

クリエイティブなデータの管理



Extensis™

Portfolio™ NetPublish

ユーザーガイド



連絡先

Extensis

1800 SW First Avenue,
Suite 500 Portland, OR 97201 USA
電話: +1 (503) 274-2020
FAX: +1 (503) 274-0530
インターネット: <http://www.extensis.com>

Extensis Europe

First Floor, Century House
The Lakes
Northampton NN4 7SJ
United Kingdom
電話: +44(0)1604 636 300
FAX: +44 (0)1604 636 366
電子メール: info@extensis.co.uk

©2005 Extensis, a division of Celartem, Inc. 本書ならびに本書に記載されているソフトウェアは著作権保護の対象であり、無断複写・複製・転載が禁止されています。本書または本書に記載されているソフトウェアの全部または一部を、Extensisの書面による承諾なしに複製することはできません。ただし、ソフトウェアの通常の利用やソフトウェアのバックアップ作成は除きます。この例外であっても他者に対する複製は許可されません。登録済みおよび申請中の米国特許に基づいてライセンス化されています。

ExtensisはExtensisの商標です。Extensisのロゴ、Extensis Library、Font Reserve、Font Reserve Server、Font Vault、Font Sense、Portfolio、Portfolio Server、Portfolio NetPublish、NetPublish、Suitecase、Suitecase ServerはExtensisの商標です。Celartem、Celartem, Inc.、Celartemのロゴ、PixelLive、PixelSafeはCelartem, Inc.の商標です。Adobe、Acrobat、Illustrator、Photoshop、PostScriptはAdobe Systems, Incorporatedの商標です。Apple、Apple Script、FontSync、Macintosh、Mac OS 9、Mac OS X、PowerPC、QuickDrawはApple Computer, Inc.の登録商標です。Microsoft、Internet Explorer、Windows、Windows XP、Windows 2000、Windows NT、Windows ME、Windows 98はMicrosoft Corporationの登録商標です。その他すべての商標は各社が所有しています。

株式会社セラーテムテクノロジー

電子メール: sales_ap@celartem.com
<http://www.celartem.com/jp/>

プレスに関するお問い合わせ

電話: (503) 274-2020 x129
電子メール: press@extensis.com

カスタマーサービス

インターネット上: <http://www.extensis.com/customerservice/>
電話: (800) 796-9798

テクニカルサポート

インターネット: <http://www.extensis.com/support/>

マニュアルに関するご意見

インターネット: <http://www.extensis.com/helpfeedback/>

本製品の一部では、さまざまなオープンソースプロジェクトで開発されたソフトウェアコンポーネントを使用しています。そうしたコンポーネントのソースコードのライセンスおよび使用可能性については、本製品に付属する著作権表示ファイルであるLICENSES.TXTに示されています。ソフトウェアコンポーネントの利用に関する情報については、該当するライセンスを参照してください。

Extensisは、製品の購入日より30日間、通常の使用状況において、ソフトウェアが記録されているディスクに材質上および製造上の欠陥が無いことを保証します。製品をExtensisから直接購入し、30日以内に障害が発生した場合、ディスクをExtensisに返送いただければ交換いたします。交換用に送付されたすべての製品は、交換の前にExtensisで登録する必要があります。販売店から購入したExtensis製品は販売店によって保証され、販売店の返品規定の対象となります。この保証は交換に限定され、その他の損害を含みません。その他の損害には、利益の損失、特殊な要求、付随的な要求、その他の同様の要求などがありますが、これらに限定されるものではありません。本ソフトウェアは、「現状のまま」提供されます。上記の明示的な保証を除き、Extensisは、同梱のソフトウェアの品質、性能、商品性、または特定目的の適合性に関して、明示的または暗黙的を問わず他のいかなる保証もいたしません。



目次

Portfolio NetPublish の紹介	1
Portfolio NetPublish によるこそ.....	1
システムとソフトウェアの必要条件.....	1
技術サポート.....	2
本書の内容.....	2
インストール.....	3
NetPublish Assistant	7
カタログ選択ステップ.....	8
開始ステップ.....	8
基本設定ステップ.....	10
検索ステップ.....	11
検索結果ステップ.....	14
詳細情報ステップ.....	16
コレクションステップ.....	17
サイトの公開ステップ.....	18
NetPublish ファイルの場所	21
NetPublish の設定	23
NetPublish Server の管理.....	23
NetPublish サイトの設定.....	25
カスタムサイトの作成	29
テンプレートの変更.....	29
JavaScript API.....	30
Mozilla JavaScript オブジェクト.....	42
NetPublish のコマンド.....	43
NetPublish のカスケーディングスタイルシート (CSS).....	52
索引	55

Portfolio NetPublish の紹介

Portfolio NetPublish によるこそ

Portfolio NetPublish (ポートフォリオ ネットパブリッシュ) は、Portfolio のカタログとしてファイルをインターネットにパブリッシュ (配信、公開) するための簡単なツールです。NetPublish を使えば、複数のユーザーがインターネットを介してファイルにアクセスしたり、必要に応じて、選択したファイルを集めてダウンロードしたりすることができます。

Web 開発者や JavaScript プログラマ以外の方のために、NetPublish には、Portfolio のカタログから動的な Web サイトを手順に従って作成できる便利なアシスタントが組み込まれています。プロがデザインしたテンプレートを使って、専門的なサイトを作ることができます。柔軟性も高く、ソースファイルを直接編集し、具体的なニーズに合わせて各ページをカスタマイズすることもできます。

サイトをさらに細かく制御する必要がある場合に備え、本書には、サイトの作成や変更に必要な JavaScript および Portfolio 固有のコマンドがよく分かる詳細な API も入っています。NetPublish は、業界標準の Mozilla サーバーサイド JavaScript Web 公開エンジンに Portfolio 固有の機能一式を加えて拡張したものです。JavaScript の知識を活用して、専用のカスタムサイトを変更および作成することができます。

Portfolio NetPublish は、初心者はもちろん、Web サイト制作のプロのニーズも満たす、豊富な機能と柔軟性を兼ね備えています。



システムとソフトウェアの必要条件

Portfolio NetPublish Server (ポートフォリオ ネットパブリッシュ サーバー) をインストールおよび使用する場合は、次の要件を満たしている必要があります。

Windows

- Windows XP Pro、Windows 2000 Server、または Windows Server 2003
- Pentium 1GHz (2GHz 以上推奨)
- 256 MB 以上のシステム RAM (512 MB 以上推奨)
- 50 MB 以上のハードディスク空き容量。NetPublish で作成したサイトとキャッシュファイルを保管する場合は、さらに容量が必要です。
- Microsoft Internet Information Services (IIS) バージョン 5 以上
- Internet Explorer 6 以上、Mozilla Firefox 1.0.4 以上

Macintosh

- Mac OS® X 10.3 または Mac OS X Server 10.3 以上
- Apple Macintosh® G4、G5、または Xserve
- 256 MB のシステム RAM (512 MB 以上推奨)
- 50MB 以上のハードディスク空き容量。
注：NetPublish で作成したサイトとキャッシュファイルを保管する場合は、さらに容量が必要です。
- Apache 1.3 (デフォルトでインストールされているもの)
- Internet Explorer 5.2.3 以上、Mozilla Firefox 1.0.4 以上、Safari 1.3 以上



Portfolio NetPublish クライアントソフトウェアは Portfolio クライアントと一緒に自動的にインストールされます。クライアントのインストール手順とシステム要件については、『Portfolio ユーザーガイド』を参照してください。

技術サポート

詳細は、登録はがきをご覧ください。

* サポートを受けるには、ユーザー登録が必要になります。

本書の内容

本書では、Portfolio NetPublish Windows 版および Macintosh 版について説明します。キーボードショートカットの説明で、「コマンド」は、Macintosh の ⌘ キーに相当します。

メニュー選択は、「メニュー」>「メニュー項目」というように表されています。

具体的な Portfolio NetPublish 機能の最新情報については、次の文書を参照してください。

- お読みくださいファイルー Portfolio NetPublish フォルダ上のこのテキストファイルに最新情報が収められています。
- ホワイトペーパーなどの情報については、Extensis 社 Web サイトの Portfolio ページを参照してください。
<http://www.extensis.co.jp>

インストール

Portfolio NetPublish のインストール

インストールしないように指定しない限り、Portfolio NetPublish は、Portfolio クライアントのインストール時に自動的にインストールされます。

Portfolio NetPublish Server のインストール

Portfolio クライアントをインストールするほかに、Web サーバーに Portfolio NetPublish Server もインストールする必要があります。NetPublish Server のインストーラを実行する前に、まずサーバー環境を正しくセットアップしておくください。

- Windows サーバー環境では、Microsoft IIS がインストールされていなければなりません。インストール手順については、Windows のマニュアルおよびヘルプファイルを確認してください。



上級ユーザー：標準以外（ポート 80 以外）の IIS ポートを指定する場合は、server.properties ファイルに新しいパラメータを追加する必要があります。テキストエディタを使って、パラメータ general.wwwPort=XXXX を追加します（ここで、XXXX は新しいポートです）。このファイルの場所は次のとおりです。C:\Program Files\Extensis\Portfolio NetPublishServer\Webroot\app

- Macintosh サーバー環境では、Apache 1.3 がインストールされていなければなりません。これは、デフォルトでオペレーティングシステムと一緒に自動的にインストールされているはずですが。

NetPublish Server をインストールするには

1. インストーラをダブルクリックして起動します。

2. インストーラの開始ステップからナビゲートします。エンドユーザーライセンスを受諾し、インストールする場所を選択します。



正しく機能させるには、Portfolio NetPublish をデフォルトの場所にインストールする必要があります。

3. Windows の場合は「完了」、Mac では「終了」をクリックします。

Macintosh サーバー環境で Web 共有を有効にするには

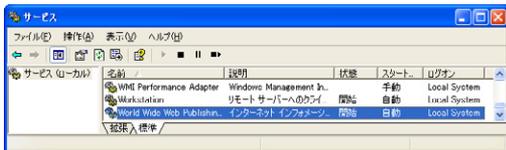
1. 「システム環境設定」パネルを開きます。
2. 「共有」を選択します。
3. リストから「パーソナル Web 共有」を選択して、「開始」をクリックします。
4. システム環境設定を終了します。

Macintosh サーバー環境で NetPublish ランチャを使用して NetPublish Server を起動するには

1. 「アプリケーション」、「Portfolio NetPublish Server」、「WebRoot」、「app」の順にナビゲートします。
2. NetPublish ランチャをダブルクリックして実行します。
3. 「開始」をクリックして、NetPublish Server を起動します。

IIS World Wide Web Publishing のサービスを再起動するには

1. 「スタート」、「コントロールパネル」、「管理ツール」、「サービス」の順にクリックし、Windows Services マネージャを開きます。



2. 「World Wide Web Publishing Service」をクリックして選択し、「アクション」、「再起動」の順にクリックします。

NetPublish Server は Windows のサービスとしてインストールされます。サーバーに、ネットワーク共有のサイト、テンプレート、ファイル、およびカタログへのアクセス権を付与する必要がある場合は、サービスのログオン権限を設定する必要があります。ログオン情報を入力しない場合、サーバーにはローカルファイルへのアクセス権のみが付与されます。

NetPublish サービスのログオン情報を入力するには

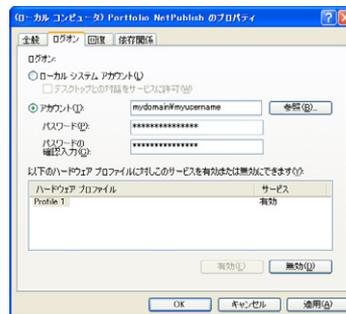
1. Windows Services マネージャを開きます。

「スタート」、「コントロールパネル」、「管理ツール」、「サービス」の順にクリックします。



2. リスト内の「Portfolio NetPublish」をダブルクリックしてプロパティのダイアログボックスを開きます。

3. 「ログオン」タブを選択し、「アカウント」オプションを選択します。



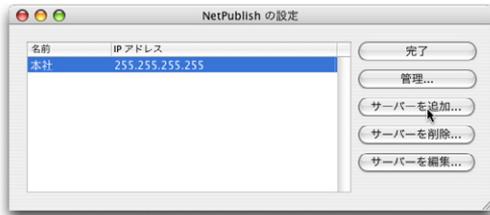
4. NetPublish Server のログオン情報を入力し、「OK」をクリックして設定を有効にします。必要に応じて、「参照」ボタンを使ってログオン情報を探します。
5. 「OK」をクリックして新しい設定を有効にします。
6. Windows Services マネージャで、Portfolio NetPublish サービスが選択されている状態で「アクション」、「再起動」の順にクリックしてサービスを再起動します。

NetPublish Server の追加

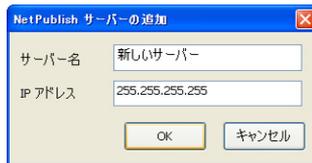
Portfolio クライアントから Web サイトを作成するには、事前にクライアントのサーバーリストに NetPublish Server を追加しておく必要があります。

NetPublish Server を追加するには

1. Portfolio クライアントを起動します。
2. 「ファイル」>「NetPublish の設定」を選択します。
3. 「NetPublish の設定」ダイアログボックスで、「サーバーの追加」をクリックします。



4. ダイアログボックスにサイト名と IP アドレスを入力します。
静かな IP アドレスに依存したくない場合は、IP アドレスのボックスに DNS 名または WINS 名を入力し、サーバー名には任意の名前を設定します。



5. 「OK」をクリックします。

NetPublish Server のシリアル番号の登録

NetPublish Server を完全に利用するには、シリアル番号を正しく入力する必要があります。シリアル番号はエンコードされて、ライセンスが付与されている同時接続数が NetPublish Server に自動的に伝えられます。

NetPublish Server にシリアル番号を登録するには

1. Portfolio クライアントを起動します。
2. NetPublish Server を追加します（すでに追加している場合は必要ありません）。
3. 「ファイル」>「NetPublish の設定」を選択します。
4. サーバーをクリックしてハイライトします。
5. 「設定（管理）」をクリックします。



6. 今回初めてサーバーを設定する場合は、NetPublish 管理者のパスワードの入力が求められます。セキュリティを確保するためには、ここでパスワードを設定して確認することを強くお勧めします。
7. 「サーバーの設定」タブで、「シリアル番号」をクリックします。

8. ダイアログボックスにシリアル番号を入力し、「OK」をクリックします。
9. 「OK」、「終了」の順にクリックします。

NetPublish Assistant

NetPublish Assistant (ネットパブリッシュ アシスタント) を利用すれば、サイトを作る HTML 知識がなくてもファイルを Web に公開することができます。

JavaScript に詳しいユーザーが、独自の NetPublish サイトを作成したい場合も、まず NetPublish Assistant を使ってサイトを作成することをお勧めします。サンプルサイトを公開したら、JavaScript と Portfolio 固有のコマンドが記載している「カスタムサイトの作成」の章を参照してください。

選択するサイトテンプレートに応じて、NetPublish Assistant では最大 8 つのステップが表示されます。各ステップに従うことで、公開するサイト内に最大 5 つの HTML ページを構成することができます。

カタログ選択 公開するカタログが表示されます。

開始 サイトの Web サイトレイアウトを選択します。

サイト ここで公開するすべての Web ページのグローバル設定を設定します。サイトに「トップページ」を追加することもできます。

検索 ここでは、検索ページの設定をします。

検索結果 ここでは、検索結果ページを設定します。

詳細情報 ここでは、詳細ページを設定します。通常、高解像度の画像が 1 つ、表示されます。

コレクション ここでは、ユーザーが複数のファイルを集め、1 つの圧縮ファイルでダウンロードすることができるコレクションページを設定します。

公開 公開する場所とカタログオプションを設定し、後で利用できるようにテンプレートの変更を保存します。

この章では、NetPublish Assistant を詳しく説明します。各ステップで利用できるオプションを学習することができます。

Portfolio クライアントにも「Web ページの作成」という別の Web ページ作成機能が用意されています。この機能は NetPublish よりもオプションの数が少なく、静的な Web ページしか作成できません。この機能の詳細については、『Portfolio ユーザーガイド』を参照してください。



NetPublish Assistant を使ってサイトを効果的に公開するには、まず NetPublish Server と、関連する Web サーバーのサポートアイテムをインストールする必要があります。詳細については、「Portfolio NetPublish の紹介」の章を参照してください。

NetPublish Assistant を起動するには

1. Portfolio クライアントを起動します。
2. 公開する Portfolio のカタログを開きます。
3. 「カタログ」>「NetPublish」を選択します。

—または—

現在のギャラリーのみを公開する場合は、「ギャラリー」>「NetPublish」を選択します。



どちらの NetPublish コマンドを選択しても NetPublish Assistant が起動されますが、「ギャラリー」>「NetPublish」コマンドを選択すると、公開ステップで現在のギャラリーが自動的に選択されます。なお、公開するギャラリーは、公開ステップでいつでも変更できます。

カタログ選択ステップ

カタログ選択ステップでは、公開対象として選択したカタログがリストされます。Assistant のステップを移動するには、「次へ」をクリックします。



開始ステップ

開始ステップでは、目的に合うテンプレートをを選択することができます。また、それらのテンプレートをコピーし、よりニーズに合わせてカスタマイズすることもできます。



NetPublish Assistant 開始ステップ

画面の上部には、NetPublish Assistant のプロセスが、現在、どここのステップにいるのかを表示します。各サイトアイコン（テンプレート）をクリックすると、そのテンプレートの詳細が表示されます。

各テンプレートには具体的なページレイアウトとデザイン情報が入っているので、NetPublish Assistant では、そのテンプレートで構成できるオプションとページのみが表示されます。



これ以降は、どのステップでも「公開」（パブリッシュ）をクリックすることができます。途中で「公開」をクリックすると、公開ステップまでの後続のすべてのステップでデフォルト設定が有効になります。

デザイン済みテンプレートを使用するには

1. 使用するテンプレートのアイコンをクリックします。
2. 「公開」をクリックして、すべてデフォルト設定を使ってサイトを公開します。
-または-
2. 「次へ」をクリックして、テンプレートのページをカスタマイズします。

テンプレートをカスタマイズした変更を保存する場合は、このステップでテンプレートをコピーし、公開ステップで「別名で保存」ボタンを使用するか、Assistant のステップごとに各 HTML ページを編集して保存します。

テンプレートの変更をデフォルトテンプレートに直接保存することはできません。これは、ユーザーが常にデフォルトテンプレートにアクセスできるようにしているからです。

テンプレートをコピーするには

1. テンプレートアイコンを右クリック (Win) または Ctrl キーを押しながらクリック (Mac) し、メニューから「複製 (コピー)」を選択します。
2. ダイアログボックスに新しいテンプレートの名前を入力し、「OK」をクリックします。
3. 新しいテンプレートアイコンをクリックしてハイライトします。
4. 「次へ」をクリックして、テンプレートのページのカスタマイズを始めます。



NetPublish Assistant の「公開」ページには、テンプレートのコピーを保存して、今後再利用できるようにするためのオプションが用意されています。

ほとんど使用しないテンプレートがある場合は、隠すことが可能です。テンプレートを隠しても削除されることはありませんので、いつでも再表示できます。

テンプレートを非表示にするには

- テンプレートアイコンを右クリック (Win) または Ctrl キーを押しながらクリック (Mac) し、メニューから「非表示」を選択します。

テンプレートを再表示するには

1. テンプレートアイコンを右クリック (Win) または Ctrl キーを押しながらクリック (Mac) します。
2. 「再表示」にカーソルを移動し、次のいずれかをクリックします。
 - 非表示のテンプレートをすべて表示する場合は「すべて」
-または-
 - 再表示するテンプレートの名前

基本設定ステップ

サイトステップでは、Web サイトに表示するロゴや、ヘッダー、フッター等のサイトオプションを指定します。これらは、公開したサイトのすべてのページに表示されるアイテムです。



NetPublish Assistant 基本設定ステップ

Web ページのタイトル

「Web ページのタイトル」ボックスには、サイトを表示したときにブラウザのタイトルバーに表示するテキストを入力します。

Web ページのロゴ

すべてのページに表示するカスタムのグラフィックを選択することができます。たとえば、社名のグラフィックや公開するカタログのグラフィックなどがあります。

JPG、GIF、PNG ファイルを使用することができます。最大サイズの推奨は 150 × 90 ピクセルです。

デフォルトのロゴグラフィックを変更する場合は、「参照」ボタンをクリックしてグラフィックの場所に移動します。



グラフィックファイル、HTML ファイルなどのファイルは、コンピュータのどこに置いても構いません。NetPublish Assistant は、サイトを公開するときにこれらをすべて自動的に収集します。

Web ページのヘッダーとフッター

「Web ページのヘッダーとフッター」にはテキストまたは HTML を入れることができます。これを使用して、現在使用中のその他のサイトと調和する外観を作成したり、Web サイトのナビゲーションボタンを取り入れたりすることができます。

ヘッダーおよびフッターの最大サイズの推奨は 600 × 90 ピクセルです。

ヘッダーまたはフッターファイルのパスを変更する場合は、「参照」ボタンをクリックします。選択した HTML またはテキストファイルが自動的にコピーされ、header.html または footer.html という名前に変更されます。



デフォルトテンプレートを使用している場合は、Assistant に特にリストされていない新しいヘッダー、フッター、またはサイトロゴを参照していない限り、ヘッダー、フッター、Web ページのロゴは常に組み込まれます。



HTML 形式は HTML ファイルでしか認識されません。プレーンテキストを追加すると、NetPublish はそれを既存の形式およびスタイルに組み込みます。



ヘッダーおよびフッターの HTML 背景色には、後続の NetPublish Assistant ページで選択するスタイルシートと一致するものを選択することをお勧めします。

トップページとリンク

トップページ

トップページは、サイトのユーザーに最初に表示するページです。サイトのナビゲート方法などの重要な情報を入れることができます。

トップページは、480 × 480 ピクセルの推奨される最大サイズまで拡大でき、テキストまたは HTML を入れることができます。



HTML 形式は HTML ファイルでしか認識されません。プレーンテキストを追加すると、NetPublish はそれを既存の形式およびスタイルに組み込みます。

ロゴ画像にリンクを設定

このオプションのチェックボックスをオンにすると、各ページにトップページへのリンクを設定することができます。ほとんどのテンプレートで、ロゴ画像がリンクボタンとして、テキストボックスに指定した URL に移動させることができます。

検索ステップ

このステップでは、ユーザーがカタログでアイテムを検索する方法を構成します。テンプレートごとに個別に作成された多くのページレイアウト、ページスタイル、および検索設定から選択することができます。検索ページは、それだけで 1 ページにするか、またはフレーム化サイトにするができます。



検索ページを含まないテンプレートを選択した場合、検索ページを編集するためのオプションは表示されません。



NetPublish Assistant 検索ステップ

ページスタイル

最初のステップは、検索ページスタイルの選択です。ページスタイルでは、ページの一般的な機能とページの「外観」が定義されます。ページスタイルは、どのアイテムをどこに表示するかを決めるものです。検索ページでは、ページスタイルを使って、クイック検索あるいは詳細検索のどちらの検索機能を組み込むかを決めることができます。

リストの先頭が、選択したテンプレートのデフォルトのページスタイルです。

ページスタイルを選択するには

- ページスタイルアイコンをクリックします。

ページスタイルアイコンをクリックすると、ページの「説明」に、そのページスタイルに関する詳細情報が表示されます。この情報を参考に、さまざまなスタイルの中から選択することができます。

スタイルシート

選択するサイトに応じて、複数のスタイルシート候補が表示されることがあります。スタイルシートでは、サイトのカラースキーム、つまり通常は最終的な検索ページ上の各アイテムの背景色とテキスト色を制御します。



ページスタイルとスタイルシートを選択したら、「Web ページをプレビューする」をクリックし、検索ページが他のページと調和するかどうかを確認することをお勧めします。



その他のカスタムのカスケーディングスタイルシートオプションがある場合、一貫性のある使用を維持するには、NetPublish サイト全体をエクスポートし、最終の HTML を編集してカスタムオプションを含めるようにします。

検索フィールドのオプション

検索ページに組み込むことができる検索には、クイック検索と詳細検索の2つのタイプがあります。

- クイック検索では、Portfolio のカタログ内でのファイル名、キーワード、および説明データのみに対して、検索が実施されます。これは基本的には Portfolio クライアントの「クイック検索」機能と同じです。
- 詳細検索では、Portfolio のカタログ内で最大 5 つの異なるフィールドに基づく検索が行われます。フィールドは指定した順に表示することができ、ブラウザ上には、選択した各フィールドのテキストを検索する機能を備えた HTML フォームとして表示されます。また、「すべてを満たすものを検索」または「いずれかのフィールドを満たすものを検索」のオプションも表示されます。これにより、「and」または「or」で複数の選択が結合されます。

詳細検索オプションを設定するには

- 「次のフィールドを使って検索を作成」チェックボックスをオンにします。
- リストで、検索対象とするデータフィールドをクリックしてチェックボックスをオンにします。



「パス」フィールドと「URL」フィールドは検索オプションとして使用できません。このことは、URL として定義したカスタムフィールドにも当てはまります。

- データ名をリスト内の希望する位置にドラッグして、データフィールドの順序を定義します。

たとえば、検索エンジンで最初に「ファイル名」フィールドをリストしたい場合は、リストの一番上にそのフィールドをドラッグします。

「すべて表示」ボタンを表示

このオプションのチェックボックスをオンにすると、Web ページに「すべて表示」リンクが挿入されます。ユーザーがこのボタンをクリックすると、NetPublish サイトで公開されているすべてのアイテムが結果ページに表示されます。

詳細検索オプション

検索エンジンが各データフィールドをどのように調べるか、さらにブラウザにデータフィールドをどのように表示するかをカスタマイズすることができます。

詳細検索オプションを構成するには

- 「検索フィールドの選択」リストで、選択したデータフィールドをクリックします。

2. 「検索条件 / オプション」 ボタンをクリックします。「詳細検索オプション」 ダイアログボックスが表示されます。

データ入力オプション

データ入力オプションは、選択したデータフィールドのタイプによって異なります。たとえば、「キーワード」フィールドの詳細オプションでは、次の項目を作成することができます。

- **テキスト入力ボックス** ここに、ユーザーはキーワードのテキストを入力することができます。
- **定義済みリストのプルダウン** カタログ内のマスターキーワードのリストを作成し、ユーザーがメニューからキーワードを選択できるようにします。
- **定義済みリストのハイパーリンク** プルダウンと同様、カタログ内のマスターキーワードのリストからハイパーリンクのリストを作成します。

検索条件

「テキスト入力ボックス」と「定義済みリストのプルダウン」のデータ入力オプションでは、検索エンジンが各データフィールドをどのように調べるかを再定義することができます。

1つの検索条件のチェックボックスのみをオンにした場合は、常にその検索条件が検索で使用されます。複数の検索条件のチェックボックスをオンにした場合は、検索条件のプルダウンリストが作成され、ユーザーはそこから選択することができます。

たとえば、「キーワード」フィールドのデータ入力オプションとして「定義済みリストのプルダウン」を選択したとします。「先頭」、「最後」、「一致」の3つの検索条件のチェックボックスをオンにします。作成される検索ページには、2つのプルダウンリストが表示されます。1つ目のプルダウンリストには3つの検索条件が入り、2つ目のプルダウンリストにはカタログのマスターキーワードが入ります。ユーザーは、各プルダウンリストから1つのアイテムを選択し、検索ボタンをクリックして検索を開始します。



検索条件として「を含む」が使用されている場合を除き、空の検索フィールドは無視されます。また、「パス」やカスタムの URL など、非インデックスフィールドは検索対象にできず、したがって検索フォームで使用することはできません。

ソースコードの編集

「ソースの表示」ボタンをクリックすると、テキストエディタに NetPublish Assistant のソースコードが表示されます。このツールを使えば、定義済みのソースコードの細かい調整も簡単に行えます。

エディタで行える操作は次のとおりです。

- ページの HTML を直接編集できます。
- デフォルトの Web ブラウザに変更内容をプレビューできます。
- ファイルに現在のソースをエクスポートできます。
- ファイルをインポートして現在のソースを置き換えることができます。
- 編集したソースファイルを保存 / 別名で保存することができます。



コピーサイトまたはカスタマイズ可能なサイトで作業している場合は、「保存」しか選択できません。また、テンプレートサイトを処理している場合は、「別名で保存」を選択して新しいファイルに保存する必要があります。



ソースファイルを編集しているときに機能を壊し、編集後のサイトが正しく動作しなくなることがあります。ソースファイルを編集する場合は注意してください。

ソースを表示する前に、ページ設定をすべて選択することをお勧めします。こうすることで NetPublish パラメータがよく分かり、ページ機能を壊す恐れが少なくなります。

NetPublish Assistant に戻るには、「閉じる」ボタンを選択します。

Web ページのプレビュー

NetPublish Assistant のどのステップでも、「Web ページのプレビュー」をクリックし、現在のページをデフォルトの Web ブラウザに表示することができます。最大でカタログ内の最初の 10 個の画像が現在のページに挿入されて、ブラウザに表示されます。



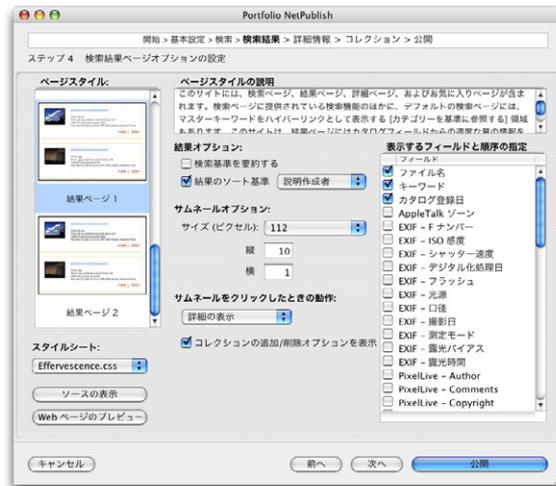
HTML コードと同様、表示される結果は、ビューアが使用しているブラウザのメーカーとバージョンによって異なります。ユーザーが使用と思われるすべてのブラウザで表示結果をテストすることをお勧めします。



デフォルトのテンプレートは、Internet Explorer、Netscape、および Safari でテストされています。既知のブラウザ制限は、サイトテンプレートの説明に記載されています。

検索結果ステップ

検索結果ステップでは、検索結果を表示するページを設定します。このステップで、ページのデザイン、サムネールのサイズ、表示するデータフィールド情報などを指定します。



NetPublish Assistant 検索結果ステップ

ページスタイル

検索ステップと同様、最初のステップは検索結果のページスタイルの選択です。ページスタイルとは、表示内容の設定済みレイアウトと設定のことです。他のページスタイルを選択するまで、デフォルトが選択されます。

ページスタイルを選択するには

- ページスタイルアイコンをクリックします。

スタイルシート

選択するサイトに応じて、複数のスタイルシート候補が表示されることがあります。スタイルシートでは、サイトのカラースキーム、つまり通常は最終ページ上の各アイテムの背景色とテキスト色を制御します。

検索結果とサムネールオプション

検索条件の表示

ユーザーが検索条件として入力した内容を表示する場合は、このオプションのチェックボックスをオンにします。これは、ユーザーにとって分かりやすい共通の言語（たとえば「犬を含んでいるキーワード」）で通常は表示されます。

検索結果のソートの基準

検索結果を具体的なデータフィールドを基準にソートする場合は、このオプションのチェックマークをオンにします。このオプションのチェックマークをオフにすると、アイテムはPortfolioのカatalog内の順序どおりに表示されます。

サムネールオプション

サムネールオプションでは、表示する画像のサイズ、縦と横のレイアウト、およびユーザーが画像をクリックした結果として発生するアクションを設定します。

サムネールサイズは、Catalog内のサムネールのサイズによります。「オリジナルサイズ」を選択した場合は、Catalogのサムネールサイズが使用されます。



Catalogのサムネールサイズよりも大きいサイズを選択すると、サムネールの画質が落ちる可能性があります。したがって、Catalogのサムネールが112ピクセルで作成されている場合は、112よりも大きいサムネールサイズを入力しないでください。新しい設定を選択した場合は、ページをプレビューして確認してください。

ドロップダウンメニューから、ユーザーがサムネールをクリックした結果として発生するアクションを選択します。「詳細の表示」を選択すると、「詳細情報」ページが表示されます。その他のオプションは、各表記のとおり動作をします。

コレクションへの追加 / 削除の表示オプション

ユーザーがファイルのコレクションを作成できるようにする場合は、このオプションのチェックボックスをオンにします。すると、結果ページ上の各アイテムの横にリンクまたはボタンが表示されます。ユーザーが各アイテムをクリックすると、そのアイテムがコレクションに追加されます。



このオプションが検索結果ページと詳細情報ページの両方で無効になっている場合は、ユーザーはアイテムをコレクションに追加できません。

表示フィールド

ページスタイルに応じて、さまざまなデータフィールドを表示することができます。ページに含める各データフィールドのチェックボックスをオンにし、それらを適切な表示順序にドラッグします。

詳細情報ステップ

詳細情報ページは、ただ1つの画像が、他のページに表示されるサムネールよりも大きいサイズ、高い画質で表示されるページです。このステップでは、詳細情報ページオプションを設定します。

ページスタイルとスタイルシート

ページスタイルとスタイルシートのオプションの働きは、これまでのページと同じです。ページスタイルをクリックし、ドロップダウンリストからスタイルシートを選択します。

画像オプション

表示する画像、その画像のサイズ、ユーザーが画像をクリックした結果として発生するアクションを選択することができます。



NetPublish Assistant 詳細情報ステップ

使用する画像

「使用する画像」とは、詳細ページで表示される画像を作成するときに使う、画像のことです。

- オリジナル画像を表示する場合は「オリジナル」を選択します。この設定を使用する場合は、下で述べている画像サイズを設定します。
- カタログのプレビュー画像を使用する場合は「プレビュー」を選択します。カタログにプレビュー画像が入っていない場合は、サムネール画像が使用されます。
- カタログのデフォルトサムネールを使用する場合は「サムネール」を選択します。

画像サイズ

オリジナル以外のサイズを設定すると、NetPublishによって画像のサイズ変更が行われ、画像は選択されたサイズで表示されます。



ファイルが Web ブラウザで表示できない形式 (TIF、PSD など) であるとき、画像サイズとソースの両方で「オリジナル」を選択すると、詳細な画像が表示されません。その場合は、具体的な数値で画像サイズを選択すると、NetPublish は Web ブラウザで表示できる形式で自動的に画像を作成します。

オリジナルのダウンロード

ユーザーがカタログからオリジナルのファイルをダウンロードすることができるリンクまたはボタンを追加するには、このオプションのチェックマークをオンにします。

コレクションへの追加 / 削除の表示オプション

ユーザーがファイルのコレクションにファイルを追加できるようにする場合は、このオプションのチェックボックスをオンにします。

表示フィールド

ページスタイルに応じて、さまざまなデータフィールドを表示することができます。ページに含める各データフィールドのチェックボックスをオンにし、それらを適切な表示順序にドラッグします。



「パス」などの、長く、折り返しのないテキストフィールドは、サイトのデザインに著しく影響することがあります。このようなタイプのフィールドの表示は避けることをお勧めします。

コレクションステップ

コレクションステップでは、コレクションページの構成を設定することができます。他のステップと同様、ページスタイル、スタイルシートを選択したり、ページのデザインや機能を変更したりすることができます。



NetPublish Assistant コレクションステップ

ページスタイルとスタイルシート

ページスタイルとスタイルシートのオプションの働きは、これまでのページと同じです。ページスタイルをクリックしてハイライトし、ドロップダウンリストからスタイルシートを選択します。

表示結果のソートの基準

検索結果を具体的なデータフィールドを基準にソートする場合は、このオプションのチェックマークをオンにします。このオプションのチェックマークをオフにすると、アイテムはPortfolioのカatalog内の順序どおりに表示されます。

サムネールオプション

サムネールオプションでは、表示する画像のサイズ、縦と横のレイアウト、およびユーザーが画像をクリックした結果として発生するアクションを設定します。

サムネールサイズは、Catalog内のサムネールのサイズに依存します。「オリジナルサイズ」を選択した場合は、Catalogのサムネールサイズが使用されます。

ダウンロードオプション

ユーザーがコレクションに追加したすべてのファイルを、1つの圧縮ファイルとして、ダウンロードできるようにするには、「**集めたアイテムのダウンロードを許可**」を選択します。圧縮ファイルは、必要に応じてNetPublishにより自動的に作成されます。

圧縮ファイルには、ZIPまたはSITを指定できます。ユーザーのオペレーティングシステムに応じて、圧縮ファイルを開くときに追加ソフトウェアが必要になる場合があります。Windowsでは、一般にWinZip®を使用してZIPファイルを開きます。Macintoshでは、StuffIt® (Allume Systems社)を使用してSITおよびZIPファイルを開くことができます。



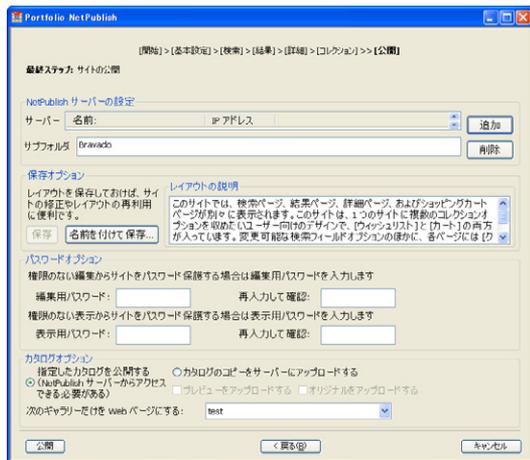
SITファイルを作成する場合、Macリソースフォークを含む画像は圧縮前にMacBinary形式に変換され、圧縮時にもリソースフォークが維持されます。

表示フィールド

これまでのステップと同様、ページに含める各データフィールドのチェックボックスをオンにし、それらの選択したデータフィールドを適切な表示順序にドラッグします。

サイトの公開ステップ

サイトの公開ステップでは、サイトを公開する場所を NetPublish に伝えます。ここで、テンプレートのコピーを保存し、パスワードを指定して、サーバーがアクセスできる場所にあるカタログを利用するかカタログの静的なコピーをアップロードするかを選択します。また、公開するギャラリーもここで指定します。



Web サーバーを追加するには

1. 「追加」をクリックします。
2. ダイアログボックスにサイト名と IP アドレスを入力します。静的な IP アドレスに依存したくない場合は、サイト名のテキストボックスに DNS 名または WINS 名を入力することができます。
3. 「OK」をクリックします。



サーバーを追加する場合は、まず NetPublish Server と、関連する Web サポートアイテムをインストールする必要があります。詳細については、「Portfolio NetPublish の紹介」の章を参照してください。

Web サーバーを選択するには

1. リストからサーバーの名前をクリックします。
2. サーバー上のサブフォルダにサイトを作成する場合は、「サブフォルダ」テキストボックスにフォルダ名を入力します。



サブフォルダ名は URL の一部に直接変換されるので、サブフォルダ名に拡張 ASCII 文字や日本語を使用することはできません。拡張文字には、Ã é ì ò ü Ø Æ などのアクセント付き文字が含まれます。

Web サーバーを削除するには

1. リスト内のサーバーをクリックします。
2. 「削除」をクリックします。
3. 確認ダイアログボックスで、「OK」をクリックします。

カタログオプション

カタログオプションでは、カタログから静的バージョン、あるいは動的バージョンのいずれかのファイルを公開することができます。

現在の場所からカタログを公開することを選択した場合は、実際には NetPublish が動的な Web サイトを作成していることとなります。カタログ（または公開しているギャラリー）を更新すると、更新した時点で NetPublish Web サイトのユーザーが、その変更内容にアクセスできるようになります。



現在の場所から公開する場合は、NetPublish Server がカタログにアクセスできることが必要です。カタログにアクセスできない場合、NetPublish Assistant は、カタログのコピーをサーバーにアップロードすることを求めるダイアログボックスを表示します。

カタログのコピーをサーバーにアップロードする場合は、実際には現在のカタログの静的なスナップショットを作成していることになります。サイトのユーザーには、カタログ（または指定したギャラリー）を公開した時点でのカタログに含まれていたものしか表示されません。また、オリジナル画像とプレビュー画像はサーバーにアップロードができます。



カタログをコピーして公開する場合には、カタログのサイズによって時間がかかることがあります。再び同じ場所にコピーを公開するときは、変更のあったファイルのみがサーバーにアップロードされます。このため、公開済みのカタログを更新するときは時間を節約できます。



公開プロセスをキャンセルしてもサーバーにコピー済みのファイルは削除されないため、オペレーティングシステムを介して手動で削除する必要があります。

次のギャラリーだけを Web ページにする

ここでは、カタログ内で公開するギャラリーを選択することができます。ギャラリーを選択すると、そのギャラリーに含まれているファイルだけが公開されます。通常のギャラリーとスマートギャラリーのどちらでも選択できますが、「最終カタログ登録日」や「検索結果」、一時ギャラリーは、NetPublish を使って公開することはできません。また、カタログ内のアイテムをすべて公開するには、「すべてのアイテム」を選択してください。

Web サイトで表示される情報の順番は NetPublish Assistant 内で指定します。選択したギャラリーでの順番は関係ありません。

保存オプション

現在の設定をすべて再利用したい場合は、現在のテンプレート設定のコピーを保存することができます。

現在の NetPublish 設定のコピーを保存するには

1. Assistant の「サイトの公開」ページで、「レイアウトの説明」テキストボックスに新しい説明を入力します。
2. 「別名で保存」ボタンをクリックします。
3. 新しいテンプレート設定に名前を付け、「OK」をクリックします。

現在の設定のコピーが保存されます。次に NetPublish を使用してサイトを作成するときは、保存したテンプレートが他の設定と共に開始ページに表示されます。

パスワードオプション

サイトには、編集用と表示用の 2 つの異なるタイプのパスワードを割り当てることができます。

編集用パスワードは、権限のないユーザーがサイトを編集できないようにします。つまり、第三者がパスワードなしで NetPublish Assistant を使用してサイトを再び公開することはできません。サイトを再び公開する場合は、編集用パスワードの入力が求められます。



既存のサイトを再び公開するたびに、サイトの編集用および表示用パスワードを再入力してパスワードを保存する必要があります。パスワードは、サイトを公開するたびに書き込まれます。

表示用パスワードでは、権限のないユーザーによる Web アクセスを防ぐことができます。Web サイトに公開したファイルを保護したい場合は、表示用パスワードを入力して確認します。Web ユーザーは、NetPublish サイトを表示する前にこのパスワードの入力が求められます。Web ユーザーが入力したパスワードが間違っていると、ブランクの HTML ページが表示されます。



複数のサイトで同じ表示用パスワードを使用している場合、サイトにアクセスするユーザーは、最初のサイトでのみパスワードの入力が求められます。ただし、ブラウザウィンドウを閉じた場合は、パスワードを再入力する必要があります。

NetPublish ファイルの場所

NetPublish を効果的に使用するには、NetPublish Server 上のファイルの場所を正確に把握しておくことが便利です。公開されたファイルが置かれている NetPublish Server 上のディレクトリは、次のとおりです。

Windows 環境のデフォルトインストールディレクトリ:

```
C:\Program Files\Extensis\
Portfolio NetPublish Server\WebRoot\
```

Macintosh 環境のデフォルトインストールディレクトリ:

```
//BootVolume/Applications/
Portfolio NetPublish Server/WebRoot
```

app%

このディレクトリには、EXE、DLL、NP、およびその他のソフトウェア関連のファイルを含む、NetPublish Server ソフトウェアのファイルが入ります。

app%server.properties

このファイルには、サーバーの通常の操作に影響する設定が入ります。

app%assets.log

このテキストファイルには、NetPublish Server からダウンロードされたコンテンツに関するログ（アセットログ）が入ります。

app%server.log

このテキストファイルには、サーバーからのメッセージのログが入ります。

app%data%jp%error.properties

このファイルには、エラー文字列のマッピングが入ります。

app%data%jp%field.properties

このファイルには、変換されたフィールド名が入ります。

cache%

このディレクトリには、ZIP ファイルおよび SIT ファイルの作成時に NetPublish Server によって生成された一時ファイルが入ります。

global%libraries%

このディレクトリには、.np テンプレートで使用されるグローバルなサーバーサイド JavaScript ファイルが入ります。グローバルアイテムは、そのアイテムを必要とするどのサイトテンプレートでも利用できます。

global%media%

このディレクトリには、グローバル画像が入ります。

global%resources%

このディレクトリには、カスケーディングスタイルシート（.css ファイル）など、グローバルなテキストファイルが入ります。

sites%site_name(ユーザー定義)%

このディレクトリには、NetPublish Assistant で作成されたサイト固有のファイルが入ります。これには、properties ファイルと、サーバーにアップロードされる静的なカタログが含まれます。

sites%site_name%alias.properties

このファイルには、この特定のサイトのカタログエイリアスを定義する設定が入ります。

sites%site_name%site.properties

このファイルには、このサイトに固有の設定が入ります。

sites%site_name%catalog.fdb

ここに、NetPublish Assistant はこの特定のサイトの静的なカタログをアップロードします。

sites%site_name%libraries%

このディレクトリには、.np テンプレートで使用されるサイト固有の JavaScript ファイルが入ります。

sites%site_name%media%

このディレクトリには、サイト固有の画像が入ります。

sites%site_name%previews%

カタログのコピーを NetPublish Assistant がアップロードする場合は、ここにプレビュー画像が置かれています。プレビュー画像のアップロードは任意です。

sites%site_name%resources%

このディレクトリには、カスケーディングスタイルシート (.css ファイル) など、サイト固有のテキストファイルが入ります。

sites%site_name%templates%

このディレクトリには、サーバーが使用するサイト固有の .np ファイルが入ります。

sites%site_name%originals%

カタログのコピーを NetPublish Assistant がアップロードする場合は、ここにオリジナル画像が置かれています。オリジナル画像のアップロードは任意です。

NetPublish の設定

NetPublish Server の管理

NetPublish Server は、「NetPublish の設定」ダイアログボックスから簡単に追加、編集、削除、管理することができます。このダイアログボックスを開くには、「ファイル」>「NetPublish の設定」を選択します。

NetPublish Server を管理する場合は、サーバーレベルのパスワードの変更、ログファイルの表示、サーバーのシリアル番号の登録、サーバーキャッシュサイズの設定、および公開済みサイトの管理を行うことができます。

NetPublish Server の追加とシリアル番号の登録の詳細については、「NetPublish の紹介」の章の「インストール」の節を参照してください。

NetPublish Server 名と IP アドレスを編集するには

1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「サーバーの編集」をクリックします。
3. 「サーバーの編集」ダイアログボックスで、サーバー名と IP アドレスを編集し、「OK」をクリックします。

静かな IP アドレスを使用したくない場合は、IP アドレスのボックスに DNS 名または WINS 名を入力し、サーバー名として任意の名前を指定します。

NetPublish Server を削除するには

1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックしてハイライトし、「サーバーの削除」をクリックします。
3. 「OK」をクリックして削除を確認します。

管理用パスワードとサイト作成用パスワード

サーバーレベルで制御できるパスワードは2つあります。管理用パスワードでは、「NetPublish の設定」ダイアログボックスへのアクセスを制御することで NetPublish Server の設定を保護します。サイト作成用パスワードでは、権限のないユーザーによる新しい NetPublish サイトの作成を防ぎます。

サーバーごとに、1つの管理用パスワードと1つのサイト作成用パスワードがあります。

各サイトに追加パスワードを具体的に割り当てることもできます。これらのパスワードのことを編集用パスワードと表示用パスワードと言います。これらのパスワードの詳細については、次ページの「NetPublish サイトの設定」を参照してください。

管理用またはサイト作成用パスワードを変更するには

1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「設定（管理）」をクリックします。
3. 「サーバーの設定」タブで、更新するパスワードに応じて、「NetPublish 管理者」または「サイト作成」のいずれかをクリックします。
4. 新しいパスワードを入力して確認し、「OK」をクリックします。
5. 「OK」、「終了」の順にクリックします。

キャッシュサイズ

サーバーは、サーバーのディスクドライブに特定のサイズを確保しています。サーバーのサイズを制限する場合は、キャッシュサイズを最小限に抑えることができます。キャッシュとは、Portfolio が、ZIP ファイルおよび SIT ファイルの作成や、プレビュー画像およびサムネール画像の動的な生成に使用するファイルを一時的に入れておく場所です。

キャッシュサイズを変更するには

1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「設定（管理）」をクリックします。



3. 「サーバーの設定」タブで、新しいキャッシュサイズをメガバイト単位で入力し、「OK」をクリックします。

ログファイル

NetPublish には、2 種類のログファイルを出力する機能があり、NetPublish Server の動作状況を解析する際に利用することができます。NetPublish Server からどのファイルがダウンロードされたかを出力するアセット（コンテンツ）ログと、サーバーに関するその他の詳しい情報を出力するサーバーログがあるので、頻繁にダウンロードするファイルを追跡する際に便利です。また、予想外のサーバーのアクティビティを診断する場合にも役立ちます。

サーバーログを表示するには

1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「設定（管理）」をクリックします。
3. 「サーバーの設定」タブで「ログを表示」をクリックします。

アセットログを表示するには

1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「ログレポート」をクリックします。
3. 「ログレポート」ダイアログボックスで、どのようにログファイルを作成するかを指定します。



「開始日時」と「終了日時」オプションを使用すると、ある期間のログだけを出力することができます。また特定のサイトに関する情報だけを出力する場合は、「対象にするサイト」オプションを使用してください。

4. 「サーバーのレポート」をクリックすると、現在のアセットログファイルが表示されます。

—または—

「以前のレポート」をクリックすると、すでに作成したアセットログを表示することができます。ログファイルの場所を指定して、「開く」をクリックしてください。

NetPublish サイトの設定

公開済みサイトのプロパティを細かく編集する場合は、「NetPublish の設定」ダイアログボックスを使用します。ここでは、具体的なサイトの有効化 / 無効化、エイリアスの使用によるサイトへの複数のカタログの追加、編集用パスワードと表示用パスワードの変更などが行えます。

公開済みサイトのアクティブ化 / 非アクティブ化

サーバーに複数のサイトを公開しておき、そのうちの一部のサイトのみをアクティブにして一般公開できるようにすることができます。

サイトをアクティブ化 / 非アクティブ化するには

1. 「ファイル」> 「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックしてハイライトし、「設定 (管理)」をクリックします。
3. 「サイトの管理」タブで、サイト名のチェックマークをオンまたはオフにします。
4. 「適用」をクリックします。チェックマークがオフのサイトは無効になり、Web ユーザーは利用できなくなります。
5. 「OK」、「終了」の順にクリックします。

編集用パスワードと表示用パスワード

編集用パスワードと表示用パスワードは、サイトレベルで設定されるパスワードです。編集用パスワードは、ユーザーが現在公開済みのサイトを再び公開できるかどうかを制御します。表示用パスワードは、Web を介してサイトにアクセスするすべてのユーザーが公開済みサイトを表示する前に入力しなければならないパスワードです。

編集用パスワードと表示用パスワードを変更するには

1. 「ファイル」> 「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「設定 (管理)」をクリックします。
3. 「サイトの管理」タブで、公開済みサイトををクリックし、「サイトの編集」をクリックします。

4. 「詳細情報」タブで、更新するパスワードのタイプに応じて、「編集用パスワードの変更」または「表示用パスワードの変更」をクリックします。
5. 新しいパスワードを入力して確認し、「OK」をクリックします。
6. 「OK」をクリックしてサイトの変更内容を有効にします。
7. 「OK」、「終了」の順にクリックします。

カタログエイリアス

エイリアスは、各 NetPublish サイトに、使用する Portfolio のカタログ、そのカタログの場所、カタログ内のどのギャラリーを公開するかを伝えるためのものです。各エイリアスで 1 つまたは複数のカタログやカタログ内のギャラリーを指定することができるので、エイリアスは便利です。これにより、1 つの NetPublish サイトが同時に複数のカタログやギャラリーにアクセスすることが可能になります。

デフォルトでは、NetPublish サイトは「catalog」というエイリアスを作成します。NetPublish サイトで複数のギャラリーまたはカタログを使用するには、エイリアスを編集して、そのエイリアスにカタログを追加します。異なるギャラリーを追加する限り、エイリアスに同じカタログを複数回追加することができます。

また、複数のエイリアスを利用するカスタムサイトを作成することもできます。これにより、サイトの領域ごとに異なるカタログエイリアスを使用することができます。なお複数のエイリアスを使用する場合は、サイトのソースコードを編集する必要があるため、JavaScript API の詳しい知識が要求されます。



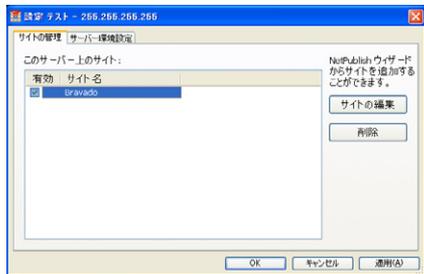
エイリアスを編集する場合は、事前に NetPublish サイトを公開しておく必要があります。



新しいカタログをエイリアスに追加する場合は、検索するすべてのカタログにカスタムフィールドが存在することを確認することが大切です。エイリアス内のすべてのカタログに存在しないフィールドを検索または表示しようとすると、Web サイトは壊れることになります。

カタログまたはギャラリーをエイリアスに追加するには

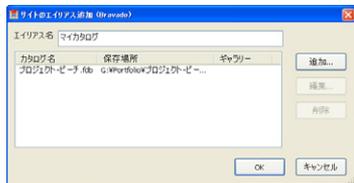
1. 「ファイル」>「NetPublish の設定」を選択します。
2. 「NetPublish の設定」ダイアログボックスで、サーバーをクリックし、「設定（管理）」をクリックします。
3. 「サイトの管理」タブで、公開済みサイトをクリックし、「サイトの編集」をクリックします。



4. 「エイリアス」タブで、現在のエイリアスをクリックし、「編集」をクリックします。

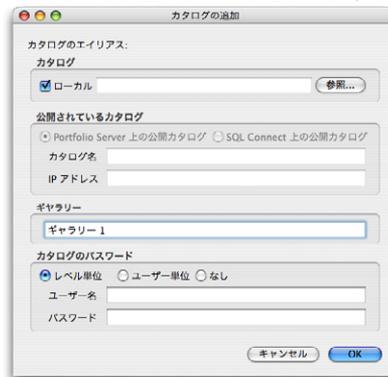


5. 「サイトのエイリアスの編集」ダイアログボックスで、「追加」をクリックします。



6. 「カタログ追加」ダイアログボックスで、次のことを行います。

- NetPublish Server が直接アクセスできるカタログでは、「ローカル」オプションのチェックマークをオンにし、「参照」ボタンをクリックしてカタログを探します。
- Portfolio Server から提供されるカタログでは、「Portfolio Server 上の公開カタログ」または「SQL Connect 上の公開カタログ」のチェックマークをオンにして、適切な IP アドレスとカタログ名を入力します。Mac 環境では、提供されているカタログを選択する前に「ローカル」オプションのチェックマークをオフにする必要があります。



7. 公開するギャラリーの名前を入力します。カタログで使用されているギャラリー名と同じ名前を入力してください。カタログ全体を公開するには、「ギャラリー」フィールドを空白のままにしておきます。

8. 新しいカタログがカタログレベルでパスワード保護されている場合：

- 適切なパスワードレベル（「ユーザー単位」または「レベル単位」）のチェックマークをオンにします。
- 適合するユーザー名とパスワードを入力します。

9. 「OK」をクリックします。新しいカタログが「サイトのエイリアスの編集」ダイアログボックス内のカタログリストに追加されます。

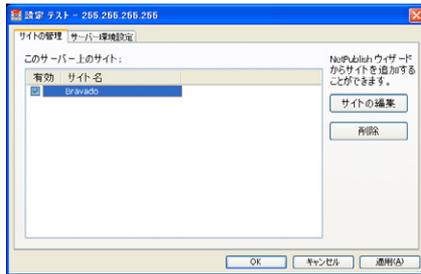
10. 「OK」を 3 回クリックしてから「終了」をクリックします。

新しいエイリアスを作成するには

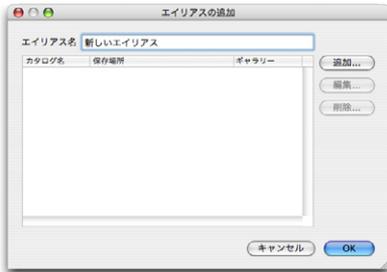


複数のエイリアスを利用する場合は、サイトのソースコードを編集する必要がありますので、JavaScript API の詳しい知識が要求されます。

- 1 「ファイル」>「NetPublish の設定」を選択します。
- 2 「NetPublish の設定」ダイアログボックスで、サーバーをクリックしてハイライトし、「設定（管理）」をクリックします。
- 3 「サイトの管理」タブで、公開済みサイトをクリックし、「サイトの編集」をクリックします。

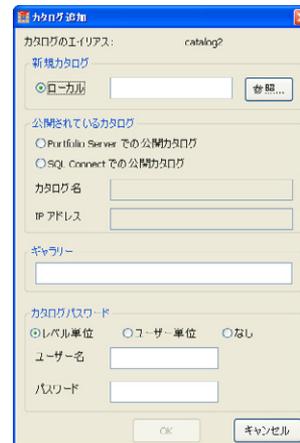


- 4 「エイリアス」タブで、「追加」をクリックします。
- 5 「サイトのエイリアス追加」ダイアログボックスで、新しいエイリアスの名前を入力し、「追加」をクリックします。



- 6 「カタログ追加」ダイアログボックスで、次のことを行います。

- NetPublish Server が直接アクセスできるカタログでは、「ローカル」オプションのチェックマークをオンにし、「参照」ボタンをクリックしてカタログを探します。
- Portfolio Server から提供されるカタログでは、「Portfolio Server 上の公開カタログ」または「SQL Connect 上の公開カタログ」のチェックマークをオンにして、適切な IP アドレスとカタログ名を入力します。



7. 公開するギャラリーの名前を入力します。カタログで使用しているギャラリー名と同じ名前を入力してください。

カタログ全体を公開するには、「ギャラリー」フィールドを空白のままにしておきます。

8. 追加したカタログがカタログレベルでパスワード保護されている場合は、以下の手順も行ってください。

- 適切なパスワードレベル（「ユーザー単位」または「レベル単位」）のチェックマークをオンにします。
- ユーザー名とパスワードを入力します。

9. 「OK」をクリックします。新しいカタログエイリアスがエイリアスリストに追加されます。

- 10.この新しいエイリアスを利用するには、「カスタムサイトの作成」の章に記載されている JavaScript API に従って、サイトのソースコードを編集します。

カスタムサイトの作成

この章では、JavaScript API と NetPublish の固有のコマンドについて詳しく説明します。この章を学習すれば、あらかじめ用意してあるテンプレートを細かく調べたり、NetPublish で使用する独自の新しいテンプレートを作成したりできるようになります。



Extensis 統合およびコンサルティングサービスチームは、NetPublish サイトの作成を支援します。詳細については、Extensis 法人営業部までお問い合わせください。また Extensis サポートは、カスタム NetPublish サイトの変更についてはサポートしていません。

テンプレートの変更

NetPublish テンプレートの構造と機能を学習する一番簡単な方法は、NetPublish Assistant が定義済みのサイトを作成する方法を調べることです。

NetPublish のサイトは .NP ファイルとして作成されます。これらのファイルには、各サイトの HTML コードとサーバーサイドの JavaScript コードがすべて含まれます。

公開された NetPublish ファイルは NetPublish Server に置かれます。Windows 環境での NetPublish Server の通常のインストールされたパスは次のとおりです。

```
C:\Program Files\Extensis  
Portfolio NetPublish Server\
```

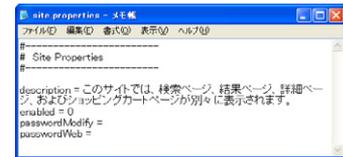
このディレクトリから、修正するファイルを `WebRoot\sites\sitename\templates` ディレクトリに置きます。これらの .NP ファイルはどのエディタでも開いて編集することができます。ファイルの場所とプロパティの詳細については、本書のこれまでの章を参照してください。

次の2つのファイルを理解することが大切です。これらのファイルは、修正したテンプレートを NetPublish Assistant で表示できるかどうか、および NetPublish がエイリアスをどのように処理するかに影響します。

site.properties

サイトのルートフォルダ (`WebRoot\sites\sitename`) にある、`site.properties` ファイル内の設定を書き留めておくことが大切です。このファイルはどのエディタでも開くことができます。

このファイルには、サイトの説明、変更用と公開用のパスワード、およびこのサイトを NetPublish Server から提供して Web 上で利用できるかどうかを制御する `enabled` の設定が入っています。



`enabled=1` の場合サイトはアクセス可能であるのに対し、`enabled=0` の場合はサイトはサーバーから提供されません。

alias.properties

`alias.properties` ファイルもサイトフォルダ内にあり、このテンプレートがアクセスできるカタログとギャラリーを制御します。エイリアスを使用する場合は、この情報が「NetPublish Server の設定」ダイアログ内の情報と一致している必要があります。



JavaScript API

グローバル関数

グローバル関数は次のとおりです。

convertHtml (html)

HTML ブロックを動的に構文解析して評価します。通常は、processRecordSet 関数と一緒にレコードブロックを処理する場合に使用します。

library (filename)

JavaScript 関数の指定ライブラリを現在の実行コンテキストにロードします。ほぼすべてのテンプレートが、次に説明する JavaScript オブジェクトにアクセスする場合にライブラリ ([global.js]) を使用します。

processRecordSet (message, html)

これは、テンプレートの RecordSet 内のすべてのテーブル行および列を処理するヘルパー関数です。message は、レコードが見つからない場合に表示されるテキストです。html には、Page.setRowsCols() で設定された行および列ごとの出力である html の <td></td> ブロックが入ります。

html 引き数は一般に、特殊な拡張引き数テンプレート構文を使って指定します。

例:

```
<!--Start of item block-->  
<% processRecordSet( No records  
found!<BR>', %>
```



閉じ括弧「」がありません。したがって閉じ括弧「」のところまですべて引き数となります。

```
<td>  
<%= RecordSet.record.get( 'Filename')%>  
</td>  
<% ); %>
```



html 引き数の終わりを表します。

```
<!--End of item block-->
```

RecordSet

RecordSet オブジェクトもグローバルです。

RecordSet.totalItems

「find」からのアイテムの総数を表す数。

RecordSet.offset

レコードセット内の現在の開始オフセットを表す数 (URL の offset=x と同じ値)。このプロパティも設定することができます。

RecordSet.record

レコードセット内の現在のアイテムに対応する、子 Record オブジェクト。



RecordSet.record オブジェクトにアクセスする場合は、事前に RecordSet.itemNext() または RecordSet.itemAt() のいずれかを呼び出す必要があります。

RecordSet.getUrl (offset)

オフセットが割り当てられた URL を生成します (Page オブジェクトによって内部で使用されます)。

RecordSet.isEmpty()

レコードが見つからない場合に true を返します。

RecordSet.hasMore()

レコードセット内にレコードが残っている場合に true を返します。RecordSet.itemNext() と一緒に呼び出す必要があります。

RecordSet.itemNext()

結果セット内の次のレコードと一緒に RecordSet.record をロードします。

RecordSet.itemPrevious()

結果セット内の直前のレコードと一緒に RecordSet.record をロードします。RecordSet の先頭に達している場合は false を返します。それ以外の場合は true を返します。

RecordSet.itemAt(offset)

結果セット内の指定されたオフセット位置にあるレコードと一緒に RecordSet.record をロードします。

Record

Record オブジェクトは、RecordSet (RecordSet.record) の子オブジェクトとしてのみ使用することができます。Record オブジェクトは、Collection.getItemRecord() から取得することもできます。

Record.itemID

アイテム ID を、単一の数字(単一カタログエイリアスの場合)または .1.132 などの一連の数字 (複数カタログエイリアスの場合) のいずれかからなる文字列として返します。

Record.original

このレコードのオリジナル画像への HTTP リンクを返します。

Record.preview

このレコードのプレビュー画像への HTTP リンクを返します。

Record.thumbnail

このレコードのサムネール画像への HTTP リンクを返します。

Record.thumbnailWidth

サムネールの幅 (ピクセル単位) を返します。

Record.thumbnailHeight

サムネールの高さ (ピクセル単位) を返します。

Record.get(fieldname, [offset], [escapestyle])

指定フィールドに対するネイティブの JavaScript の文字列型、数値型、または日付型オブジェクトを返します。フィールドに複数の値が入っている場合は、フィールド内の各値に対する値の配列を (前述と同じ型で) 返します。offset を指定すれば、複数値フィールド内の具体的な値にアクセスすることができます。フィールドが存在しない場合は、ヌルの型を返します。

escapestyle には、次のいずれかを指定できます。

0 - エスケープなし - デフォルトです。

1 - 二重引用符で囲まれた HTML にテキストが入っている場合に役立ちます。HTML は、アンパサンド、二重引用符、および不等号括弧 (&;, ";, <;, >) をエンコードします。制御文字はスペースに置き換えます。これは、エスケープスタイルを指定している場合および複数値フィールド内のすべての値が必要な場合に使用します。

2 - 二重引用符で囲まれた JavaScript にテキストが入っている場合に役立ちます。JavaScript は、二重引用符と不等号括弧をエンコードします。バックスラッシュで、単一引用符と不等号括弧をエスケープします。制御文字をスペースに置き換えます。

Record.getType(name)

指定フィールドの FieldType オブジェクトを返します。

Record.getTypeAll([type])

すべてのフィールドの FieldType オブジェクトの配列を返します。type には、次のいずれかの値を指定できます。

- default カスタム以外のフィールドのみ
- custom カスタムフィールドのみ
- string すべての文字列型フィールド
- date すべての日付型フィールド
- number すべての数字型フィールド
- decimal すべての 10 進型フィールド
- url すべての url 型フィールド

レコードの親エイリアスに対し複数のカタログがある場合、返されるフィールドはすべてのカタログに共通するフィールドです。

Record.getDetailLink(template)

現在のレコードに対する指定テンプレート名 (つまり detail.np) を使って詳細ページへの HTTP リンクを返します。詳細テンプレートは、RecordSet 内にレコードが 1 つしかないものと見なします。

例:

```
<%= RecordSet.record.getDetailLink
( detail.np' );%>
```

Record.getCollectionPostForm(command, template, [name], format)

指定コマンド (add, show, remove, removeAll, archive) の後の操作に対する適正な <form> タグを返します。カタログレコードの表示前に、結果セットのページの一番上でのみ使用されます。

呼び出し側は、独自の HTML を作成し、各アイテム、および後の操作を開始するボタンに対するフォーム <input> タグを指定する必要があります。

template には、コレクションの表示に使用するテンプレート名を指定します。name を使用して名前付きコレクションを指定することができます。format は、オプションのパラメータで、ZIP または SIT になります。

Record.getCollectionLink(command, template, [name], format)

コレクションコマンド add, show, remove, removeAll, archive のいずれかを実行する場合に href リンク内で使用できるリンクを返します。

template には、コレクションの表示に使用するテンプレート名を指定します。name を使用して名前付きコレクションを指定することができます。format は、オプションのパラメータで、ZIP または SIT になります。

FieldType

FieldType オブジェクトは、Record.getType()、Record.getTypeAll()、Catalog.getTypeAll() によって作成されます。

FieldType.type

string、date、number、decimal、url、undefined のいずれかの値が入ります。

FieldType.name

フィールドの名前が入ります。

FieldType.length

フィールドの最大長が入ります。

FieldType.custom

フィールドがカスタムフィールドである場合は true、それ以外の場合は false が入ります。

FieldType.preDefinedList

このフィールドが定義済みリストである場合は true、それ以外の場合は false が入ります。

FieldType.multivalued

複数值フィールドである場合は true、それ以外の場合は false が入ります。

FieldType.getPreDefinedList()

このフィールドが定義済みの値である場合は（基本型を使って）定義済みの値からなる配列を返します。それ以外の場合は、0 の要素からなる配列が返されます。

CatalogSet

CatalogSet オブジェクトもグローバルで、NetPublish で定義されているカタログエイリアスをカプセル化します。

CatalogSet.get(alias, [catalogOffset])

指定エイリアス名に対する Catalog オブジェクトを返します。複数のカタログを持つエイリアスの場合は、catalogOffset を指定する必要があります（先頭は 1 です）。

例: CatalogSet.get('homes', 1);

カタログエイリアスが見つからない場合は、エラーが表示されます。

CatalogSet.getAliases()

システムで定義されているすべてのエイリアスの名前が入っている文字列の配列を返します。

CatalogSet.getMasterKeywords()

CatalogSet 内にあるすべてのカタログのすべてのマスターキーワードを返します。

CatalogSet.getCatalogCount(alias)

指定されたエイリアスに対して定義されているカタログの数を返します。

Catalog

Catalog オブジェクトは、`CatalogSet.get()` によって作成されます。Catalog オブジェクトは、検索テンプレートと検索結果テンプレート内で使用することができます。フィールド値、フィールドに有効な検索演算子、複数値フィールドに有効な値など、カタログのスキーマ情報を調べる場合に使用します。

Catalog.dateCreated

カタログが作成された日付 (Date オブジェクト) を返します。

Catalog.dateModified

カタログが最後に変更された日付 (Date オブジェクト) を返します。

Catalog.diskPreview

カタログにプレビュー画像が入っているかどうかに応じて true または false を返します。

Catalog.diskPreviewMaxSize

カタログにプレビュー画像が入っている (diskPreview が true) の場合は、プレビューの最大サイズ (ピクセル単位) を返します。それ以外の場合は、ヌルを返します。

Catalog.diskPreviewPath

diskPreview が true の場合は、プレビュー画像フォルダのパスからなる文字列を返します。それ以外の場合は、ヌルを返します。

Catalog.fullPath

カタログの絶対パス (または URL) からなる文字列が入ります。例: `C:%data%houses.fdb`

Catalog.thumbnailSize

カタログ内のサムネールのデフォルトサイズを返します。有効な値は、112 (112 × 112 ピクセルのサムネール) と 256 (256 × 256 ピクセルのサムネール) の 2 つです。

Catalog.totalItems

カタログ内のアイテムの総数を返します。

Catalog.totalKeywords

カタログ内のキーワードの総数を返します。

Catalog.getType(name)

指定フィールドの FieldType オブジェクトを返します。

Catalog.getTypeAll(type)

すべてのフィールドの FieldType オブジェクトの配列を返します。type には、次のいずれかの値を指定できます。

- default カスタム以外のフィールドのみ
- custom カスタムフィールドのみ
- string すべての文字列型フィールド
- date すべての日付型フィールド
- number すべての数字型フィールド
- decimal すべての 10 進型フィールド
- url すべての url 型フィールド

このカタログの親エイリアスに対し複数のカタログがある場合、返されるフィールドはすべてのカタログに共通するフィールドです。

Catalog.getMapping(name)

IPTCopyright など、指定タイトルにマップするフィールドの名前を返します。マップするタイトルが存在しない、またはこのタイトルにマップするフィールドが見つからない場合は、ヌルを返します。

Catalog.getMappingAll()

マップされたすべてのタイトルの文字列の配列を返します (定義されている場合)。定義されていない場合は、0 の要素からなる配列を返します。

Catalog.getMasterKeywordsAll()

マスターキーワードリストの文字列の配列を返します (定義されている場合)。定義されていない場合は、0 の要素からなる配列を返します。

CollectionSet

CollectionSet オブジェクトもグローバルで、NetPublish で定義されている名前付きコレクションをカプセル化しません。

CollectionSet.get([alias], [namedSet])

現在のユーザー、サイト、およびカタログエイリアスの Collection オブジェクトを返します。必要に応じて、具体的なエイリアスまたはコレクションセット名のいずれかを指定することができます。namedSet を指定しないと、グローバルな定義済みのコレクションが返されます。

コレクションレコードが存在しない場合は、空のレコードが作成されます。

クライアントの Cookie を使わずにコレクションにアクセスするには、テンプレートページに「セッション」の GET または POST の引き数が入っていることを確認してください。



Base コマンドを使用するページでは、エイリアスを指定する必要があります。



この関数は、特殊な Cookie 値を設定してコレクションの有効期限切れを防ぎます。Cookie は応答ヘッダーとして実装されるので、この関数は最初の Response.write() の出力後に呼び出すことはできません。呼び出すと、HTML 形式で「応答ヘッダーはすべて、HTML の前のテンプレートの一番上で作成する必要があります」というエラーメッセージが作成されます。

この関数はテンプレートの一番上で呼び出し、その結果を、テンプレート内のその他の場所で参照されるグローバルな JavaScript 変数に代入する必要があります。

例:

```
<% Cost = CollectionSet.get().getRecord  
(RecordSet.record.itemID).get(" Cost");%>
```

CollectionSet.getNamedSetAll([alias])

この関数は、コレクション内で、現在のユーザー、サイト、およびカタログエイリアスに対する名前付きセットが定義されている場合に、その名前付きセットの配列を返します。必要に応じて、具体的なエイリアスを指定することができます。

クライアントの Cookie を使わずにコレクションにアクセスする場合は、テンプレートページに「セッション」の GET あるいは POST の引き数が含まれていなければなりません。



Base コマンドを使用するページでは、エイリアスを指定する必要があります。



この関数は、配列に、特殊なグローバルセット名「_GLOBAL_」を必ず取り込みます。



この関数は、特殊な Cookie 値を設定してコレクションの有効期限切れを防ぎます。Cookie は応答ヘッダーとして実装されるので、この関数は最初の `Response.write()` の出力後に呼び出すことはできません。呼び出すと、HTML 形式で「応答ヘッダーはすべて、HTML の前のテンプレートの一番上で作成する必要があります」というエラーメッセージが作成されます。

この関数はテンプレートの一番上で呼び出し、その結果を、テンプレート内のその他の場所で参照されるグローバルな JavaScript 変数に代入する必要があります。

例:

```
<% Response.write(CollectionSet.getNamedSetAll()[0]); %>
```

CollectionSet.addNamedSet(name, [alias])

現在のユーザー、サイト、およびカタログエイリアスの新しい空の名前（「name」）付きセットを追加します。必要に応じて、具体的なエイリアスを指定することができます。

クライアントの Cookie を使わずにコレクションにアクセスする場合は、テンプレートページに「セッション」の GET 引き数または POST 引き数が含まれていなければなりません。



この関数は、配列に、特殊なグローバルセット名「_GLOBAL_」を必ず取り込みます。



この関数は、特殊な Cookie 値を設定してコレクションの有効期限切れを防ぎます。Cookie は応答ヘッダーとして実装されるので、この関数は最初の `Response.write()` の出力後に呼び出すことはできません。呼び出すと、HTML 形式で「応答ヘッダーはすべて、HTML の前のテンプレートの一番上で作成する必要があります」というエラーメッセージが作成されます。

この関数はテンプレートの一番上で呼び出し、その結果を、テンプレート内のその他の場所で参照されるグローバルな JavaScript 変数に代入する必要があります。

例:

```
<% CollectionSet.addSetName( otherSet' );
```

CollectionSet.removeNamedSet(name, [alias])

現在のユーザー、サイト、およびカタログエイリアスに対するコレクションセット名（「name」）およびそのセットのすべての内容を削除します。必要に応じて、具体的なエイリアスを指定することができます。

クライアントの Cookie を使わずにコレクションにアクセスする場合は、テンプレートページに「セッション」の GET あるいは POST の引き数が含まれていなければなりません。



Base コマンドを使用するページでは、エイリアスを指定する必要があります。



この関数は、配列に、特殊なグローバルセット名「_GLOBAL_」を必ず取り込みます。



この関数は、特殊な Cookie 値を設定してコレクションの有効期限切れを防ぎます。Cookie は応答ヘッダーとして実装されるので、この関数は最初の Response.write() の出力後に呼び出すことはできません。呼び出すと、HTML 形式で「応答ヘッダーはすべて、HTML の前のテンプレートの一番上で作成する必要があります」というエラーメッセージが作成されます。

この関数はテンプレートの一番上で呼び出し、その結果を、テンプレート内のその他の場所で参照されるグローバルな JavaScript 変数に代入する必要があります。

例:

```
<% CollectionSet.removeSetName( otherSet' );
%>
```

Collection

Collection オブジェクトは CollectionSet.get() のみによって作成されます。NetPublish の組み込みのコレクションデータベースから（現在のユーザー、サイト、カタログ、および必要に応じてセット名で定義されている）具体的なコレクションをカプセル化します。

Collection.add(itemID)

この関数は、コレクションにレコードの itemID を追加します。

Collection.getItemIDAll()

コレクションに対するすべてのレコードの itemID の配列を返します。

Collection.remove(itemID, [removeAll])

この関数は、コレクションからレコードの itemID を削除します。RemoveAll はオプションのブール値です。true の場合、コレクションからすべてのアイテムが削除されます。

Collection.getRecord(itemId)

指定された itemId に対する Record オブジェクトを返します。

Request

Request オブジェクトはグローバルで、http のリクエスト変数へのアクセスを許可します。これにより、ユーザーは url に指定されているすべての値（つまり、カタログとテンプレート）と、検索で POST フォームを介して入力されたすべてのフィールドとその値を調べることができます。また、クライアントの Cookie にもアクセスできます。

Request.post

この要求が post 要求の場合は true、get 要求の場合は false を返します。

Request.getCookie(name)

指定されたクライアント Cookie の値を返します。Cookie が見つからない場合は空の文字列を返します。

Request.getHeader(name)

指定された http ヘッダー名の値、つまり「ユーザーエージェント」を返します。Cookie が見つからない場合は空の文字列を返します。

Request.getHeaderAll()

この関数は、現在の要求に対するすべての http ヘッダー名の配列を返します。

Request.getParameter(parameter)

この関数は、指定された get または post パラメータの値を返します。パラメータが見つからない場合は空の文字列を返します。

Request.getParameterAll()

get または post 要求に対するすべてのパラメータ名の文字列配列を返します。

Request.getQueryString()

url 上の「?」の右側にあるものをすべて返します。見つからない場合はヌルを返します。

Request.getRequestURL()

url 全体を返します。見つからない場合はヌルを返します。

Response

Response オブジェクトはグローバルで、これを使って http の応答変数を設定することができます。XML の回答で、カスタムの制御コードとクライアント Cookie を設定する場合に役立ちます。

Response.setContentType()

この関数は、戻される Content-Type ヘッダー（テキストや XML など）を取得します。見つからない場合は空の文字列を返します。

Response.getStatusCode()

http レスポンスステータスのコード列を取得します。見つからない場合は空の文字列を返します。

Response.logMessage(priority, message)

この関数は、指定されたレベル (priority) のメッセージを NetPublish Server エラーログファイルに記録します。使用できるレベルは fatal、error、info、debug です。

Response.setCookie(name, value, [expires], [path], [domain], [secure])

この関数は、有効期限、パス、ドメイン、およびセキュアの設定をオプションとして、名前と値を持つ Cookie を設定または作成します。「expires」引数は、GMT 形式にする必要がある文字列です。これは、Date.toGMTString() を使って実行することができます。



Cookie ヘッダーは必ず付加され、置き換えられることはありません。

例:

```
var today = new Date();
var expires = new Date(today.getTime() +
28 * 24 * 60 * 60 * 1000); // plus 28 days
Response.setCookie( 'name', 'value',
expires.toGMTString(), ' / ', '.cnn.com',
false);
```



最初の `Response.write()` 呼び出しですべての応答ヘッダーが作成されます。`Response.setCookie()` は、HTML の前で、テンプレートの一番上の最初の `Response.write()` 呼び出しの前に呼び出します。

Response.setContentType (type)

この関数は、戻される Content-Type ヘッダーを設定します。デフォルトは `text/html` です。



最初の `Response.write()` 呼び出しで、すべての応答ヘッダーが作成されます。`Response.write()` は、HTML の前で、テンプレートの一番上の最初の `Response.setContentType()` 呼び出しの前に呼び出します。

Response.setHeader (name, value, [append])

指定された http の応答ヘッダーに値を設定します。`append` が `true` の場合は (デフォルトは `false`)、同じ名前のヘッダーがすでに存在する場合でも新しいヘッダーが追加されます。



最初の `Response.write()` 呼び出しですべての応答ヘッダーが作成されます。`Response.setHeader()` は、HTML の前で、テンプレートの一番上の最初の `Response.write()` 呼び出しの前に呼び出します。

Response.sendRedirect (url)

Location 応答ヘッダーを送り、それによってブラウザを指定 url に接続させます。



最初の `Response.write()` 呼び出しで、すべての応答ヘッダーが作成されます。`Response.sendRedirect()` は、HTML の前で、テンプレートの一番上の最初の `Response.write()` 呼び出しの前に呼び出します。

例:

```
if (RecordSet.isEmpty())
    Error.redirect( 'errors/norecords.html' )
```

Response.setStatusCode (code)

http 回答状況コードにコード文字列を設定します。デフォルトは「200 OK」です。その他のコードについては、次を参照してください。

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html>



最初の `Response.write()` 呼び出しで、すべての応答ヘッダーが作成されます。`Response.setStatusCode()` は、HTML の前で、テンプレートの一番上の最初の `Response.write()` 呼び出しの前に呼び出します。

Response.showErrorHtml (filename, message)

この関数は、テンプレートの内容の代わりに指定されたエラーメッセージ html ファイルを表示し、テンプレートの実行をただちに停止します。

`filename` に、カスタム html ファイルを指定します。現在のサイトのリソースフォルダでカスタムファイルを探してから、グローバル / リソースフォルダを検索します。

`message` には、html ファイル内にある `MESSAGE_DESCRIPTION` と置き換える文字列を指定します。

Response.write(string)

文字列データをクライアントのブラウザに送り返します。これは、html テンプレートが巨大な場合に内部で呼び出されます。

Page

Page オブジェクトはグローバルで、global.np ファイルに入っています。これらの関数を使用する場合は、事前に totalRows と totalCols のグローバル変数をテンプレートの上で設定する必要があります。

Page.getCurrent()

現在のページ番号を返します。

Page.getList(total)

ページ番号のリストをハイパーリンク付きで返します。total には、ページ数を指定します。

Page.getPrevious(text)

この関数は、該当する場合に説明文を使って直前のページへのリンクを出力します。

Page.getNext(text)

この関数は、該当する場合に説明文を使って次のページへのリンクを出力します。

Page.getSpecified(page, text)

この関数は、該当する場合に説明文を使って指定ページへのリンクを出力します。

Page.getTotal()

この関数は、ページの総数を返します。

Page.isFirst()

この関数は、このページが最初のページかどうかを調べます。

Page.isLast()

この関数は、このページが最後のページかどうかを調べます。

Page.setRowsCols(rows, cols)

Page オブジェクトの行と列の数を設定します。Page オブジェクトを使用する場合は事前にこの関数を呼び出す必要があります。

Path

Path オブジェクトはグローバルで、global.np ファイルに入っています。ユーザーによるパス変換を支援する関数のみが含まれます。

Path.getOS(path)

指定されたパスを調べ、ソースとなっているオペレーティングシステムを判別します。mac、win、unix のいずれかを返します。

Path.getParts(path)

指定されたパスの部分文字列の配列を返します。

たとえば、C:\%Images%\Pool%\Swimmers%\image1.jpg というパスの場合、次の配列が返されます。

```
Result[0] = ' C: '
```

```
Result[1] = ' Images '
```

```
Result[2] = ' Pool '
```

```
Result[3] = ' Swimmers '
```

```
Result[4] = ' image1.jpg '
```

Units

Units オブジェクトはグローバルで、`global.np` ファイルに入っています。サイズと寸法の変換を処理する関数が含まれます。

Units.convertDate(value, format)

指定された形式の文字列が提供された場合に、日付を文字列に変換します。

形式文字列に有効な値は、次のとおりです。

`yyyy` は 4 桁の年号です (2002 など)。

`yy` は 2 桁の年号です (02 など)。

`month` は月の完全な名前です (September など)。

`mmm` は月の数字です (9 など)。

`mon` は、月の名前の最初の 3 文字です (Sep など)。

`hh` は時間です (3 など)。

`mm` は分 (必ず 2 桁) です (05 など)。

`ss` は秒 (必ず 2 桁) です (08 など)。

`ddd` は、曜日の最初の 3 文字です (Wed など)。

`dd` は月の日付です (25 など)。

`day` は曜日の完全な名前です (Wednesday など)。

`timezone` は、GMT の時間単位のタイムゾーン (GMT+5 など) です。

`time24` は 24 時間表記の時間です (18:24 など)。

`time` は AM/PM 表記の時間です (6:24PM など)。

たとえば、`Friday, March28, 03 4:12pm` という書式で日付文字列を設定するには、次の関数を使用します。

```
<%= Units.convertDate(RecordSet.record.  
get(' Created'), " day, monthdd, yyyytime") %>
```

Units.convertLength (value, to, dpi)

この関数は、形式 (`to`) として `inches`、`cm`、`mm`、`points`、`picas` のいずれかを使って、指定されたピクセル長 (`value`) を変換します。1 インチあたりのドット数 (`dpi`) が必要です。

例:

レコードの幅をインチ単位で取得する場合:

```
<%= Units.convertLength(RecordSet.record.  
get(' Width'),  
' inches',  
RecordSet.record.get(' Horizontal  
Resolution')) %>
```

Units.convertMoney(value, currency, [comma], [decimal], [negativeParens])

この関数は、指定された値を通貨文字列に変換します。通貨は、通貨記号（「\$」など）を含む文字列です。

comma が true（デフォルト）の場合、結果はコンマ付きで書式設定されます。decimal には、小数点の右側に表示する桁数を指定します。negativeParens が true の場合、負の値は括弧で囲まれて返されます。それ以外の場合は、先行のマイナス（-）文字のみが使用されます。

Cost フィールド（カスタム通貨フィールド）を英国ポンドでフォーマットする場合：

```
<%= Units.convertMoney(RecordSet.record.  
get( 'Cost'), unescape( '%A3'), true, 0, true)  
%>
```

Units.convertSize(value, to, [comma], [decimal])

この関数は、形式 (to) として bytes、kbytes、mbytes のいずれかを使って、指定されたサイズ (value) を変換します。

comma が true（デフォルト）の場合、数字は、適切な場所にコンマが挿入された文字列としてフォーマットされます。decimal には、小数点の右側に表示する桁数を指定します。

たとえば、ファイルのサイズをメガバイト単位で取得する場合は、次のようになります。

```
<%= Units.convertSize(RecordSet.record.  
get( 'File Size') * 1024, " mbytes", true) %>
```

Mozilla JavaScript オブジェクト

次に、Mozilla JavaScript API で定義され、NetPublish で使用することができる 6 つのオブジェクトを示します。詳細については、次のサイトを参照してください。

<http://www.mozilla.org>

Array

Array オブジェクトは、Mozilla JavaScript の実装で定義されています。組み込みの配列処理と、ソートおよびリバース機能を提供します。

Date

Date オブジェクトは、Mozilla JavaScript の実装で定義されています。

Math

Math オブジェクトは、Mozilla JavaScript の実装で定義されています。

String

String オブジェクトは、Mozilla JavaScript の実装で定義されており、サブストリング、検索、小文字、大文字、および文字区切り文字をベースとする配列への分割を作成することができます。

RegExp

RegExp オブジェクトは、Mozilla JavaScript の実装で定義されています。このオブジェクトは、正規表現の構文解析を処理するので、必要な場合は非常に強力です。

File

File オブジェクトは、Mozilla JavaScript の実装で定義されています。open、remove、copy、rename、read、write、list、mkdir などのローカルファイル操作を処理します。

NetPublish のコマンド

NetPublish のコマンドでは、関数の検索、ユーザーのコレクションからのアイテムの追加と削除、ZIP や SIT アーカイブファイルの作成の開始など、さまざまな操作を実行することができます。これらのコマンドは、html コード内で `get` アクションと `post` アクションのフォームで使います。

これらの NetPublish コマンドの働きは、以前の PortWeb 製品の同一のコマンドと同じです。前に作成した PortWeb テンプレートを NetPublish で使用できるようにするには、適切にコマンドを調整する必要があります。

Base コマンド

`base` コマンドは、JavaScript `RecordSet` オブジェクトを使用する必要はないが他の API は必要である検索テンプレート（およびその他のテンプレート）に使用します。`base` コマンドを使用するテンプレートで `RecordSet` オブジェクトを利用することはできません。

`base` コマンドは通常、カタログのスキーマ情報にアクセスしたり、検索フォームでフィールド名を指定したりする場合に使用します。

base コマンドのパラメータ

`base` コマンドの名前です。

`site` サイト名です。

`template` ページを生成するときに使うスクリプトテンプレート（拡張子は `.np`）です。

フォームが `Get` の例（文字列のクエリー）：

```
<A HREF="/netpub/server.np?base&site=sales&template=search.np">...</A>
```

フォームが `Post` の例：

```
<FORM ACTION="/netpub/server.np?base"
METHOD=POST >
<INPUT NAME="base" TYPE="HIDDEN" VALUE="">
<INPUT NAME="site" TYPE="HIDDEN" VALUE="
sales">
<INPUT NAME="template" TYPE="HIDDEN" VALUE="
search.np">
<INPUT TYPE="submit">
</FORM>
```

QuickFind コマンド

quickfind コマンドは、基本キーワード検索用の簡易化された検索メカニズムです。指定されたテキストを使って「先頭キーワード」検索を実行します。これは Portfolio クライアントのクイック検索機能と同じです。

QuickFind コマンドのパラメータ

quickfind コマンドの名前ですが、検索対象のキーワードを渡すためのパラメータでもあります。

site リソースを使用するサイト。

catalog 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

template ページを生成するときに使うスクリプトテンプレート (拡張子は .np) です。

sorton (オプション) レコードセットをソートするフィールドを識別します。ソートフィールドは単一値のインデックスフィールドでなければなりません。指定されない場合、レコードセットの順序は Find から返された順序になります。



ソートは複数カタログエイリアスではサポートされていません。

ascending (オプション) レコードセットをソートする順序を指定します。値 1 は昇順、値 0 は降順を示します。sorton パラメータが指定されていない場合、このパラメータは無視されます。sorton パラメータが指定され、このパラメータが指定されていない場合、デフォルトのソート順序は 1 (昇順) です。



ソートは複数カタログエイリアスではサポートされていません。

offset (オプション) コレクションからすでに表示されたレコードの数を示す正の整数。このパラメータが指定されない場合、表示はコレクションの最初のレコードから開始されます。

フォームが get の例 (文字列のクエリー) :

```
<A HREF="/netpub/server.np?quickfind&site=sales&catalog=fall&template=results.np&sorton=Filename&ascending=1&offset=10">...</A>
```

フォームが post の例 :

```
<FORM ACTION="/netpub/server.np?quickfind" METHOD=POST>
<INPUT NAME="quickfind" TYPE="Text"VALUE="">
<INPUT NAME="site" TYPE="HIDDEN"VALUE="sales">
<INPUT NAME="catalog" TYPE="HIDDEN"VALUE="fall">
<INPUT NAME="template" TYPE="HIDDEN"VALUE="results.np">
<INPUT NAME="sorton" TYPE="HIDDEN"VALUE="Filename">
<INPUT NAME="ascending" TYPE="HIDDEN"VALUE="1">
<INPUT NAME="offset" TYPE="HIDDEN"VALUE="10">
<INPUT TYPE="submit">
</FORM>
```

Find コマンド

find コマンドでは、特定の Portfolio のカタログで検索が実行され、その結果が特定のレイアウトで返されます。find コマンドでは次の値を渡すことができます。

Find コマンドの値

find コマンドの名前です。

site リソースを使用するサイト。

catalog 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

template ページを生成するときに使うスクリプトテンプレート（拡張子は .np）です。

sorton（オプション）レコードセットをソートするフィールド識別します。ソートフィールドは単一値のインデックスフィールドでなければなりません。指定されない場合、レコードセットの順序は Find から返された順序になります。



ソートは複数カタログエイリアスではサポートされていません。

ascending（オプション）レコードセットをソートする順序を指定します。値 1 は昇順、値 0 は降順を示します。sorton パラメータが指定されていない場合、このパラメータは無視されます。sorton パラメータが指定され、このパラメータが指定されていない場合、デフォルトのソート順序は 1（昇順）です。



ソートは複数カタログエイリアスではサポートされていません。

offset（オプション）コレクションからすでに表示されたレコードの数を示す正の整数。このパラメータが指定されない場合、表示はコレクションの最初のレコードから開始されます。

defaultjoin（オプション）このパラメータは、join の引き数がフィールドごとに指定されない場合に、使用します。これにより、ユーザーはすべてのフィールドに対しグローバルとなる join パラメータの値を指定することができます。ここには、and または or のいずれかの値を入れることができます。

find コマンドでは次のパラメータを使用して、NetPublish による検索の実行方法を定義することができます。これらは、必要に応じてグループ化して繰り返すことができます。find の各句は、Field、Operator、Value の 3 つの変数で定義されます。後続の句は join 変数（and または or）で前の句と結合する必要があります。最大 10 個の句を結合して複数条件の検索を作成することができます。

field 指定データベース内のインデックスフィールド。非インデックスフィールドが指定された場合、find は異常終了します。

op 有効な演算子は次のとおりです。

- **Matches / Does Not Match**
Date フィールドを除くすべてのフィールドで動作します。
- **Equals / Does Not Equal**
Date フィールドでのみ動作します。
- **Greater Than / Greater Than または Equal To**
Number フィールドでのみ動作します。
- **Less Than / Less Than または Equal To**
Number フィールドでのみ動作します。
- **Starts With / Does Not Start With**
String フィールドでのみ動作します。

value 検索対象の値。

join (オプション) find に複数の句がある場合のみ使用します。これには、「and」または「or」のいずれかの値を入れることができます。このパラメータが指定されない場合は、「and」が使用されます。最大 10 個の句を結合して複数条件の検索を作成することができます。

NetPublish では Portfolio クライアントでサポートされているすべての演算子を使用できますが、各フィールドですべての演算子が動作するわけではありません。選択したフィールドで演算子が動作しない場合、検索句はスキップされます。具体的なフィールドで演算子がサポートされているかどうか分からない場合は、Portfolio クライアントの検索機能を使って確認することができます。Portfolio クライアントの検索ダイアログでフィールド名を選択し、演算子のプルダウンを使ってそのフィールド名でサポートされている演算子を確認します。

find は、次の条件に基づきプラグインによって検証されます。

- 各句に 3 つの変数すべて (field、operator、value) が存在しなければならない。
- 最初の句以降のすべての句で、join 変数も (「and」または「or」のいずれかの値で) 指定されていなければならない。
- 各 field 変数と operator 変数は前述の規則を満たしていなければならない。

特定の句でこれらの検証のいずれかが失敗すると、その句はスキップされます。find 内のすべての句で検証が失敗すると、プラグインは、「一致するレコードが見つかりませんでした」ページを返します。検証が成功した場合は、残りのパラメータが評価されます。

find には、最大 10 個の句を入れることができます。

次の例では、2 つの句からなる find コマンドを示します (分かりやすいようにコメントを入れています)。各種パラメータは、サイトのニーズに応じて、非表示または表示することができます。この例では、ユーザーは最初の句の値フィールドには入力しますが、他の値はすべて非表示になっています。

```
<FORM ACTION="/netpub/server.np?find" METHOD=
POST>
<INPUT NAME="site" TYPE="HIDDEN" VALUE="
sales">
<INPUT NAME="catalog" TYPE="HIDDEN" VALUE="
fall">
<INPUT NAME="template" TYPE="HIDDEN" VALUE="
results.np">
<!--This is the first clause,and prompts the
user for a keyword-->
<INPUT NAME="field" TYPE="HIDDEN" VALUE="
Keywords">
<INPUT NAME="op" TYPE="HIDDEN" VALUE="
matches">
<INPUT NAME="value" TYPE="TEXT" VALUE="">
<!--This is a second clause (with the
required join) that is entirely hidden.-->
<INPUT NAME="join" TYPE="HIDDEN" VALUE="and">
<INPUT NAME="field" TYPE="HIDDEN" VALUE="
ExtensionWin">
<INPUT NAME="op" TYPE="HIDDEN" VALUE="
matches">
<INPUT NAME="value" TYPE="HIDDEN" VALUE="jpg"
>
<INPUTTYPE="Submit">
</FORM>
```

Add コマンド

add コマンドは、1つ以上のレコードをユーザーのコレクションセットに追加し、指定されたテンプレートを使ってコレクションページを表示します。特定のカタログのユーザーに対するコレクションセットが存在しない場合は、新しいコレクションセットが作成され、Cookie がユーザーに返されます。

Add コマンドのパラメータ

add コマンドの名前です。

site 使用するリソースのソースであるサイト。

catalog 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

name (オプション) ユーザーが複数の異なるコレクションを作成してアクセスできるようにしたい場合に指定します。name の値は希望するどのような文字列でも構いません。

template ページを生成するときに使うスクリプトテンプレート (拡張子は .np) です。

itemid カatalog内アイテム ID を指定します。アイテム ID はカatalog内で一意です。これは、1つのレコードでも、レコードのリストでも構いません (たとえば、...&itemid=70&itemid=75&itemid=85&...)。無効なアイテム ID は無視されます。

sorton (オプション) レコードセットをソートするフィールドを識別します。ソートフィールドは単一の値のインデックスフィールドでなければなりません。指定されない場合、レコードセットの順序は Find から返された順序になります。



ソートは複数カタログエイリアスではサポートされていません。

ascending (オプション) レコードセットをソートする順序を指定します。値 1 は昇順、値 0 は降順を示します。sorton パラメータが指定されていない場合は、このパラメータは無視されます。sorton パラメータが指定され、このパラメータが指定されていない場合、デフォルトのソート順序は 1 (昇順) です。



ソートは複数カタログエイリアスではサポートされていません。

offset (オプション) コレクションからすでに表示されたレコードの数を示す正の整数。このパラメータが指定されない場合、表示はコレクションの最初のレコードから開始されます。

session (オプション) 設定している httpCookie を使わずにコレクションのサポートを実装する場合に使用する一意の整数。一意の値を選択し、それをパラメータとしてすべてのコレクションコマンドに渡すことはユーザーが行います。

Show コマンド

show コマンドは、指定されたテンプレートを使ってユーザーのコレクションセット内のレコードを表示します。

Show コマンドのパラメータ

show コマンドの名前です。

site 使用するリソースのソースであるサイト。

catalog 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

name (オプション) ユーザーが複数の異なるコレクションを作成してアクセスできるようにしたい場合に指定します。name の値は希望するどのような文字列でも構いません。

`template` ページを生成するときに使うスクリプトテンプレート (拡張子は `.np`) です。

`sorton` (オプション) レコードセットをソートするフィールドを識別します。ソートフィールドは単一の値のインデックスフィールドでなければなりません。指定されない場合、レコードセットの順序は `Find` から返された順序になります。



ソートは複数カタログエイリアスではサポートされていません。

`ascending` (オプション) レコードセットをソートする順序を指定します。値 `1` は昇順、値 `0` は降順を示します。`sorton` パラメータが指定されていない場合は、このパラメータは無視されます。`sorton` パラメータが指定され、このパラメータが指定されていない場合、デフォルトのソート順序は `1` (昇順) です。



ソートは複数カタログエイリアスではサポートされていません。

`offset` (オプション) コレクションからすでに表示されたレコードの数を示す正の整数。このパラメータが指定されない場合、表示はコレクションの最初のレコードから開始されます。

`session` (オプション) 設定している `httpCookie` を使わずにコレクションのサポートを実装する場合に使用する一意の整数。一意の値を選択し、それをパラメータとしてすべてのコレクションコマンドに渡すことはユーザーが行います。

Remove コマンド

`remove` コマンドは、1つ以上のレコードをユーザーのコレクションセットから削除し、指定されたテンプレートを使ってコレクションページを表示します。Cookie が存在しない場合、ユーザーの Cookie 内の ID がコレクションデータベース内のいずれのレコードとも一致しない場合、または Cookie 内の ID が要求で指定されたカタログと一致しない場合は、空のレコードセットが返されます。

Remove コマンドのパラメータ

`remove` コマンドの名前です。

`site` 使用するリソースのソースであるサイト。

`catalog` 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

`name` (オプション) ユーザーが複数の異なるコレクションを作成してアクセスできるようにしたい場合に指定します。`name` の値は希望するどのような文字列でも構いません。

`template` ページを生成するときに使うスクリプトテンプレート (拡張子は `.np`) です。

`itemid` カタログ内のアイテム ID を指定します。アイテム ID はカタログ内で一意です。これは、1つのレコードでも、レコードのリストでも構いません (たとえば、`...&itemid=70&itemid=75&itemid=85&...`)。無効なアイテム ID は無視されます。

`all` (オプション) これが指定された場合は、コレクション内のすべてのレコードが削除されます。このオプションを使用する場合、`Itemid` を指定する必要はありません。

`sorton` (オプション) レコードセットのソート基準とするフィールドを識別します。ソートフィールドは単一のインデックスフィールドでなければなりません。指定されない場合、レコードセットの順序は `Find` から返された順序になります。



ソートは複数カタログエイリアスではサポートされていません。

ascending (オプション) レコードセットをソートする順序を指定します。値 1 は昇順、値 0 は降順を示します。sorton パラメータが指定されていない場合は、このパラメータは無視されます。sorton パラメータが指定され、このパラメータが指定されていない場合、デフォルトのソート順序は 1 (昇順) です。



ソートは複数カタログエイリアスではサポートされていません。

offset (オプション) コレクションからすでに表示されたレコードの数を示す正の整数。このパラメータが指定されない場合、表示はコレクションの最初のレコードから開始されます。

session (オプション) 設定している httpCookie を使わずにコレクションのサポートを実装する場合に使用する一意の整数。一意の値を選択し、それをパラメータとしてすべてのコレクションコマンドに渡すことはユーザーが行います。

Archive コマンド

archive コマンドは、指定されたコレクションセット内のレコードに対するすべてのオリジナル画像を ZIP または SIT アーカイブに (圧縮なしで) 入れ、クライアントのマシンへのこのアーカイブファイルのダウンロードを開始します。コレクションにレコードがない場合は、空のアーカイブファイルが返されます。

archive コマンドの名前です。

site 使用するリソースのソースであるサイト。

catalog 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

name (オプション) ユーザーが複数の異なるコレクションを作成してアクセスできるようにしたい場合に指定します。name の値は希望するどのような文字列でも構いません。

filename (オプション) アーカイブに使用する具体的なファイル名 (拡張子を含む)。指定されない場合、ファイル名はデフォルトで archive.zip または archive.sit に設定されます。

format zip または sit のいずれかを指定します。sit を指定した場合、リソースフォークを含むオリジナル画像はアーカイブへの追加時に MacBinary 形式に変換されます。

session (オプション) 設定している httpCookie を使わずにコレクションのサポートを実装する場合に使用する一意の整数。一意の値を選択し、それをパラメータとしてすべてのコレクションコマンドに渡すことはユーザーが行います。

Thumbnail コマンド

thumbnail コマンドは、指定されたアイテムのサムネール画像をブラウザに戻します。

Thumbnail コマンドのパラメータ

`thumbnail = itemID` コマンドの名前です。このパラメータは、返されるレコードのサムネールの `itemID` も取ります。

`site` 使用するリソースのソースであるサイト。

`catalog` 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは **Portfolio Server** から提供されているカタログのいずれかにマップされます。

`download` (オプション) これが指定された場合は、ブラウザに画像を表示せずに「別名で保存」ダイアログボックスを表示するように画像の応答ヘッダーが変更されません。

`filename` (オプション) `download` が指定された場合に使用する具体的なファイル名を指定します。

`height` (オプション) これが指定された場合、画像は指定された高さに拡大/縮小されてからブラウザに転送されます。この値が -1 の場合、拡大/縮小は発生しません。

`width` (オプション) これが指定された場合、画像は指定された幅に拡大/縮小されてからブラウザに転送されます。この値が -1 の場合、拡大/縮小は発生しません。

`aspect` (オプション) これが指定された場合、画像は、画像の縦横比に合わせて調整された、高さと同幅に拡大/縮小されます。



画像をすべての NetPublish サイトでより見やすく表示するには、`aspect` オプションを使用して、最大の画像寸法を最大の幅または高さに指定します。これにより、縦向きおよび横向きのオリジナル画像がサイズ変更されて、同様のサイズ領域に表示されます。たとえば、次のコードは、縦向きの画像の高さを 600 ピクセルに設定します。

```
...&width=200&height=600&aspect
```

```
...&width=600&height=200&aspect
```

`format` (オプション) 生成された画像のファイル形式。
有効な値は、`jpg`、`png`、`gif` です。

Preview コマンド

preview コマンドは、指定されたアイテムのプレビュー画像をブラウザに戻します。プレビュー画像が存在しない場合は、サムネール画像が戻されます。

Preview コマンドのパラメータ

`preview = itemID` コマンドの名前です。このパラメータは、返されるレコードのプレビュー画像の `itemID` も取ります。

`site` 使用するリソースのソースであるサイト。

`catalog` 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは **Portfolio Server** から提供されているカタログのいずれかにマップされます。

`download` (オプション) これが指定された場合は、ブラウザに画像を表示せずに「別名で保存」ダイアログを表示するように画像の応答ヘッダーが変更されます。

`filename` (オプション) `download` が指定された場合に使用する具体的なファイル名を指定します。

`height` (オプション) これが指定された場合、画像は指定された高さに拡大/縮小されてからブラウザに転送されます。この値が -1 の場合、拡大/縮小は発生しません。

`width` (オプション) これが指定された場合、画像は、指定された幅に拡大/縮小されてからブラウザに転送されます。この値が -1 の場合、拡大/縮小は発生しません。

`aspect` (オプション) これが指定された場合、画像は、画像の縦横比に合わせて調整された、高さと同幅に拡大/縮小されます。



画像をすべての NetPublish サイトでより見やすく表示するには、`aspect` オプションを使用して、最大の画像寸法を最大の幅または高さに指定します。これにより、縦向きおよび横向きのオリジナル画像がサイズ変更されて、同様のサイズ領域に表示されます。たとえば、次のコードは、縦向きの画像の高さを 600 ピクセルに設定します。

```
...&width=200&height=600&aspect
```

```
...&width=600&height=200&aspect
```

`format` (オプション) 生成された画像のファイル形式。
有効な値は、`jpg`、`png`、`gif` です。

Original コマンド

`original` コマンドは、指定されたアイテムのオリジナル画像をブラウザに戻します。オリジナル画像が存在しない場合は、サムネールが戻されます。

Original コマンドのパラメータ

`original = itemID` コマンドの名前です。このパラメータは、返されるレコードのオリジナル画像の `itemID` も取ります。

`site` 使用するリソースのソースであるサイト。

`catalog` 使用するカタログのエイリアスを指定します。エイリアスは、設定ファイルを介してディスク上のカタログまたは Portfolio Server から提供されているカタログのいずれかにマップされます。

`download` (オプション) これが指定された場合は、ブラウザに画像を表示せずに「別名で保存」ダイアログを表示するように画像の応答ヘッダーが変更されます。

`filename` (オプション) `download` が指定された場合に使用する具体的なファイル名を指定します。

`macbinary` (オプション) このパラメータが指定され、オリジナル画像にリソースフォークが含まれ、`download` が指定されている場合、ファイルは MacBinary としてエンコードされます。

`height` (オプション) これが指定された場合、画像は指定された高さに拡大 / 縮小されてからブラウザに転送されます。この値が `-1` の場合、拡大 / 縮小は発生しません。

`width` (オプション) これが指定された場合、画像は指定された幅に拡大 / 縮小されてからブラウザに転送されます。この値が `-1` の場合、拡大 / 縮小は発生しません。

`aspect` (オプション) これが指定された場合、画像は、画像の縦横比に合わせて調整された、高さと幅に拡大 / 縮小されます。



画像をすべての NetPublish サイトでより見やすく表示するには、`aspect` オプションを使用して、最大の画像寸法を最大の幅または高さに指定します。これにより、縦向きおよび横向きのオリジナル画像がサイズ変更されて、同様のサイズ領域に表示されます。たとえば、次のコードは、縦向きの画像の高さを 600 ピクセルに設定します。

```
...&width=200&height=600&aspect
```

```
...&width=600&height=200&aspect
```

`format` (オプション) 生成された画像のファイル形式。
有効な値は、`jpg`、`png`、`gif` です。

NetPublish のカスケーディングスタイルシート (CSS)

デザイン済みの NetPublish テンプレートは、色の使用の点で、スタイルシート依存サイトとスタイルシート独立サイトの2つのカテゴリに分かれます。

スタイルシート依存サイトは、そのサイト専用で作成された1つの固有のスタイルシートのみを使用してデザインされます。NetPublish Assistant では、1つのスタイルシートオプションしか表示されません。

スタイルシート非依存サイトは、色の使用に関してもっと柔軟にデザインされており、共通する共有のスタイルシートの1つと組み合わせることができます。これらの共通スタイルシートには、「red_theme.css」や「blue_theme.css」のような分かりやすい名前が付けられています。スタイルシート非依存サイトで使用できる新しいスタイルシートを作成することもできます。

カスタムスタイルシートの作成

カスタムスタイルシートを作成し、それを NetPublish Assistant 内のすべてのスタイルシート非依存サイトで利用できるようにすることができます。

カスタムスタイルシートを作成するには

1. master.css ファイルを探します。このファイルは、変更したり、新しいファイルに保存できる読み取り専用テンプレートです。

Windows:

```
C:\¥Documents and Settings¥All Users¥Application Data¥Extensis¥Portfolio¥NetPublish¥Data¥style sheets¥master¥
```

Mac では、このファイルはいろいろなパッケージの中に入っています。Portfolio アプリケーションパッケージには NetPublish プラグインが入っており、この NetPublish プラグインパッケージに master.css ファイルが入っています。Portfolio アプリケーションパッケージ内の NetPublish プラグインを Ctrl ボタンを押しながらクリックし、「パッケージの内容を表示」を選択します。

```
<Portfolio Package>/Contents/Plugins/  
<NetPublish Plugin.plugin>/Contents/  
Resources/Dada/en/istyle sheets/master/
```

- HTML またはテキストエディタで master.css を開きます。
- 次の4つの仮カラー値を検索して置き換えます。スタイルカラーの選択については、この章の最後の節を参照してください。
 - 背景色を変更するには、#BG を検索し、選択する色の16進値と #XXXXXX を置き換えます。
 - 1次色を変更するには、#ONE を検索し、選択する色の16進値と #XXXXXX を置き換えます。
 - 2次色を変更するには、#TWO を検索し、選択する色の16進値と #XXXXXX を置き換えます。
 - アクセント色を変更するには、#ACCENT を検索し、選択する色の16進値と #XXXXXX を置き換えます。



シャープ記号 # をこれらの検索および置換操作に含めることが大切です。

4. エディタで「ファイル」>「別名で保存」を選択し、編集後のCSSファイルを新しい名前と同じディレクトリに保存します。master.cssファイルは置き換えないください。

次にNetPublish Assistantを立ち上げたときに、この新しいスタイルシートはすべてのスタイルシート非依存テンプレートで利用できるようになります。

適切なスタイルシートカラーの選択

スタイルシート非依存サイトは、色の柔軟性を維持するだけでなく、カラー値の特定のコントラストも維持するように、慎重にデザインされています。

すべてのスタイルシート非依存サイトで読みやすい結果が生まれるカスタムスタイルシートを作成するには、特定の値のガイドラインに従ったカラー値を選択することが大切です。つまり、新しい色は、どのような色相や輝度であっても、明るさと暗さの点で特定の要件を満たしている必要があります。

適切な色を選択する場合は、次の値の仕様を参考にしてください。色は、輝度(パーセント)としてリストされています。

Photoshopでは、輝度は、カラーピッカーのHSBインジケータ内のB%の下に表示されます。#マークが付いたフィールドの下部に16進値も表示されます。

背景色: 約20%の輝度値を選択します。0～30%が許容範囲です。

メイン色: 最適な輝度は約60%、50～80%が許容範囲です。

サブ色: 最適な輝度は約80%、70～95%が許容範囲です。

アクセント色: 最適な輝度は約80%、60～95%が許容範囲です。

デフォルトのスタイルシートはすべて、アクセント色を除き、単一色のバリエーションを使用します。単一色ベースのスキームを選択すれば、調和の取れた結果が生まれることは確かですが、これは必要条件ではありません。



定義済みの色をすべて使っていないCSS独立サイトもたくさんあります。したがって、すべてのスタイルシート非依存サイトで色の変更に必要なはありません。

索引

A

Add コマンド 47
alias.properties ファイル 29
Archive コマンド 49

B

Base コマンド 43

C

Catalog 34
CatalogSet 33
Collection 37
CollectionSet 35
CSS 52

E

Extensis 社の連絡先 ii

F

FieldType 33
Find コマンド 45

J

JavaScript API 30

M

Mozilla JavaScript オブジェクト 42

N

NetPublish のインストール 3
NetPublish のコマンド 43
NetPublish Server のシリアル番号の登録 5
NetPublish Server の追加 5

O

Original コマンド 51

P

Page 40
Path 40

Q

QuickFind コマンド 44

R

Record 31
RecordSet 30
Remove コマンド 48
Request 37
Response 38

S

Show コマンド 47
site.properties ファイル 29

T

Thumbnail コマンド 50

U

Units 41

W

Web ページのヘッダーとフッター 10
Web ページのロゴ 10

あ

アイテムのコレクション 15, 16
アクセント色 53

え

エイリアス
カタログ 25

か

開始ステップ 8
カスタマーディングスタイルシート 52
カスタム NetPublish サイト 29
カタログ
公開、1つのエイリアスによる複数の
25
カタログ選択ステップ 8
管理
NetPublish Server 23
NetPublish サイト 25

き

技術サポート 2
 基本設定ステップ 10
 キャッシュサイズ 24

く

クイック検索 44
 グローバル JavaScript 関数 30

け

検索結果オプション 15
 検索結果ステップ 14
 検索ステップ 11
 検索フィールドのオプション 12

こ

公開ステップ 18
 コレクションステップ 17
 コレクションへの追加 / 削除 15, 16

さ

サーバーの選択 18
 サーバーの追加 18
 サイトの削除 25
 サブ色 53
 サムネールオプション 15, 17

し

システムの必要条件 1
 詳細検索オプション 12
 詳細検索オプションを設定する場合 12
 詳細情報ステップ 16
 新規テンプレートの作成 9

す

スタイルシート 12, 15, 16, 17, 52
 スタイルシートカラー 53
 「すべて表示」ボタン 12

せ

設定
 NetPublish Server 23
 NetPublish サイト 25

そ

ソフトウェアの必要条件 1

た

ダウンロードオプション 17

て

デザイン済みテンプレート 9
 テンプレート
 変更 29
 テンプレートの再表示 9
 テンプレートの非表示 9

と

トップページ 11

は

背景色 53
 パスワード
 管理 23
 サイト作成 23
 表示 25
 編集 25
 パスワードのオプション 19

ひ

必要条件
 システムとソフトウェア 1

ふ

ファイルの場所 21

へ

ページスタイル 11, 14, 16

め

メイン色 53

り

リンク 11

れ

連絡先 ii

ろ

ログファイル 24

