# SSL サーバー証明書

SSL(Secure Socket Layer) とは、 Netscape Communications 社（米）が開発した安全な通信を行うためのプロトコルです。

**POINT**

＜SSL の必要性＞
現在、インターネットでは通信手段に「暗号化」の規定がないため、悪意と技術を持った第三者によって個人のプライバシーに関わる情報等のデータの盗聴や改ざんが行われる可能性があります。 インターネットは多くのネットワークを接続した巨大ネットワークであり、中継地点が多いほど覗かれる可能性も高くなるため、プライバシーに関する情報を送受信する場合は十分に注意が必要です。
SSL ではデータを暗号化して通信を行うため、第三者による盗聴や改ざんを防止し、安全にデータの送受信が行えます。


## コモンネームについて

コモンネームは SSL 接続の際にブラウザにアドレスとして入力する URL(FQDN) です。
SSL 接続の際、ブラウザは入力された URL(FQDN) とサーバー ID のコモンネームが一致しているかを検証します。 コモンネームとは異なる URL や IP アドレスなどでアクセスしたすると、警告メッセージが表示されます。
たとえばコモンネームを【example.jp】で取得した場合、 SSL 接続時に【www.example.jp】と入力されると警告メッセージが表示されます。
コモンネームのネーミングは自由ですが【www. 独自ドメイン名】が一般的です。
なお、一度コモンネームを取得団体に申請してしまうと、変更ができない場合があります。その場合、変更は再取得となり再度料金が発生します。

# 共用 SSL（無料）

CHM-Z プランでは共用 SSL がご利用いただけます。
お申し込みは不要です。

| 使用ドメイン | 共用 SSL |
| --- | --- |
| 主契約ドメイン | ご利用いただけます |
| バーチャルドメイン<br>(IP バーチャル) | 共用 SSL はネームバーチャル (SNI) での提供となります。 |
| バーチャルドメイン<br>(ネームバーチャル) | 共用 SSL はネームバーチャル (SNI) での提供となります。 |

アクセス用 URL はお客さまがご利用のドメイン名を元に生成されます。
実際の URL は、 CHM-Z のコントロールパネルにログインしていただき、「お客様情報」＞「プログラムのパスとサーバーの情報」のページ内の「共用 SSL URL」をご確認ください。

**POINT**

【アクセス用 URL】
https:// {お客様ドメイン名を一部変換} .cpi-common.jp

【変換例】
お申し込みドメインが example.com の場合
https://ssl-example-com.cpi-common.jp

お申し込みドメインが test-example.com の場合
https://ssl-test--example-com.cpi-common.jp

※ご利用制限
・お客様ドメイン名より変換された部分が63文字を超える場合、本機能はご利用いただけません。
・共用 SSL の URL は変更できません。
・共用 SSL はすべて SHA-2 で発行されます。
・日本語ドメインをご利用の場合共用 SSL はご利用いただけません。

# CPI SSL サーバー証明書

CPI SSL サーバー証明書は、自分のドメインで SSL を使いたい！という方にオススメです。
取得したドメインで SSL を使う場合にはデジタル ID の取得が必要です。CPI SSL サーバー証明書は、CPI が提供する SSL サーバー証明書の中でもっとも安価で、 256bit 対応の暗号強度を持ち、法人、個人どなたでも取得可能です。

# シマンテック SSL サーバー証明書（シマンテック セキュア・サーバー ID）

シマンテックの SSL サーバー証明書は、世界で最も支持されているサーバー証明書として、国内において中央省庁、地方自治体、大手金融機関など、より強固なセキュリティーを必要とされている企業に導入実績があります。

**POINT**

シマンテックのサーバー証明書取得申請の際には、正社員以上の方を責任者にする必要があります。
この場合の責任者とは、下記のいずれかに該当する社員を指します。

・シマンテックの申請・取得に際し、決裁権を持つ
・シマンテックの申請・取得に際し、管理の責を負う

上記の権限を持つ社員であれば、役職、肩書きは問いません。
ただし、「権限・責務の有無」については、合同会社シマンテック・ウェブサイトセキュリティが申請団体様の人事・総務担当に電話にて直接確認連絡を行い、「権限・責務があるか」の確認を行います。
なお、「自称」で「団体公認」ではない場合は審査ができませんのでご注意ください。

# シマンテック SSL サーバー証明書（シマンテック グローバル・サーバー ID）

シマンテックの SSL サーバー証明書は、世界で最も支持されているサーバー証明書として、国内において中央省庁、地方自治体、大手金融機関など、より強固なセキュリティーを必要とされている企業に導入実績があります。

シマンテック SSL サーバー証明書申請に際して、合同会社シマンテック・ウェブサイトセキュリティへの書類提出が必要な場合は、 CPI よりご連絡いたしますので、事前に書類をご準備になる必要はありません。

- 企業コードがある場合でも、合同会社シマンテック・ウェブサイトセキュリティで使用している帝国データバンク COSMOSNET2000 に基本情報が登録されていない団体の場合は、書類の提出が必要になる場合があります。
- 合同会社シマンテック・ウェブサイトセキュリティより電話にて申請責任者の在籍確認および申請意志確認を行います。
- 更新時にも更新申請の意思確認が必要になります。
- 法人企業のみ申請可能となります。個人での申請はできません。

# シマンテック SSL サーバー証明書（シマンテック セキュア・サーバー ID EV）

弊社ではシマンテック EV SSL サーバー証明書 セキュア・サーバー ID EV の取得を代行します。
SSL の機能はもとより、よく知られているからこそお客様に安心感を与えます。

シマンテック EV SSL サーバー証明書は、証明書を発行するプロセスがこれまでよりさらに強化されます。ブラウザのバーが緑色に変化するため、厳格な認証基準と手続きに従い発行された証明書であるうことをブラウザ上で一目で確認できるので、視覚的にも安全性をアピールできます。

**POINT**

シマンテックのサーバー証明書取得申請の際には、正社員以上の方を責任者にする必要があります。
この場合の責任者とは、下記のいずれかに該当する社員を指します。

シマンテックの申請・取得に際し、決裁権を持つ
シマンテックの申請・取得に際し、管理の責を負う

上記の権限を持つ社員であれば、役職、肩書きは問いません。
ただし、「権限・責務の有無」については、合同会社シマンテック・ウェブサイトセキュリティが申請団体様の人事・総務担当に電話にて直接確認連絡を行い、「権限・責務があるか」の確認を行います。
なお、「自称」で「団体公認」ではない場合は審査ができませんのでご注意ください。

**目次**

## 申請にあたって

### 申請可能な団体

- 日本に登記のある法人・団体
  └一般企業、財団法人、国立大学法人、学校法人、社団法人、組合、相互会社、その他法人などの単位
  「××大学」のような大学名での申請はできません
- 中央省庁および国の機関・地方公共団体およびその機関
  └「職員録」に記載のある団体または局単位

### 申請できない団体

- 法人登記されていない組織・任意団体（公共団体を除く）
- 日本国外で登記された組織・団体
- 個人事業主、個人

## 必要書類

シマンテック EV SSL サーバー証明書申請に際して、「利用規約同意書」「印鑑証明書」「申請責任者確認書」の三点の提出が必須となります。

- 「帝国データバンク企業コード」をリンク先の TDB 企業コード検索システム よりご確認ください。
- 企業コードがある場合でも、合同会社シマンテック・ウェブサイトセキュリティで使用している帝国データバンク COSMOSNET2000 に基本情報が登録されていない団体の場合は、書類の提出が必要になる場合があります。
- 合同会社シマンテック・ウェブサイトセキュリティより電話にて申請責任者の在籍確認および申請意志確認を行います。
- 更新時にも更新申請の意思確認が必要になります。

# シマンテック SSL サーバー証明書（シマンテック グローバル・サーバー ID EV）

弊社ではシマンテック EV SSL サーバー証明書 グローバル・サーバー ID EVの取得を代行します。
SSL の機能はもとより、よく知られているからこそお客様に安心感を与えます。

シマンテック EV SSL サーバー証明書は、証明書を発行するプロセスがこれまでよりさらに強化されます。ブラウザのバーが緑色に変化するため、厳格な認証基準と手続きに従い発行された証明書であるうことをブラウザ上で一目で確認できるので、視覚的にも安全性をアピールできます。

**POINT**

シマンテックのサーバー証明書取得申請の際には、正社員以上の方を責任者にする必要があります。
この場合の責任者とは、下記のいずれかに該当する社員を指します。

シマンテックの申請・取得に際し、決裁権を持つ
シマンテックの申請・取得に際し、管理の責を負う

上記の権限を持つ社員であれば、役職、肩書きは問いません。
ただし、「権限・責務の有無」については、合同会社シマンテック・ウェブサイトセキュリティが申請団体様の人事・総務担当に電話にて直接確認連絡を行い、「権限・責務があるか」の確認を行います。
なお、「自称」で「団体公認」ではない場合は審査ができませんのでご注意ください。

**目次**

## 申請にあたって

### 申請可能な団体

- 日本に登記のある法人・団体
  └一般企業、財団法人、国立大学法人、学校法人、社団法人、組合、相互会社、その他法人などの単位
  「××大学」のような大学名での申請はできません
- 中央省庁および国の機関・地方公共団体およびその機関
  └「職員録」に記載のある団体または局単位

### 申請できない団体

- 法人登記されていない組織・任意団体（公共団体を除く）
- 日本国外で登記された組織・団体
- 個人事業主、個人

## 必要書類

シマンテック EV SSL サーバー証明書申請に際して、「利用規約同意書」「印鑑証明書」「申請責任者確認書」の三点の提出が必須となります。

- 「帝国データバンク企業コード」をリンク先の TDB 企業コード検索システム よりご確認ください。
- 企業コードがある場合でも、合同会社シマンテック・ウェブサイトセキュリティで使用している帝国データバンク COSMOSNET2000 に基本情報が登録されていない団体の場合は、書類の提出が必要になる場合があります。
- 合同会社シマンテック・ウェブサイトセキュリティより電話にて申請責任者の在籍確認および申請意志確認を行います。
- 更新時にも更新申請の意思確認が必要になります。

# ジオトラスト SSL サーバー証明書

弊社ではジオトラスト Quick SSL Premium の取得を代行します。
Quick SSL Premium は、登記簿謄本などの書類が不要なので、個人事業主でも取得できます。

# グローバルサイン クイック認証 SSL

グローバルサイン クイック認証 SSL は、オンライン審査による取得が可能な SSL サーバー証明書です。
書類の郵送などの面倒な手続きは必要なく、郵送費用などのコストも削減できます。個人でも取得が可能となり、幅広い層の方にてご利用いただけます。

# セコムパスポート for WebSR3.0

ウェブサイト運営企業とホームページ利用者間のデータ通信を暗号化する機能と「その企業・組織の公開しているホームページが確実に存在していること」をセコムトラストシステムズが審査・確認し、実在を証明します。また、PC サイトはもちろん、全キャリアの携帯電話に対応しているので、携帯サイトの情報漏えい対策にも最適です。

# データベース

目次



## データベース情報

CHM-01Z と CHM-11Z ではバージョン情報などに違いがございます。
⇒ ご契約中のプランが CHM-01Z か CHM-11Z か確認する

## CHM-01Z

### MySQL5.5.42

| MySQL5.5.42 | |
|---|---|
| ホスト名 | 127.0.0.1 |
| ポート番号 | 3306 |
| データベース名 | ユーザー ID |
| ユーザー名 | ユーザー ID |
| パスワード | コントロールパネルの初期ログインパスワード |

## MySQL5.6.30

| MySQL5.6.30 | |
|---|---|
| ホスト名 | 127.0.0.1 |
| ポート番号 | 3307 |
| データベース名 | ユーザー ID |
| ユーザー名 | ユーザー ID |
| パスワード | コントロールパネルの初期ログインパスワード |

## PostgreSQL9.3.6

| PostgreSQL9.3.6 | |
|---|---|
| ホスト名 | 127.0.0.1 |
| ポート番号 | 5432 |
| データベース名 | ユーザー ID |
| ユーザー名 | ユーザー ID |
| パスワード | コントロールパネルの初期ログインパスワード |

# CHM-11Z

**重要**

CHM-11Z は、初期の状態ではデータベースは作成されておりません。
データベースをご利用いただく場合には、新規作成から行っていただく必要がございます。

## MySQL5.6.37

| MySQL5.6.37 | |
|---|---|
| ホスト名 | 127.0.0.1 |
| ポート番号 | 3306 |
| データベース名 | ユーザー ID_任意の文字列 |
| ユーザー名 | ユーザー ID |
| パスワード | コントロールパネルの初期ログインパスワード |

※追加された際は、「ユーザー ID_任意の文字列」となります。

## PostgreSQL9.5.9

| PostgreSQL9.5.9 | |
|---|---|
| ホスト名 | 127.0.0.1 |
| ポート番号 | 5432 |
| データベース名 | ユーザー ID |
| ユーザー名 | ユーザー ID |
| パスワード | コントロールパネルの初期ログインパスワード |

# MySQL データベースパスワード変更機能

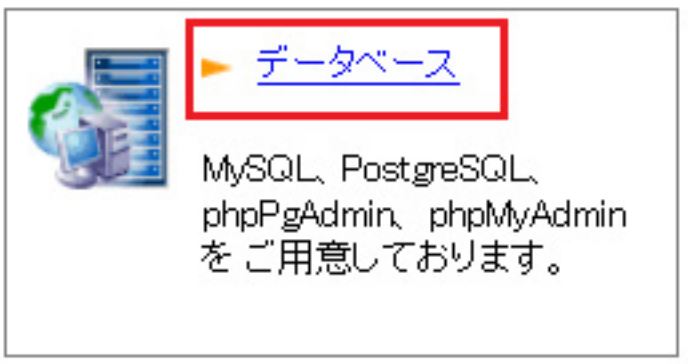
MySQL データベースのパスワードを変更できます。

**重要**

パスワードを変更しますと、MySQL を利用しているシステム側でもパスワードを変更する必要があります。

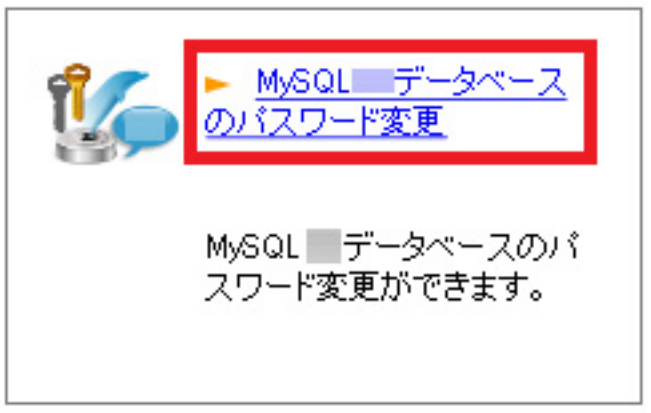
## データベースのパスワードの変更

**1** コントロールパネルの【制作ツール】から【データベース】をクリックします。

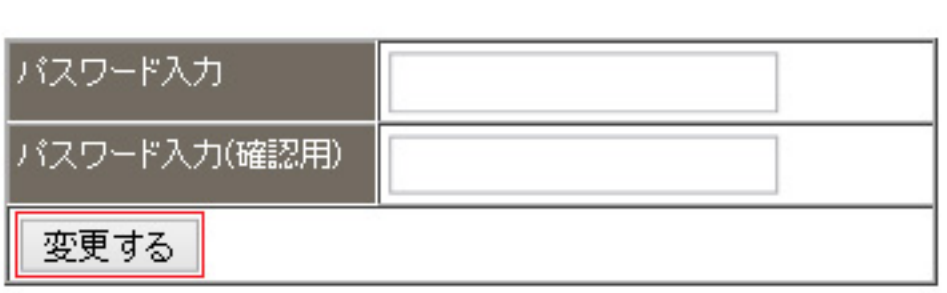



**2** パスワードを変更したいデータベースの【MySQL データベース設定】をクリックします。

## 3 【MySQL データベースのパスワード変更】をクリックします。




## 4 変更後のパスワードを入力し、【変更する】ボタンをクリックします。




**POINT**

◆パスワード設定に関しての制限は下記になります。

・文字数: 8～32 文字の文字列を設定してください。
・文字種：半角のアルファベット、数字、記号が利用できます。
　・ アルファベット：A～Z、a～z
　・ 数字：0～9
　・ 一部の半角記号：!#%&()*+,-./;<=>?@[]_{|}~^'"\

※2 バイト文字（日本語、全角など）は利用できません。

# MySQL データベースの追加・削除

MySQL データベースの追加・削除ができます。アプリケーション別に使い分けることも可能です。
設定できるデータベースの数は無制限です。

**重要**

弊社にてご用意している初期データベースの文字コードは、 EUC-JP となります。

**目次**



## データベースの追加

### 1 コントロールパネルの【制作ツール】から【データベース】をクリックします。

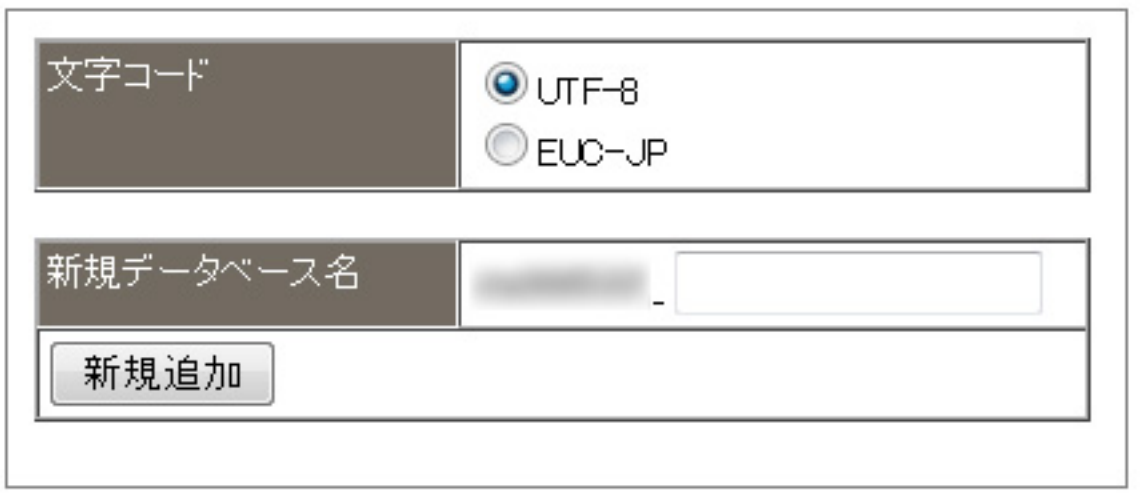

### 2 追加したいデータベースのバージョンを選選択します。

## 3 追加したいデータベースのバージョンの【MySQL データベースの追加・削除】をクリックします。


## 4 【データベース新規作成】ボタンをクリックします。


## 5 入力フォームに任意の名称を入力し【新規追加】ボタンをクリックします。

データベース名は「ユーザー ID_任意の文字列」となります。
入力文字数は 半角英数字 8 文字まで ご入力いただけます。



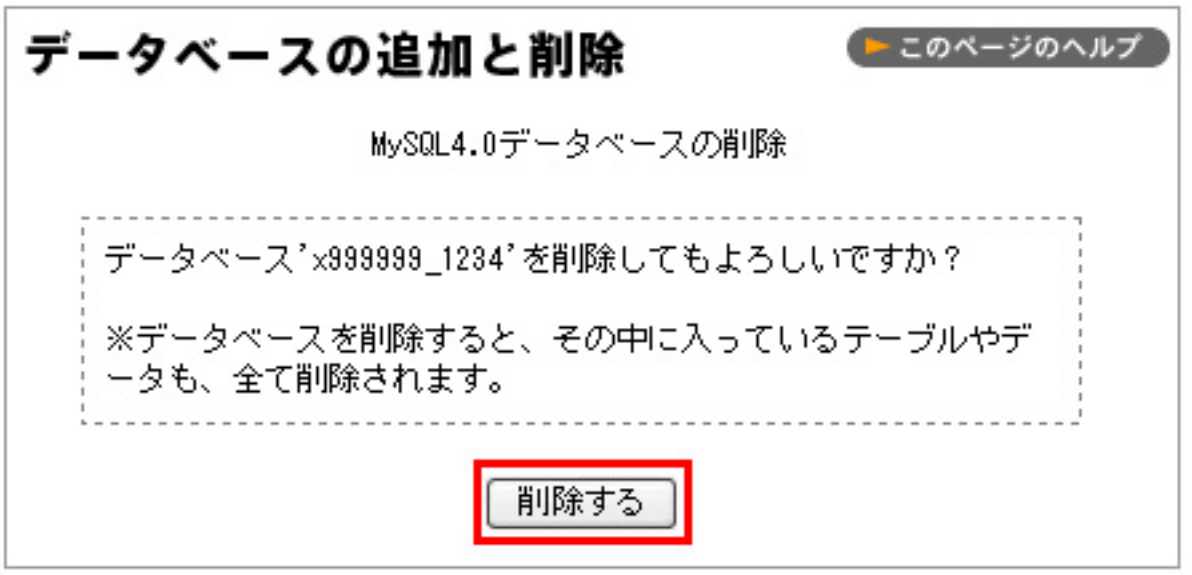
## 6 設定した名称でデータベースが追加されます。


# データベースの削除

## 1 削除するデータベースの【削除】ボタンをクリックします。

## 2 内容を確認のうえ【削除する】ボタンをクリックします。

削除したデータベースの復旧はできませんので、【削除する】ときは充分注意してください。




## 3 データベースが削除されます。

# MySQL 5.5 管理画面

MySQL をブラウザ上から管理するためのツールとして、 phpMyAdmin をご用意しています。
アクセスの際には、データベースの ID とパスワードを入力してください。
※phpMyAdmin はサポートの対象外となります。ご了承ください。

**重要**

MySQL 管理画面には、 Basic 認証が設定されております。
Basic 認証のユーザー ID／パスワードは、コントロールパネルにログインする際のユーザー ID／パスワードとなります。
コントロールパネルのログインパスワードを変更されますと、 Basic 認証のパスワードも連動して変更されますので、ご注意ください。

**POINT**

MySQL 管理画面の URL は以下の通りとなります。
■MySQL5.5 管理画面
https://お客様サーバー/public/database/phpMyAdmin/index.php
各バージョン間のデータの互換性および共通性はありません。それぞれ独立したデータとして存在します。

コントロールパネルの【制作ツール】タブをクリックし、【データベース】＞【MySQL 管理画面】のリンクをクリックします。

# MySQL 5.6 管理画面

MySQL をブラウザ上から管理するためのツールとして、 phpMyAdmin をご用意しています。
アクセスの際には、データベースのIDとパスワードを入力してください。
※phpMyAdmin はサポートの対象外となります。ご了承ください。

**POINT**

MySQL 管理画面の URL は以下の通りとなります。

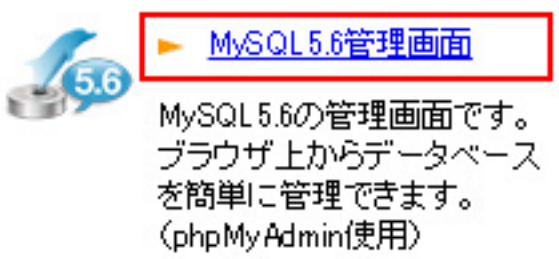
■MySQL5.6 管理画面
https://お客様サーバー名/public/database/phpMyAdmin56/index.php

各バージョン間のデータの互換性および共通性はありません。
それぞれ独立したデータとして存在します。

**重要**

MySQL 管理画面には、 Basic 認証が設定されております。
Basic 認証のユーザー ID／パスワードは、コントロールパネルにログインする際のユーザー ID／パスワードとなります。
コントロールパネルのログインパスワードを変更されますと、 Basic 認証のパスワードも連動して変更されますので、ご注意ください。

コントロールパネルの【制作ツール】タブをクリックし、【データベース】→【MySQL 管理画面】のリンクをクリックします。



# PostgreSQL データベースパスワード変更機能

**重要**

パスワードを変更しますと、PostgreSQL を利用しているシステム側でもパスワードを変更する必要にあります。

## データベースのパスワードの変更

### 1 【データベース】をクリックします。

► データベース

MySQL、PostgreSQL、phpPgAdmin、phpMyAdmin を ご用意しております。

### 2 PostgreSQL データベース設定 をクリックします。

► PostgreSQL データベース設定

データベースのパスワード変更、管理画面(phpPgAdmin)がご利用いただけます。

### 3 PostgreSQL データベースのパスワード変更をクリックします。

► PostgreSQL データベースのパスワード変更

PostgreSQL データベースのパスワード変更ができます。

## 4 変更をしたパスワードを入力し【変更する】ボタンをクリックします。

| | |
|---|---|
| パスワード入力 | |
| パスワード入力(確認用) | |
| 変更する | |

**POINT**

◆パスワード設定に関しての制限は下記になります。

・文字数: 8～32 文字の文字列を設定してください。
・文字種：半角のアルファベット、数字、記号が利用できます。
　・アルファベット：A～Z、a～z
　・数字：0～9
　・一部の半角記号：!#%&()*+,-./;<=>?@[]_{|}~^'"\

※2 バイト文字（日本語、全角など）は利用できません。

# PostgreSQL データベースの追加・削除

PostgreSQL データベースの追加・削除ができます。アプリケーション別に使い分けることも可能です。
設定できるデータベースの数は１件です。

**重要**

CHM-11Z をご利用のお客さまは、PostgreSQL データベースをご利用される場合には、追加をしていただく必要がございます。

CHM-01Z のお客さまは、あらかじめ作成した状態となっており、追加・削除を行う事ができません。

CHM-01Z と CHM-11Z ではバージョン情報などに違いがございます。

**目次**



## データベースの追加

### 1 コントロールパネルの【制作ツール】から【データベース】をクリックします。

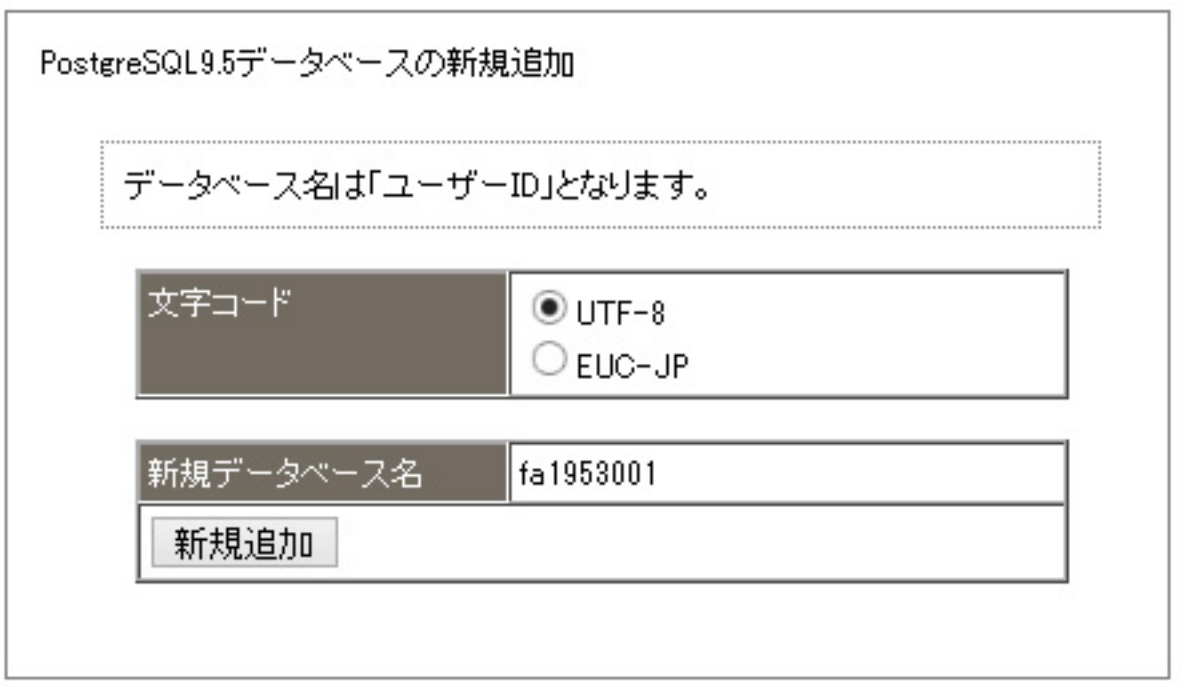

### 2 追加したいデータベースのバージョンを選選択します。

## 3 追加したいデータベースのバージョンの【PostgreSQL データベースの追加・削除】をクリックします。

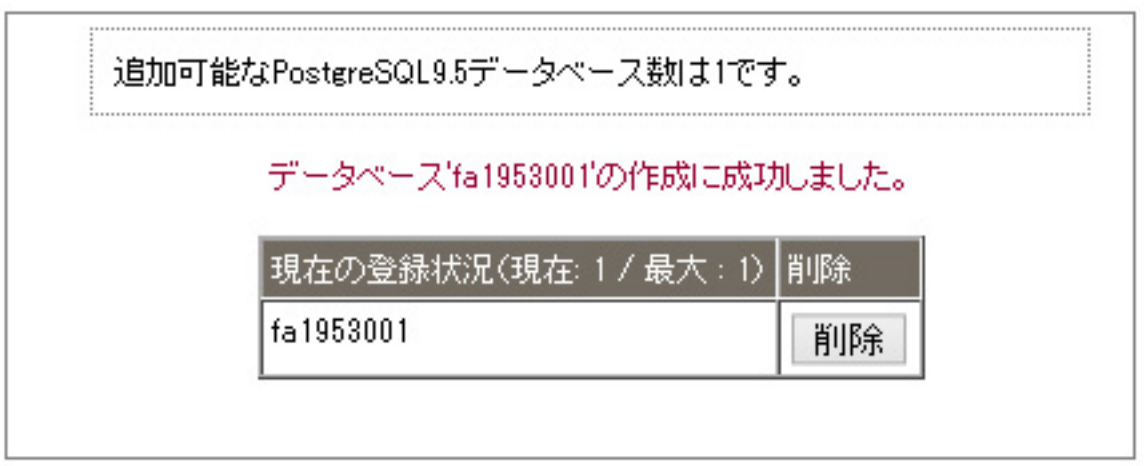

## 4 【データベース新規作成】ボタンをクリックします。


## 5 【新規追加】ボタンをクリックします。

データベース名は「ユーザー ID」となります。

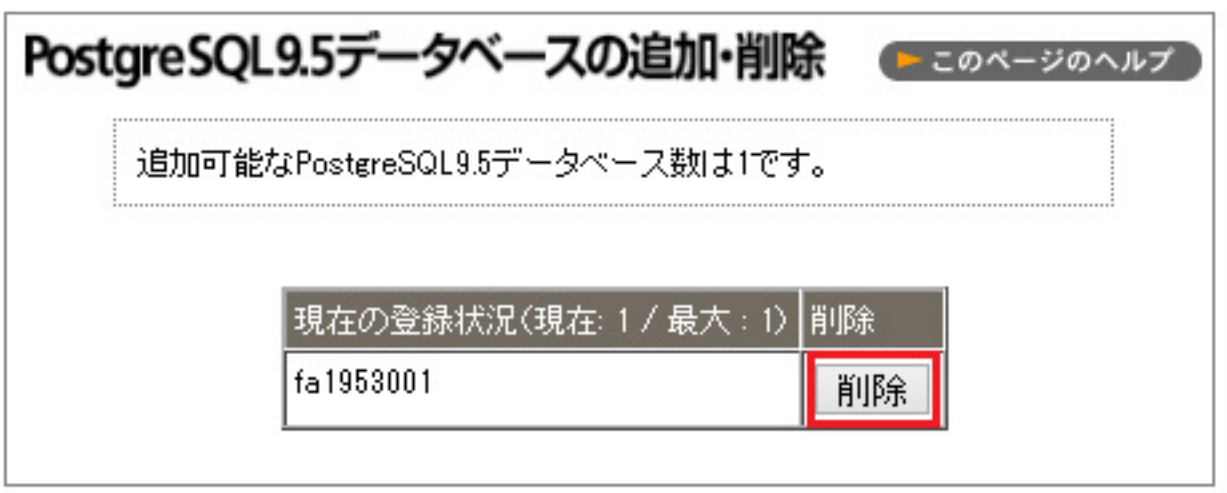



## 6 設定した名称でデータベースが追加されます。



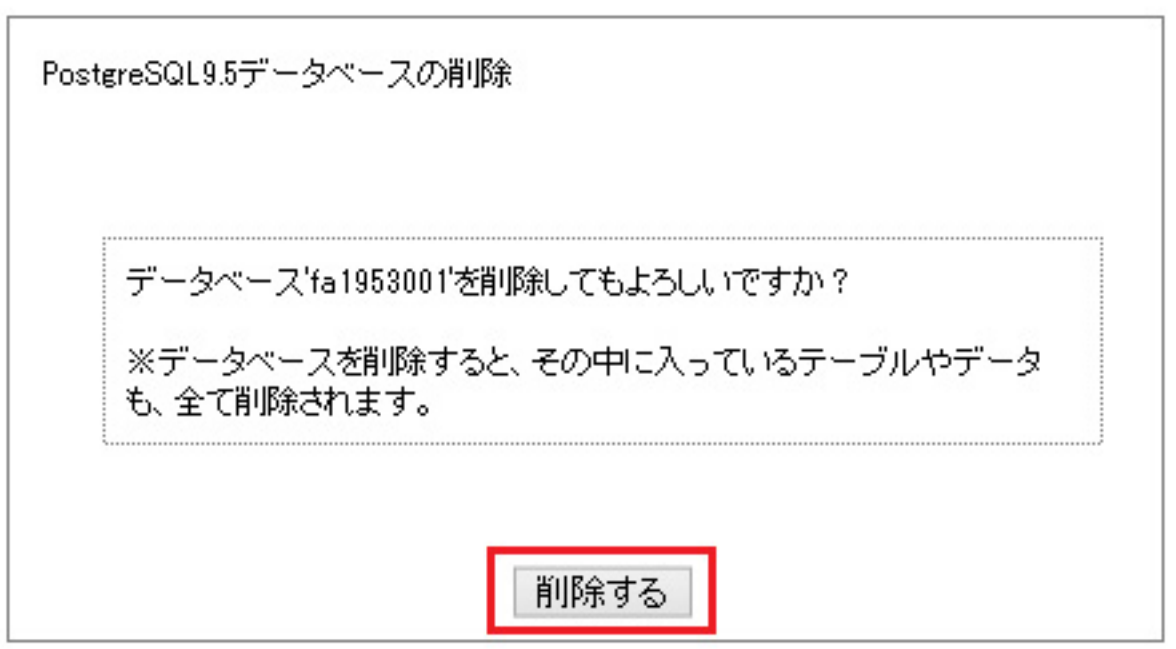
# データベースの削除

## 1 削除するデータベースの【削除】ボタンをクリックします。

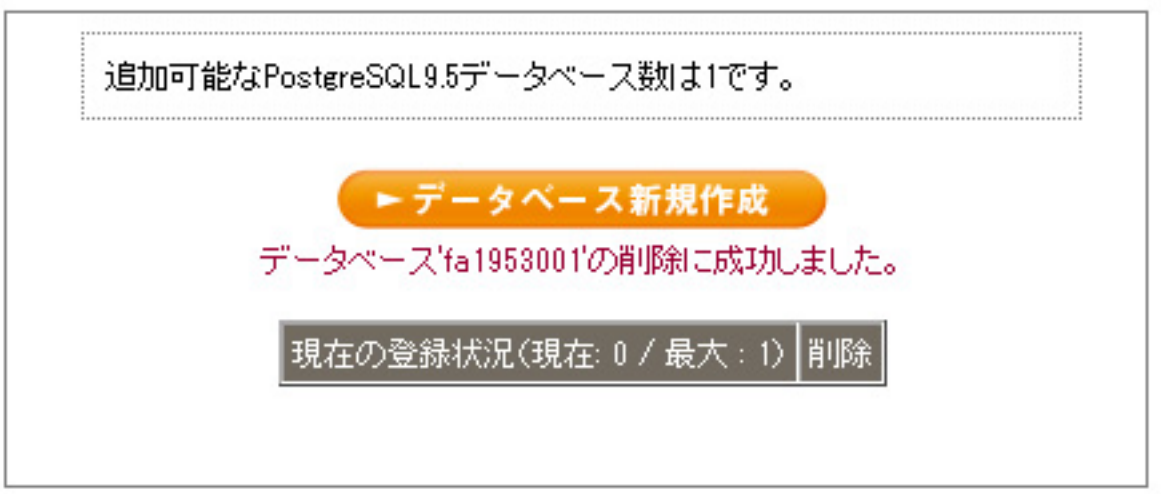



## 2 内容を確認のうえ【削除する】ボタンをクリックします。

削除したデータベースの復旧はできませんので、【削除する】ときは充分注意してください。




## 3 データベースが削除されます。



# PostgreSQL 9.3 管理画面

PostgreSQL をブラウザ上から管理するためのツールとして、 phpPgAdmin をご用意しています。
アクセスの際には、データベースの ID とパスワードを入力してください。
※phpPgAdmin はサポートの対象外となります。ご了承ください。

**POINT**

■PostgreSQL9.3 管理画面
https://お客様サーバー名/public/database/phpPgAdmin/index.php

各バージョン間のデータの互換性および共通性はありません。
それぞれ独立したデータとして存在します。

※こちらのバージョンは「CHM-01Z」のみの提供となります。

**重要**

PostgreSQL 管理画面には、 Basic 認証が設定されております。
Basic 認証のユーザー ID／パスワードは、コントロールパネルにログインする際のユーザー ID／パスワードとなります。
コントロールパネルのログインパスワードを変更されますと、 Basic 認証のパスワードも連動して変更されますので、ご注意ください。

CHM-01Z と CHM-11Z ではバージョン情報などに違いがございます。
⇒ご契約中のプランが CHM-01Z か CHM-11Z か確認する

コントロールパネルの【データベース】＞【PostgreSQL9.3データベース設定】＞【PostgreSQL●管理画面】をクリックすると phpPgAdmin のログイン画面にアクセスできます。
必要に応じてブックマークに追加してください。

PostgreSQL9.3管理画面

PostgreSQL9.3の管理画面です。ブラウザ上からデータベースを簡単に管理できます。（phpPgAdmin使用）

# PostgreSQL 9.5 管理画面

PostgreSQL をブラウザ上から管理するためのツールとして、 phpPgAdmin をご用意しています。
アクセスの際には、データベースの ID とパスワードを入力してください。
※phpPgAdmin はサポートの対象外となります。ご了承ください。

※こちらのバージョンは「CHM-11Z」のみの提供となります。

**POINT**

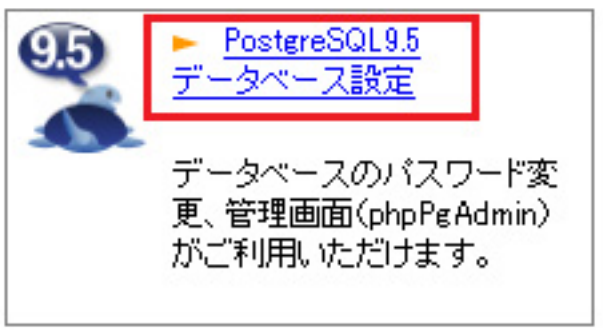
■PostgreSQL9.5 管理画面
https://お客様サーバー名/public/database/phpPgAdmin95/index.php
各バージョン間のデータの互換性および共通性はありません。
それぞれ独立したデータとして存在します。

**重要**

PostgreSQL 管理画面には、 Basic 認証が設定されております。
Basic 認証のユーザー ID／パスワードは、コントロールパネルにログインする際のユーザー ID／パスワードとなります。
コントロールパネルのログインパスワードを変更されますと、 Basic 認証のパスワードも連動して変更されますので、ご注意ください。

CHM-01Z と CHM-11Z ではバージョン情報などに違いがございます。
⇒ご契約中のプランが CHM-01Z か CHM-11Z か確認する

コントロールパネルの【データベース】＞【PostgreSQL9.5データベース設定】＞【PostgreSQL9.5管理画面】をクリックすると phpPgAdmin のログイン画面にアクセスできます。
必要に応じてブックマークに追加してください。



# 外部バックアップアカウントの設定

外部バックアップアカウントの設定では、外部バックアップサーバーへ接続するための FTP アカウントを設定します。

**POINT**

■作成可能な FTP アカウント数は 1 個です。
■FTP アカウントとパスワードの変更ができません。パスワードを変更したい場合には、アカウントを一度削除して再作成する必要があります。
■また、FTP アカウントを削除すると「接続元制限」にて設定している設定内容も削除されますので、再度接続元制限の設定を行ってください。

## FTP アカウントの作成

**1** コントロールパネルの【制作ツール】から、【外部バックアップサービス】＞【外部バックアップアカウントの設定】の順番でクリックします。


**2** 【FTP アカウント新規作成】ボタンをクリックします。

# FTP アカウントの設定変更・削除

外部バックアップアカウントの設定では、 FTP アカウントとパスワードの変更ができません。パスワードを変更したい場合には、アカウントを一度削除して再作成する必要があります。

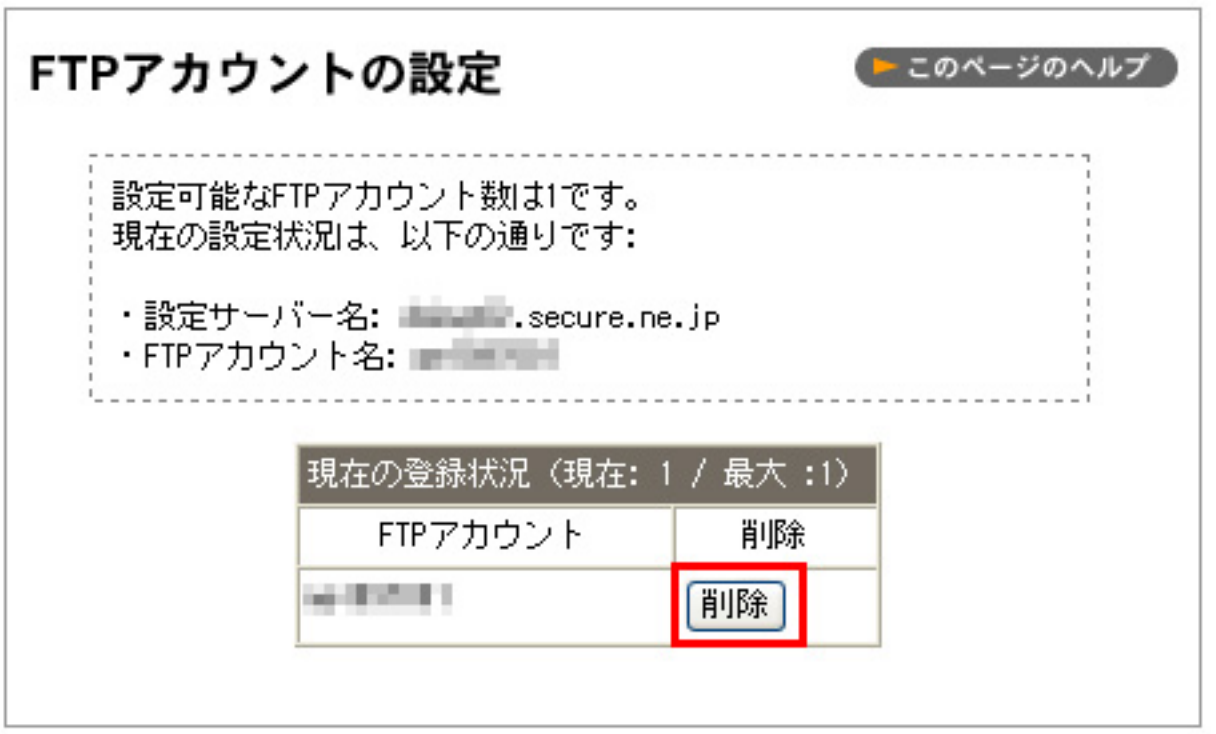
**重要**

FTP アカウントを削除すると「接続元制限」にて設定している設定内容も削除されますので、再度接続元制限の設定を行ってください。

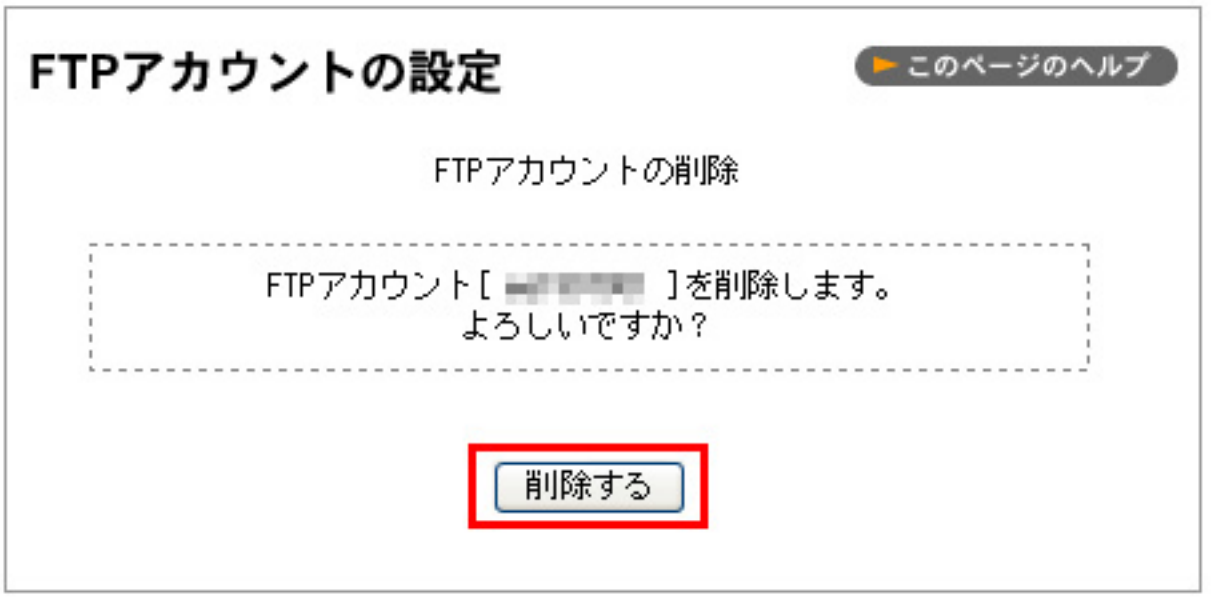
## 1 コントロールパネルの【外部バックアップアカウントの設定】から【FTP アカウントの設定】をクリックします。


## 2 FTP アカウントの【削除】ボタンをクリックします。

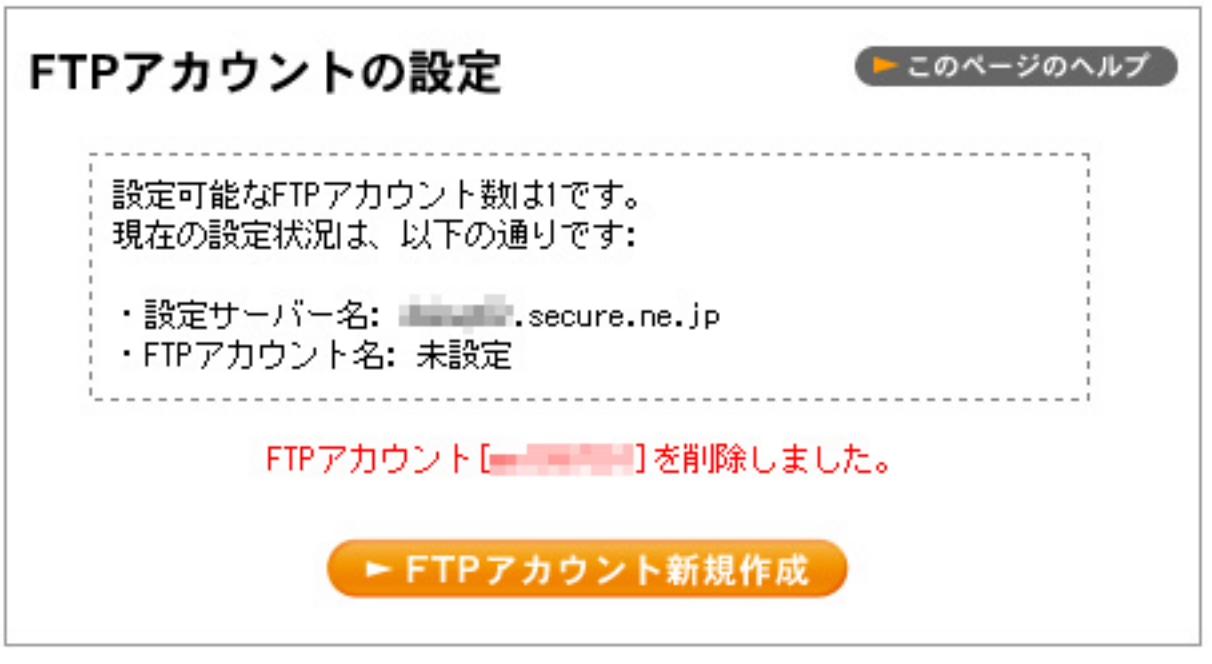
パスワードを変更したい場合、一度削除を行わないと再設定ができません



## 3 確認画面が表示されますので【削除する】ボタンをクリックします。




## 4 削除されると【FTP アカウント［*****］を削除しました。】と表示されます。

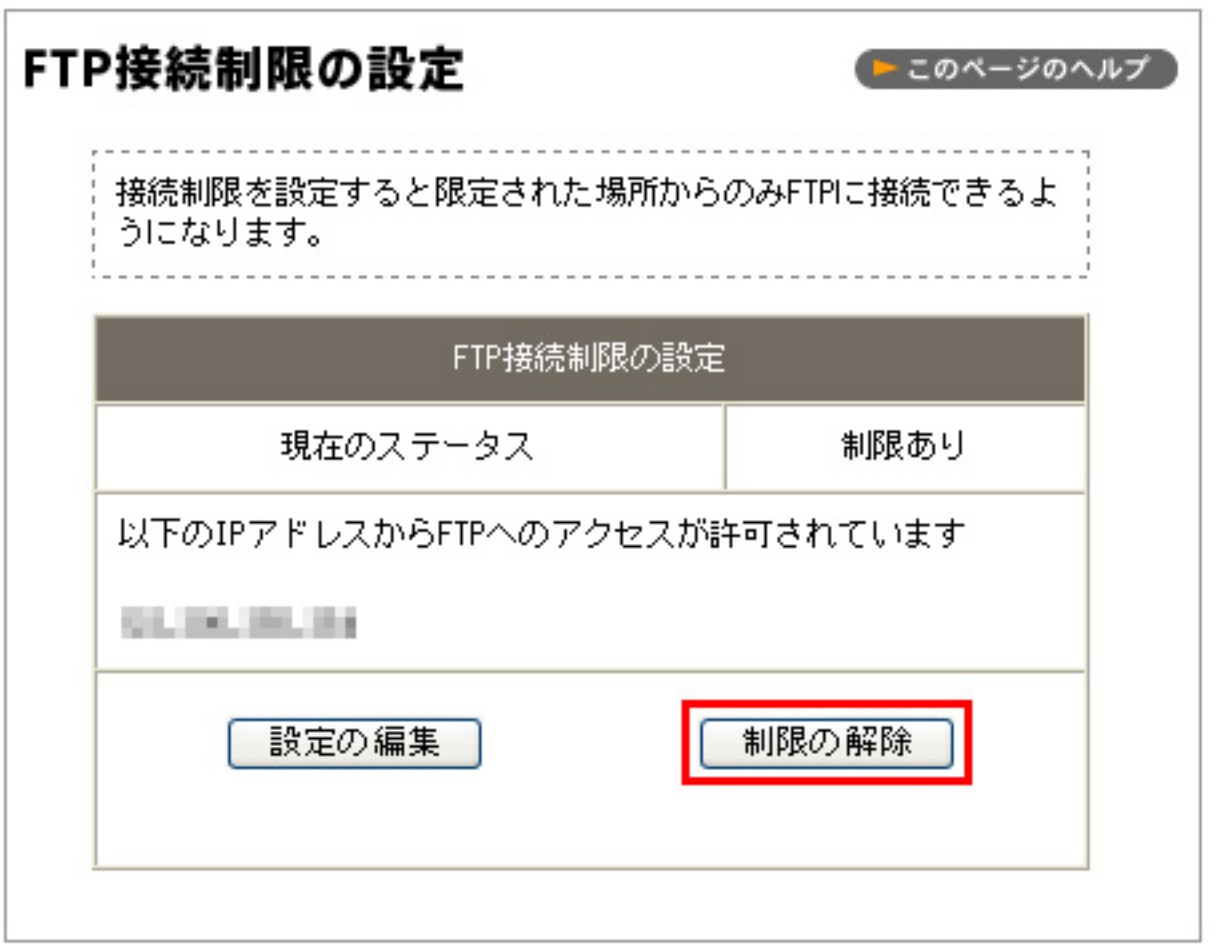


# FTP 接続制限の解除

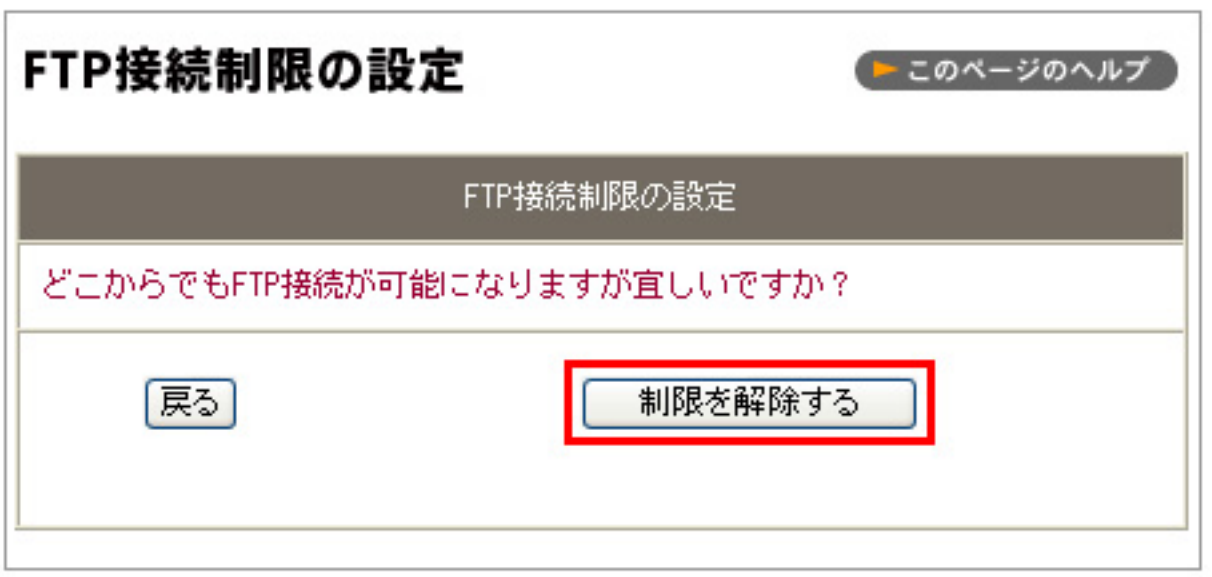
## 1 コントロールパネルの【制作ツール】から【外部バックアップサービス】⇒【外部バックアップ接続制限の設定】の順番でクリックします。


## 2 「制限の解除」をクリックします。

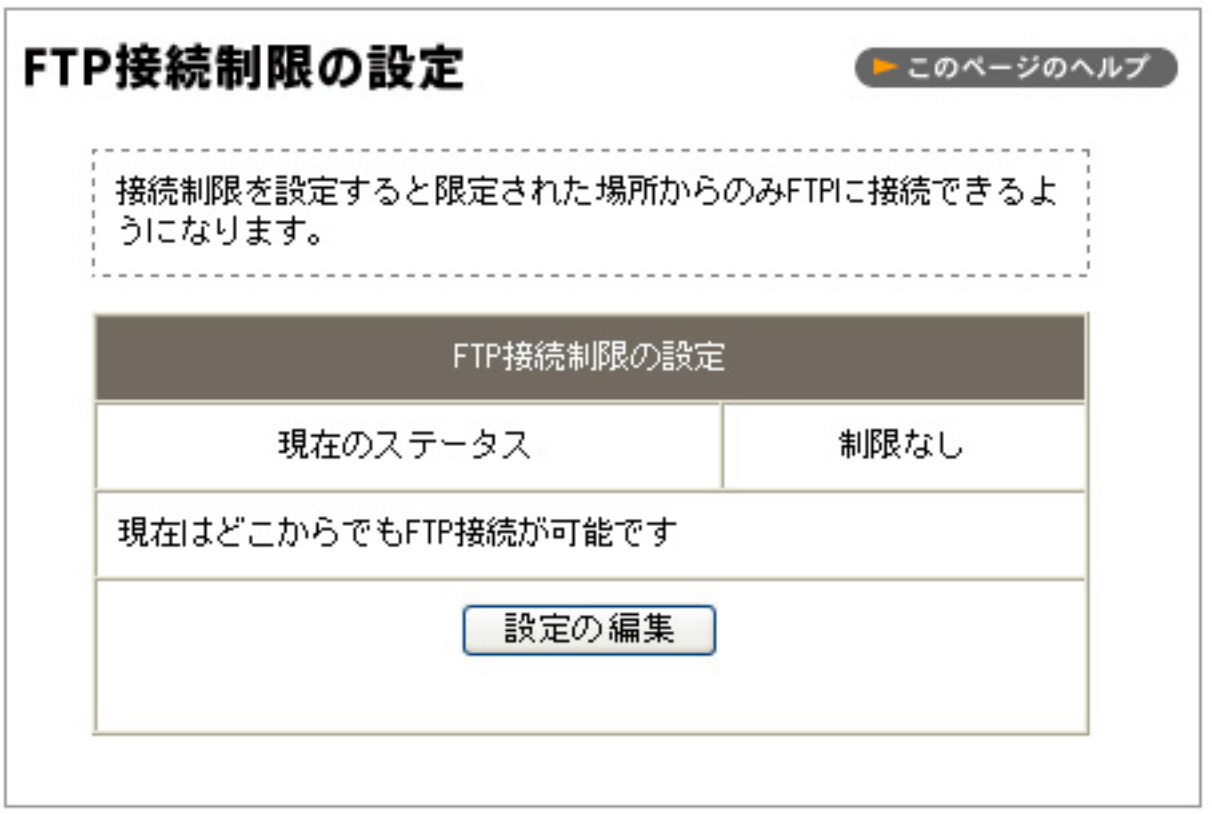
「FTP 接続制限」が有効になっていると、「現在のステータス」は「制限あり」「以下の IP アドレスから FTP へのアクセスが許可されています」となり、設定されている IP アドレスが表示されています。



3 「どこからでもFTP接続が可能になりますが宜しいですか？」と表示されますので「制限を解除する」をクリックします。




4 解除されると、「現在のステータス」は「制限なし」「現在はどこからでもFTP接続が可能です」に変わります。



# FTP 接続制限の設定

コントロールパネルから設定した IP アドレス以外からの接続を拒否する機能です。
これにより、万が一 FTP アカウント名やパスワードが外部に流出したとしても、設定した IP アドレスでなければ外部バックアップサーバーへ接続できなくなります。
「誰でもどこからでも接続できる」というインターネット本来の利便性は失われますが、例えば本部と特定の支社からしか外部バックアップサーバーへ接続できないといったような、クローズドな外部バックアップ環境が構築できます。
外部バックアップサービスをよりセキュアに使いたい方にお勧めの機能です。

FTP 接続制限を設定されますと、 FTP の他に、 SFTP も同じ設定が反映されます。

**POINT**

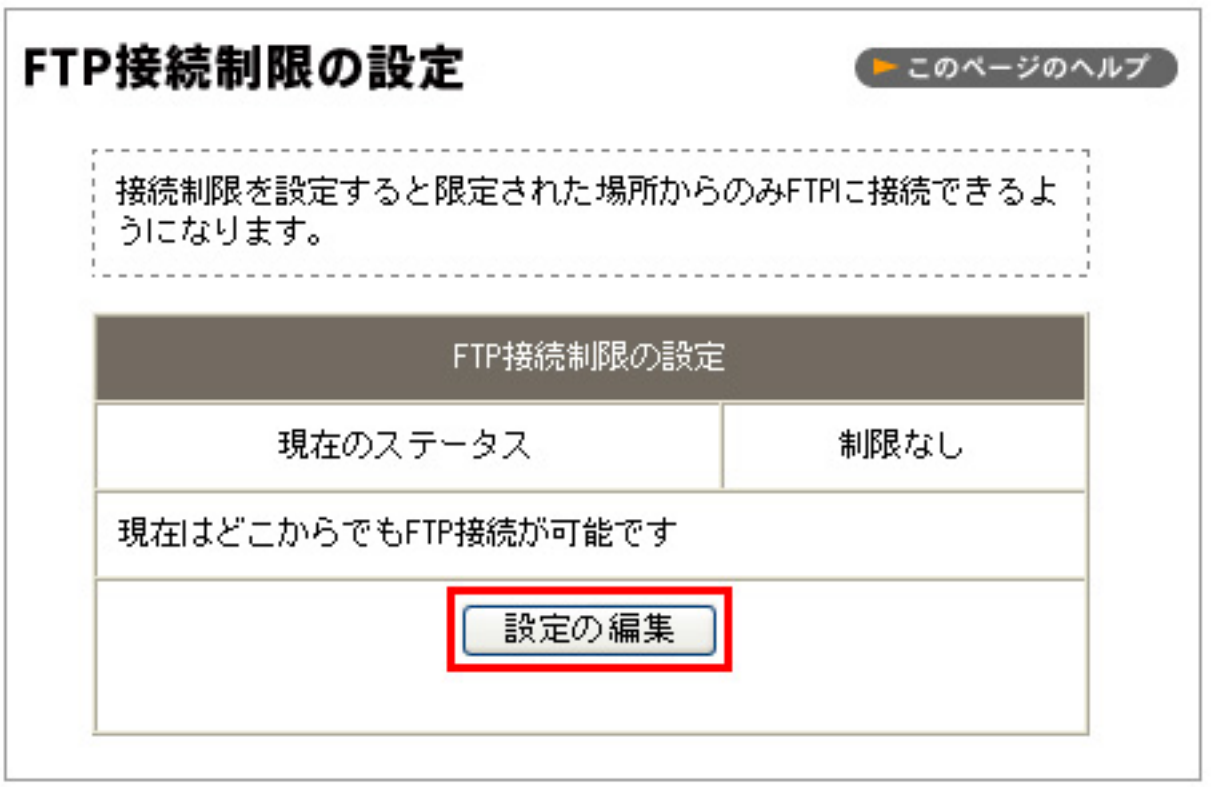
■設定した IP アドレスからしか接続できなくなります。
■IP アドレスでの制限のみ有効です。ホスト名での制限には対応していません。

## FTP 接続制限の設定

### 1 コントロールパネルの【制作ツール】から【外部バックアップサービス】＞【外部バックアップ接続制限の設定】の順番にクリックします。

## 2 【設定の編集】ボタンをクリックします。

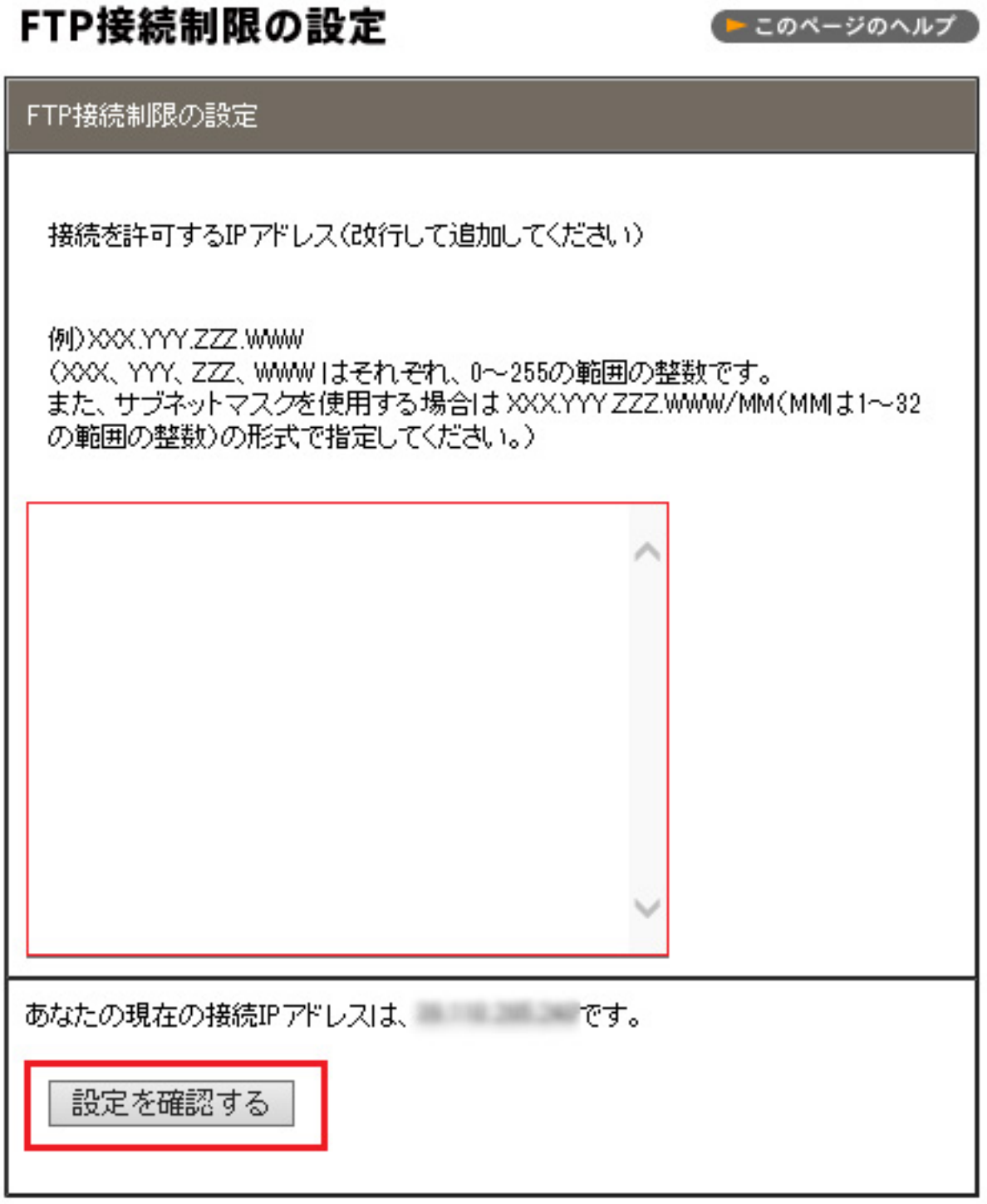
なにも設定されていない場合は、「現在のステータス」は「制限なし」「現在はどこからでもFTP接続が可能です」になっています。



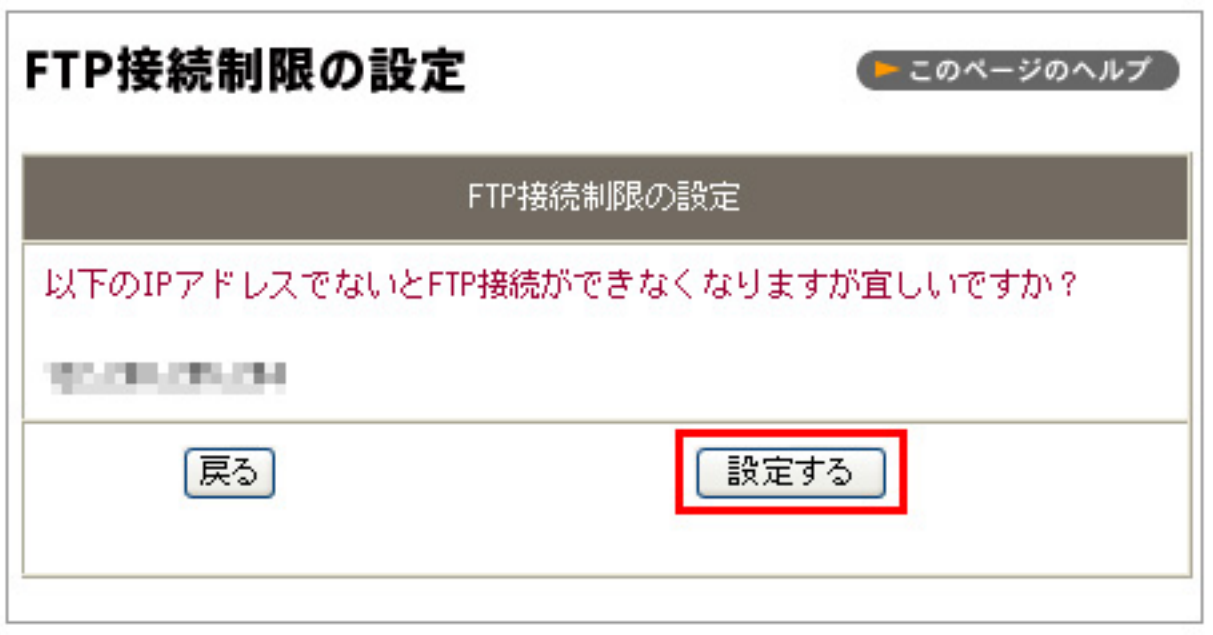

## 3 テキストエリアに「接続を許可するIPアドレス」を入力し、「設定を確認する」をクリックします。

**POINT**

・現在の利用しているネットワークのIPアドレスは、フォーム下に「あなたの現在の接続IPアドレスは ***** です」と表示されます。
改行して追記することで複数のIPアドレスを登録できます。
・ホスト名での制限には対応しておりません。
・【192.168.0.】のようなレンジ指定や、【192.168.0.1/24】のようなセグメント指定でも設定できます。

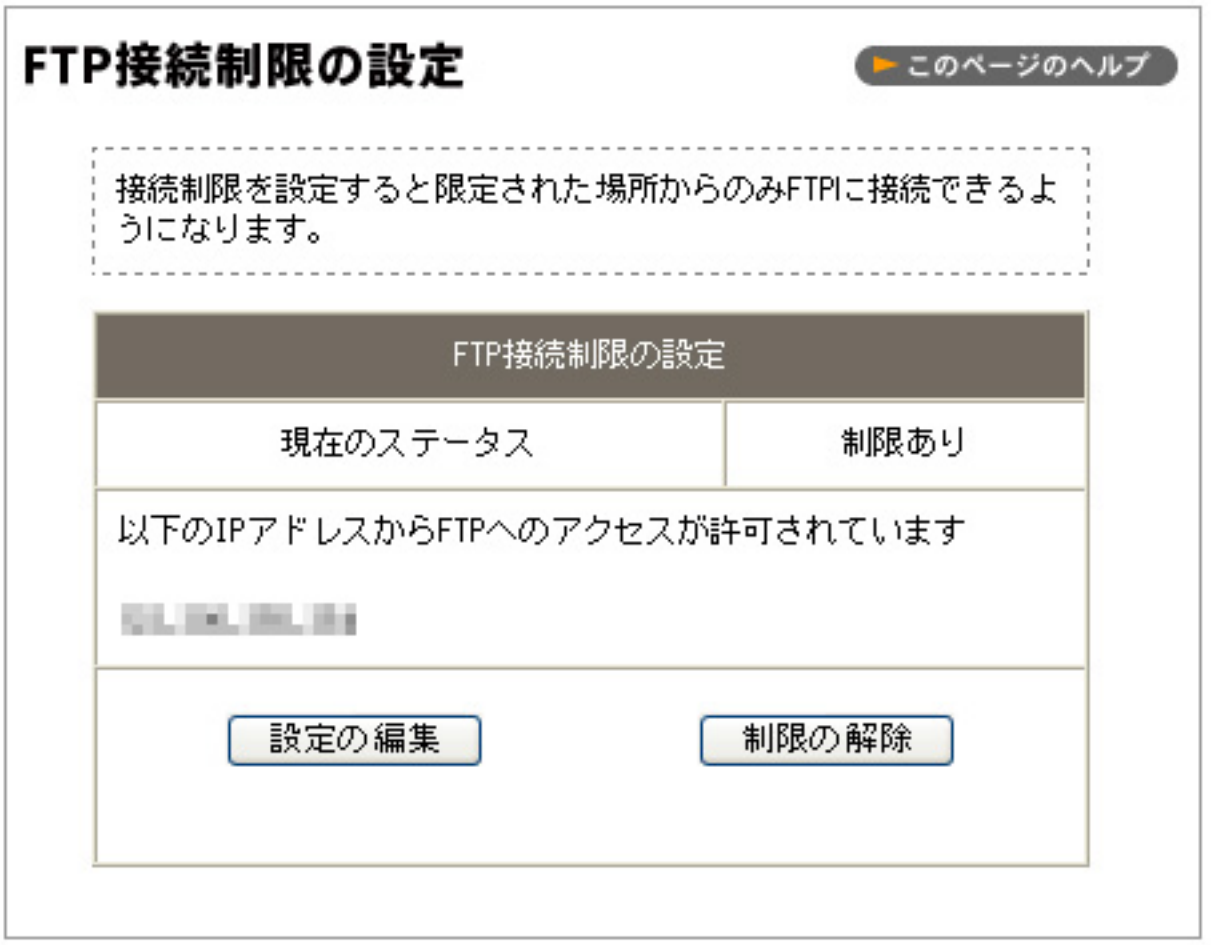



4 外部バックアップサーバーへ FTP 接続を許可する IP アドレスを確認のうえ「設定する」をクリックします。



## 5 設定完了

「現在のステータス」は「制限あり」、設定されている IP アドレスが表示されます。



# ドメインエイリアス

ドメインエイリアスをご利用いただきますと、ご契約ドメインのサイトを別のドメインでも表示させることができます。
たとえば、「example.jp」のサイトへ「example.co.jp」や「example.ne.jp」などの別ドメインでのアクセス時に同一のトップページを表示させます。

ドメインエイリアスの申し込みは、マイページからお手続きください。

**重要**

■本機能で設定したドメインではメールをご利用いただくことはできません。
■バーチャルドメインでは本機能をご利用いただけません。
■サブドメインで本機能をお申し込みすることはできません。
■本機能は CPI の DNS サーバーを使用されている場合にご利用いただけます。
弊社以外の DNS サーバーを使用される場合にはご利用いただけません。
■本機能では「www.ドメイン名」「ドメイン名」の２つでアクセスが可能となります。
例：「example.com」で本機能をお申し込みの場合
http://www.example.com/
http://example.com/
のどちらでアクセスしてもご契約ドメインのサイトを表示させることができます。

# アクセス制御（BASIC 認証）

アクセス制御（BASIC 認証）は、特定のディレクトリの閲覧に対して ID とパスワードを設定する機能です。

## アクセス制御の設定

アクセス制御（BASIC 認証）では、ディレクトリ単位でアクセス制御を設定できます。 アクセス制御のかけられたディレクトリ内を閲覧するには ID とパスワードが必要になりますので会員制のページなどに使うと便利です。
アクセス制御を設定したいディレクトリは、あらかじめ FTP クライアントアプリケーションで作成しておきます。

### アクセス制御の設定

アクセス制御は一つのディレクトリに複数のユーザーを設定することも可能です。

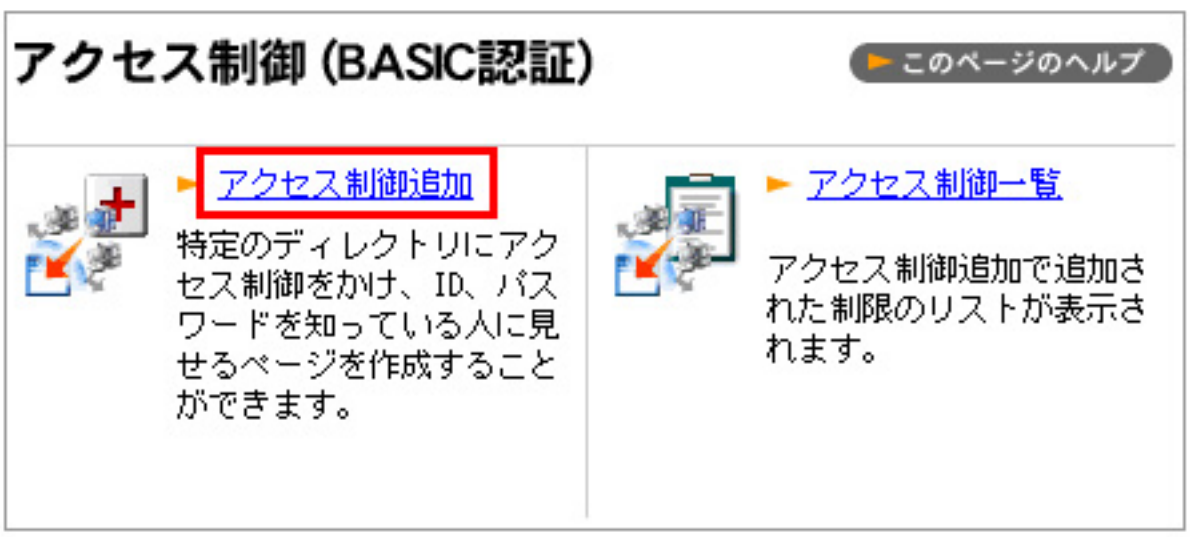
**POINT**

■登録したパスワードを確認することはできません。 パスワードを忘れたとき、または変更したいときは、同じユーザー名で設定しなおしてください。
■弊社のアクセス制御機能は「.htaccess」、「.htpasswd」にて制御しております。 お客様にて「.htaccess」、「.htpasswd」を設置されている場合、コントロールパネルからの当機能の設定に影響を与える場合があります。

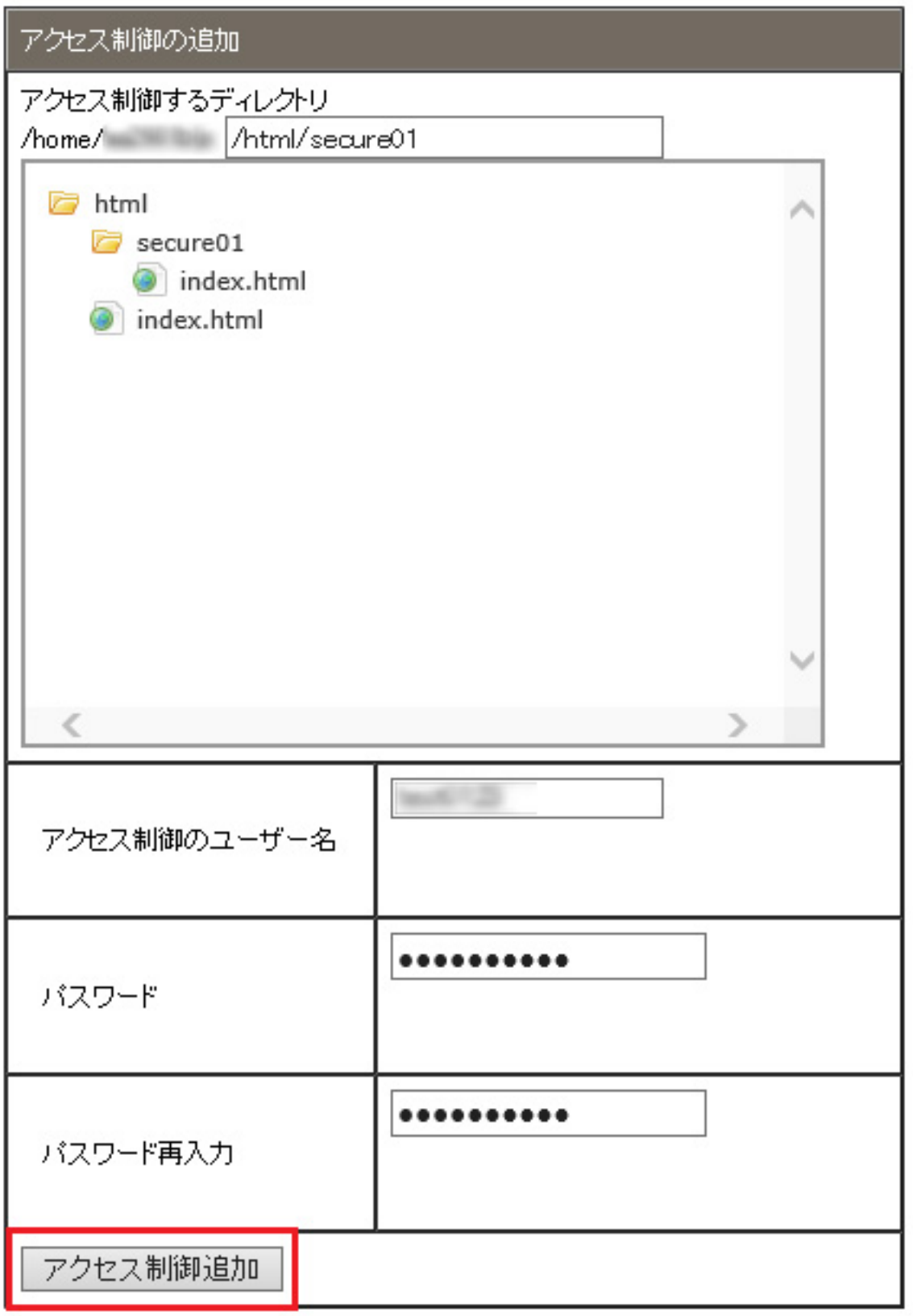
ここでは、http://ドメイン/secure01/ にアクセス制御を設定してみます。

**1** FTP クライアントソフトを使ってサーバーにアクセスし、/html/の下に secure01 ディレクトリを作成します。

**2** コントロールパネルの【制作ツール】から【アクセス制御 (BASIC 認証)】＞【アクセス制御追加】の順番にクリックします。



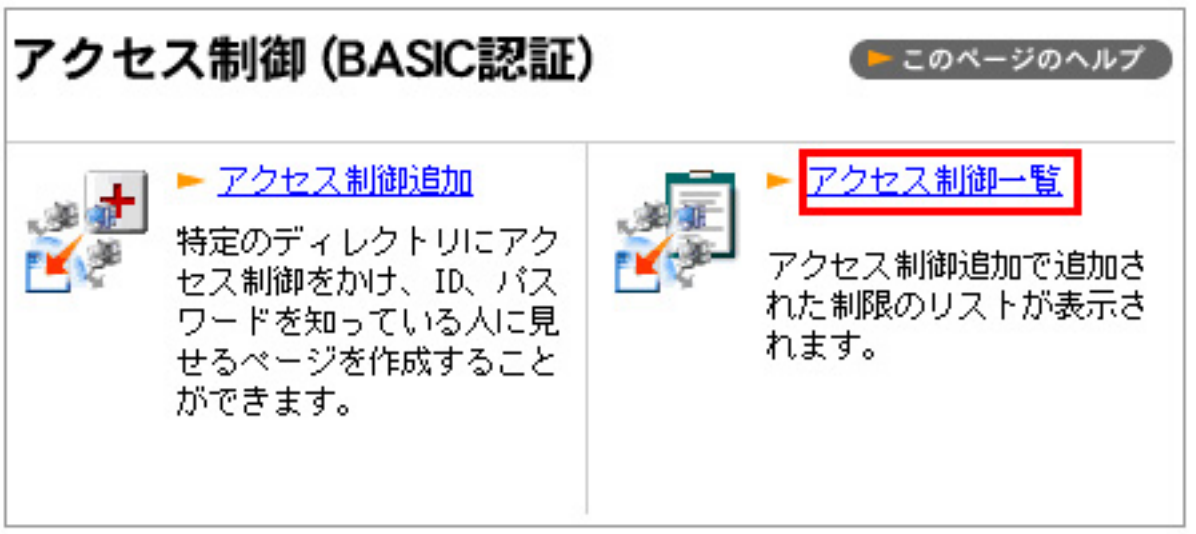
3 ディレクトリから【/html/secure01】を選択のうえ、ユーザー ID と パスワードを入力して【アクセス制御追加】ボタンをクリックします。




4 「ディレクトリ（/html/secure01）にユーザー（secure01）を追加しました。」と表示され、アクセス制御が設定されます。

## アクセス制御の解除

**1** コントロールパネルの【制作ツール】から【アクセス制御 (BASIC 認証)】＞【アクセス制御一覧】の順番にクリックします。




**2** 解除したいアクセス制御の【解除】ボタンをクリックします。

| No | アクセス制限のユーザー名 | アクセス制限するディレクトリ | 解除 |
|---|---|---|---|
| 1 | secure01 | /html/secure01 | 解除 |
| 2 | secure02 | /html/secure02 | 解除 |

**POINT**

PHP をご利用の方へ
【アクセス制御】機能は、 .htaccess ファイルを使用しています。
設定したいディレクトリに既に .htaccess ファイルが存在している場合は、コントロールパネルからアクセス制御を設定するのではなく、 .htaccess ファイルに直接追記して設定してください。
また運用には充分ご注意ください。

# サブドメイン追加設定

サブドメイン追加設定では、任意のサブドメインをサーバー上に設定できます。
サブドメインは無制限に設定できます。

サブドメインを追加し、規定のフォルダにウェブコンテンツを分けてアップロードすることで、異なるコンテンツを公開できます。
テスト公開としてのご利用から、サービス内容に分けてのウェブサイト展開など、幅広くご活用ください。

**POINT**

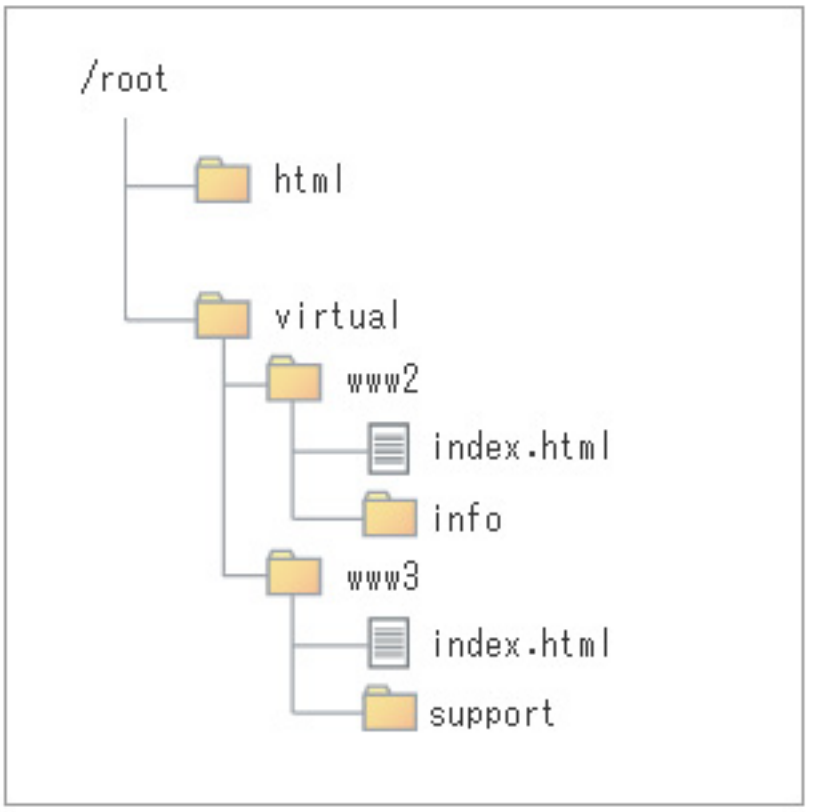
＜サブドメイン追加設定ではできないこと＞
■本機能でメールは利用できません。
■本機能で設定したサブドメイン専用のコントロールパネルは発行されません。
■本機能で設定したサブドメイン領域には CMS インストーラーを含むコントロールパネルの各種ツールはご利用できません。 FTP アクセスのみ可能です。
■本機能ではアクセスログは出力されますが、アクセスログ分析機能は利用できません。
■本機能では SSL は利用できません。
■本機能では www, ftp, mail, pop, smtp, blog の各サブドメインは利用できません。
■追加したサブドメインは、 http://www.サブドメイン/ では利用できません。
http://www.domain.com/ にサブドメイン sub を追加した場合、 http://sub.domain.com/ でのご利用となり、
http://www.sub.domain.com/ ではご利用いただけません。

**重要**

■本機能で追加したサブドメイン領域に FTP アクセスするには、 FTP アカウントのログインディレクトリを/（ルート）もしくは /virtual に設定してください。
■設定可能なサブドメイン名は 32 文字以内の半角英数のみです。
■サブドメイン追加設定では DNS サーバーへの登録が必要となりますので、サブドメインで利用できるまでにお時間がかかります。
■登録するサブドメインのアップロード先ディレクトリ(/virtual/登録したサブドメイン名/)が既に存在する場合はエラーになります。該当ディレクトリを削除してから登録作業を行ってください。

# ディレクトリ構造について

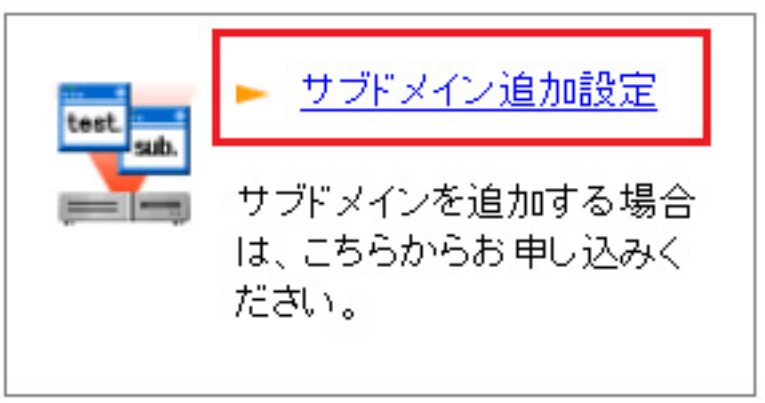
サブドメイン追加設定では、既に存在している html, ftp, log ディレクトリと同階層に virtual ディレクトリを生成し使用します。
さらに virtual ディレクトリ以下にサブドメイン名のディレクトリを生成し、そこをサブドメインでのドキュメントルートとして使用します。
例えば http://www2.example.jp/ と http://www3.example.jp/ というサブドメインを運用したい場合には、以下のようなディレクトリ構造になります。




/virtual/www2/index.html が http://www2.example.jp/ のトップページとなり、さらに /virtual/www2/info/index.html というファイルがあった場合には、URLは http://www2.example.jp/info/index.html となります。

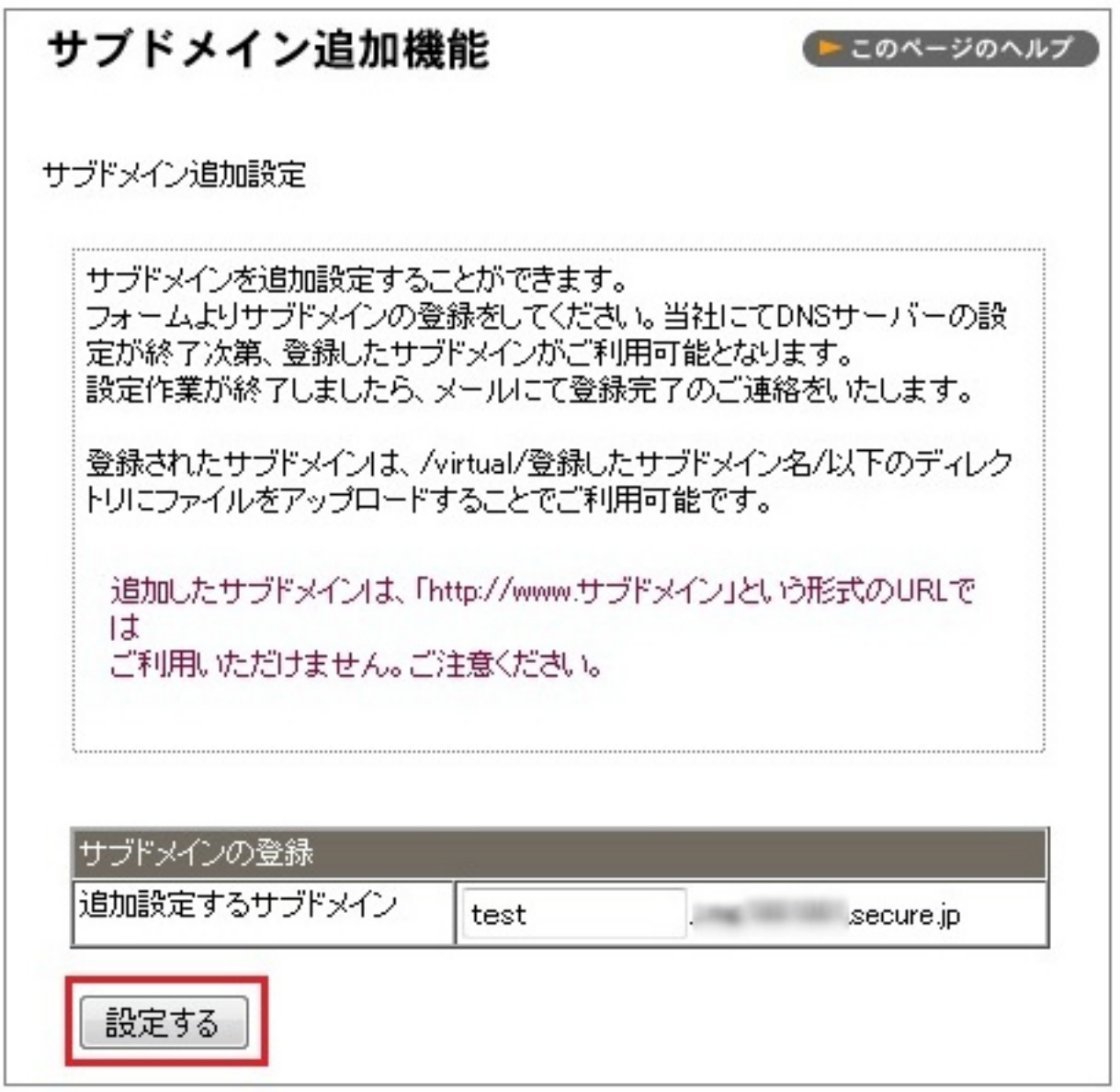
# サブドメインの追加

## 1 コントロールパネルの【制作ツール】から【サブドメイン追加設定】をクリックします。

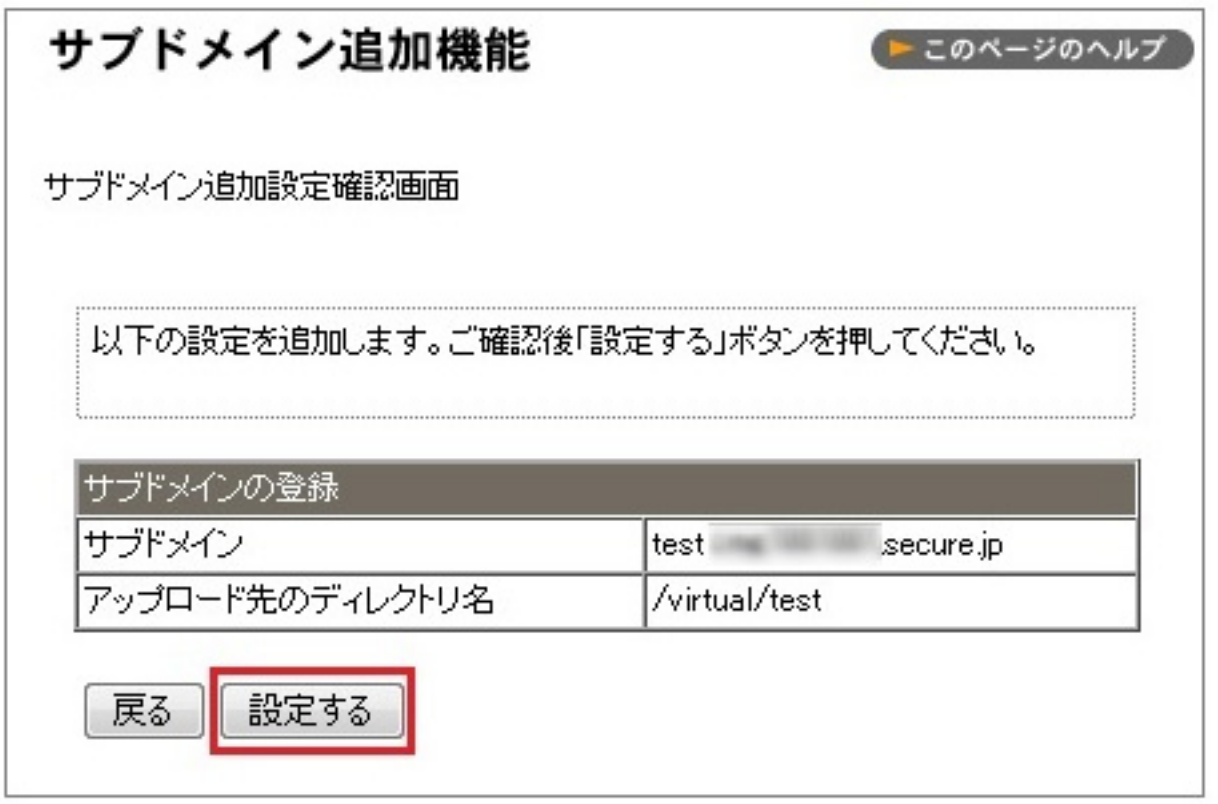



## 2 追加したいサブドメイン名を入力し【設定する】ボタンをクリックします。

※設定可能なサブドメイン名は 32 文字以内の半角英数のみです。



## 3 確認画面で問題がなければ【設定する】ボタンをクリックします。




| | |
|---|---|
| サブドメイン | サブドメインで公開される URL です。 |
| アップロード先のディレクトリ名 | このディレクトリ以下に設置したファイルが公開されます。<br>詳しくはディレクトリ構造についてをご覧ください。 |

## 4 サブドメイン登録完了画面が表示されます。

コントロールパネルでの設定は完了です。

## 5 コントロールパネルでの設定後、必要に応じて弊社にて DNS サーバーへの登録を行います。

詳細につきましては弊社よりご連絡いたします。
DNS サーバーへの登録があるため、コントロールパネルの設定から、実際にドメイン名でアクセスできるまで数日間のタイムラグが発生します。あらかじめご了承ください。

**POINT**

コントロールパネルの「サブドメイン追加設定」で、削除の欄が---となっている場合は、まだサブドメインの DNS 情報が有効になっていないことを表しています。
この状態ではまだサブドメインをご利用いただけません。

| 現在の登録状況（現在:1/最大:10） | | |
|---|---|---|
| サブドメイン | アップロード先ディレクトリ | 削除 |
| test | /virtual/test | --- |

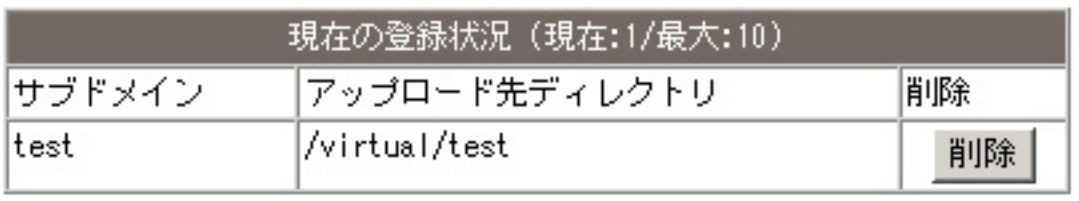
## 6 ファイルのアップロードを行います。

サブドメインが利用できるようになったあと、FTP アカウントのログインディレクトリを「/」（ルート）もしくは「/virtual」 に設定のうえ、ファイルのアップロードを行なってください。

# サブドメインの削除

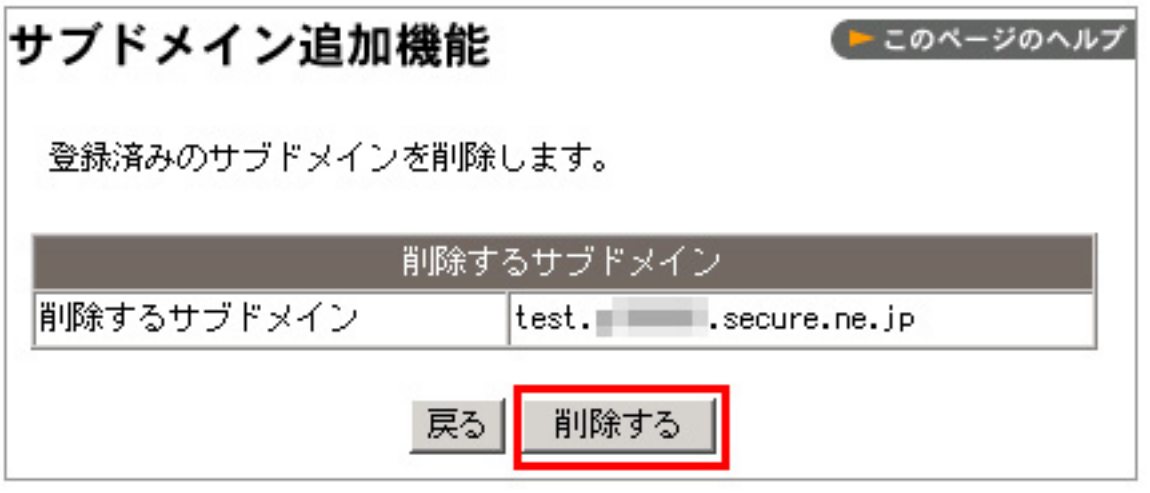
## 1 コントロールパネルの【制作ツール】から【サブドメイン追加設定】をクリックします。


## 2 削除したいサブドメインの右横にある【削除】ボタンをクリックします。




## 3 【削除する】ボタンをクリックします。

確認画面が表示されるので、このサブドメインを削除して問題ないかご確認のうえ、削除ボンタンをクリックしてください。




**重要**

サブドメインを削除してもデータは削除されません。ご不要の場合には使用していたアップロードディレクトリを FTP にて削除してください。

## 4 コントロールパネルでの削除後、必要に応じて弊社にて DNS サーバーの削除を行います。

詳細につきましては弊社よりご連絡いたします。 DNS サーバーはキャッシュを有しているため、コントロールパネルの削除から、実際にドメイン名でアクセスできなくなるまで数日間のタイムラグが発生します。あらかじめご了承ください。

# サブドメインのログの見方

## 1 FTP で【log】ディレクトリにアクセスします。

必要に応じて FTP アカウントのログインディレクトリを変更してください。

## 2 下記は FFFTP で【log】ディレクトリにアクセスした状況です。

【sub-サブドメイン名-access_log】や【sub-サブドメイン名-error_log】といった名称で出力されます。


| | |
|---|---|
| sub-サブドメイン名-access_log | サブドメインで公開される URL です。 |
| sub-サブドメイン名-access_log.yyyymmdd | 当日分のログです。 |
| sub-サブドメイン名-access_log.yyyymm.gz | 先月以降の過去のログです。 |

過去のログは最大３ヶ月まで保存されます。以降は自動ローテーションされサーバーから削除されます。
必要に応じて、早めにダウンロードしてください。

## 3 見たいログをダウンロードし、テキストエディタやエクセルなどでご覧ください。（.gz ファイルは解凍後にご覧ください。）

また、ログは Apache 準拠のログ（apache combined log）ですので、対応しているログ解析ソフトであれば、ローカル PC での分析も可能です。

# スクリプト定期実行ツール

指定した日時に CGI スクリプトを自動的に実行する機能です。

**POINT**

各種集計やレポートのメール送信などを定期的に行いたい場合に使用します。

**目次**

## 注意事項

スクリプト定期実行ツールで CGI スクリプトを確実に実行するためには、以下の点をご注意ください。

・該当 CGI スクリプトのコマンドは絶対パスで記述する。
・該当 CGI スクリプトの入出力ファイルは絶対パスで記述する。
・HTTP_XXXX_XXXXの ような環境変数は受け取れない。

### × 実行できない CGI スクリプト例

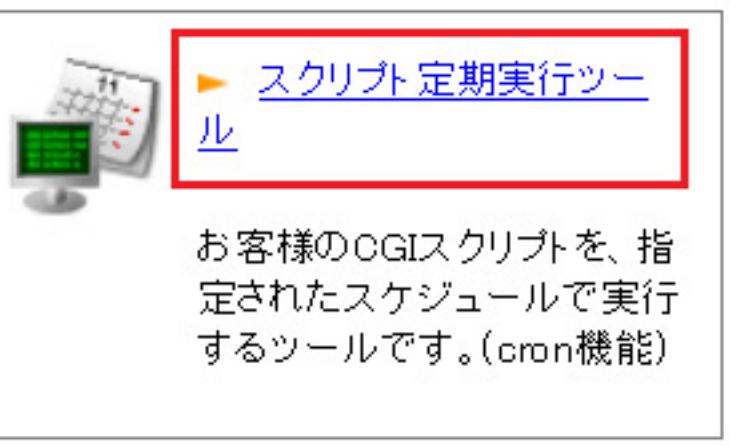
```
#!/usr/local/bin/perl
while(<DATA>){ system($_); }
__DATA__
cd /usr/home/ユーザ ID/html/temp
tar cvfz backup.tgz /usr/home/ユーザ ID/html/example/*
```

### ○ 実行可能な CGI スクリプト例

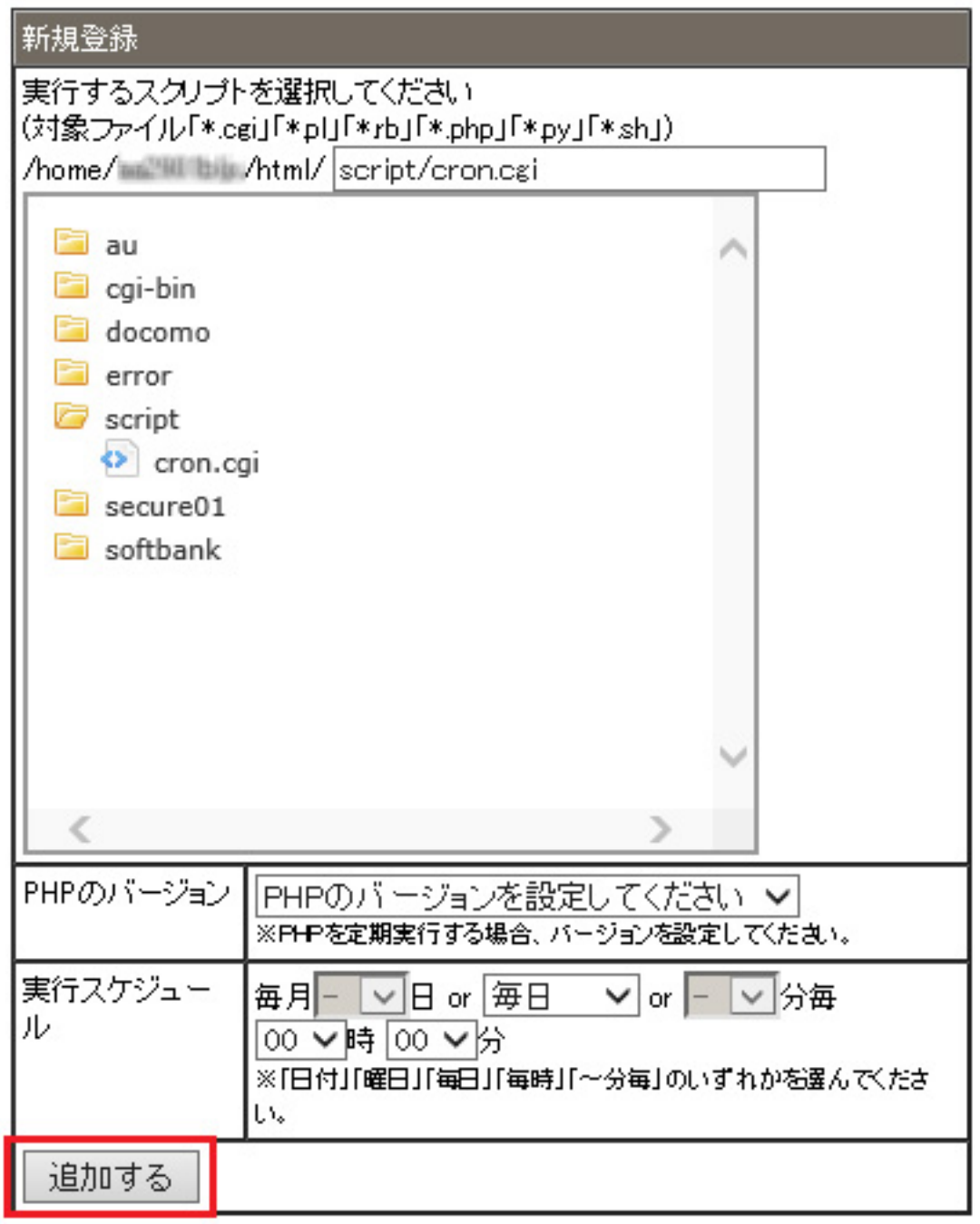
```
#!/usr/local/bin/perl
while(<DATA>){ system($_); }
__DATA__
/usr/bin/tar cvfz /usr/home/ユーザ ID/html/temp/backup.tgz /usr/home/ユーザ ID/html/example/*
```

## スクリプト定期実行ツールの設定

### 1 コントロールパネルの【制作ツール】から【スクリプト定期実行ツール】をクリックします。




### 2 実行するスクリプトとスケジュールを設定のうえ【追加する】ボタンをクリックします。



| | |
|---|---|
| 実行するスクリプト | プルダウンメニューには、html ディレクトリ以下にあるスクリプトファイルが表示されます。ここで、実行したいスクリプトファイルを選択します。<br>対象となる拡張子は、【.cgi】【.pl】【.rb】【.php】【.py】【.sh】です。 |
| PHPのバージョン | PHP のスクリプトを選択した場合は、必ず実行するPHP のバージョンを選択してください。<br>.htaccess で指定しているバージョンと違うと不具合が出る可能性があります。 |
| 実行スケジュール | 1 分毎、 2 分毎、 3 分毎、 4 分毎、 5 分毎、 6 分毎、 10 分毎、 12 分毎、 15 分毎、 20 分毎、 30 分毎<br>毎時＊分<br>毎日＊時＊分<br>＊曜日　＊時＊分<br>毎月＊日　＊時＊分<br>にて設定が可能です。<br>※「20 分毎」を選択されますと、毎時 00 分、 20 分、 40 分に実行されます。<br>例) 10 時 30 分に登録されますと、最初の実行は 10 時 40 分となります。 |

**重要**

・サーバーの状況により、開始時間が遅れることもあります。ご了承ください。
・設定した内容は変更できません。変更する場合は、一度削除してから設定しなおしてください。
・.htaccess ファイルが設置されている場合、コントロールパネルからのスクリプト定期実行ツールの設定が正常に反映されないことがございます。
・回避方法といたしましては、既存の .htaccess をリネームしていただき、コントロールパネルからスクリプト定期実行ツールを設定後に、リネームした .htaccess を元に戻してご利用ください。

## 設定の削除

現在の登録状況の右側にある【削除】ボタンをクリックします。

| 現在の登録状況（最大登録件数:[無制限]） | | |
|---|---|---|
| スケジュール | 実行ファイル | 削除 |
| 毎月1日 3時5分 | /html/test.php | 削除 |
| 毎週水曜日 10時0分 | /html/test.sh | 削除 |

# SSI

SSI (Server Side Include）は、 HTML ドキュメント内に CGI の出力結果やファイル情報等を埋め込む機能です。

**POINT**

ブラウザで SSI が埋め込まれた HTML ファイルにアクセスすると、サーバー上で SSI の構文を解析し、その結果を返します。
このため動的な出力が可能ですが、 HTML ファイルが読み込まれるたびにサーバー側で SSI の構文解析が行われるため、通常の HTML ファイルと比較して、サーバーにより大きな負荷がかかります。
サーバーに大きな負荷を与えるような SSI は記述しないよう、お願いいたします。

**目次**



## SSI の利用方法

SSI を使用する場合は HTML ファイルに次のように記述します。

```
<!--#コマンド 引数-->

例）世界標準時（グリニッジ標準時）による現在の日時を表す変数を表示する記述は次のようになります。

<!--#echo var="DATE_GMT"-->
```

SSI を使用しているファイルは HTML ファイルと区別するために、拡張子を【.html】ではなく【.shtml】にする必要があります。
パーミッションは HTML ファイル同様【644】に設定します。
.htaccess ファイルを設置する必要はありません。

※SSI を拡張子【.html】でご利用される場合は、.htaccess の設定方法をご参照ください。

## 引数の渡し方

SSI から CGI スクリプトに引数を渡す場合には、下記のように記述します。

```
<!--#include virtual="test.cgi?arg1+arg2"-->
```

※arg1 arg2 が引数となります。

# エラーページ設定ツール（テストサイトのみ）

ウェブサーバーは、存在しないページへのアクセスや、アクセス権限のないページへのアクセスなどが発生するとエラーページを表示します。
エラーページ設定ツールは、それらのエラーページを任意のデザインに変更します。

以下のエラーについて、それぞれ任意の HTML ファイルを表示するよう指定します。

401 Authorization Required （認証失敗）
403 Forbidden （権限が無い）
404 Not Found （ページが見つかりません）
500 Internal Server Error （サーバー内部エラー）

**重要**

/html にアクセス制御を設定している場合や独自に記述した .htaccess を設置している場合、エラーページ設定ツールから設定することはできません。

**目次**

## エラーページの設定

**1** あらかじめ、エラーページ用の HTML を作成して、サーバーにアップロードしておきます。

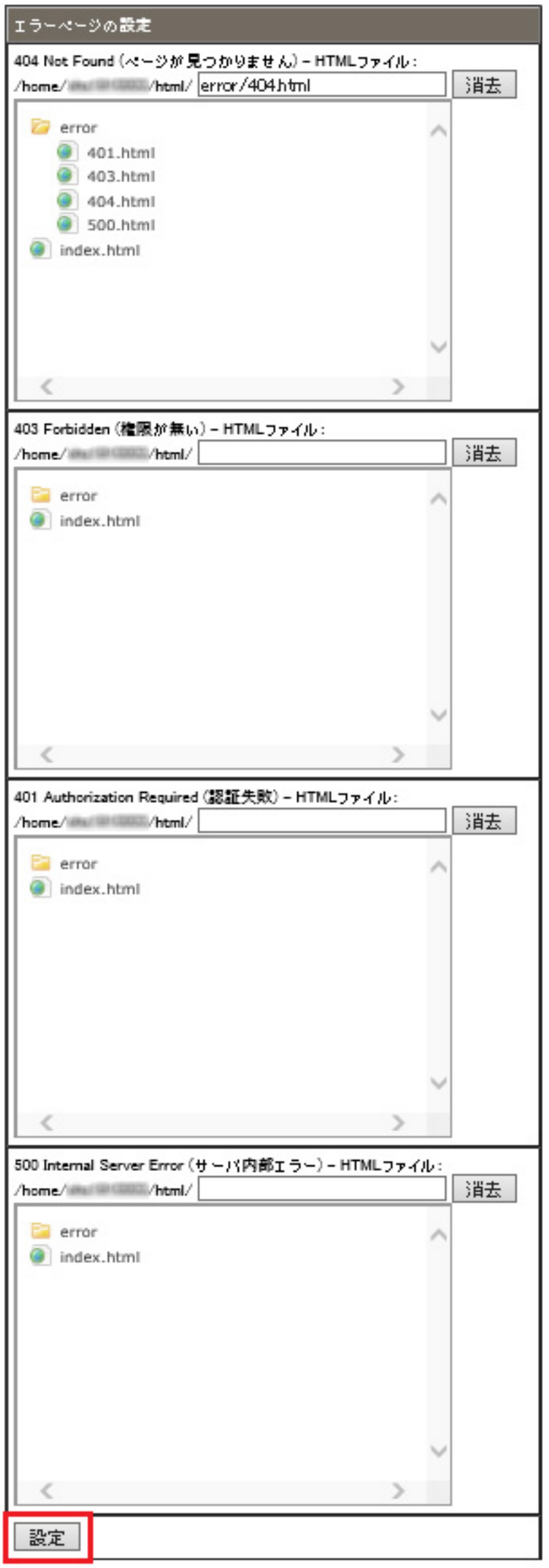
**2** コントロールパネルの【制作ツール】から【エラーページ設定ツール】をクリックします。

► エラーページ設定ツール

404、403等のエラーページの表示を、初期設定のものから、お客様が作成されたHTMLファイルに変更するツールです。

## 3 アップロードしてあるエラーページ用のファイルを指定して【設定】ボタンをクリックします。

※すでに独自の .htaccess が存在する場合は設定できません。



### 4 【設定】ボタンをクリックすると、 html ディレクトリ直下に .htaccess が作成されます。

.htaccess ファイルの内容は以下のとおりです。

ErrorDocument 401 /ファイル名 0.html
ErrorDocument 404 /ファイル名 1.html
ErrorDocument 403 /ファイル名 2.html
ErrorDocument 500 /ファイル名 3.html

## 設定の解除

html ディレクトリに FTP でアクセスし、 .htaccess ファイルを削除します。

# PHP

**目次**

## ご利用可能な PHP のバージョン

ご利用可能な PHP のバージョンは以下のとおりになります。

| プラン | PHP バージョン | PHP のバージョンを指定する場合の .htaccess ファイルへの記述内容 |
|---|---|---|
| CHM-01Z | PHP 5.4.39 | AddHandler x-httpd-php5439 .php |
| | PHP 5.5.23 ※1 | AddHandler x-httpd-php5523 .php |
| | PHP 5.6.7 | AddHandler x-httpd-php567 .php |
| | PHP 5.6.19 | AddHandler x-httpd-php5619 .php |
| | PHP 5.6.30 | AddHandler x-httpd-php5630 .php |
| | PHP 5.6.31 | AddHandler x-httpd-php5631 .php |
| | PHP 5.6.34 | AddHandler x-httpd-php5634 .php |
| | PHP 5.6.38 | AddHandler x-httpd-php5638 .php |
| | PHP 7.0.32 ※2 | AddHandler x-httpd-php70 .php |
| | PHP 7.1.30※2 | AddHandler x-httpd-php71 .php |
| | PHP 7.2.20※2 | AddHandler x-httpd-php72.php |
| CHM－11Z | PHP 5.6.38 ※2 | AddHandler x-httpd-php56 .php |
| | PHP 7.0.32※1※2 | AddHandler x-httpd-php70 .php |
| | PHP 7.1.30※2 | AddHandler x-httpd-php71 .php |
| | PHP 7.2.20※2 | AddHandler x-httpd-php72 .php |
| | PHP 7.3.7※2 | AddHandler x-httpd-php73 .php |

（すべてのバージョンで暗号化通信のバージョン TLS1.2 に対応しています）
※1 バージョン指定しない場合に実行される標準のバージョンです。
CHM-01Z では PHP 5.5.23 、CHM-11Zでは、PHP 7.0 系が実行されます。
⇒ご契約中のプランが CHM-01Z か CHM-11Z か確認する

※2 .htaccess ファイルの記述方法は、プランや PHP のバージョンによって異なります。
詳しくは「PHP バージョン指定の記述方法」をご参照ください。

## PHP のご利用方法

ご契約のサーバーで PHP を利用する方法は 2 つございます。

### .htaccess ファイルを利用せず、標準の PHP バージョンを使用する場合

お客さまのサーバー領域内の html ディレクトリ内の任意のディレクトリに PHP ファイルをアップロードしてください。設置された PHP プログラムは、上記表の【実行バージョン】に※1 がついているバージョンで実行されます。

### .htaccess ファイルを利用して、 PHP のバージョンを指定する場合

お客さまのサーバー領域内の html ディレクトリ内の任意のディレクトリに PHP ファイルをアップロードしてください。その後、【.htaccess】ファイルに実行バージョンを指定する記述を追加してください。実行可能なバージョンは上記表をご確認ください。

## PHP バージョン指定の記述方法

CHM-01Z の PHP 7 系と、CHM-11Z の全バージョンの PHP は、下記のように .htaccess ファイルに記述します。
下記は PHP 7.0.2 を使用するときの記述例です。
ご利用のバージョンに合わせて変更してください。

**提供バージョン　PHP 7.0.2**

```
.htaccess の記述　AddHandler x-httpd-php70 .php
```

**重要**

.htaccess の記述は、～php702.php ではなく　～php70.php となりますのでご注意ください。

提供されている PHP のバージョンは、コントロールパネルの『お客様情報』＞『プログラムのパスとサーバの情報』でご確認いただけます。

今後、 PHP7.0.X の「X」の数字が上がっても（バージョンアップしても）、 .htaccess の記述を変更する必要はございません。 PHP7.Y.Z の「Y」の数字が上がり、PHP7.Y.Z をご利用されたい場合、 .htaccess には『AddHandler x-httpd-php7Y .php』と記述する必要がございます。

## PHP の設定の変更方法

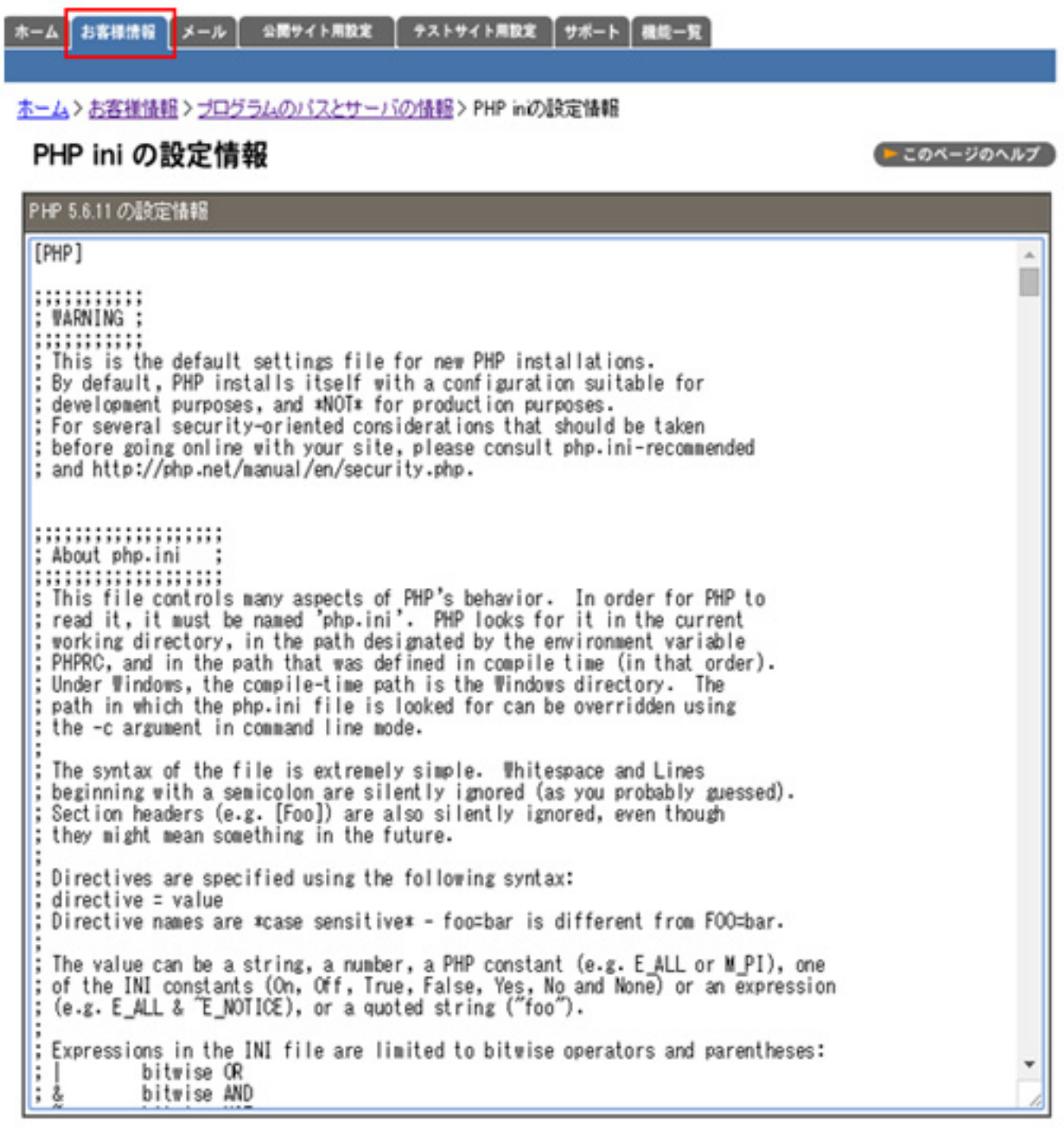
PHP の設定を変更することができます。 PHP の設定は設定ファイル (php.ini) を編集し、 PHP プログラムが存在するディレクトリと同じ場所に設定ファイル (php.ini) を設置することにより、設定を変更することができます。また、設定ファイル (php.ini) が設置されているディレクトリ以下の階層すべてに同じ設定を適用することも可能です。

### 1 設定ファイル (php.ini) をコントロールパネルから入手する。

コントロールパネルの【お客さま情報】＞【プログラムのパスとサーバの情報】＞【PHP ini の設定情報】に PHP のバージョンごとに、 php.ini ファイルの内容がテキスト形式で表示されます。

| PHP ini の設定情報 |
| --- |
| PHP 5.4.39 |
| PHP 5.5.23 |
| PHP 5.6.7 |
| PHP 7.0.2 |

以下の情報を、お手元のテキストエディタにコピー＆ペーストし、php.ini というファイル名で保存してください。



## 2 保存した設定ファイル (php.ini) をサーバーにアップロードする。

PHP 5.4 未満の場合は、以下のファイル内の以下の記述を変更してください。
(register_globals は PHP 5.4.0 で削除されました)

変更前 register_globals = Off
↓
変更後 register_globals = On

設定ファイル (php.ini) をアップロードしたディレクトリ内で、設定が有効となります。
アップロードしたディレクトリ内の下位ディレクトリにも同じ設定を反映させたい場合は下位ディレクトリにも同じ設定ファイル (php.ini) を設置するか、 .htaccess ファイルに 1 行情報を追記する必要があります。

## 3 設定ファイル (php.ini) を設置したディレクトリ以下に同じ設定を反映させる方法

設定ファイル (php.ini) を設置したディレクトリ内に、 .htaccess ファイルを作成します。
ファイルの内容は以下の情報を記載してください。

【例】
設定ファイル (php.ini) のフルパス　/usr/home/ユーザーID/conf/php.ini
※conf は例として記述したディレクトリ名です。
.htaccess に以下を記述して、設定ファイル (php.ini) と同じ場所にアップロードします。
suPHP_ConfigPath /usr/home/ユーザーID/conf/

最終的に以下の構成になります。
/usr/home/ユーザー ID/conf/php.ini
/usr/home/ユーザー ID/conf/.htaccess

## PHP の文字コードについて

PHP 5.5.16 以降のバージョンではデフォルトの文字コード（mbstring.internal_encoding）は「UTF-8」です。
その他のバージョンのデフォルトの文字コード（mbstring.internal_encoding）は「EUC-JP」です。
また、新しくリリースされた PHP7 では、文字コードはmbstring.internal_encoding で指定するのではなく、default_charset で指定すべきとの情報もございます。
http://php.net/manual/ja/ のサイトや関連情報を参照のうえ、ご利用願います。

## PHP ファイル、ディレクトリのパーミッション

PHP を実行する際には PHP プログラムを記述したファイル、 PHP ファイルが含まれるディレクトリのパーミッションの設定にご注意ください。以下のようにファイルは【604】、または【644】、ディレクトリは【705】、【755】としていただき動作をご確認願います。

PHP ファイル
- 604　rw----r--
- 644　rw-r--r--

PHP ファイルが含まれるディレクトリ
- 705　rwx---r-x
- 755　rwxr-xr-x

## 利用制限

.htaccess ファイル内に php_value および php_flag を記述することは出来ません。 500 エラー（Internal Server Error）が発生します。

CPI が提供しているサービス内で提供している PHP は、CPI サービス向けに独自に実装をしております。そのため、すべての機能が動作することを保証しておりません。

PHP や設定ファイル (php.ini) 、.htaccessのご利用に際しては、事前の動作検証やデバッグ、エラーのご確認などを経てご利用いただきますようお願い申し上げます。

PHP プログラム、設定ファイル(php.ini)、 .htaccess ファイルは、記述内容が 1 文字でも間違っていますと、実行結果や動作に影響が発生します。
また、ファイルを設置したディレクトリと、配下のディレクトリすべてに影響が及びますので十分ご注意ください。

# PHP（CGI 版）について

このページでは CGI 版での PHP の実行方法について説明します。

**目次**

## PHP プログラムの準備

### 1 PHPファイルの拡張子は、perl で作成した CGI と同じように「.cgi」としてください。

### 2 PHP ファイルの1行目に実行する PHP のパスを記述します。

PHP のバージョンごとにパスが異なりますので以下の表を参考にしてください。

| プラン名 | PHP バージョン | PHP を CGI 版でご利用する場合の PHP のパスの記述内容<br>※ファイルの１行目に記述してください。 |
|---|---|---|
| CHM-01Z | PHP 5.4.39 | #!/usr/local/bin/php-cgi-5.4.39 |
| | PHP 5.5.23 | #!/usr/local/bin/php-cgi-5.5.23 |
| | PHP 5.6.7 | #!/usr/local/bin/php-cgi-5.6.7 |
| | PHP 5.6.19 | #!/usr/local/bin/php-cgi-5.6.19 |
| | PHP 5.6.30 | #!/usr/local/bin/php-cgi-5.6.30 |
| | PHP 5.6.31 | #!/usr/local/bin/php-cgi-5.6.31 |
| | PHP 5.6.34 | #!/usr/local/bin/php-cgi-5.6.34 |
| | PHP 5.6.38 | #!/usr/local/bin/php-cgi-5.6.38 |
| | PHP 7.0.32 | #!/usr/local/bin/php-cgi-7.0 |
| | PHP 7.1.30 | #!/usr/local/bin/php-cgi-7.1 |
| | PHP 7.2.20 | #!/usr/local/bin/php-cgi-7.2 |
| CHM-11Z | PHP 5.6.38 | #!/usr/local/bin/php-cgi-5.6 |
| | PHP 7.0.32 | #!/usr/local/bin/php-cgi-7.0 |
| | PHP 7.1.30 | #!/usr/local/bin/php-cgi-7.1 |
| | PHP 7.2.20 | #!/usr/local/bin/php-cgi-7.2 |
| | PHP 7.3.7 | #!/usr/local/bin/php-cgi-7.3 |

（すべてのバージョンで暗号化通信のバージョン TLS1.2 に対応しています）

## 3 CGI ファイルをサーバーへアップロードしてください。

ファイルのパーミッションは 705(rwx---r-x) にしてください。

## PHP ファイルの拡張子を .php のまま CGI 版で動作させる方法

.htaccess ファイルに以下の記述を追記してください。

```
AddType application/x-httpd-cgi .php
```

# PHP の設定を変更する

弊社サーバーでは任意の設定内容でPHP を動作させることができます。独自の PHP の設定には、【php.ini】ファイルを使用します。
一例として、 register_globals の設定を変更する方法を紹介します。

## 1 コントロールパネルの【お客様情報】から【プログラムのパスとサーバーの情報】をクリックます。

使用する PHP と同じバージョンの php.ini 情報をテキストファイルにコピーします。

## 2 テキストファイルの下記の記述を修正します

```
中略
～
変更前    register_globals = Off
変更後    register_globals = On
～
中略
```

## 3 ファイル名を【php.ini】として保存し、任意のディレクトリにアップロードします。

## 4 phpinfo を参照のうえ、実際にアクセスして動作検証を行います。

**重要**

弊社サーバーでは .htaccess ファイルにて php_value および php_flag はご利用いただけません。
記述が .htaccess ファイルにある場合、 500 エラー（Internal Server Error）となってしまいますのでご注意ください。

**POINT**

弊社サーバーの PHP は php.ini ファイルがカレントディレクトリにある場合、その php.ini ファイルの設定を最優先で読み込みます。ただし下位ディレクトリに対しては効果は及びませんので、下位ディレクトリ全体に独自の php.ini ファイルを有効したい場合には .htaccess ファイルを利用するか、各ディレクトリに 1 つずつ php.ini ファイルを設置してください。

## データベースをご利用の場合

MySQL の標準ポート番号は 3306 ですが、ご契約プランでは、 MySQL は 5.5系 と 5.6 系 をご利用いただけます。
そのため下記のとおり、バージョンごとに接続ポート番号が異なります。

MySQL5.5 系：3306
MySQL5.6 系：3307

お客様にてご用意された CMS などで、 PHP から MySQL に接続する場合は、php.ini や CMS 側の設定ファイルなどで、 MySQL のバージョンに応じたポート番号をご指定ください。

## .htaccess を利用して下位ディレクトリ全体に PHP の独自設定を反映させる

以下に.htaccess を利用する場合の設置方法を紹介します。

**1** 下記のように、 php.ini ファイルを設置したディレクトリパスを記述した .htaccess ファイルを用意します。パスは間違いのないよう、正確に記述してください。

```
例）  suPHP_ConfigPath /usr/home/ユーザー ID/html/
```

**2** 独自設定の php.ini ファイルで動作させたいディレクトリに .htaccess ファイルをアップロードします。

## 3 実際にアクセスして動作検証を行います。

**重要**

■php.ini ファイルは、そのままドキュメントルート以下のディレクトリに設置するとブラウザから閲覧されてしまう場合があります。 .htaccess ファイルで以下の記述を追加することにより、 ini ファイルの閲覧を制限できます。

---

```
<Files ~ "\.ini$">
deny from all
</Files>
```

---

■.htaccess ファイルでは php.ini ファイルを直接指定すると変更が有効に適用されませんので、ご注意ください。

---

○　suPHP_ConfigPath /usr/home/ユーザー ID/html/sample/

×　suPHP_ConfigPath /usr/home/ユーザー ID/html/sample/php.ini

---

■php.ini ファイルに必要な箇所しか記述しない場合（ここでは register_globals= On）は、正常に PHP が動作しません。
php.ini ファイルを作成するときには、弊社標準の php.ini ファイルを編集されることをお勧めします。

（コントロールパネルの【お客様情報】＞【プログラムのパスとサーバーの情報】を選択し、【PHP ini の設定情報】をクリックして、内容をコピー＆ペーストしてご利用ください。）

■.htaccess ファイルは設置したディレクトリ以下すべてに影響します。
特定のディレクトリで動作させる場合には、 .htaccess ファイルを使わず、 php.ini ファイルを該当ディレクトリに設置してください。

■.htaccess ファイルは設置したディレクトリ以下すべてに影響します。
記述方法を間違えますとお客様のサイト全体に多大な影響を及ぼしますので、設置タイミング、動作検証など充分に行ってください。

■該当するディレクトリに .htaccess を設置している場合には、ファイルを上書きせずに既存の .htaccess ファイルに追記してください。上書きをしますと既存の設定内容が無効となりますのでご注意ください。

■同一ディレクトリに .htaccess ファイルと php.ini ファイルが両方設置された場合、 .htaccess ファイルの設定が優先されます。

■.htaccess ファイルの詳細については、検索エンジンや関連書籍を参照してください。

■.htaccess ファイルに関しては CPI サポート対象外となります。あらかじめご了承ください。

# Perl について

以下のバージョンの Perl をご利用いただけます。

CHM-01Z をご利用のお客さま： perl のバージョンは、 5.16.3 です。
CHM-11Z をご利用のお客さま： perl のバージョンは、 5.24.3 / 5.18.4 です。

CHM-01Z と CHM-11Z ではバージョン情報などに違いがございます。
⇒ご契約中のプランが CHM-01Z か CHM-11Z か確認する

**目次**



## Perl のパス

```
#!/usr/local/bin/perl
```

もしくは

```
#!/usr/bin/perl
```

と記述します。

**重要**

CHM-11Z のお客様は、どちらのバージョンもご利用可能ですが、Perlのパスで一般的に利用される「/usr/bin/perl」は Perl5.24 系が実行されます。
Perl5.18 系を利用する場合のパスは「/usr/local/bin/perl5.18」となります。

## パーミッション

パーミッションは 755 が一般的です。

設置マニュアルなどにパーミッションが記載されている場合には、その設定内容を優先してください。

また CPI サーバーではセキュリティーを考慮するうえで、 705 を推奨しています。

## CGI の設置

CGI ファイル専用として【cgi-bin】ディレクトリを用意しておりますが、【cgi-bin】ディレクトリ以外にファイルを設置しても CGI は動作します。

**重要**

【cgi-bin】は CGI ファイル専用ディレクトリであるため、画像ファイルや、 html ファイルを設置してもブラウザには表示されません。

## CGI のサポート

**重要**

以下の CGI につきましては、弊社サポートの対象外です。ご了承ください。

・お客様が独自に作成された CGI
・CPI で提供しているものであっても、お客様が独自にカスタマイズされた CGI
・CPI で提供しているもの以外の CGI

# SSH について

遠隔地のマシンからコマンドを実行したり、他のマシンへファイルを移動したりできます。
ネットワーク上を流れるデータは暗号化されるため、インターネット経由でも一連の操作を安全に行えます。

CHM-01Z と CHM-1Z では、接続方法が変わります。
CHM-11Z は、鍵認証が必要となります。

CHM-11Z で SSH をご利用の場合はこちらをご確認ください。

⇒ご契約中のプランが CHM-01Z か CHM-11Z か確認する

## ご利用上の注意

- SSH のご利用には、お客様サーバー情報に記載されている、【ご利用サーバー名】、【ユーザー ID】、【パスワード】で接続してください。
- 接続ポート番号は、コントロールパネルにログイン後、【お客様情報】＞【プログラムのパスとサーバーの情報】＞【SSH ポート番号】にてご確認ください。
- ユーザー権限でのご利用となります。ルート権限ではありませんので、ルート権限が必要なソフトウェアのインストール等は行えません。ご了承ください。
- SSH 接続時の認証は【チャレンジレスポンス認証を使う（キーボードインタラクティブ）】を選択してください。

# SSH（鍵認証）

目次



## SSH について

SSH(Secure Shell) は、ご契約サーバーに対し遠隔（リモート）から接続するためのプロトコルです。 SSH クライアントソフトを利用することにより、サーバーでコマンドを実行し、ファイル操作、プログラム実行などを行うことができます。

CHM-01Z と CHM-11Z では、接続方法が変わります。

CHM-01Z で SSH をご利用の場合はこちらをご確認ください。

## SSH の利用方法

SSH 接続をするためには、鍵ペア(公開鍵と秘密鍵)を生成し、 CHM-11Z のサーバーに公開鍵を設定し、接続する端末 (PC 、各種デバイスなど) に秘密鍵を持ちます。鍵ペア生成時にパスフレーズを設定すると、接続の際にパスフレーズの入力が必要になります。
CHM-11Z のコントロールパネルでは

- 鍵ペアの生成
- 公開鍵を CHM-11Z のサーバーへ設定
- 秘密鍵のダウンロード

の作業を行うことができます。

## SSH 接続に必要な情報

### ログイン情報

| | |
|---|---|
| サーバー名(ホスト名) | ご契約のサーバー名<br>CHM-11Z 主契約ドメイン：faXXXX.secure.ne.jp |
| パスフレーズ | 鍵ペアを作成する際に、パスフレーズを設定した場合は、 SSH 接続時にパスフレーズの入力を求められます。<br>パスフレーズを設定していない場合は、パスフレーズの入力なしで SSH 接続をすることができます。 |
| ユーザー名 | コントロールパネルのユーザー ID |
| ポート番号 | コントロールパネルの【お客様情報】の【プログラムのパスとサーバーの情報】の【SSH ポート番号】に記載しています。 |
| サーバーの公開鍵保管ディレクトリとファイル名 | /home/ウェブコントロールパネルのユーザー ID/.ssh/authorized_keys |

## コントロールパネルの利用方法

CHM-11Z のコントロールパネルで以下の機能を提供しています。

| | |
|---|---|
| SSH 利用開始／利用停止の切り替え | 【利用開始する】<br>公開鍵が登録されている状態で【利用開始する】ボタンをクリックすると、 SSH 接続が利用可能になります。 |
| | 【利用停止する】<br>【利用停止する】ボタンをクリックすると、 SSH 接続が利用できなくなります。公開鍵は削除されません。公開鍵は登録された状態で SSH 接続を無効にしたい場合などにご利用ください。 |
| 公開鍵一覧表示 | サーバーに登録されている公開鍵の一覧（フィンガープリントとコメントが表示されます。）を確認することができます。<br>【削除】ボタンをクリックすると公開鍵を削除することができます。<br>【公開鍵一括削除】ボタンをクリックすると複数の公開鍵を一括で削除することができます。 |
| 鍵ペア作成機能 | 秘密鍵ファイル名(必須)、パスフレーズ、コメントを入力して【作成する】ボタンをクリックすると、鍵ペア(秘密鍵と公開鍵)が作成されます。<br>その後、公開鍵は /home/コントロールパネルのユーザー ID/.ssh /authorized_keys に自動的に書き込まれて公開鍵一覧で確認できるようになります。<br>公開鍵は複数登録できます。<br>秘密鍵はブラウザ経由でお手元の PC にダウンロードされ、 サーバー上から秘密鍵は削除されます。 |
| 公開鍵登録機能 | お客さまがお手元の PC などで生成された公開鍵を、ウェブサーバーに登録するための機能です。<br>公開鍵は　/home/コントロールパネルのユーザー ID/.ssh /authorized_keys　に書き込まれて公開鍵一覧で確認できるようになります。 |

## SSH クライアントソフトの設定方法、 SSH 接続方法

- PuTTY、TeraTerm、Mac OS での SSH 接続方法について説明します。
- コントロールパネルで鍵ペアを生成する場合　と　ローカル PC で鍵ペアを生成する場合のそれぞれについて手順書(PDF ファイル)にまとめました。

| コントロールパネルで鍵ペアを生成する場合 | |
|---|---|
| PuTTY | TeraTerm |
| Mac OS | |

| ローカル PC で鍵ペアを生成する場合 | |
|---|---|
| PuTTY | TeraTerm |
| Mac OS | |

| SSH 接続マニュアル応用編 |
|---|
| PDF |

## SSH 鍵認証の仕様

ウェブコントロールパネルで生成される鍵ペアの仕様

| | 利用可能な文字種 | 文字数制限 | 必須 |
|---|---|---|---|
| 秘密鍵ファイル名 | 半角英数、ハイフン、アンダースコア | 1～30 文字以内 | 必須 |
| パスフレーズ | 半角英数、_#@.,+-;:$%&!= | 5 文字以上～50 文字以内 | 省略可能※ |
| コメント | 制限なし | 50 文字以内 | 省略可能 |

※省略可能ですが指定する場合は文字種、文字数制限がございます。

### ウェブコントロールパネルで生成される秘密鍵の仕様

| | |
|---|---|
| 種類 | RSA2 |
| 鍵長 | 2048 bit |
| パスフレーズ | 指定は任意 |
| コメント | 指定は任意 |

### ウェブコントロールパネルの公開鍵登録機能の仕様

| | |
|---|---|
| 種類 | RSA2 もしくは ECDSA |
| 鍵長 | RSA2 の場合のみ 2048以上 |
| パスフレーズ | 指定は任意 |
| コメント | 指定は任意 |

## 注意事項

### SSH はサーバー導入時は SSH 接続は利用停止状態です。

公開鍵をウェブサーバーに登録し、【利用開始する】ボタンをクリックしてください。

### SSH はユーザー権限でご利用いただけます

SSH は root 権限ではなくユーザー権限で提供しています。 root 権限が必要な操作は行えません。
また、上述したコマンド以外はご利用いただくことができません。

# Git

Git とは分散型のバージョン管理を行うシステムです。
バージョン管理システムとは、ファイルの変更履歴（いつ、誰が、何を更新したか？）を記録し、過去のある時点の変更点などを表示するシステムです。

バージョン管理システムは、「分散管理型」と「集中管理型」の二種類あります。 Git は分散型であり、リポジトリ※をお客様環境やサーバーに分散して作成・管理ができます。
そのため、各作業者の環境（PC など）でリポジトリを作成し作業を進めて、最終的にサーバーのリポジトリに反映するといったことも可能です。

※リポジトリとは、英語で「貯蔵庫」という意味です。リポジトリにはファイルの内容やその変更履歴、誰が変更したかといった情報が保存されています。

# Git クライアント「SourceTree」の利用方法（Windows）

**目次**



## CPI サーバーにリポジトリの作成する方法

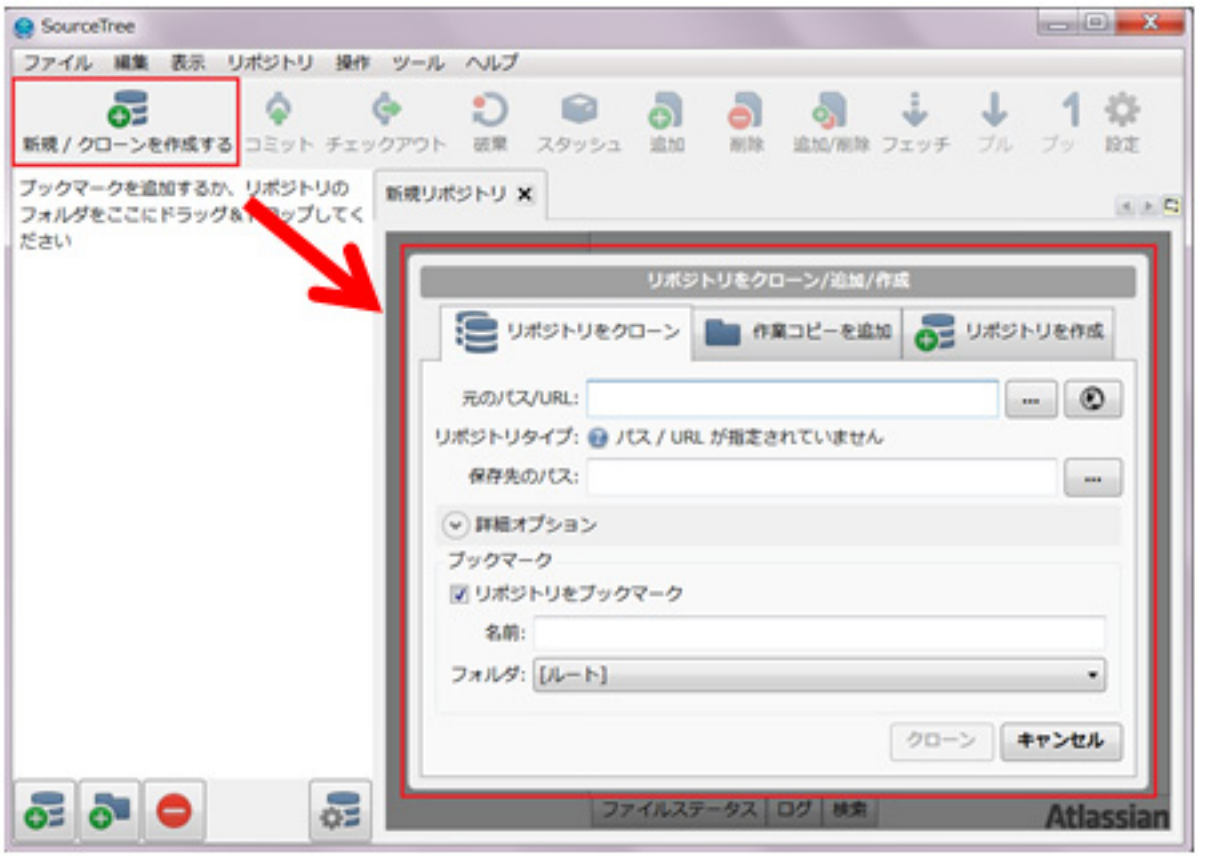
ここでは Git クライアント「SourceTree」を使って、お客様環境（PC）にローカルリポジトリを作成する方法をご説明します。

リモートリポジトリの作成方法はCPI サーバーにリモートリポジトリを作成する方法をご参照ください。

お客様領域（/usr/home/ユーザー ID/）直下に「remote-repo」というディレクトリ（リモートリポジトリ）があることを前提にご説明します。

### 1 SourceTree を起動します。

### 2 画面左上の「新規/クローンを作成する」をクリックします。



## 3 「リポジトリをクローン」タブをクリックし、各項目を入力します。


### 元のパス/URL

**リモートリポジトリを指定します。**

```
ssh://SSH ユーザー @ ドメイン名(または IP アドレス):ポート番号/usr/home/ユーザー ID/remote-repo
```

**※「ユーザー ID/」以降は、お客様が CPI サーバーで作成されたリモートリポジトリの場所によって異なります。**

**リモートリポジトリの場所は、 CPI サーバーに SSH 接続して、リモートリポジトリのディレクトリに移動したあと「pwd コマンド」を実行してご確認ください。**

```
$ pwd


/usr/home/ユーザー ID/remote-repo  ←現在のディレクトリの場所が表示されます
```

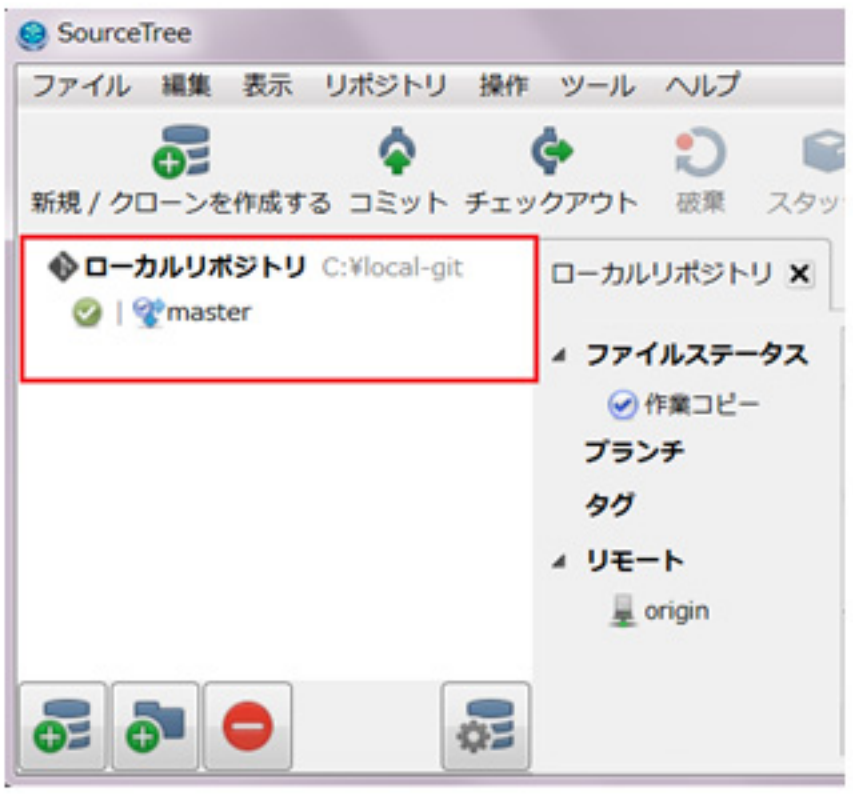
## 保存先のパス

お客様環境（PC）の任意のパスをご入力ください。ローカルリポジトリが作成される場所となります。

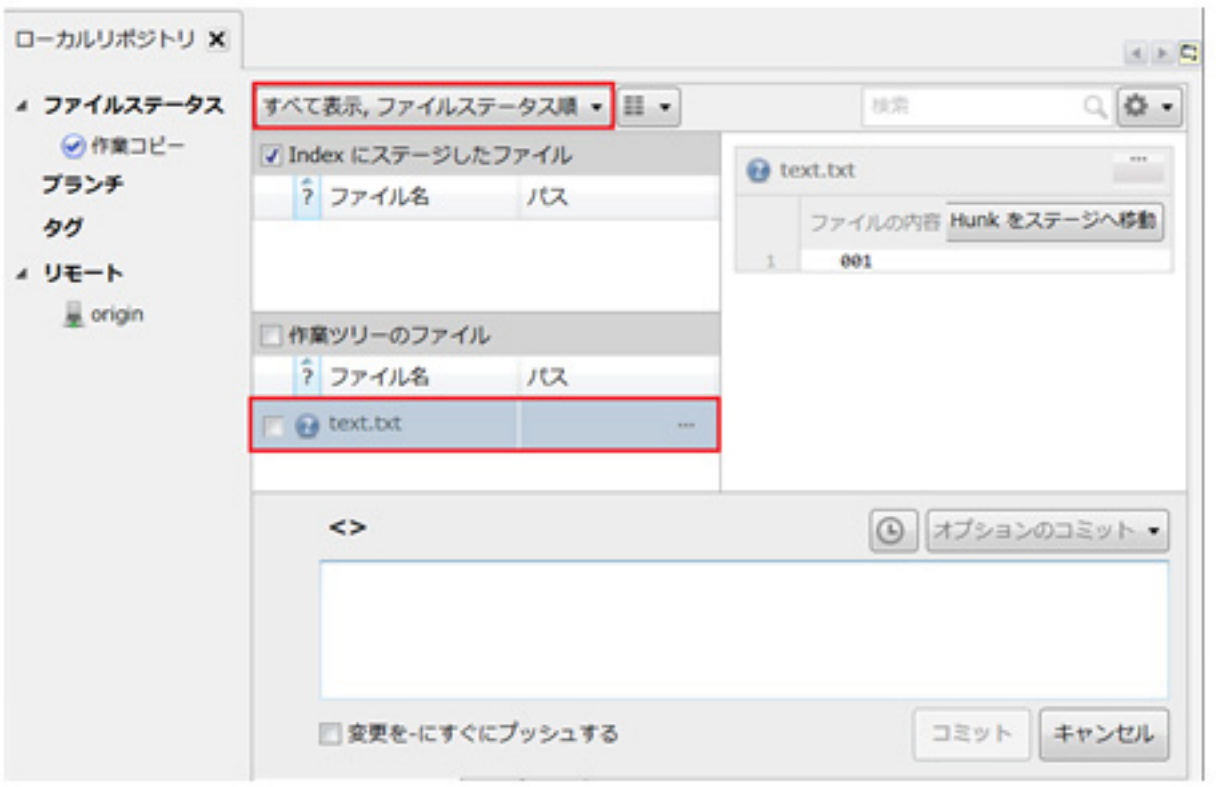
## 名前

任意の名前をご入力ください

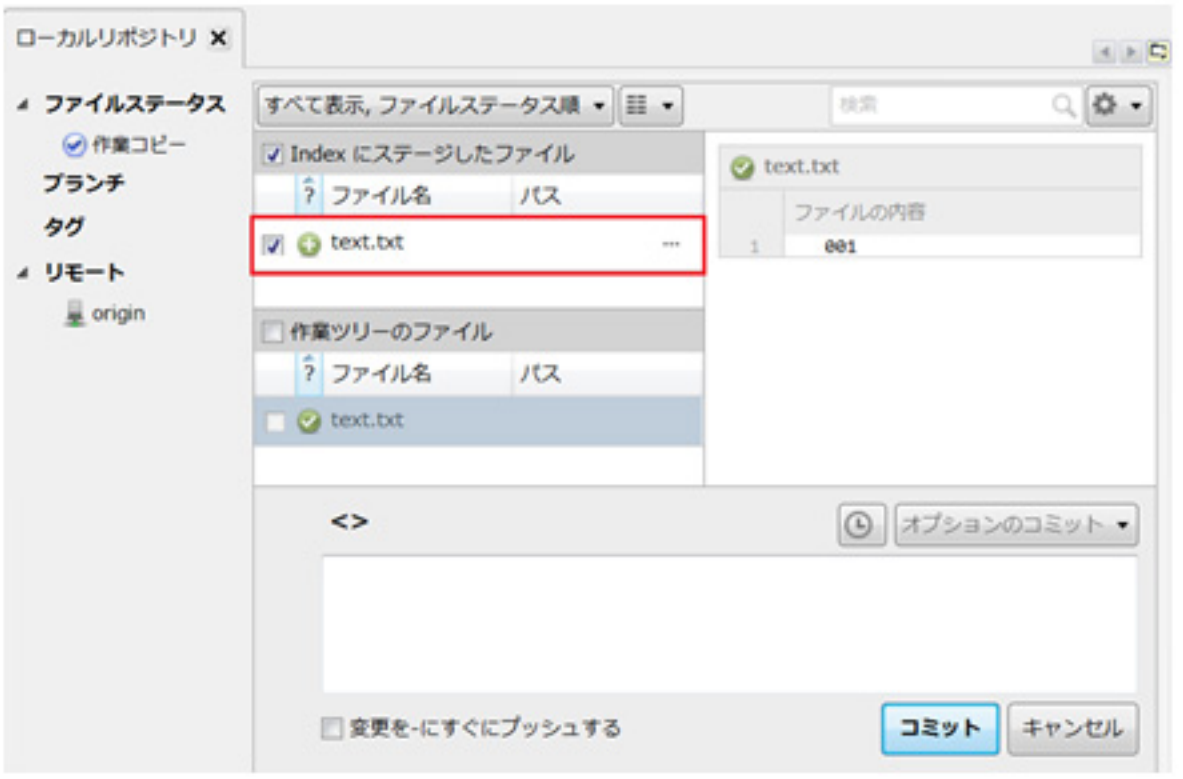
**4** ローカルリポジトリが作成されますと、画面左側に表示されます。



**5** 表示の絞り込みで「すべて表示」を選択すると、ローカルリポジトリ内でファイルの変更（新規作成/修正など）があれば、下図のように対象ファイルが表示されます。

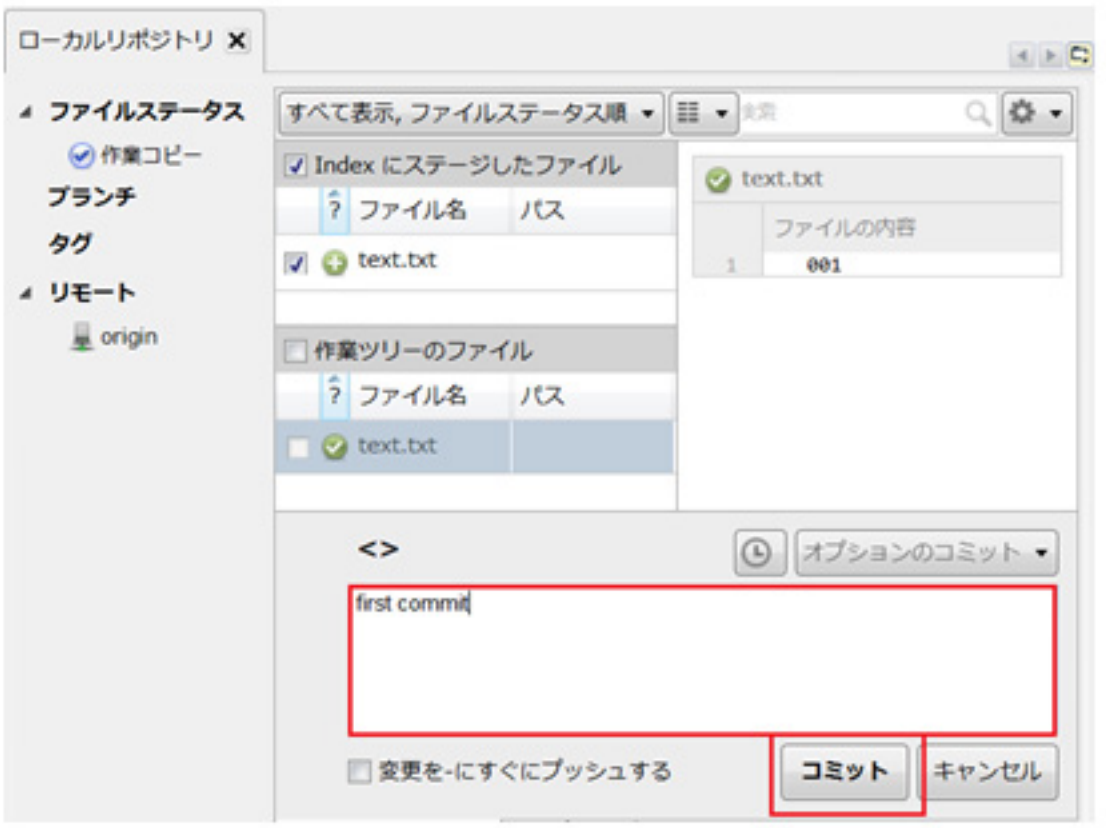



**6** ファイル名の左のチェックボックスにチェックを入れると、「Index にステージしたファイル」に同じファイルが表示されます。



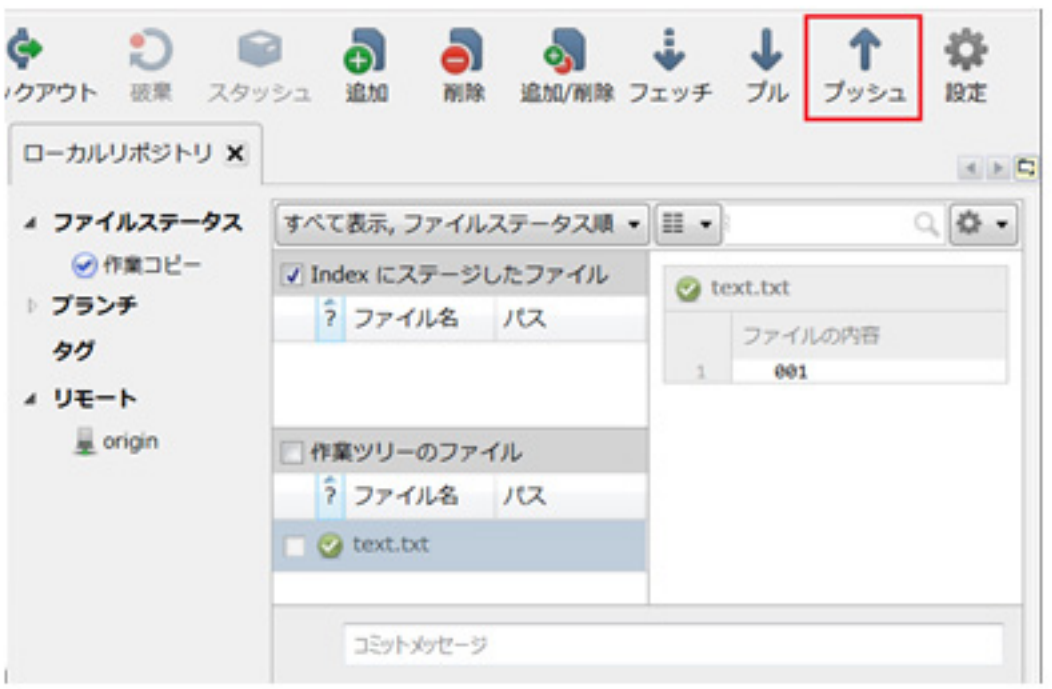
## 7 コメント欄にコメントを記入し、「コミット」ボタンをクリックするとローカルリポジトリ内でファイルの変更がコミットされます。

※コメントが日本語の場合、文字化けする場合があります。

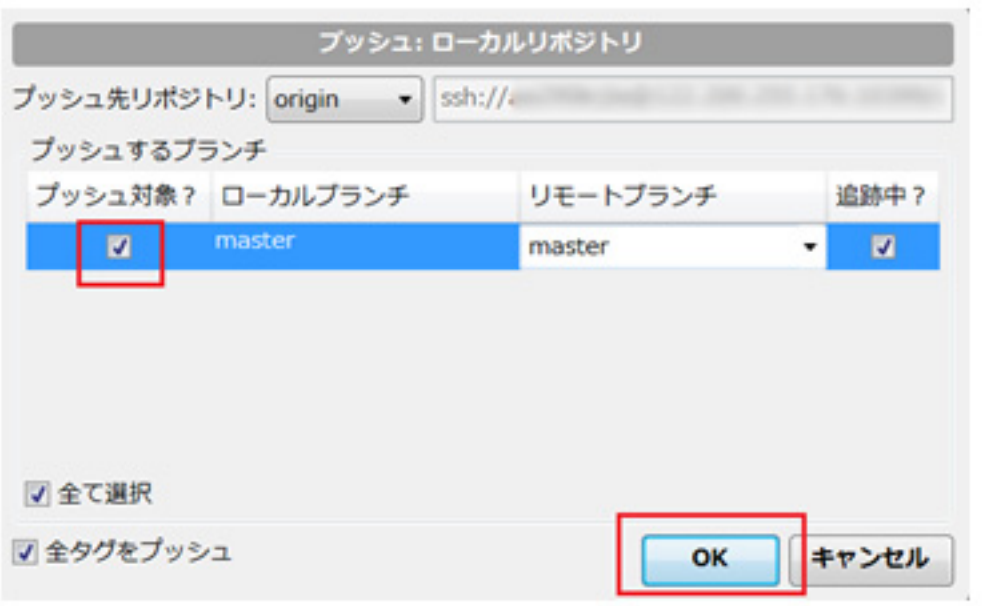



# ローカルリポジトリをリモートリポジトリに push します

## 1 画面上段の「プッシュ」をクリックします。



## 2 「プッシュ対象？」にチェックを入れ「OK」ボタンをクリックすると、リモートリポジトリに push できます。




**POINT**

リモートリポジトリに push したあと、CPI サーバーに SSH 接続し、リモートリポジトリのディレクトリで以下コマンド（git log）を実行すると push されたことが分かります。

```
$ git log

commit b7f5c65fe46c929e51231s6ac6397f35s0a62fe3
Author: Your Name
Date: Wed Oct 8 18:34:05 2014 +0900

first commit
```

# CPI サーバーにリモートリポジトリを作成する方法（Mac）

**目次**



## CPI サーバーにリモートリポジトリを作成する方法

ここでは Mac のターミナルを使って、 CPI サーバーにリモートリポジトリを作成する方法を説明します。

以下、コマンド文は行頭に「$」、コマンド実行後の応答文は行頭に「>」で表示します。

### 1 Mac 上で「移動」→「ユーティリティ」→「ターミナル」から「ターミナル」を起動し、ローカルリポジトリにするフォルダに移動します。

### 2 以下のコマンドを実行し、お客様環境で Git の最低限の設定をします。

※予め Mac に Git プログラムがインストールされている必要があります。
※「Your Name」と「you@example.com」にはお客様にてご指定ください。

```
$ git config --global user.name "Your Name"
$ git config --global user.email you@example.com
```

### 3 CPI サーバーに SSH でログインします。

**※それぞれの区切りに半角スペースが必要ですのでご注意ください。**

```
$ ssh -l SSH アカウント名 -p SSH のポート番号 ドメイン名（または IP アドレス）
```

```
> Password:
```

## 4 CPI サーバーでリモートリポジトリにするディレクトリに移動します。

※ここではお客様環境（/usr/home/ユーザー ID/）直下に「repo」というディレクトリを作成し、リモートリポジトリの設定を行ないます。

### 1.「repo」ディレクトリを作成

```
$ mkdir repo
```

### 2.「repo」ディレクトリに移動

```
$ cd  repo
```

### 3.リモートリポジトリを作成

bare の手前の「--」は半角ハイフンが 2 つ

```
$ git init --bare
```

正常に作成できたら以下のメッセージが表示されます

```
> Initialized empty Git repository in /usr/home/*******/repo/
```

※「*******」にはユーザー ID が表示されます。

**POINT**

CPI サーバーで、HTTP/HTTPS プロトコル経由の clone や pull をする場合には、 CPI サーバーでリモートリポジトリを作成したあと、以下のコマンドを実行してください。

$ git update-server-info

$ mv ./hooks/post-update.sample ./hooks/post-update

$ chmod +x ./hooks/post-update

# お客様環境（ローカルリポジトリ）に CPI サーバー のリモートリポジトリを clone する方法

## 1 お客様環境でローカルリポジトリにするディレクトリを作成します。

※ここでは「local-repo」というディレクトリを作成します。

### 1.お客様環境の任意の場所に「local-repo」ディレクトリの作成

```
$ mkdir local-repo
```

### 2.「local-repo」ディレクトリに移動

```
$ cd local-repo
```

## 2 リモートリポジトリを clone する。

### ■SSH 経由で clone する場合

```
$ git clone ssh://SSH ユーザー名 @ ドメイン名（または IP アドレス）:SSH ポート番号/usr/home/ユーザー ID/repo/
```

**※上記の「/usr/home/ユーザーID/repo/」で、「ユーザーID/」以降はCPIサーバーのリモートリポジトリの場所によって異なりますので、ご注意ください。**

### ■HTTP/HTTPS 経由で clone する場合

```
$ git clone http://ドメイン名（または IP アドレス）/repo
```

**※HTTP/HTTPS 経由の場合は、リモートリポジトリがドキュメントルート配下に作成されている必要があります。**

## お客様環境のローカルリポジトリでファイルを作成し add（索引に追加） したのち commit する方法

### 1 お客様環境（ローカルリポジトリ）で、ファイル（test.txt）を作成。

```
$ touch test.txt
```

ステータスを確認するには「git status」を実行します。

```
$ git status
```

```
> On branch master

> Initial commit

> Untracked files:

>   (use "git add <file>..." to include in what will be committed)

> test.txt



> nothing added to commit but untracked files present (use "git add" to track)
```

### 2 作成したファイルを add （索引に追加）します。

```
$ git add test.txt

$ git status
```

```
>On branch master

>Initial commit

>Changes to be committed:

  (use "git rm --cached <file>..." to unstage)



>new file:   test.txt
```

## 3 コミットします。

```
$ git commit -m "first commit"
```

※「first commit」の箇所は、任意の文字をご指定ください。

日本語を入力されますと文字化けする場合があります。

```
$ git status
```

```
>On branch master

>Your branch is based on 'origin/master', but the upstream is gone.

  (use "git branch --unset-upstream" to fixup)

>nothing to commit, working directory clean
```

## 4 お客様環境のローカルリポジトリに加えた変更を CPI サーバー のリモートリポジトリに push します。

```
$ git push -u origin master
```

```
>Password:
```

※パスワードは、 SSH アカウントのパスワードです。

## 5 CPI サーバー（リモートリポジトリ）に SSH 接続し、該当ディレクトリに移動後、以下のコマンドを実行すると変更履歴（ログ）をご確認いただけます。

```
$ git log
```

```
>commit 1a37473053c0b57r616f86df3de2e24135ef070e

>Author: Your Name <you@example.com>

>Date:   Wed Oct 8 14:27:52 2014 +0900


    > first commit
```

# CPI サーバーにローカルリポジトリを作成する方法（Windows/Mac）

**目次**



## CPI サーバーにローカルリポジトリを作成する方法

ここでは Windows の SSH クライントソフトや Mac のターミナルを使って、CPI サーバーにローカルリポジトリを作成する方法を説明します。

リモートリポジトリは外部の Git リポジトリ（GitHub 等）に存在することを前提とします。

以下、コマンド文の行頭を「&」は囲み枠、コマンド実行後の応答文の行頭を「>」で表示します。

### 1 Windows の SSH クライントソフトまたは Mac のターミナルを起動します。

### 2 CPI サーバーに SSH でログインします。

```
$ ssh -l SSHアカウント名 -p SSHのポート番号 ドメイン名（または IP アドレス）
```

※それぞれの区切りに半角スペースが必要ですのでご注意ください。

```
>Password:
```

※SSH のパスワードを入力します。入力しても文字は表示されません。

SSH クライントソフトの場合は、SSH アカウント名やパスワード、SSH のポート番号を所定の箇所に入力し、SSH 接続してください。

## 3 以下のコマンドを実行し Git の最低限の設定をします。

```
$ git config --global user.name "Your Name"
$ git config --global user.email you@example.com
$ chmod go-rwx ~/.gitconfig
```

※「Your Name」と「you@example.com」にはお客様にてご指定ください。

**POINT**

HTTP/HTTPS で clone やpush　する場合は以下の設定を行なってください。

CPI サーバーのホームディレクトリ（/usr/home/ユーザー ID）直下に .netrc ファイルを配置する
$ vim ~/.netrc
$ chmod go-rwx ~/.netrc

.netrc ファイルの内容（GitHub の場合）
machine github.com
login ${GITHUB_USERNAME}
password ${GITHUB_PASSWORD}

## 4 CPI サーバーに GitHub 上のリモートリポジトリを clone する。

### ■HTTP/HTTPS 経由で clone する場合

```
$ git clone https://github.com/${GITHUB_USERNAME}/パス
```

### ■Git経由で clone する場合

```
$ git clone git://github.com/パス
```

※CPI サーバーのローカルリポジトリの変更を外部の Git リポジトリ（GitHub 等）上のリモートリポジトリに push するには、HTTP/S プロトコルを使用する必要があります。

**重要**

上記は GitHub の設定例であり、変更される場合があります。
詳細は外部の Git リポジトリ（GitHub 等）にてご確認ください。

## CPI サーバーのローカルリポジトリでファイルを作成し add（索引に追加）したのち commit し、最後にリモートリポジトリに push します

### 1 CPI サーバーでローカルリポジトリにするディレクトリを作成します。

※ここでは「local-repo」というディレクトリを作成します

```
$ mkdir local-repo
```

### 2 お客様環境の任意の場所に「local-repo」ディレクトリの作成し移動します。

```
$ cd local-repo
```

### 3 CPI サーバー（ローカルリポジトリ）で、ファイル（test.txt）を作成し、 add したのち commit します。

#### 1.ファイル（test.txt）の作成します

```
$ touch test.txt
```

#### 2.作成したファイルを add （索引に追加）します

```
$ git add test.txt
```

#### 3.commit します

```
$ git commit -m "first commit
```

### 5 CPI サーバーのローカルリポジトリの変更をリモートリポジトリに push します。

```
$ git push -u origin master
```

# CPI サーバーにリモートリポジトリとローカルリポジトリを作成する方法（Windows/Mac）

**目次**



## CPI サーバーにリモートリポジトリを作成する方法

ここでは Windows の SSH クライントソフトや Mac のターミナルを使って、CPI サーバーにローカルリポジトリを作成する方法を説明します。

リモートリポジトリは外部の Git リポジトリ（GitHub 等）に存在することを前提とします。

以下、コマンド文の行頭は「&」、コマンド実行後の応答文の行頭は「>」で表示します。

### 1 Windows の SSH クライントソフトまたは Mac のターミナルを起動します。

### 2 CPI サーバーに SSH でログインします。

```
$ ssh -l SSH アカウント名 -p SSH のポート番号 ドメイン名（または IP アドレス）
```

**※それぞれの区切りに半角スペースが必要ですのでご注意ください。**

```
>Password:
```

**※SSH のパスワードを入力します。入力しても文字は表示されません。**

**SSH クライントソフトの場合は、 SSH アカウント名やパスワード、 SSH のポート番号を所定の箇所に入力し、SSH 接続してください。**

## 3 以下のコマンドを実行し Git の最低限の設定をします。

```
$ git config --global user.name "Your Name"
$ git config --global user.email you@example.com
$ chmod go-rwx ~/.gitconfig
```

※「Your Name」と「you@example.com」にはお客様にてご指定ください。

## 4 CPI サーバーでリモートリポジトリを作成する。

### 1.任意の場所に「remoto-repo」ディレクトリを作成

※ディレクトリ名も任意で問題ございません。
※ここではお客様領域（/usr/home/ユーザー ID/）直下に「remote-repo」というディレクトリを作成します。

```
$ mkdir remote-repo
```

### 2.「remote-repo」ディレクトリに移動

```
$ cd  remote-repo
```

### 3.リモートリポジトリを作成

```
$ git init --bare
```

（bare の手前の「--」は半角ハイフンが 2 つ）

正常に作成できたら以下のメッセージが表示されます

```
>Initialized empty Git repository in /usr/home/*******/remote-repo/
```

※「*******」にはユーザーIDが表示されます。

**POINT**

HTTP/HTTPS プロトコル経由の clone や pull をする場合には、リモートリポジトリを作成したあと、以下のコマンドを実行してください。

```
$ git update-server-info
$ mv ./hooks/post-update.sample ./hooks/post-update
$ chmod +x ./hooks/post-update
```

※HTTP/HTTPS プロトコル経由で push はご利用できません。

## CPI サーバーにローカルリポジトリを作成する方法

### 1 CPI サーバーでローカルリポジトリを作成しリモートリポジトリを clone します。

#### 1.任意の場所に「local-repo」ディレクトリを作成

ここではお客様領域（/usr/home/ユーザーID/）直下に「local-repo」というディレクトリを作成します。
※ディレクトリ名も任意で問題ございません。

```
$ mkdir local-repo
```

#### 2.「local-repo」ディレクトリに移動

```
$ cd  local-repo
```

### 2 リモートリポジトリを clone します。

#### ■Local プロトコルを使用する場合

```
$ git clone /usr/home/ユーザーID/remote-repo
```

#### ■HTTP/S プロトコルを使用する場合

```
$ git clone http://ドメイン名（またはIPアドレス）/remote-repo
```

※上記の「/usr/home/ユーザー ID/remote-repo/」で、「ユーザー ID/」以降は CPI サーバーのリモートリポジトリの場所によって異なりますので、ご注意ください。

## CPI サーバーのローカルリポジトリでファイルを作成し、add したのち commit して最後にリモートリポジトリに push する

**1** 「local-repo」直下にリモートリポジトリと同じ名前のディレクトリ「remote-repo」が作成されているので移動します。

```
$ cd ~/local-repo/remote-repo
```

**2** ファイルを作成し add したのち commit して最後にリモートリポジトリに push します。

### 1.ファイル（test.txt）を作成します

```
$ touch test.txt
```

### 2.作成したファイルを add （索引に追加）します

```
$ git add test.txt
```

### 3.コミット（commit）します

```
$ git commit -m "first commit"
```

### 4.リモートリポジトリに push します

```
$ git push -u origin master
```

# WAF（Web アプリケーションファイアウォール）

Web サイトの改ざんやデータベース情報の不正入手など Web アプリケーションの脆弱性を狙うサイバー攻撃から Web サイトを防御するソフトウェアまたはハードウェアです。
CPI では株式会社ジェイピー・セキュアが提供するソフトウェアタイプの SiteGuard Server Edition を採用しております。
※弊社が提供する WAF には攻撃検出時のアラートメール通知機能はございません。

WAF は従来のファイアウォールや IDS/ADS では防御しきれなかった攻撃の検知・防御が可能となります。

ファイアウォールは、主に不要なサービス（サービスポート）へのアクセスを制限し、不正なアクセスの防御を行っております。
また IDS/ADS では不正なアクセスを検知するとアクセス元の通信を遮断します。

しかし Web サイトなど、公開されているサービス（HTTP や HTTPS）はファイアウォールでは制限されない形となるため Web アプリケーションに脆弱性があると攻撃の脅威となります。

WAF ではファイアウォールや IDS/ADS では検知できない攻撃を検出することができます。
例として、データベースを不正に操作する「SQL インジェクション」のパラメーターが含まれたアクセスがあった場合、その通信を遮断するといった対策をとることができます。

**重要**

・WAF（Web アプリケーションファイアウォール）の有償オプションサービスです。
　ご利用に際しては、マイページからお申し込みをお願いいたします。
・サーバーのご契約途中からでもお申込みいただけます。
　主契約ドメイン、バーチャルドメインそれぞれでご利用いただけます。
・共用 SSL のアクセス用 URL への通信に対しても、WAF（Web アプリケーションファイアウォール）は動作いたします。

# WAF 機能の ON/OFF 方法

## 1 コントロールパネルの【制作ツール】＞【WAF】をクリックします。


## 2 「WAF の設定状況」の「有効」「無効にする」をクリックすることで切り替えられます。

初期設定は「無効」になっています。


## 3 「有効」にすると以下のように表示が切り替わります。


**POINT**

「サブドメイン」の「有効/無効」は、親ドメイン（契約ドメインまたはバーチャルドメイン）の「有効/無効」の設定に準じます。サブドメイン単位での「有効/無効」の設定は出来かねます。ご了承ください。

# WAF の検出ログの確認方法

## 1 コントロールパネルの【制作ツール】＞【WAF】をクリックします。


## 2 「対象期間」を選択し、「ログを表示する」ボタンをクリックします。

## 3 検知している場合は、詳細が表示されます。


検出ログ

対象期間 当月

検索文字列

ログを再表示する　表示されているログをダウンロードする

| 日時 | 動作 | クライアントホスト | メソッド | URL | 検出シグネチャ |
|---|---|---|---|---|---|
| 2014-09-25 21:39:36 | BLOCK | | GET | http:// | |
| 2014-09-27 00:44:59 | BLOCK | | GET | http:// | |

| | |
|---|---|
| 日時 | クライアントから接続された時刻 |
| 動作 | MONITOR：監視（ログには残りますが、リクエストは拒否しません）<br>BLOCK：拒否（リクエストを拒否し、検出メッセージをクライアントに応答します） |
| クライアントホスト | クライアントの IP アドレス |
| メソッド | HTTP の要求メソッド(GET, POST 等) |
| URL | 接続先 URL |
| 検出シグネチャ | シグネチャ名 |

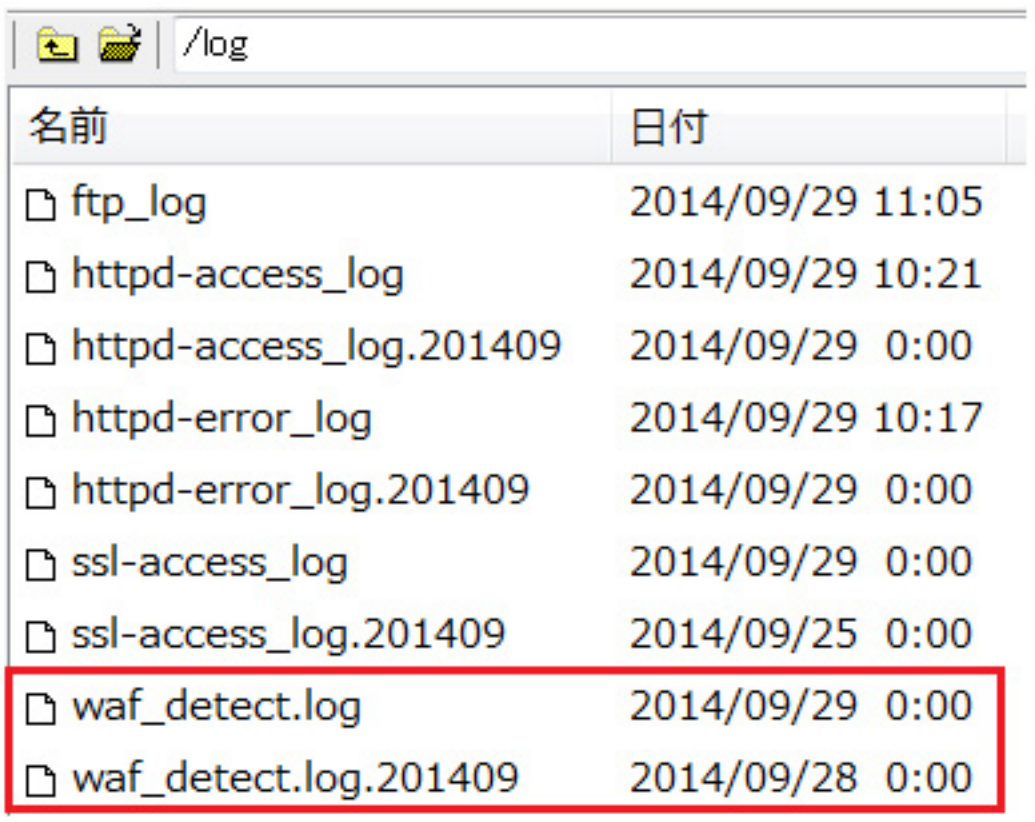
# WAF のログのダウンロード方法

## 1 FTP アカウントの設定で FTP アカウントを作成します。

※FTP アカウントのログインディレクトリは「/」または「/log」になっている必要があります。

## 2 「/log」ディレクトリ配下の「waf-detect.log」が当日分、「waf-detect.log.yyyymm」が１ ヶ月分のログファイルとなります。

※WAF のログは３ヶ月前まで保存されています。それより過去のログは自動で削除されますので、必要な場合はお客様にてダウンロードして保存してください。




**重要**

ログの保存ディレクトリ「/log」やログファイルの名前を変更（リネーム）した場合、 WAF が正常に機能しなくなりますのでご注意ください。

# WAF のログの見方

お客様領域の「/log」配下に保存されている WAF のログは以下のフォーマットにて記録されています。

## ログフォーマット

検出状況が 1 行ずつ記録されます。
以下の各項目がスペースで区切られています。

### 時刻

クライアントから接続された時刻を表します。
※仕様上、エポックタイム (1970/01/01 00:00:00(UTC)) からの秒数をミリ秒単位で表示します。

### 接続時間

クライアントとの接続時間をミリ秒単位で表します。

### クライアントホスト

クライアント（接続元）の IP アドレスです。

### 処理結果

常に TCP_MISS/000 となります。

### ファイルサイズ

転送したファイルのサイズ（単位はバイト）です。
（クライアントからサーバーに送られてくるデータサイズ）

## 要求メソッド

HTTP の要求メソッド(GET, POST 等) を表します。

## URL

接続先の URL を表します。

## ユーザ名

常に "-" となります。

## hierarchy code

"DIRECT/接続先 IP アドレス" を表します。

## Content-Type

送受信するファイルの Content-Type を表します。利用できない場合は "-" となります。

## 検出情報

DETECT-STAT:[検査結果]:[検出名]::[検出文字列]:[検出文字列(全体)]:

| | |
|---|---|
| 検出結果 | WAF(攻撃検出) |
| 検出名 | 攻撃検出名称<br>- シグネチャ検査:<br>RULE_SIG/[検出箇所]/[名前]/[シグネチャファイル]/[シグネチャ ID]/[シグネチャ名]<br><br>- URLデコードエラー:<br>RULE_URLDECODE/[検出箇所]/[名前]<br><br>- パラメータ数の上限値の制限:<br>RULE_PARAMS_NUM/[パラメータ数の上限値]<br><br>・[検出箇所]は以下のいずれかとなります。<br>PART_PARAM_NAME|[種別] ⇒ パラメータ変数の名前<br>PART_PARAM_VALUE|[種別] ⇒ パラメータ変数の値<br>PART_URL ⇒ URL<br>PART_PATH ⇒ パス名<br>PART_METHOD ⇒ メソッド<br>PART_REQHEAD ⇒ 要求ヘッダ<br>PART_CLIENT_ADDR ⇒ クライアントアドレス<br>PART_SERVER_ADDR ⇒ サーバーアドレス<br>PART_POST_FILENAME ⇒ 送信ファイル名 (filename 変数で指定される名前)<br><br>・[名前]は、パラメータ変数、ヘッダフィールド名を表示します。<br><br>・[シグネチャファイル]は、 OFFICIAL/CUSTOMの いずれかで表記されます。<br><br>・[種別]は、 PART_GET_PARAM(クエリストリング)、PART_REQBODY (要求本文)、PART_COOKIE (クッキー)のいずれかです。 |
| 検出文字列 | 検出した文字列(シグネチャ検査で検出した場合) |
| 検出文字列（全体） | 検出した箇所の文字列全体(パラメータ、要求ヘッダで検出した場合)<br>290 バイトまで記録されます。 |

## 動作

ACTION:[動作]:

| | |
|---|---|
| 動作 | ・MONITOR　⇒　監視（ログには残りますが、リクエストは拒否しません）<br>・BLOCK　⇒　拒否（リクエストを拒否し、検出メッセージをクライアントに応答します） |

## 判定情報

JUDGE:[判定]:[監視URLの設定]:

| | |
|---|---|
| 判定 | MONITOR　⇒　監視（ログには残りますが、リクエストは拒否しません）<br>BLOCK　⇒　拒否（リクエストを拒否し、検出メッセージをクライアントに応答します） |
| 監視URLの設定 | 0　⇒　固定となります |

## 検索キー

SEARCH-KEY:[検索キー]:

| | |
|---|---|
| 検索キー | 時刻(エポックタイム).apacheコネクション ID で表します。 |

# 通信を遮断した際の表示画面

通信を遮断した際にはブラウザに以下の画面が表示されます。


**POINT**

WAF が有効な状態で "WAF-TEST-SIGNATURE" を含む 以下の URL にアクセスするとブロックが発動しますので、事前にご確認いただけます。

http:// IP アドレス/WAF-TEST-SIGNATURE

http://ドメイン/WAF-TEST-SIGNATURE

※ドメインでテストされる場合は、該当ドメインが CPI サーバーに向いている必要があります。
※「サブドメイン追加設定」で設定したドメインは、 IP アドレスでのご確認はできませんのでご注意ください。

# 特定シグネチャの除外方法

お客様にて以下の書式で「.htaccess」ファイルに記述のうえ、ドキュメントルート配下に設置していただくことで、特定のシグネチャを除外することができます。
※すでに「.htaccess」ファイルを設置されている場合は、既存の「.htaccess」ファイルに追記してください。

以下は、例として「signature1」と「signature2」の二つのシグネチャを除外した場合の書式です。

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig signature1,signature2
</IfModule>
```

※上記の「signature1」と「signature2」の箇所には【シグネチャ名】が入ります。

シグネチャ名については、「WAF の検出ログの確認方法」で確認方法をご案内します。

**目次**

## 親ディレクトリと子ディレクトリの関係

親ディレクトリで設定した除外シグネチャは、子ディレクトリに継承されます。
ただし、子ディレクトリにおいても除外設定を行った場合は、親ディレクトリの除外設定は継承されません。

### (親ディレクトリ)

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig signature1,signature2
</IfModule>
```

### (子ディレクトリ)

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig signature3
</IfModule>
```

上記の場合、子ディレクトリでは、 signature3 のみ除外設定されます。

## 親ディレクトリの除外設定の継承を解除する方法

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig clear
</IfModule>
```

上記の設定を子ディレクトリにすると、親ディレクトリ（上位にあるすべて）の除外設定は継承されません。
子ディレクトリを含む下層ディレクトリは除外設定無しの状態になります。

## 除外設定方応用編

**1** 攻撃が検知されると、コントロールパネルの【制作ツール】＞【WAF】よりログを確認することができます。


**2** WAF の除外設定は Web サーバーに .htaccess を設置し除外設定を行います。
記述は下記の通りです。

### 『特定のシグネチャを指定して除外』

```
SiteGuard_User_ExcludeSig
[ シグネチャ ID|シグネチャ名|urldecode|all|clear ]
```

### 【.htaccess　記述例】

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig xss-tag-1,xss-tag-filter
</IfModule>
urldecode：URL デコードエラーによる検出を除外
all：すべてのシグネチャと URL デコードエラーによる検出を除外
clear：上位階層設定を解除
```

## 『特定ファイルの除外』

```
<Files ~ "filename¥.php$">
SiteGuard_User_ExcludeSig
[ シグネチャ ID|シグネチャ名|urldecode|all|clear ]
</Files>
```

設置した階層以下の filename.php が除外されます。

### 【.htaccess　記述例】

```
<Files ~ "sample¥.php$">
SiteGuard_User_ExcludeSig all
</Files>
```

## 『クエリを指定した除外』

```
SiteGuard_User_ExcludeSig_With_ParamName
[ シグネチャ ID|シグネチャ名|urldecode|all|clear] [パラメータ名 ]
```

### 【.htaccess　記述例】

```
SiteGuard_User_ExcludeSig_With_ParamName all xss
```

例の「all xss」 は、キー（xss）に対してすべてのパラメータ（all）を許可しています。実際にどのようなパラメータを受け取るかが分かっている場合は、 all を param に変更ください。

## 『IP アドレスを指定した除外方法』

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig ip(IP アドレス)
</IfModule>
```

### 接続元 IP アドレス「192.168.0.1」からのアクセスを除外 【.htaccess　記述例】

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig ip(192.168.0.1)
</IfModule>
```

### 接続元 IP アドレス「192.168.0.1」、「192.168.0.2」、「192.168.0.3」からのアクセスを除外 【.htaccess　記述例】

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig ip(192.168.0.1)
SiteGuard_User_ExcludeSig ip(192.168.0.2)
SiteGuard_User_ExcludeSig ip(192.168.0.3)
</IfModule>
```

IP アドレスが複数ある場合は、１行ずつ IP アドレスを記述してください。

### 接続元 IP アドレス「192.168.0.1」、「192.168.0.2」、「192.168.0.3」からのアクセスを除外し、/usr/home/ユーザー ID/html/ディレクトリ名/test　は除外しない 【.htaccess　記述例】

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig ip(192.168.0.1)
SiteGuard_User_ExcludeSig ip(192.168.0.2)
SiteGuard_User_ExcludeSig ip(192.168.0.3)
</IfModule>
```

### 【.htaccess　記述例】 (例では /usr/home/ユーザー ID/html/ディレクトリ名/testに .htaccess を設置)

```
<IfModule siteguard_module>
SiteGuard_User_ExcludeSig clear
</IfModule>
```

# 外部バックアップサービス

本サービスは、「ユーザ領域」「メール領域」「データベース領域」のコピーを行います。 バックアップ用サーバーに日付ごとにディレクトリを作成し、その中に各データをコピーします。 本サービス利用時には、バーチャルドメインのデータも含めて１台のサーバーのバックアップを行いますので、ドメイン単位での外部バックアップの適用の選択は行えません。 バックアップの取得方法は、「FTP」の他に「SFTP」もご利用いただけます。

**重要**

■3 日に 1 回バックアップを行います。（保存は 3 世代まで行われます。）
■バックアップデータは圧縮せずに保存されます。

## 外部バックアップサービスによってバックアップされたデータのリストア方法について。

**POINT**

外部バックアップデータのうち、ウェブ領域のデータはお客様にて FTP ソフト等をご利用いただき、サーバへアップロードをしてください。メールデータはサーバへリストアすることはできません。データベースのデータについては下記資料を参考にしてください。

当社では、お客様がご利用のデータ、ならびにバックアップにより取得されたデータの取り扱いはお客様にお願いしております。また、バックアップデータのリストアの一例を下記にまとめさせていただきましたが、お客様がご利用のデータ容量やファイル数、データ形式等によってはリストアできない場合もございます。あらかじめご了承願います。