログラムが遮断していないかどうか確認してください。

McAfee Enterprise V8.0 ではプログラムとポートのブロッキングを含めるよう設定できます。このような設定を行うと、CatTools でレポート送信用のメールサーバーに接続できなくなります。

9.7 XP ファイアウォール

XP システム上で CatTools を実行している場合、Service Pack 2 以降では XP ファイアウォールがデフォルトでオンになっていますので注意が必要です。XP ファイアウォールをオンにしたまま CatTools を実行するには、CatTools に対する例外設定を行うかファイアウォールが CatTools の処理を妨害しないよう設定を調整する必要がある場合があります。

9.8 サービスは稼働しているがスケジューラーが機能しない

CatTools は正常に実行されておりスケジューラーもオンになっているようだが、アクティビティが実行されないというケースが時々あります。

infolog ファイルを確認すればおそらく、アクティビティがハングアップし終了していないことが示されているはずです。この現象は論理ループが発生しているか通信がハングアップしており CatTools 側でタイムアウトを検出できないことが原因で発生しています。通常はこのようなことが毎回発生するということはありませんが、電流障害等の偶発的なトラブルで発生する可能性はあります。

[Kiwi CatTools Setup]画面の[Misc]タブに、このようなトラブルが発生した後も CatTools が正常に実行を続けられるようにするために役立つ設定項目 ([Terminate orphaned activity running longer than])があります。この項目を設定することによって、指定した待ち時間(分単位)経過後に孤立アクティビティを強制終了させることができます。強制終了後 CatTools はリカバリを実行しスケジュール化されているアクティビティを再起動します。

通常、問題が発生していない限りこの設定は 0 のままにしてください。長い実行時間を要するアクティビティを強制終了させずに最後まで実行させるためです。

この現象が発生する、もう一つの原因は DCOM コンポーネントの許可設定の問題である可能性があります。このことが原因であることを確認するには、[Info Log]タブに次のメッセージが表示されていないかどうかチェックしてください。

"Problem starting CatTools_Client instance #1. Error:70: Permission denied"

10 インストール手順

本章では、CatTools のインストール手順についてのヒントを記載します。

- CatTools の自動インストール
- インストール/アップグレード後の問題

10.1 CatTools の自動インストール

CatTools では、ユーザーが介在せずにソフトウェアのインストールと起動を自動的に行うことができます。

CatTools を標準の対話式アプリケーションとしてインストールし起動するには、次の情報が記述されたバッチファイルを作成する必要があります。

```
"*AppPath%Kiwi_CatTools_X.X.X.exe" /S INSTALL=APP /D=*InstallPath
"*InstallPath%CatTools.exe"
```

バッチファイルの指定例

```
"C:%Kiwi_CatTools_3.2.15_Beta.setup.exe" /S INSTALL=APP
/D="C:%Program Files%CatTools3"
"C:%Program Files%CatTools3%CatTools.exe"
```

CatTools を Windows サービスとしてインストールし起動するには、次の情報が記述されたバッチファイルを作成する必要があります。

```
"*AppPath%Kiwi_CatTools_X.X.X.exe" /S INSTALL=SERVICE
/D=*InstallPath
"*InstallPath%CatTools_Manager.exe"
```

バッチファイルの指定例

```
"C:%Kiwi_CatTools_3.2.15_Beta.setup.exe" /S INSTALL=SERVICE
/D="C:%Program
Files%CatTools3"
"C:%Program Files%CatTools3%CatTools_Manager.exe"
```

10.2 インストール/アップグレード後の問題

CatTools をインストールした後あるいはアップグレードした後に問題が生じた場合は、弊社 Web サイトの [Knowledge Base](#) の記事を参照してください。

それでも対応策がわからない場合は、CatTools 技術サポートチームまで弊社 Web

サイトの製品[サポートフォーム](#)を使ってご連絡ください。