

ADOBE® COLDFUSION® BUILDER™ 2.0 ユーザーガイド

法律上の注意

法律上の注意について詳しくは、http://help.adobe.com/ja_JP/legalnotices/index.htmlを参照してください。

コンテンツ

第 1 章：ColdFusion Builder の新機能

ColdFusion Builder 2.0.1 の新機能	1
ColdFusion Builder 2.0 の新機能	2

第 2 章：はじめに

ColdFusion および ColdFusion Builder のマニュアル	7
ColdFusion および ColdFusion Builder の学習用リソース	8

第 3 章：ColdFusion Builder について

ColdFusion Builder の設定	9
ColdFusion Builder のライセンス認証	9
サードパーティー Eclipse プラグインのインストール	10
ファーストステップ	11
ColdFusion Builder のヘルプシステム	11
ColdFusion Builder の概要	14

第 4 章：ColdFusion Builder ワークベンチの基本

ワークベンチについて	17
ワークベンチのメニューとツールバー	18
ColdFusion Builder パースペクティブについて	22
ColdFusion Builder 開発パースペクティブ	23
ColdFusion デバッガパースペクティブ	31
ワークベンチへのビューの追加	37
ColdFusion Builder のエディター	37
パースペクティブのカスタマイズ	37

第 5 章：ColdFusion Builder でのコード編集

CFML エディター	39
コードアシスト	40
任意のタグブロックの選択	47
対応するタグへのジャンプ	47
クイックフィックス	47
コードフォーマッター	48
自動整形	50
コードの色付け	51
SQL エディター	51
コードの折りたたみと展開	53
シンタックスのチェックと強調表示	54
コードのナビゲート	55

コードリファクタリング	57
タグエディター	59
入力設定	59
CFML エディターのキーボードショートカット	59
ColdFusion Builder の検索	61
タスクビューへのタスクの追加	63
第 6 章：サーバーの管理	
Web サーバーの用語について	65
ColdFusion サーバーの追加	66
サーバービューの使用	72
RDS サーバー設定のインポート	73
第 7 章：プロジェクトの管理	
プロジェクトについて	74
ColdFusion プロジェクトの作成	74
プロジェクトのプロパティおよびサーバーの設定	75
ColdFusion ページ、インターフェイス、およびコンポーネントの追加	76
CFM または CFC ファイルのワークスペース外への作成	78
起動ページの設定	78
URL 接頭辞の設定	78
プロジェクトのインポート、エクスポート、および削除	78
プロジェクトおよびファイルの除外	79
ワークスペース外のリソースへのリンク	80
FTP および Secure FTP 接続によるプロジェクトのデプロイ	81
Ajax ライブラリのインポート	81
Flash Builder プロジェクトの使用	82
AIR アプリケーションの開発	82
第 8 章：アプリケーションのデバッグ	
ColdFusion デバッガーの使用	84
アプリケーションのデバッグ	86
リモートアプリケーションのデバッグ	90
第 9 章：ColdFusion Builder 拡張機能	
拡張機能について	91
ColdFusion Builder 2 の拡張機能の新機能	91
拡張機能の開発	91
拡張機能のユーザーインターフェイスの作成	106
ビューの作成	108
ハンドラーからのコールバックコマンドの指定	110
拡張機能ビューの使用	115
ColdFusion Builder Extension Creator を使用した拡張機能の作成とパッケージ化	116

拡張機能からコードアシストへのコントリビュート	118
拡張機能による起動ページ設定のサポート	119
第 10 章：拡張機能の使用	
Adobe CFC Generator	121
AS Class Generator	124

第 1 章：ColdFusion Builder の新機能

ColdFusion Builder 2.0.1 の新機能

ColdFusion Builder 2.0.1 には、ColdFusion 10 の言語とサーバー、複数のバグ修正、および次の機能強化が含まれます。

パフォーマンスの向上

- Application.cfc を含む大きいプロジェクトの強化されたインポート。
- 大きいプロジェクトでのコードアシストの効率の向上。
- エディターの効率の向上。

Adobe CFC Generator のユーザーエクスペリエンスの向上

Adobe CFC Generator は、よりよいユーザーエクスペリエンスを提供するように強化されました。123 ページの「[Adobe CFC Generator の使用 \(ColdFusion Builder 2.0.1 の場合\)](#)」を参照してください。

コールバックコマンドの新規追加と拡張

新しいコマンド

getProjects、reloadExtensions、sendMessageToExtension

拡張されたコマンド

コールバックコマンド getfunctionsandvariables によって返される応答には、変数スコープ関連の情報が含まれるようになっています。例えば変数が関数で var スコープになっている場合、スコープを特定する拡張機能を作成できます。

110 ページの「[ハンドラーからのコールバックコマンドの指定](#)」を参照してください。

ビューコントリビューションでの新しい keepfocus 属性

属性 keepFocus を true に設定すると、拡張機能ビューが動的に生成されるときに、フォーカスが移動しません。108 ページの「[ビューの作成](#)」を参照してください。

新しく追加されたイベント

イベント	説明
onfilechangeineditor	拡張機能はエディターでファイルが変更されたときのイベントを listen できます
onRDSDataViewSelectionChange	拡張機能は、RDS データビューで選択範囲が変更されたときのイベントを listen できます
OnFileSaved	ファイルが保存される際のイベント通知を追加します

94 ページの「[ColdFusion Builder 2.0.1 で追加された新しいイベント](#)」を参照してください。

複数ノードでのメニューコントリビューションのサポート

RDS データビューでは、テーブルノードに加えて、サーバー、データベース、ファイルの各ノードにメニューコントリビューションを追加できます。102 ページの「[複数ノードでのメニューコントリビューションのサポート](#)」を参照してください。

拡張機能間でのメッセージの送信

拡張機能では、別の拡張機能にメッセージを送信して、アクションを実行したり、イベントを通知したりできます。メッセージを受信した拡張機能は、アクションを実行することも、メッセージを無視することもできます。104 ページの「[拡張機能間でのメッセージの送信](#)」を参照してください。

その他の機能強化

- ファイルをローカルに保存するときに FTP でアップロードするための、FTP connector ダイアログボックスの新しい「Upload On Save」オプション。
- ショートカットを使用した同期およびアップロード (Ctrl + Alt + W U) とダウンロード (Ctrl + Alt + W D) の同時実行。
- ショートカットを利用した同期の実行 (Ctrl + Alt + W Y)。
- 新しいオプションを使用することで容易になったクイックフィックスのオン/オフ。このオプションを使用するには、**環境設定 / ColdFusion / プロファイル / エディター / シンタックスチェック**の順に選択し、「**クイックフィックスを使用可能にする**」オプションを選択します。
- リモートでの ColdFusion サーバーの起動と停止。
- プロジェクトを作成するときのコードアシスト用の辞書のバージョンの設定。

ColdFusion Builder 2.0 の新機能

ColdFusion Builder 2.0 Express Edition

トライアル版の有効期限が切れた後は、機能が制限された ColdFusion Builder の Express Edition を使用するようになります。ライセンスを購入するまでは、エディター、コードアシスト、シンタックスをハイライトなどの基本機能を引き続き使用できます。9 ページの「[ColdFusion Builder のライセンス認証](#)」を参照してください。

ファーストステップ

動的なファーストステップでは、チュートリアルやワークフロー、ムービー、通知、ヒント、テクニックへのアクセスと役に立つリソースが紹介されます。これは、ColdFusion Builder を起動したときに毎回表示されます。11 ページの「[ファーストステップ](#)」を参照してください。

エディターの機能強化

タスクビューへのタスクの追加

タスクを CFML コメントとして定義すると、設定した優先度に基づいてタスクビューにタスクが表示されます。タスクは、デフォルトの TODO および FIXME タグ、またはカスタムタスクタグの接頭辞を使用して表示します。63 ページの「[タスクビューへのタスクの追加](#)」を参照してください。

任意のタグブロックの選択

Ctrl + Alt + B (Windows) または Command + Option + B (Mac) のショートカットを使用して、キャレット位置から囲まれるタグブロックを選択できます。選択状態を維持したままショートカットを繰り返すと、コードの選択を累積的に行うことができます。47 ページの「[任意のタグブロックの選択](#)」を参照してください。

対応するタグへのジャンプ

Ctrl + Alt + M (Windows) または Command + Option + M (Mac) のショートカットを使用して、キャレット位置をタグブロックの最初から最後に、または最後から最初に移動できます。47 ページの「[対応するタグへのジャンプ](#)」を参照してください。

コードアシストの機能強化

必須属性の自動挿入

環境設定に基づいて、必須属性を自動挿入します (デフォルトのオプション)。環境設定が設定されていない場合は、属性は挿入されず、単に提案候補として表示されます。45 ページの「[必須属性の自動挿入](#)」を参照してください。

必須のタグ属性とオプションのタグ属性の分割リストの表示

属性のリストを表示します。必須属性は提案候補リストの上部にチェックマーク付きで表示され、その属性が必須であることが示されます。必須の属性とオプションの属性は、線によって区切られます。45 ページの「[必須属性とオプション属性の分割リスト](#)」を参照してください。

createObject の提案候補の表示

createObject 関数でサポートされているすべてのオブジェクト (および対応する引数) の提案候補が表示されます。45 ページの「[createObject の提案候補](#)」を参照してください。

コードアシストの提案候補の循環

Ctrl + スペースバーを使用して、コードアシストの提案候補を循環することができます。一度に 1 つの提案候補グループを次のグループに (例えば、変数から関数のように) 切り替えることができます。46 ページの「[コードアシストの提案候補の循環](#)」を参照してください。

テキストを含む提案候補のフィルタリング

指定したテキストに基づいて、提案候補をフィルタリングします。フィルターテキストを含むすべての候補が表示され、指定したテキストで始まる候補が選択状態になります。46 ページの「[テキストを含む提案候補のフィルタリング](#)」を参照してください。

コードアシストのデータ型認識

コードアシストの提案候補は、予想されるデータ型に基づいて表示されます。46 ページの「[コードアシストのデータ型認識](#)」を参照してください。

関数の引数の自動挿入

関数を挿入すると、必須のすべての引数が自動的に挿入されます。46 ページの「[関数の引数の自動挿入](#)」を参照してください。

関数コンテキストアシストの機能強化

- 関数のパラメーター内でコードアシストを読み込むと、毎回、関数コンテキストアシストが表示されます。
- 現在編集中的パラメーターは、太字で強調表示されます。
- オプションのパラメーターは、[] の中に表示されます。

46 ページの「[関数コンテキストアシスト](#)」を参照してください。

接続属性のスマートコードアシスト

cfexchangeconnection や cftp などのタグ用に作成された接続変数が表示されます。46 ページの「[接続属性のスマートコードアシスト](#)」を参照してください。

cfloop の使用法に基づいたコードアシスト

ループのタイプに基づいて、必須属性が自動的に挿入されます。47 ページの「[cfloop の使用法に基づいたコードアシスト](#)」を参照してください。

コードの折りたたみの機能強化

任意のコードブロックを非表示にしたり表示したりできます (Ctrl + Alt + F (Windows) または Command + Option + F (Mac))。定義したコードの折りたたみは保存されるため、それ以降のセッションでも使用できます。53 ページの「[コードの折りたたみと展開](#)」を参照してください。

コードの整形

コードフォーマッター

インデントや行の長さ、タグおよび属性名の大文字 / 小文字表記を標準化して、CFML コードの見た目を変更します (Ctrl + Shift + F (Windows) または Command + Shift + F (Mac))。コードフォーマッターでは、タグベースとスクリプトベースの両方のシンタックスが整形されます。コードフォーマッターでは、汎用的な環境設定が提供されます。この環境設定は、カスタマイズ、読み込みおよび書き出しすることができます。48 ページの「[コードフォーマッター](#)」を参照してください。

自動整形

自動的にコードの行をインデントして、終了タグを調整します。50 ページの「[自動整形](#)」を参照してください。

クイックフィックス

ファイルを開いたり編集したりすると、定義されていない関数呼び出し、CFC および CFM が自動的に識別されます。エディターの左余白に電球のアイコンが表示され、このアイコンをクリックすると、適切なクイックフィックスを実行するかどうかメッセージが表示されます。47 ページの「[クイックフィックス](#)」を参照してください。

ColdFusion Builder の検索

専用の検索と置換の機能 (Ctrl + F (Win) または Command + F (Mac)) で、テキストまたはタグの検索を、現在のドキュメントや開いているドキュメント、選択したリソース、プロジェクト、作業セット、ワークスペース、ローカルディレクトリ、FTP の場所、RDS の場所に対して実行します。正規表現と複数行検索もサポートされています。時間を有効に活用するために、バックグラウンドでリモート検索を実行して、他の作業を進めることもできます。ColdFusion Builder の検索は、CFC、CFM、HTML、XML、CSS および JS のファイルタイプに適用されます。61 ページの「[ColdFusion Builder の検索](#)」を参照してください。

キーボードショートカットの機能強化

- クイックアシストを使用して、ウィザードやタグ、カスタムショートカットに関連するショートカットのカタログを表示
- カスタムキーボードショートカットの作成

- ショートカットの変更および削除
- ショートカットを検索し、フィルターテキストに基づいて特定のアクションを実行
- ショートカットを CSV 形式で書き出しおよび読み込み
- デフォルトのショートカット設定の復元

59 ページの「[CFML エディターのキーボードショートカット](#)」を参照してください。

デフォルトのキーボードショートカットの変更

アクション	ColdFusion Builder のデフォルトのショートカット	ColdFusion Builder 2.0 のデフォルトのショートカット
cfabort の挿入	Ctrl + Shift + A	Ctrl + T, A
cfdump の挿入	Ctrl + Shift + D	Ctrl + T, D
CFScript ブロックの挿入	Ctrl + Alt + R	Ctrl + T, S
cfset の挿入	Ctrl +=	Ctrl + T, =
タグエディターへのアクセス	Ctrl + T	Ctrl + Shift + T
ColdFusion コメントで囲む / 解除	Ctrl + Shift + M	Ctrl + T, M
## で囲む	Ctrl + Shift + H	Ctrl + T, H
cfoutput で囲む	Ctrl + Shift + O	Ctrl + T, O
try/catch で囲む	Ctrl + Shift + T	Ctrl_T, T

ナビゲーションの機能強化

コードナビゲーション

ユーザー定義関数から、次または前のユーザー定義関数にナビゲートしたり、(CFC ファイル内の) CFC の最上位レベルのメンバーにナビゲートします (Ctrl + Shift + 上向き矢印 / 下向き矢印キー (Windows) または Command + Shift + 上向き矢印 / 下向き矢印キー (Mac))。CFScript で提供されるフロー制御ステートメントの場合、CFC ページと CFM ページの両方で、キャレット位置がこれらのステートメント内にあるときに、次または前のコードブロックにナビゲートします。55 ページの「[コードナビゲーション](#)」を参照してください。

スマートタブナビゲーション

次または前の関数の引数やタグ属性にすばやくナビゲートします。次にナビゲートするには、Ctrl +] (Windows) または Command +] (Mac) を、前にナビゲートするには、Ctrl + [(Windows) または Command + [(Mac) を使用します。スマートタブナビゲーションは、関数定義の場合の cffunction の引数にも適用できます。57 ページの「[スマートタブナビゲーション](#)」を参照してください。

拡張機能の強化

ハンドラーからのコールバックコマンドの指定

実行フェーズで次の操作を完了できます。可能な操作：ファイル、フォルダー、プロジェクトの更新、ファイルを開く、テキストの挿入、サーバーまたはデータソースのリストの取得、テーブルの詳細の取得、ファイルの内容の検索と結果の取得、ファイル内の関数と変数の一覧表示。110 ページの「[ハンドラーからのコールバックコマンドの指定](#)」を参照してください。

ビューの作成

ビューを作成し、拡張機能を使用して作成するビューにツールバーのアイテムを指定します。このビューは、データを表示したまま IDE の機能を使用する場合に便利です。108 ページの「[ビューの作成](#)」を参照してください。

拡張機能からコードアシストへのコントリビュート

拡張機能から、コードアシストに提案候補を追加します。118 ページの「[拡張機能からコードアシストへのコントリビュート](#)」を参照してください。

拡張機能による起動ページ設定のサポート

拡張機能を使用して、フレームワークアプリケーションの開始ページの URL を動的に生成します。ColdFusion Builder の機能（ColdFusion デバッガー、外部ブラウザの使用、ColdFusion アプリケーションとしての実行など）では、この URL が開始ページの URL として使用されます。生成される URL には、URL に追加されるクエリパラメーターも含まれません。119 ページの「[拡張機能による起動ページ設定のサポート](#)」を参照してください。

バージョンとサーバーの詳細

ColdFusion Builder からハンドラー CFM ファイルに送信される `ideeventinfo.xml` に、ColdFusion Builder のバージョン情報とサーバーの詳細（ホスト名やポート番号など）が含まれます。103 ページの「[ColdFusion Builder とハンドラーの通信について](#)」を参照してください。

その他の機能強化

CFM または CFC ファイルのワークスペース外への作成

ファイルビューを使用して、CFM または CFC ファイルをワークスペースの外に作成します。78 ページの「[CFM または CFC ファイルのワークスペース外への作成](#)」を参照してください。

起動ページの設定

特定のファイルをプロジェクトの起動ページとして指定します。ファイルを実行したりデバッグしたりすると、このページが読み込まれます。78 ページの「[起動ページの設定](#)」を参照してください。

ヘルプの機能向上

- マウスポインターを合わせると、状況依存型ヘルプが表示されます。これには、関数やタグ、UDF が該当します。UDF の場合は、`hint` 属性の値がヘルプウィンドウに表示されます。
- コードアシストを実行して、パラメーターまたは属性を強調表示したときに状況依存型ヘルプを使用できます。

その他のリソース

• [ColdFusion Builder 2 の新機能](#)

この記事では、Ramchandra Kulkarni が ColdFusion Builder 2 で提供されるすべての生産性向上ツールを活用するのに役立つ概要情報について説明します。

• [Getting started with ColdFusion Builder 2](#)

この Adobe TV ビデオチュートリアルでは、Adobe プラットフォームのエバンジェリストである Terry Ryan が、ColdFusion Builder 2 の最新機能の使用方法を説明します。

• [ColdFusion Builder team blog](#)

ColdFusion Builder チームが最新のニュース、更新、その他の役に立つ情報を提供します。

第2章：はじめに

Adobe® ColdFusion® Builder™ は、Adobe® ColdFusion® アプリケーションを開発するための IDE (Integrated Development Environment：統合開発環境) です。

ColdFusion Builder の使用方法を解説する資料は数多く提供されています。背景となる知識や、ColdFusion Builder の利用目的に応じて、適切なマニュアル資料に目を通すようにしてください。

ColdFusion を初めて使用する場合

- ❖ ColdFusion Builder を使用する前に ColdFusion の基本知識を習得しておく、後で非常に役に立ちます。最初に、ColdFusion Administrator から参照できる『ファーストステップ』マニュアルに目を通しておくことをお勧めします。

ColdFusion での開発経験がある場合

- ColdFusion Builder は Eclipse™ (オープンソース IDE) をベースに開発されているため、ColdFusion Builder を効率的に使用するには、Eclipse 固有の用語や概念を理解しておく必要があります。ただし、今までに Eclipse ベースの環境を使用した経験がある場合、ColdFusion Builder ワークベンチの操作方法はそれらの環境とほとんど変わりありません。

詳細については、17 ページの「[ColdFusion Builder ワークベンチの基本](#)」を参照してください。

- 今までテキストエディターを使用してコードを開発していた場合、高性能な ColdFusion Builder を使用すると、コードの開発がはるかに容易になります。詳細については、9 ページの「[ColdFusion Builder について](#)」を参照してください。
- ColdFusion Builder の機能の概要については、14 ページの「[ColdFusion Builder の概要](#)」を参照してください。
- ColdFusion Builder を使用して効率的に ColdFusion アプリケーションを開発するには、『CFML リファレンス』と『ColdFusion アプリケーションの開発』を参照してください。『CFML リファレンス』では、CFML 言語要素について詳しく説明しています。この資料は、ColdFusion Builder のヘルプから参照できます。詳細については、13 ページの「[CFML リファレンスヘルプの表示](#)」を参照してください。
- ColdFusion および ColdFusion Builder に関連して入手できるマニュアルについては、7 ページの「[ColdFusion および ColdFusion Builder のマニュアル](#)」を参照してください。

ColdFusion および ColdFusion Builder のマニュアル

ColdFusion および ColdFusion Builder のマニュアルは、統合ヘルプシステムを使用して ColdFusion Builder 内で参照できます。詳細については、11 ページの「[ColdFusion Builder のヘルプシステム](#)」を参照してください。

オンラインマニュアルを表示するには、ColdFusion ヘルプ&サポートページ (www.adobe.com/go/learn_cfu_support_jp) にアクセスしてください。

これらのマニュアルでは、新しいコメントを追加したり、既存のコメントを参照することもできます。

Adobe ColdFusion のマニュアル

ColdFusion のマニュアルセットには、次のマニュアルが含まれています。

マニュアル	説明
『Adobe® ColdFusion® 9 インストール』	Windows、Macintosh、Solaris、Linux、および AIX 環境でのシステムインストールおよび基本設定について説明します。
『Adobe ColdFusion® 9 設定と管理』	サーバー設定の管理、データソースと Web サーバーの設定、セキュリティの管理、ColdFusion アプリケーションのデプロイなど、ColdFusion の管理方法について説明します。 また、キャッシュの設定、CFX タグの設定、ColdFusion サーバーモニターを使用したサーバーアクティビティの監視などについても説明しています。
『Adobe® ColdFusion® 9 アプリケーションの開発』	ダイナミック Web アプリケーションの開発方法について説明します。 このガイドでは、CFML プログラミング言語と ColdFusion の機能（ColdFusion Web サービス、ColdFusion ポートレット、ColdFusion ORM、AJAX サポート、Flex および AIR との統合など）の使用方法と、他社の製品およびテクノロジー（Microsoft Office、OpenOffice、SharePoint など）との統合に関する詳細情報が記載されています。
『Adobe® ColdFusion® 9 CFML リファレンス』	すべての ColdFusion タグ、関数、変数に関する説明、シンタックス、使用方法、コード例が含まれています。

Adobe ColdFusion Builder のマニュアル

ColdFusion Builder のマニュアルセットには、次のマニュアルが含まれています。

マニュアル	説明
『Adobe® ColdFusion® Builder 2.0 インストール』	Windows プラットフォームおよび Macintosh プラットフォームにおいて、ColdFusion Builder をスタンドアロン設定でインストールする方法、または既存の Eclipse ワークベンチのプラグインとしてインストールする方法について説明します。
『ColdFusion® Builder 2.0 ユーザーガイド』	ColdFusion アプリケーションを開発するための IDE (Integrated Development Environment : 統合開発環境) として ColdFusion Builder を使用方法について説明します。

ColdFusion および ColdFusion Builder の学習用リソース

ColdFusion および ColdFusion Builder の学習用に、次のリソースがオンラインで提供されています。

リソース	リンク
ColdFusion ヘルプ&サポート	www.adobe.com/go/learn_cfu_support_jp
ColdFusion デベロッパーセンター	www.adobe.com/go/learn_cfu_cfdevcenter_jp
Adobe ColdFusion オンラインフォーラム	www.adobe.com/go/learn_cfu_forums_jp

第3章：ColdFusion Builder について

Adobe® ColdFusion® Builder™ は、オープンソース IDE（Integrated Development Environment：統合開発環境）である Eclipse™ をベースに開発されました。IDE は、開発者が必要とするツールの大部分を 1 つにまとめたアプリケーションです。ColdFusion Builder には、コードの編集と検証、ファイルとプロジェクトの管理、およびデバッグ用のツールが用意されています。ColdFusion Builder のプラグインアーキテクチャを利用すると、必要に応じて機能を拡張できます。

ColdFusion Builder は Microsoft® Windows® プラットフォームおよび Apple® Macintosh プラットフォームで動作します。ColdFusion Builder のインストール設定オプションでは、スタンドアロン設定で ColdFusion Builder をインストールするか、既存の Eclipse ワークベンチインストール内にプラグインセットとして ColdFusion Builder をインストールするかを選択できます。詳細については、『Adobe ColdFusion Builder インストール』を参照してください。

ColdFusion Builder の設定

ColdFusion Builder インストーラーには、次の 2 つの設定オプションがあります。

スタンドアロン設定の ColdFusion Builder IDE インストール Eclipse 3.6.2 をベースに構築されたスタンドアロン IDE（Integrated Development Environment：統合開発環境）として ColdFusion Builder をインストールします。

Eclipse IDE 内の ColdFusion Builder プラグイン 既存の Eclipse 環境または Adobe® Flash® Builder™ 環境内のプラグインとして ColdFusion Builder をインストールします。既に Eclipse ワークベンチを使用していて、Eclipse プラグインのツールキットに ColdFusion Builder プラグインを追加したいユーザーに適した設定です。

注意：プラグインで ColdFusion Builder をインストールするには、コンピューターに Eclipse 3.6.2 以上がインストールされている必要があります。

どちらの設定を使用すればよいか分からない場合は、次のガイドラインに従ってください。

- 既に Eclipse 3.6.2 以上がインストールされている場合は、プラグイン設定を選択してください。
- まだ Eclipse 3.6.2 以上がインストールされていない場合は、スタンドアロン設定を選択してください。この設定では他の Eclipse プラグインもインストールできるので、将来の開発作業の幅が広がります。

どちらの設定も機能的には変わりません。ただし、プラグインでは、メニューの名前や、メニューコマンドへのアクセス方法が少し異なる場合があります。

ColdFusion Builder のライセンス認証

ColdFusion Builder を起動すると、製品シリアル番号の入力を求めるメッセージが表示されます。有効な製品シリアル番号を入力すると、インターネット接続が最初に検出されたときに、ソフトウェアのライセンス認証がバックグラウンドで開始します。詳細については、Adobe 製品ライセンス認証センター（www.adobe.com/go/activation_jp）にアクセスしてください。

ColdFusion Builder の起動時に製品シリアル番号を入力しない場合は、ColdFusion Builder 2.0 の機能をすべて備えた体験版を 60 日間使用できます。体験版の有効期限が切れた後もすべての機能を引き続き使用するためには、ライセンスを購入する必要があります。購入しない場合は、機能が制限された ColdFusion Builder Express Edition に切り替わります。

ColdFusion Builder 2 Express Edition

Express Edition では、エディター、コードアシスト、シンタックスの強調表示などの、ColdFusion Builder の基本機能を引き続き使用できます。

しかし、Express Edition では、次の重要な機能を使用できません。

- 拡張機能に対するコードアシスト
- コードのヒント
- 拡張機能のコールバック
- リモートサーバーへの接続
- クイックフィックス
- リモートプロジェクトのデバッグ
- リファクタリング
- ColdFusion 検索
- コードの整形
- FTP サポート
- ログビューア
- ローカルファイルブラウザー
- コードハイパーリンク
- ホバーヘルプ

Express Edition は、ライセンスを購入して製品シリアル番号を指定すると、すべての機能を備えた製品版に移行できます。

ColdFusion Builder ライセンスの管理

シングルライセンスユーザーの場合は、最大 2 台のコンピューターで ColdFusion Builder を使用できます。最初の 2 台のコンピューター以外のコンピューターで ColdFusion Builder のライセンスを使用するには、2 台のコンピューターのいずれかの ColdFusion Builder のライセンス認証を解除します。その後、別のコンピューターでライセンスを使用できます。

ColdFusion Builder のライセンス認証を解除するには、ヘルプ / Adobe ColdFusion Builder のライセンス認証解除を選択します。

注意： ColdFusion Builder をアンインストールする場合は、まずライセンス認証を解除してから、ColdFusion Builder をアンインストールすることをお勧めします。

サードパーティー Eclipse プラグインのインストール

ColdFusion Builder は、開発環境の拡張やカスタマイズを可能にするサードパーティー製 Eclipse プラグインとの統合をサポートしています。サードパーティー製 Eclipse プラグインは、Windows および Mac OS X プラットフォームにインストールできます。

注意： プラグインが正しくインストールされていることを確認するには、Helios 更新サイトの URL <http://download.eclipse.org/releases/helios> (または同等の Eclipse 更新サイトの URL) を追加します。

- 1 Adobe ColdFusion Builder で、ヘルプ / 新規ソフトウェアをインストールを選択します。
- 2 「追加」をクリックします。

- 3 インストールするプラグインの名前と URL を入力します。例えば、Subclipse バージョン 1.0 をインストールするには、http://subclipse.tigris.org/update_1.6.x という URL を入力します。
- 4 インストールするプラグインを選択します。
- 5 「次へ」をクリックし、インストールウィザードの指示に従います。
- 6 「終了」をクリックします。

ファーストステップ

ColdFusion Builder では、動的なファーストステップが提供されます。ColdFusion Builder を起動すると、毎回、ファーストステップ画面で次へのアクセスが紹介されます。

- ファーストステップのチュートリアル
- ワークフロー
- ムービーのチュートリアル
- リリースにおける新機能や、ColdFusion および ColdFusion Builder のリリースとホットフィックスの詳細などの通知
- ヒントとテクニク
- 役に立つリソース
- よく使用するオプション（最近開いたファイルなど）や、プロジェクトウィザードの起動、サーバーの追加、推奨される拡張機能、拡張機能のインストールへのリンク

ファーストステップ画面は、表示されないようにしたり、後でまた表示されるようにしたりできます。

ファーストステップ画面の表示

- ❖ ヘルプ / ColdFusion Builder : はじめにを選択します。

ファーストステップ画面の非表示

- ❖ 次のいずれかの操作を行います。
 - ファーストステップ画面の「再び表示しない」オプションを選択します。
 - 環境設定ダイアログボックス（ウィンドウ / 環境設定 / ColdFusion / スタートアップ）で、「起動時に「ようこそ」ページを表示」オプションのチェックを解除します。

その他のリソース

- [ColdFusion Builder Start Page](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sandeep Paliwal が、ファーストステップについて説明します。

ColdFusion Builder のヘルプシステム

ColdFusion Builder で作業する際には、ワークベンチのユーザーインターフェイス要素（ビューとダイアログボックス）に関する状況依存型ヘルプと、コード要素に関する言語リファレンスヘルプを表示できます。

Adobe Community Help Client

Adobe CHC (Community Help Client) は、ColdFusion Builder のヘルプコンテンツを表示する AIR アプリケーションです。ヘルプコンテンツは Web から直接提供されます。CHC を初めて開く場合は、インターネット接続が必要です。その後、ヘルプコンテンツをデスクトップにダウンロードすると、オフラインで参照または検索できます。

ColdFusion Builder のヘルプシステムを開いて表示するには、ヘルプ / ColdFusion Builder ヘルプを選択します。

状況依存型ヘルプの使用

- 1 ワークベンチで、ビューやダイアログボックスなどのユーザーインターフェイス要素を選択します。
- 2 Windows の場合は、F1 キーを押すと、関連するヘルプページがコミュニティヘルプクライアントで開きます。

Mac OS で状況依存型ヘルプを使用するには、ダイナミックヘルプコマンドへのキーボードショートカットを設定します。ダイナミックヘルプコマンドへのキーボードショートカットを指定するには、次のようにします。

- 1 Adobe ColdFusion Builder / 環境設定を選択します。
- 2 ツリービューで、一般 / キーを選択します。
- 3 ダイナミックヘルプコマンドを選択します。
- 4 設定するキーの組み合わせを押します。例えば、Ctrl + Shift + 1 と入力するには、Ctrl キーと Shift キーを押した状態で 1 キーを押します。キーの間のプラス記号 (+) は、キーを連続して押す必要があることを示しています。

Community Help Client の設定

CHC の設定を調整すると、ヘルプコンテンツの表示を制御できます。

- 1 CHC で、編集 / 環境設定を選択します。
- 2 環境設定ダイアログボックスで、必要に応じて次のオプションを設定します。

一般設定

- CHC のテーマとして、明るいテーマや暗いテーマを選択できます。テーマは CHC の外観を決定する形式設定です。例えば、暗いテーマを適用すると、CHC で暗い背景とテーマ固有のフォントが使用されます。
- デフォルトでは、CHC は AIR アプリケーションとして起動されます。ただし、Web ブラウザー内で CHC を起動することもできます。その場合は、「ブラウザーでヘルプを開きます」を選択します。

ダウンロードの環境設定

オフラインで利用する製品とヘルプパッケージを選択できます。

アップデーターの設定

CHC では、AIR アプリケーションを再インストールしなくても、コンテンツと機能を更新できます。CHC を開くとき、または閉じるときに、新しい更新に関するメッセージを表示するには、「アップデートが提供されたら通知する」を選択します。更新を手動で検索してインストールすることもできます。

ローカルコンテンツ

デスクトップにダウンロードしたヘルプパッケージを手動で削除できます。

検索とフィルターのオプション

CHC では、複数のヘルプパッケージや検索場所にまたがる検索を実行できます。

「検索オプション」をクリックすると、次の検索場所が表示されます。

ローカルヘルプ オフラインで表示するためにデスクトップにダウンロードしたヘルプパッケージを検索します。「フィルター結果」をクリックすると、検索結果がヘルプパッケージ内の特定のブックに絞り込まれます。

Adobe.com デベロッパーセンターの記事、チュートリアル、ナレッジベースの記事など、adobe.com で入手できるすべてのオンラインリソースから信頼性の高い最新のリファレンスコンテンツを検索します。

コミュニティヘルプ adobe.com で入手できるすべてのリソースに加え、Adobe コミュニティのエキスパートから投稿された実用的なコンテンツを検索します。

コード検索 固有のコード検索機能を使用して、adobe.com で入手可能なコードスニペットを検索できます。

ダイナミックナビゲーション

CHC には、関連するコンテンツを検索するために、検索結果に基づくダイナミックナビゲーション機能が用意されています。関連するコンテンツはすべて、「関連トピック」タブに表示されます。

フィードバックと評価

有効な Adobe ID を持っている場合は、ヘルプコンテンツに対するコメントの追加、評価、寄稿などを行えます。Adobe ID をお持ちでない場合は、<https://www.adobe.com/jp/membership> で登録することで、ID を取得できます。

CHC の左下隅にある「フィードバックと評価」をクリックし、フィードバックを記入します。フィードバックは、ヒントなどの形式で記入することも、関連するオンラインリソースへのリンクを示すこともできます。

Eclipse のヘルプシステム

Eclipse のヘルプシステムには、コード要素に関する CFML リファレンスヘルプが表示されます。また、Eclipse のヘルプシステムには、インストールされているサードパーティープラグインや、ColdFusion Builder に付属するプラグインのヘルプも表示されます。

Eclipse のヘルプシステムを開いて表示するには、ヘルプ/ヘルプ目次を選択します。

CFML リファレンスヘルプの表示

ColdFusion Builder には『CFML リファレンス』が組み込まれているので、CFML タグや関数のリファレンスヘルプをすばやく参照できます。

- CFML タグや関数のヘルプトピックを開くには、次のようにします。
 - 1 CFML エディターで、ビルトインタグや関数名を強調表示するか、その場所にマウスポインターを置いて、CFML タグ要素または関数を選択します。
 - 2 Windows の場合は F1 キー、Mac OS の場合は相当するキーボードショートカットを押します。選択したタグに関するヘルプページへのリンクが、ドッキングされたヘルプウィンドウに表示されます。

注意： Mac OS の場合は、Command + Shift + / キーを押して CFML タグまたは関数のヘルプを表示すると、ドッキングされたヘルプウィンドウに全般的なエディターのヘルプが表示されます。タグまたは関数に固有のヘルプを表示するには、CFML エディターで選択したタグまたは関数名を再度クリックします。
 - 3 ヘルプページリンクをクリックします。選択したタグまたは関数に関するヘルプページが表示されます。

ヘルプの設定

ヘルプの設定を調整すると、Eclipse ヘルプシステムでのヘルプの表示方法を制御できます。

- 1 ColdFusion Builder でウィンドウ/環境設定を選択します。

- 2 環境設定ダイアログボックスの左側にツリービュー構造が表示されます。「ヘルプ」を選択します。
- 3 必要に応じて、次のオプションを設定します。

ヘルプ検索を開く ヘルプ検索をダイナミックヘルプビューウィンドウで表示するか、ブラウザで表示するかを決定します。デフォルトでは、ダイナミックヘルプでの表示が選択されています。

ヘルプビュードキュメントを開く ダイナミックヘルプのリンクから開かれたドキュメントをどこに表示するかを決定します。デフォルトでは、「インプレース」が選択されており、ヘルプドキュメントはダイナミックヘルプビューウィンドウに表示されます。ヘルプドキュメントを ColdFusion Builder IDE の編集領域で開くには、「エディター領域内」を選択します。

ヘルプコンテンツを開く 外部の Web ブラウザーにヘルプを表示できます。デフォルトでは、IDE に組み込まれているブラウザにヘルプが表示されます。Web ブラウザーを選択するには、「外部ブラウザ内」を選択し、「Web ブラウザー」リンクを選択します。

ウィンドウコンテキストヘルプを開く ウィンドウを開く状況依存型ヘルプリンクの表示方法を決定します。デフォルトでは、状況依存型ヘルプリンクは、ダイナミックヘルプビューに表示されます。ダイナミックヘルプビューは、他のビューと同様に、現在のパースペクティブにドッキングされます。状況依存型ヘルプリンクを情報ポップアップ（ツールヒントに類似したもの）に表示するには、「情報ポップアップ内」を選択します。

ダイアログコンテキストヘルプを開く ダイアログボックスを開く状況依存型ヘルプリンクの表示方法を決定します。デフォルトでは、ダイアログボックスにヘルプが表示されます。状況依存型ヘルプリンクを情報ポップアップ（ツールヒントに類似したもの）に表示するには、「情報ポップアップ内」を選択します。

ダイナミックヘルプの使用

ダイナミックヘルプは現在のパースペクティブにドッキングされ、関連するビューとダイアログボックスについてのトピックが表示されます。

❖ ヘルプ/ダイナミックヘルプを選択します。

デフォルトのキーボードショートカットバインディングを変更するには、ウィンドウ/環境設定/一般/キーを選択し、「ダイナミックヘルプ」コマンドに割り当てられたバインディングを変更します。キーボードショートカットを変更する方法については、59 ページの「[CFML エディターのキーボードショートカット](#)」を参照してください。

ColdFusion Builder の概要

ColdFusion Builder を使用すると、次の作業を行うための機能をすべて備えた IDE で ColdFusion アプリケーションを開発できます。

注意：作業を開始する前に、ColdFusion Builder ワークベンチとその機能について理解しておいてください。詳細については、17 ページの「[ColdFusion Builder ワークベンチの基本](#)」を参照してください。

- **ColdFusion プロジェクトの作成：**プロジェクトウィザードを使用してプロジェクトを作成し、そのプロジェクトで使用するローカルサーバーまたはリモートサーバーを設定できます。詳細については、74 ページの「[ColdFusion プロジェクトの作成](#)」を参照してください。
- **サーバーの追加と管理：**ColdFusion Builder にはサーバー管理機能が組み込まれており、ColdFusion Builder 内で ColdFusion サーバー、ColdFusion Administrator、およびサーバーモニターを管理できます。

JRun サーバーと JRun 以外のサーバーの両方を設定できます。アプリケーションを効率的に管理するために、ColdFusion Builder には仮想ホスト、仮想ディレクトリ、URL 接頭辞などの機能が用意されています。

これらの機能の使用方法与サーバー管理の詳細については、65 ページの「[サーバーの管理](#)」を参照してください。

- **FTP 接続によるリモートプロジェクトのデプロイ**：ColdFusion Builder では、ローカルサーバーとリモートサーバーの両方を管理できます。リモートサーバーへの接続には、RDS (Remote Data Services) 接続または FTP (File Transfer Protocol：ファイル転送プロトコル) 接続を使用できます。FTP 接続を介したリモートプロジェクトのデプロイと同期の詳細については、81 ページの「[FTP および Secure FTP 接続によるプロジェクトのデプロイ](#)」を参照してください。
- **ColdFusion ページ、コンポーネント、およびインターフェイスの作成**：ColdFusion Builder には、ColdFusion ページ、ColdFusion コンポーネント (CFC)、インターフェイスなどのリソースを作成するためのウィザードが用意されています。これらのリソースはプロジェクトに関連付けることができます。詳細については、76 ページの「[ColdFusion ページ、インターフェイス、およびコンポーネントの追加](#)」を参照してください。
- **CFML エディターを使用した ColdFusion アプリケーションのソースコードの記述と編集**：ColdFusion Builder には、CFML、HTML、JavaScript、および CSS のコードを編集するためのコードエディターが用意されています。編集するコードのタイプに応じて、適切なエディターが開きます。詳細については、37 ページの「[ColdFusion Builder のエディター](#)」を参照してください。

これらのエディターには、コードの色付け、コードアシスト、アウトラインビューなど、コードのナビゲーションを容易にするための機能が数多く含まれています。

CFML エディターには、コード補完、コードリファクタリング、効率的なコードナビゲーションなどの機能が用意されています。ColdFusion Builder では、様々な色やフォントを使用してエディター内のコードを表示できます。CFML エディターの機能の詳細については、39 ページの「[ColdFusion Builder でのコード編集](#)」を参照してください。

- **SQL エディターを使用した SQL ステートメントの作成と編集**：CFML エディターには SQL エディターが組み込まれており、SQL ステートメントを作成または編集できます。サポートされている SQL ステートメントのコードの色付けを設定することもできます。SQL エディターの詳細については、51 ページの「[SQL エディター](#)」を参照してください。
- **CFML 辞書の使用と作成**：ColdFusion Builder には、CFML コードの入力を支援する CFML 辞書が組み込まれています。CFML 辞書は、ColdFusion バージョン 7、8、および 9 でサポートされています。独自のカスタム CFML 辞書を作成することもできます。辞書の使用と作成の詳細については、44 ページの「[CFML 辞書](#)」を参照してください。
- **ColdFusion Builder 開発ビューの使用**：ColdFusion Builder には、アプリケーションの開発を容易にする様々なビューがあります。

RDS ファイルビューでは、ローカルサーバーとリモートサーバーのファイルシステムにアクセスして、ファイルを検索できます。RDS データビューでは、ローカルまたはリモートのデータソースにアクセスして、データを検索できます。詳細については、24 ページの「[RDS ファイルビューおよび RDS データビュー](#)」を参照してください。

また、RDS クエリビューアを使用して、ローカルデータサーバーまたはリモートデータサーバー上のデータを照会することもできます。詳細については、25 ページの「[RDS クエリビューア](#)」を参照してください。

サービスブラウザービューでは、サーバーの Web ルート内にある CFC とそのメソッドを参照できます。ローカルとリモート両方の CFC および Web サービスにアクセスできます。詳細については、28 ページの「[サービスブラウザービュー](#)」を参照してください。

スニペットビューとアウトラインビューを使用すると、コードの再利用や簡素化が容易になります。詳細については、29 ページの「[スニペットビュー](#)」および 23 ページの「[アウトラインビュー](#)」を参照してください。

すべての開発ビューの詳細については、23 ページの「[ColdFusion Builder 開発パースペクティブ](#)」を参照してください。

- **ColdFusion debugger を使用したアプリケーションのデバッグ**：ColdFusion Builder には、アプリケーションをデバッグできるデバッグビューやブレークポイントビューなどのデバッグビューを備えた統合デバッガーが用意されています。これらのビューの詳細については、31 ページの「[ColdFusion デバッグパースペクティブ](#)」を参照してください。
- コードに対するブレークポイントの追加、関数のステップインやステップオーバー、式の検査と評価を行えます。ローカルとリモートの両方のサーバーにあるファイルをデバッグできます。詳細については、84 ページの「[アプリケーションのデバッグ](#)」を参照してください。

問題ビュー、TailView ビュー、およびコンソールビューでは、シンタックスエラー、サーバーエラー、およびコンパイルエラーが検出されて表示されます。

- **ColdFusion Builder 拡張機能の使用と開発**：ColdFusion Builder 拡張機能を管理するには、拡張機能ビューを使用します。

コードを生成したり、ユーザーインターフェイスを設計したり、データベースに対する基本的な CRUD（作成、読み込み、更新、削除）操作を実行できます。CFML エディターでのファイルのオープン、開いたファイルへのテキストの挿入などのカスタムアクションを実行する拡張機能も開発できます。拡張機能の作成の詳細については、91 ページの「[ColdFusion Builder 拡張機能](#)」を参照してください。

ColdFusion Builder には、拡張機能の作成とパッケージ化に必要な手順をガイドする ColdFusion Builder Extension Creator が用意されています。詳しくは、116 ページの「[ColdFusion Builder Extension Creator を使用した拡張機能の作成とパッケージ化](#)」を参照してください。

ColdFusion Builder には次の拡張機能が付属しています。

- Adobe CFC Generator
- ActionScript Class Generator

これらの拡張機能のインストールと使用について詳しくは、121 ページの「[拡張機能の使用](#)」を参照してください。

第4章：ColdFusion Builder ワークベンチの基本

ColdFusion Builder は Eclipse ベースの開発環境であり、これを利用すると Adobe ColdFusion アプリケーションを開発して ColdFusion サーバーで実行できます。ColdFusion Builder では、コーディングツールやデバッグツールを利用して ColdFusion アプリケーションを開発できます。

ワークベンチについて

ColdFusion Builder ワークベンチは、Adobe ColdFusion アプリケーションの開発に必要な機能をすべて備えた環境です。ColdFusion Builder IDE の基本機能の多くは、Eclipse から受け継いでいます。ColdFusion Builder プラグインをインストールすると、ColdFusion アプリケーションの作成に必要な機能を追加できます。また、これらのプラグインには、IDE のユーザーインターフェイスを変更するためのツールもあり、アプリケーションのビルドをサポートする主要機能が提供されます。

ワークベンチ ワークベンチは ColdFusion Builder の開発環境です。ワークベンチには、パースペクティブ、エディターおよびビューという 3 つの主要要素があります。アプリケーション開発プロセスの各種段階で 3 つの要素を様々な組み合わせで使用します。ワークベンチには、アプリケーションの開発に必要なツールがすべて用意されています。

パースペクティブ パースペクティブは、ワークベンチにあるビュー、エディター、メニュー、およびツールバーのグループです。これは、基本的には特定の種類の作業に役立つ特別な作業環境です。例えば、ColdFusion Builder には 2 つのパースペクティブがあります。アプリケーションの開発には ColdFusion Builder 開発パースペクティブを使用し、アプリケーションのデバッグには ColdFusion デバッグパースペクティブを使用します。

パースペクティブの詳細については、22 ページの「[ColdFusion Builder パースペクティブについて](#)」を参照してください。

エディター エディターでは様々な種類のファイルを編集できます。使用できるエディターは、インストールされている Eclipse プラグインの数と種類によって異なります。ColdFusion Builder には、CFML、HTML、JavaScript、および CSS (Cascading Style Sheets : カスケーディングスタイルシート) のコードを記述するためのエディターが含まれています。ColdFusion Builder でのコード編集の詳細については、37 ページの「[ColdFusion Builder のエディター](#)」および 39 ページの「[ColdFusion Builder でのコード編集](#)」を参照してください。

ビュー ビューは一般的にエディターをサポートします。例えば、CFML を編集するときには、ColdFusion Builder 開発パースペクティブにアウトラインビューとスニペットビューも表示されます。これらのビューは、ColdFusion アプリケーションの開発をサポートするため、CFML ファイルが編集用に開かれると表示されます。

ビューの中にはワークベンチ自体の主要な機能をサポートするものがあります。例えば、ファイルエクスプローラービューでは、ワークベンチ内のファイルおよびフォルダーを管理できます。RDS データビューおよび RDS ファイルビューには、リモートサーバーとローカルサーバーのデータソース、ファイル、およびディレクトリが表示されます。

ビューという用語は、Adobe Dreamweaver® やその他の Adobe 開発ツールで使用されるパネルという用語と同義です。

ColdFusion Builder 開発パースペクティブのビューの詳細については、23 ページの「[ColdFusion Builder 開発パースペクティブ](#)」を参照してください。

ColdFusion デバッグパースペクティブのビューの詳細については、31 ページの「[ColdFusion デバッグパースペクティブ](#)」を参照してください。

ワークスペース ワークベンチと混同しないようにしてください。ワークスペースとは、ファイルシステム内の特定の領域です。ワークスペースには、アプリケーションプロジェクトの構成要素となるリソース (ファイルとフォルダー) が格納されます。1 つのワークスペースに複数のプロジェクトを格納することもできます。一度に使用できるワークスペースは 1 つだ

けですが、ColdFusion Builder を起動するたびに異なるワークスペースを選択できます。詳しくは、74 ページの「プロジェクトの管理」を参照してください。

リソース リソースという一般用語は、ワークスペース内のプロジェクトに含まれるファイルとフォルダーを意味します。詳細については、76 ページの「ColdFusion ページ、インターフェイス、およびコンポーネントの追加」および 77 ページの「その他のファイルの追加」を参照してください。

プロジェクト アプリケーションを構成するリソースはすべてプロジェクト内にあります。最初にプロジェクトを作成しないと、ColdFusion Builder でアプリケーションを構築できません。詳しくは、74 ページの「プロジェクトの管理」を参照してください。

ワークベンチのメニューとツールバー

ワークベンチのコマンドにアクセスするには、メニューバー、右クリックのコンテキストメニュー、ツールバー、およびキーボードショートカットを使用します。

ワークベンチのメニュー

ColdFusion Builder ワークベンチには次のメインメニューがあります。

ファイルメニュー

ファイルメニューでは、ワークベンチリソースの作成、保存、エディターの終了、印刷、インポート、エクスポート、およびワークベンチの終了を行います。

メニューコマンド	説明
新規	リソースを作成します。ColdFusion コンポーネント、インターフェイス、またはページを作成する前に、それらのリソースを保存するためのプロジェクトを作成しておく必要があります。
ファイルを開く	編集するファイルを開きます。ワークスペースに含まれないファイルを開くこともできます。
閉じる	アクティブなエディターを閉じます。ファイルを閉じる前に、変更を保存するかどうかを確認するメッセージが表示されます。
すべて閉じる	開いているエディターをすべて閉じます。ファイルを閉じる前に、変更を保存するかどうかを確認するメッセージが表示されます。
保管	アクティブなエディターの内容を保存します。
別名保存	アクティブなエディターの内容を別のファイル名で他の場所に保存できます。
すべて保管	開いているすべてのエディターの内容を保存します。
前回保存した状態に戻す	アクティブなエディターの内容を、前回保存した内容に戻します。
移動	現在選択されているリソースを別のプロジェクトに移動します。
名前変更	選択されているリソースの名前を変更します。
更新	リソースにファイルシステムの現在の内容を反映します。
行区切り文字の変換	開発プラットフォームまたはデプロイ先プラットフォームのオペレーティングシステムに合わせて行区切り文字を変換します。 <ul style="list-style-type: none">Windows (デフォルト)UNIXMacOS 9

メニューコマンド	説明
印刷	アクティブなエディターの内容を印刷します。
ワークスペースの切り替え	別のワークスペースに切り替えるためのワークスペースランチャーを開きます。ワークベンチは、別のワークスペースに切り替えるたびに再起動されます。
再開	ColdFusion Builder を再起動します。
インポート	インポートウィザードを使用して、ワークベンチにリソースをインポートします。
エクスポート	エクスポートウィザードを使用して、ワークベンチからリソースをエクスポートします。
プロパティ	選択されているリソースのプロパティダイアログボックスを表示します。プロパティダイアログボックスには、リソースのパスと最終変更日が表示されます。また、プロジェクトのリソースが親のエンコードや行区切り文字を継承しているか、特定の値に設定されているかに関する情報も表示されます。
最近使ったファイルのリスト	最近ワークベンチでアクセスしたファイルのリストです。これらのファイルを開くには、ファイルメニューでファイル名をクリックします。このリストに表示するファイルの数は、環境設定ダイアログボックスで設定できます。
終了	開いているリソースを閉じて、ワークベンチを終了します。

ナビゲートメニュー

ナビゲートメニューでは、ワークベンチ内のリソースを探して、該当する場所に移動できます。

メニューコマンド	説明
次へジャンプ	選択したリソースがルートに来るように、リソースの階層内を移動します。つまり、プロジェクト内のフォルダーを選択してから「次へジャンプ」を選択すると、選択したフォルダー（およびフォルダー内のアーティファクト）のみがナビゲータービューに表示されます。 このコマンドは、複雑な階層構造を持つ大規模なプロジェクト内を移動するときに便利です。
ジャンプ	特定のリソースにジャンプするか、前に表示されていた階層に戻ります。 例えば、特定のリソースを選択して「次へジャンプ」コマンドを選択した後に、ジャンプ/戻るを選択するとします。「戻る」コマンドを実行すると、「次へジャンプ」コマンドの実行前と同一の階層が表示されます。
宣言を開く	選択したコード要素の宣言を開きます。
クイックアウトライン	アウトラインビューの簡易版を表示します。ページのコード構造が階層状になってポップアップメニューに表示されます。
リソースを開く	リソースを開くダイアログボックスを使用して、ワークスペース内のリソースを選択し、エディターで開きます。
表示	別のビューで現在選択されているリソースを探して選択できます。エディターがアクティブになっている場合は、これらのコマンドを使用して、別のビューで現在編集されているリソースを選択できます。
次へ	アクティブなビューのリストまたはテーブル内で次の項目に移動します。
前へ	アクティブなビューのリストまたはテーブル内で前の項目に移動します。
最後の編集位置	アクティブなエディター内で最後に編集した位置に移動します。
指定行へジャンプ	アクティブなエディター内の特定の行に移動します。
戻る	以前に開いたファイルにエディターのフォーカスを移動します。
進む	以前のファイルからエディターのフォーカスを戻します。

プロジェクトメニュー

プロジェクトメニューでは、ワークベンチのプロジェクトを管理します。

メニューコマンド	説明
プロジェクトを開く	現在選択されているプロジェクトを開きます。このコマンドは、現在選択されているプロジェクトが閉じられている場合にのみ使用できます。
プロジェクトを閉じる	現在開いているプロジェクトを閉じます。
プロパティ	プロジェクトのプロパティダイアログボックスを表示します。

ウィンドウメニュー

ウィンドウメニューでは、ワークベンチの各種ビュー、パースペクティブ、およびアクションの表示、非表示、管理を行います。

メニューコマンド	説明
新規ウィンドウ	現在開いているパースペクティブと同じパースペクティブで、新しいワークベンチウィンドウを開きます。
新規エディター	現在アクティブなエディターと同じタイプのエディターを開きます。
パースペクティブを開く	ワークベンチ内に新しいパースペクティブを開きます。
ビューの表示	ワークベンチに表示するビューを選択できます。
パースペクティブのカスタマイズ	現在選択されているパースペクティブをカスタマイズできます。
パースペクティブの別名保管	現在選択されているパースペクティブを保存し、カスタマイズした独自のパースペクティブを作成します。
パースペクティブのリセット	カスタマイズしたパースペクティブを元の状態に戻します。
パースペクティブを閉じる	現在開いているパースペクティブを閉じます。
すべてのパースペクティブを閉じる	ワークベンチで開いているパースペクティブをすべて閉じます。
ナビゲーション	ワークベンチのビュー、パースペクティブ、エディターを切り替えるためのショートカットがあります。
環境設定	ワークベンチを使用するための設定を指定できます。ワークベンチ本体だけでなく、ワークベンチに含まれるビューやエディターの外観に関連する様々な設定項目があります。

ツールバー

ワークベンチツールバー

ワークベンチツールバーには、頻繁に使用される重要なコマンドのボタンがあります。これらのコマンドは、ColdFusion Builder の各種メニューからも使用できます。



ワークベンチツールバー

ワークベンチツールバーには、次のボタンが（左から右の順に）表示されます。

ボタン/コマンド	説明
新規	作成できるプロジェクトとドキュメントのすべてのタイプを表示するポップアップメニューが表示されます。
保存	エディターで開かれていて現在選択されているドキュメントを保存します。
印刷	エディターで開かれていて現在選択されているドキュメントを印刷します。

ボタン / コマンド	説明
新規 ColdFusion プロジェクト	ColdFusion プロジェクトを作成するためのウィザードを開きます。
CFC を開く	ナビゲータービューに表示されているプロジェクトで使用できるすべての CFC を表示します。
デバッグ	プロジェクトで現在開かれているファイルを使用してデバッグセッションを開始します。付帯しているポップアップメニューから、プロジェクト内の他のアプリケーションファイルを選択することもできます。
実行	プロジェクトで現在開かれているファイルを実行します。付帯しているポップアップメニューから、プロジェクト内の他のアプリケーションファイルを選択することもできます。
外部ツール	カスタム起動設定を選択します。
検索	プロジェクト内のファイルのテキスト文字列およびファイル名のパターンを検索します。
次の注釈	先の位置にあるコード注釈を選択して移動できます。
前の注釈	前の位置にあるコード注釈を選択して移動できます。
最後の編集位置	現在開いているファイルで最後に編集したコード要素に移動します。
戻る	以前に開いたファイルにエディターのフォーカスを移動します。
進む	以前のファイルからエディターのフォーカスを戻します。

CFML エディターのツールバー

CFML エディターのユーザーインターフェイスには、使用頻度の高いコマンドをすばやく実行するためのボタンがあります。これらのコマンドは、ColdFusion Builder の各種メニューからも使用できます。このツールバーには使用頻度の高いコード要素へのショートカットとして機能するボタンもあり、それらのボタンを使用できるかどうかは CFML エディターで入力したコードの内容に依存します。



CFML エディターのツールバー

ボタン / コマンド	説明
タグエディターを開く	タグエディターを開きます。 タグエディターについて詳しくは、59 ページの「 タグエディター 」を参照してください。
cfabort の挿入	cfabort タグを挿入します。 このコマンドを使用できるかどうかはスクリプトコードの内容に依存します。
cfdump の挿入	cfdump タグを挿入します。このコマンドを使用できるかどうかはスクリプトコードの内容に依存します。
cfoutput で囲む	選択したコードを cfoutput タグで囲みます。 このコマンドを使用できるかどうかは、CFML エディターで入力したスクリプトコードの内容に依存します。
cfset の挿入	cfset タグを挿入します。
cfscript ブロックの挿入	cfscript コードブロックを挿入します。
CF Try/Catch で囲む	内容に応じて、選択したコードを cftry または cfcatch タグで囲みます。 このコマンドを使用できるかどうかはスクリプトコードの内容に依存します。
## で囲む	選択したコードを「#」マークで囲みます。
CF コメント <!-- で囲む / 解除	選択したコードをコメント化するか、選択したコードのコメント化を解除します。

ボタン / コマンド	説明
cfscript コメント /* */ で囲む / 解除	選択したコードを「/* */」マークで囲むか、選択したコードから「/* */」マークを外します。
一重引用符で囲む	選択したコードを一重引用符で囲みます。
二重引用符で囲む	選択したコードを二重引用符で囲みます。
大文字へ	選択したコードのテキストを大文字に変更します。
小文字へ	選択したコードのテキストを小文字に変更します。

ColdFusion Builder パースペクティブについて

パースペクティブとは、作業に必要なエディターとビューを1つにまとめたものを指します。ColdFusion Builder には、ColdFusion Builder 開発パースペクティブと ColdFusion Builder デバッグパースペクティブという2つのパースペクティブがあります。

パースペクティブの表示と切り替え

特定のパースペクティブに関連付けられているファイルを開くと、そのパースペクティブが自動的に開きます。つまり、パースペクティブは、実行しようとしているタスクに合わせて自動的に変化します。例えば、ColdFusion プロジェクトを作成する場合は、ワークベンチに開発パースペクティブが表示されます。同様に、デバッグセッションを始めるときには、ColdFusion Builder がデバッグパースペクティブに切り替わります。

デフォルトでは、同じウィンドウ内に複数のパースペクティブが開きます。パースペクティブを新しいウィンドウで開くには、次のようにします。


- 1 ウィンドウ／環境設定を選択します。
- 2 ツリービュー構造で、一般／パースペクティブを選択します。
- 3 「新規パースペクティブを開く方法」で、「同じウィンドウ」を選択します。

手でパースペクティブを切り替えるには、次のいずれかの操作を行います。

- メインメニューでウィンドウ／パースペクティブを開く／その他を選択します。
- ワークベンチのメインツールバーにあるパースペクティブバーを使用します。

Eclipse には定義済みのパースペクティブが数多く用意されています。したがって、プラグイン設定の ColdFusion Builder を使用している場合は、他のパースペクティブも使用できます。

Eclipse パースペクティブを表示するには、次のいずれかの操作を行います。

- メインメニューでウィンドウ／パースペクティブを開く／その他を選択します。
- ワークベンチのメインツールバーにあるパースペクティブバーで  をクリックし、「その他」を選択します。

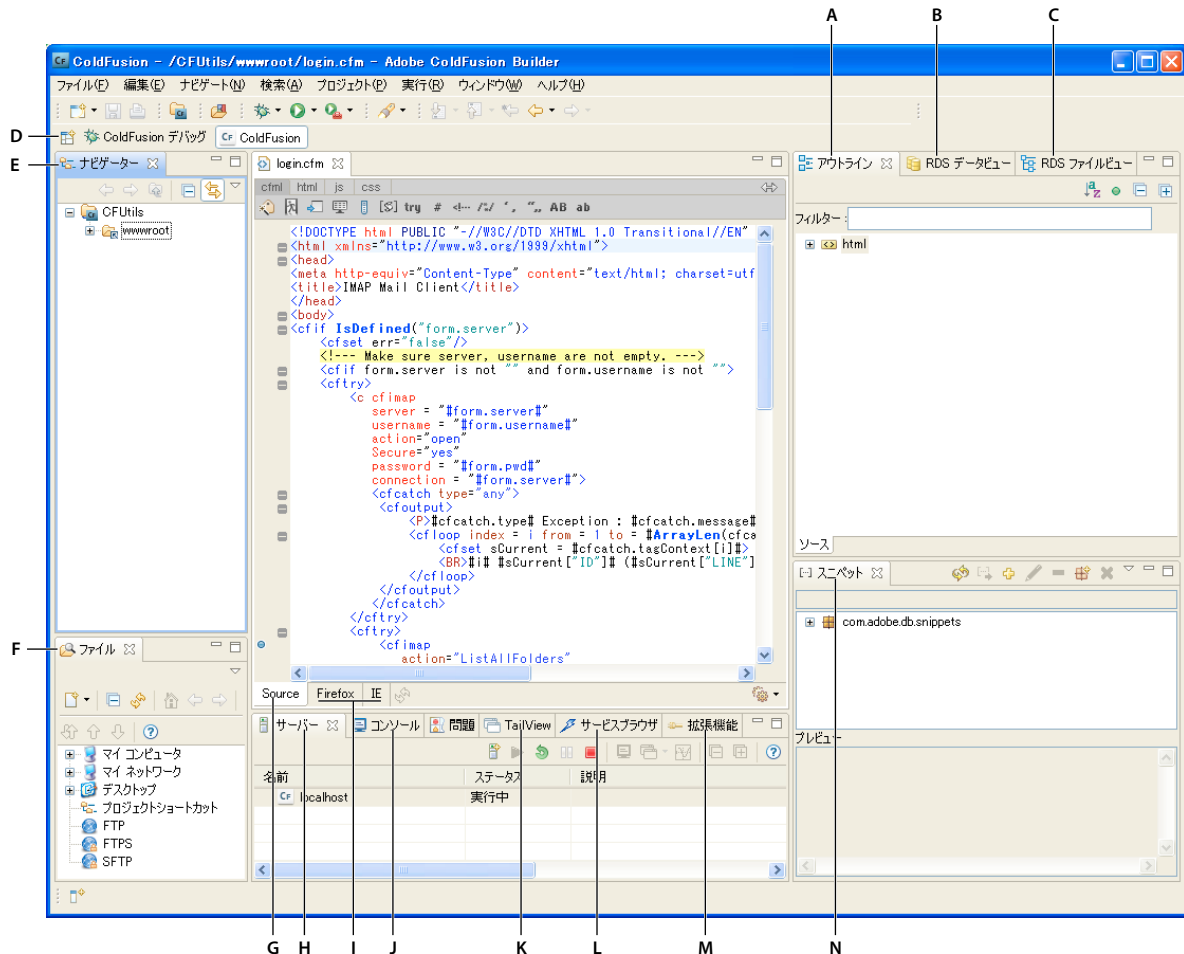
デフォルトのパースペクティブの設定

- 1 ウィンドウ／環境設定を選択します。
- 2 ツリービュー構造で、一般／パースペクティブを選択します。
- 3 「選択可能なパースペクティブ」でデフォルトとして設定するパースペクティブを選択し、「デフォルトにする」をクリックします。
- 4 「OK」をクリックします。

デフォルトに設定したパースペクティブの名前の後には「(デフォルト)」と表示されます。

ColdFusion Builder 開発パースペクティブ

ColdFusion Builder 開発パースペクティブは、ColdFusion Builder で ColdFusion アプリケーションを作成、編集、設定、または実行するときに使用します。さらに、ローカルやリモートの ColdFusion サーバーの設定と管理もできます。



ColdFusion パースペクティブ

A. アウトラインビュー B. RDS データビュー C. RDS ファイルビュー D. パースペクティブバー E. ナビゲータービュー F. ファイルビュー G. ソースビュー H. サーバービュー I. デフォルトブラウザビュー J. コンソールビュー K. TailView ビュー L. サービスブラウザー M. 拡張機能ビュー N. スニペットビュー

ColdFusion Builder 開発パースペクティブには次のビューがあります。

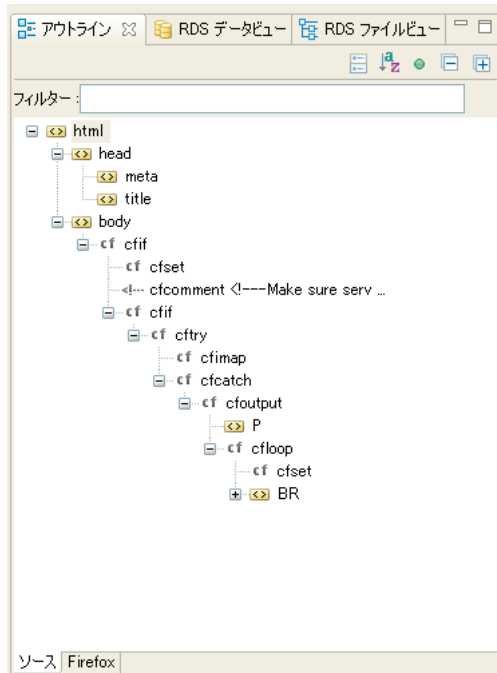
アウトラインビュー

アウトラインビューには、エディターで現在開いているファイル内の要素の階層が表示されます。例えば、CFC ファイル内の関数や HTML ファイル内のタグが表示されます。

アウトラインビューでは、CFML、HTML、JavaScript、および CSS ページの構造の調査とナビゲートを行います。複数のコード要素を含むページでは、アウトラインビューを使用して、ページのコード構造を階層形式で表示できます。

アウトラインビューの要素は、アルファベット順、またはページ内の要素定義の順に並べ替えることができます。このビューでは、すべてのコード要素を表示するか、特定のコード要素のみを表示するかを選択できます。

- アウトラインビューには、CFML コードの構造が表示されます。例えば、アウトラインビューの各アイテムは CFML タグを表す場合があります。



アウトラインビュー

- コード全体をスクロールせずに、エディター内の要素に直接移動するには、アウトラインビューで要素をダブルクリックします。アウトラインビューでアイテムを選択すると、そのアイテムがエディター内で強調表示されるので、コードのナビゲートがより簡単になります。
- アウトラインビュー内のタグや要素を検索するには、フィルターを使用します。フィルターフィールドに、検索するタグまたは要素の名前を入力します。一致する文字列がアウトラインビューに表示されます。

アウトラインビューで表示する CFML タグを指定するには、ColdFusion / エディタープロファイル / エディター / アウトラインを選択します。必要に応じて、タグを選択、追加、または削除します。

RDS ファイルビューおよび RDS データビュー

Remote Data Services (RDS) を使用するには、ColdFusion サーバーのインストール時に RDS を有効にします。ColdFusion Builder には、リモートサーバー上のファイルやデータソースにアクセスするためのビューが用意されています。

RDS ファイルビューには、リモートサーバーおよびローカルサーバーのファイルとディレクトリが表示されます。RDS データビューには、リモートサーバーで設定されたデータソースが表示されます。

ColdFusion Builder で追加した ColdFusion サーバーインスタンスは自動的に RDS ファイルビューおよび RDS データビューからアクセス可能になります。

RDS ファイルビューでのリモートサーバーの設定

RDS ファイルビューを使用してリモートサーバーを手動で設定し、そのサーバー上のファイルにアクセスするには、次のようになります。

- 1 ColdFusion Builder パースペクティブの右上隅にある RDS ファイルビューをクリックします。
- 2 RDS ファイルビュー内で右クリックし、RDS 構成を選択します。
- 3 「新規」をクリックします。
- 4 リモートサーバーの情報（ホスト名、ポート番号、ユーザー名、パスワードなど）を指定します。
- 5 「パスワードを要求」を選択します。
- 6 「接続のテスト」をクリックして、RDS 構成が正しいことを確認します。「OK」をクリックします。
- 7 RDS ファイルビューからリモートサーバーを選択します。
- 8 リモートサーバー上のファイルとディレクトリを表示するためのパスワードを指定します。

RDS データビューでのリモートサーバーの設定


RDS データビューを使用してリモートサーバーを設定し、リモートサーバー上のデータソースにアクセスするには、次のようになります。

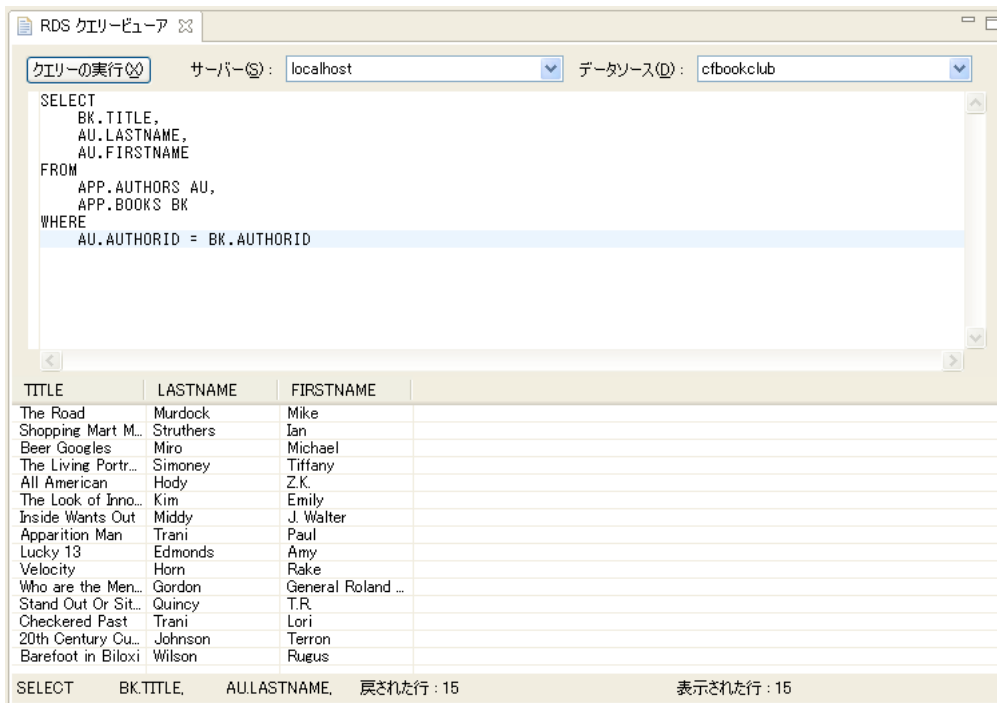
- 1 RDS データビューをクリックします。
- 2 RDS データビュー内で右クリックし、「RDS 構成」を選択します。
- 3 「新規」をクリックします。
- 4 サーバーの情報（ホスト名、ポート番号、リモートサーバーを使用するためのユーザー名、パスワードなど）を指定します。
- 5 「パスワードを要求」を選択します。
- 6 「接続のテスト」をクリックして、RDS 構成が正しいことを確認します。「OK」をクリックします。
- 7 新しく設定したサーバーを RDS データビューで展開して、データソースを表示します。

RDS クエリビューア

RDS クエリビューアでは、選択したデータソースに対するクエリを作成して実行できます。

RDS クエリビューアは RDS データビューツールバーから起動できます。RDS クエリビューアを使用してクエリを作成または実行するには、次のようになります。

- 1 RDS データビューで  をクリックして、RDS クエリビューア領域を開きます。
- 2 空白の領域にクエリを入力します。
RDS データビュービューから RDS クエリビューアにテーブルと列をドラッグしてクエリを作成することもできます。
- 3 ドロップダウンリストからサーバーとデータソースを選択します。
- 4 「クエリの実行」をクリックします。RDS クエリビューア領域にクエリの結果が表示されます。



RDS クエリビューア

RDS データビュービューのテーブルを右クリックして、「テーブルの内容の表示」を選択します。RDS クエリビューアでは、選択したテーブルにあるすべてのレコードを表示できます。

ナビゲータービュー

ナビゲータービューには ColdFusion プロジェクトやその他のプロジェクトが表示され、プロジェクトのファイルやフォルダーの間をナビゲートできます。

ファイルビュー

ファイルビューには ColdFusion Builder がインストールされているローカルコンピューター上のファイルとディレクトリが表示され、新しいファイルを作成できます。

ColdFusion Builder の外部でファイルを編集した場合は、そのファイルを ColdFusion Builder で自動的に更新できます。これを行うには、ウィンドウ/環境設定/一般/ワークスペースを選択し、「自動的に更新」を選択します。

また、FTP サイトを追加したり、FTP サーバー上のファイルを参照することもできます。ColdFusion プロジェクトを FTP 接続に関連付ければ、ファイルのアップロード、ダウンロード、および同期を簡単に行えます。詳細については、81 ページの「FTP および Secure FTP 接続によるプロジェクトのデプロイ」を参照してください。

サーバービュー

ColdFusion サーバーの詳細（サーバー名、ステータス、説明、サーバータイプ、サーバーホスト、ポートなど）が表示されます。サーバービューでは次の作業を行えます。

- ColdFusion サーバーの追加と削除
- サーバーの起動と停止

- サーバーモニターの起動
- ColdFusion Server Administrator の起動
- サーバーログを開く

詳細については、65 ページの「[サーバーの管理](#)」を参照してください。

ソースビューとデフォルトブラウザービュー

ソースビューには、ファイルのソースコードが表示されます。デフォルトブラウザービューには、HTML または CFML ページの出力が表示されます。

コンソールビュー

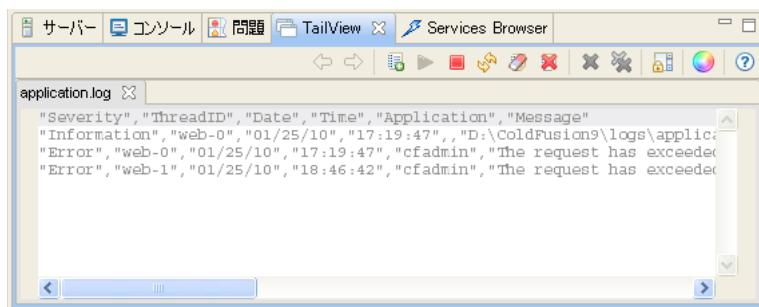
コンソールビューには、ColdFusion サーバーの状態と、ColdFusion サーバーからのメッセージが表示されます。

TailView ビュー

TailView ビューには、サーバーログとワークスペースログが表示され、ログを簡単にナビゲートできます。ログを自動的に表示するには、RDS を構成して有効にする必要があります。

サーバーを選択した後は、Server Manager のボタンバーにあるショートカットからログを開くことができます。ログを選択すると、TailView ビューにログの内容が表示されます。デフォルトでは、TailView ビューに例外ログが表示されます。サーバービューを使用している場合は、ビュー内を右クリックして「ログを開く」を選択すると、例外ログが開きます。

ログを手動で開くには、TailView ビューで右クリックして、「追加」を選択します。コンピューターのファイルシステムにあるログファイルを参照して選択します。



TailView ビュー

TailView ビューのツールバーには、(左から右の順に) 次のボタンがあります。これらのボタンを使用すると、複数のログを個別のタブで管理できます。



TailView ツールバー

ボタン/コマンド	説明
左に移動	サーバーログまたはワークスペースログを左に移動します。
右に移動	サーバーログまたはワークスペースログを右に移動します。
ログを追加します	コンピューターのファイルシステムでログファイルを開きます。
ログファイルの監視を開始します	ログファイルの監視を開始します。

ボタン / コマンド	説明
ログファイルの監視を停止します	ログファイルの監視を停止します。
ログファイルを再ロードします	ログファイルを再ロードします。
表示されているログファイルをクリアします	TailView ビューに表示されているログファイルを削除します。ただし、コンピューターのファイルシステムからは削除されません。
ディスク上のログファイルを消去	コンピューターのファイルシステムからログファイルを削除します。
ログタブを削除します	コンピューターのファイルシステムからログファイルを削除します。
すべてのログタブを削除します	TailView ビューからすべてのログタブを削除します。
スクロールロック	コンソールビューがスクロールしないように固定します。
色の設定	TailView ビューにログファイルを表示するときの色を設定します。

サービスブラウザービュー

サービスブラウザービューには、Server Manager で追加されたサーバーが一覧表示され、CFC (ColdFusion Component : ColdFusion コンポーネント)、および Web サービスとそのメソッドを参照できます。

ColdFusion コンポーネントの表示

サービスブラウザービューには、次のコンポーネントが一覧表示されます。

- ColdFusion コンポーネントブラウザーに一覧表示されるコンポーネント . ColdFusion コンポーネントブラウザーの場所は、`cf_root\wwwroot\CFIDE\componentutils\componentdoc.cfm` です。
- ColdFusion Administrator のマッピングページで指定されたディレクトリに配置されているコンポーネント
- ColdFusion Administrator のカスタムタグのパスページで指定されたディレクトリに配置されているコンポーネント

サービスブラウザービューのツールバーには、(左から右の順に) 次のボタンがあります。これらのボタンを使用すると、CFC 内のメソッドをアクセスタイプ (リモート、パブリック、パッケージ、プライベート) に基づいてフィルタリングできます。






サービスブラウザーツールバー

ボタン / コマンド	説明
システム CFC の表示	システム CFC の表示と非表示を切り替えるトグルボタン。システム CFC は、サーバーの Web ルート内に保存されます。
リモートの表示	アクセスタイプがリモートのメソッドの表示と非表示を切り替えるトグルボタン。
Public の表示	アクセスタイプがパブリックのメソッドの表示と非表示を切り替えるトグルボタン。
パッケージの表示	アクセスタイプがパッケージのメソッドの表示と非表示を切り替えるトグルボタン。
Private の表示	アクセスタイプがプライベートのメソッドの表示と非表示を切り替えるトグルボタン。

Web サービスを管理する

サービスブラウザービューでは、Web サービスの一覧に対して WSDL URL の追加や削除を行うことで、Web サービスの一覧を管理できます。

- Web サービスの一覧を表示するには、サービスブラウザービューで  をクリックします。

- 新しい Web サービスを一覧に追加するには、 をクリックし、有効な WSDL URL を入力して「OK」をクリックします。
- Web サービスを削除するには、該当する Web サービスを選択して  をクリックします。

ColdFusion ファイルを編集する際には、サービスブラウザービューを使用して、Web サービスの実行や Web サービスオブジェクトの作成を行う CFML コードを生成できます。同様に、ActionScript ファイルを編集する際にも、サービスブラウザーを使用して ActionScript を生成できます。

Web サービスの実行または作成を行うには、次のようにします。

- 1 コードを挿入する位置にポインターを配置します。
- 2 Web サービスの一覧を表示します。
- 3 Web サービスまたは Web サービス内のメソッドを強調表示し、右クリックして次のいずれかを選択します。
 - Web サービスを挿入する場合は「CFOBJECT の挿入」
 - Web サービスを作成する場合は「CreateObject の挿入」

拡張機能ビュー

拡張機能ビューでは次の作業を行えます。

- 115 ページの「[拡張機能のインストールとアンインストール](#)」
- 116 ページの「[拡張機能のインポートとリロード](#)」

拡張機能ビューには、拡張機能名やインストールされている拡張機能の説明などの詳細も表示されます。

拡張機能ビューのツールバーには（左から右の順に）次のボタンがあります。これらのボタンを使用すると、拡張機能ビューで拡張機能を管理できます。



拡張機能ビューのツールバー

ボタン/コマンド	説明
アーカイブファイルから拡張機能をインストール	インストールするアーカイブファイルを選択して、拡張機能のインストールウィザードを開きます。
フォルダーから拡張機能をインポート	拡張機能ファイルを含むフォルダーを選択します。
選択した拡張機能をアンインストール	選択した拡張機能をアンインストールします。
選択した拡張機能の詳細を表示	選択した拡張機能の詳細を表示します。

スニペットビュー

コードスニペットとは、ファイルに挿入して何度でも再利用できるテキストであり、これを作成することにより、同じテキストを毎回入力する必要がなくなります。コードスニペット、変数、およびプレーンテキストを作成して保存するには、ColdFusion スニペットビューを使用します。コードスニペットは、既存のドキュメントまたは新規のドキュメントに追加できます。


スニペットビューのツールバーには（左から右の順に）次のボタンがあります。これらのボタンを使用すると、スニペットビューでコードスニペットを管理できます。




コードスニペットツールバー

ボタン/コマンド	説明
スニペットビューの更新	スニペットビューの表示を更新します。
選択されているスニペットをドキュメントに挿入	選択されているスニペットをドキュメントに挿入します。
新規スニペットの作成	コードスニペットを作成します。
選択されているスニペットの編集	選択されているコードスニペットを編集できます。
選択されているスニペットの削除	選択されているコードスニペットを削除します。
スニペットパッケージの新規作成	コードスニペットを追加するためのスニペットパッケージを作成します。
選択されているスニペットパッケージの削除	選択されているスニペットパッケージを削除します (パッケージが空の場合のみ)。スニペットを含んでいるスニペットパッケージを削除することはできません。

コードスニペットの作成

- 1 コードスニペット領域で  をクリックします。
- 2 新規スニペットダイアログボックスで、スニペット名、コードの開始ブロック、および終了ブロックを指定します。
- 3 「OK」をクリックするとコードスニペットが作成されます。

コードスニペットの挿入

- 1 挿入ポイントにカーソルを移動します。
- 2 コードスニペット領域でコードスニペットを選択します。
- 3 コードスニペットツールバーで  をクリックします。

エディターでコードを選択してスニペットを挿入すると、選択したコードの開始ブロックと終了ブロックの間にスニペットがラップされます。エディターでコードを選択せずにスニペットを挿入すると、現在のキャレットの位置にスニペットが挿入されます。キャレットとは、次の文字が出現する位置を示す CFML エディター内のマーカーです。

コードスニペット領域では既存のコードスニペットやコードスニペットパッケージ全体を編集したり削除したりすることもできます。

トリガーテキストを使用したスニペットの挿入

トリガーテキストを使用して CFML エディターにスニペットを挿入することもできます。例えば、スニペットを作成し、そのスニペットに「abc」というトリガーテキストを設定します。このスニペットをエディターに挿入するには、「abc」と入力し、Ctrl + J キー (Windows の場合) または Command + J キー (Mac の場合) を押します。テキスト abc がスニペットに置換されます。

スニペットを保存するパスの指定

スニペットは、スニペットのディレクトリに XML ファイルとして保存されます。デフォルトでは、スニペットはプロジェクトワークスペースの `workspace/.metadata/snippets` ディレクトリに格納されます。別のディレクトリを指定するには、次のようにします。

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 環境設定ダイアログボックスのツリービュー構造で、ColdFusion / スニペットを選択します。
- 3 スニペットを保存するディレクトリのパスを指定します。

システム定義スニペット変数

スニペット変数の大文字と小文字は区別されます。次のシステム定義変数が用意されています。

変数	説明
\$\$ {DATE}	4桁の年号を含むシステム日付。例：12/01/2009
\$\$ {TIME}	12時間表記のシステム時刻。例：03:15:05 PM
\$\$ {DATETIME}	システム日時。例：12/01/2009 15:15:05 PM
\$\$ {DAYOFWEEK}	完全な曜日名。例：Monday
\$\$ {CURRENTFILE}	現在開いているファイル名。例：application.cfm
\$\$ {CURRENTFOLDER}	現在開いているフォルダーの完全修飾パス。例：C:\workspace\myproject
\$\$ {CURRENTPATH}	現在開いているファイルの完全修飾パス。例：C:\workspace\myproject\application.cfm
\$\$ {CURRENTPRJPATH}	現在開いているプロジェクトのプロジェクト名。例：myproject
\$\$ {USERNAME}	現在のユーザーの名前。
\$\$ {MONTHNUMBER}	月 (01 ~ 12)。
\$\$ {DAYOFMONTH}	日 (01 ~ 31)。
\$\$ {DAYOFWEEKNUMBER}	1 ~ 7の数値で表した曜日。日曜日が週の始まりになります。したがって、月曜日は2になります。
\$\$ {DATETIME24}	24時間表記の日時。例：12/01/2009 15:30:00
\$\$ {YEAR}	4桁の数値で表した現在の年。例：2009
\$\$ {YEAR2DIGIT}	2桁の数値で表した現在の年。例：09

例えば、現在の日付を挿入するには、次のシステム定義変数を使用します。

Date: \$\$ {DATE}

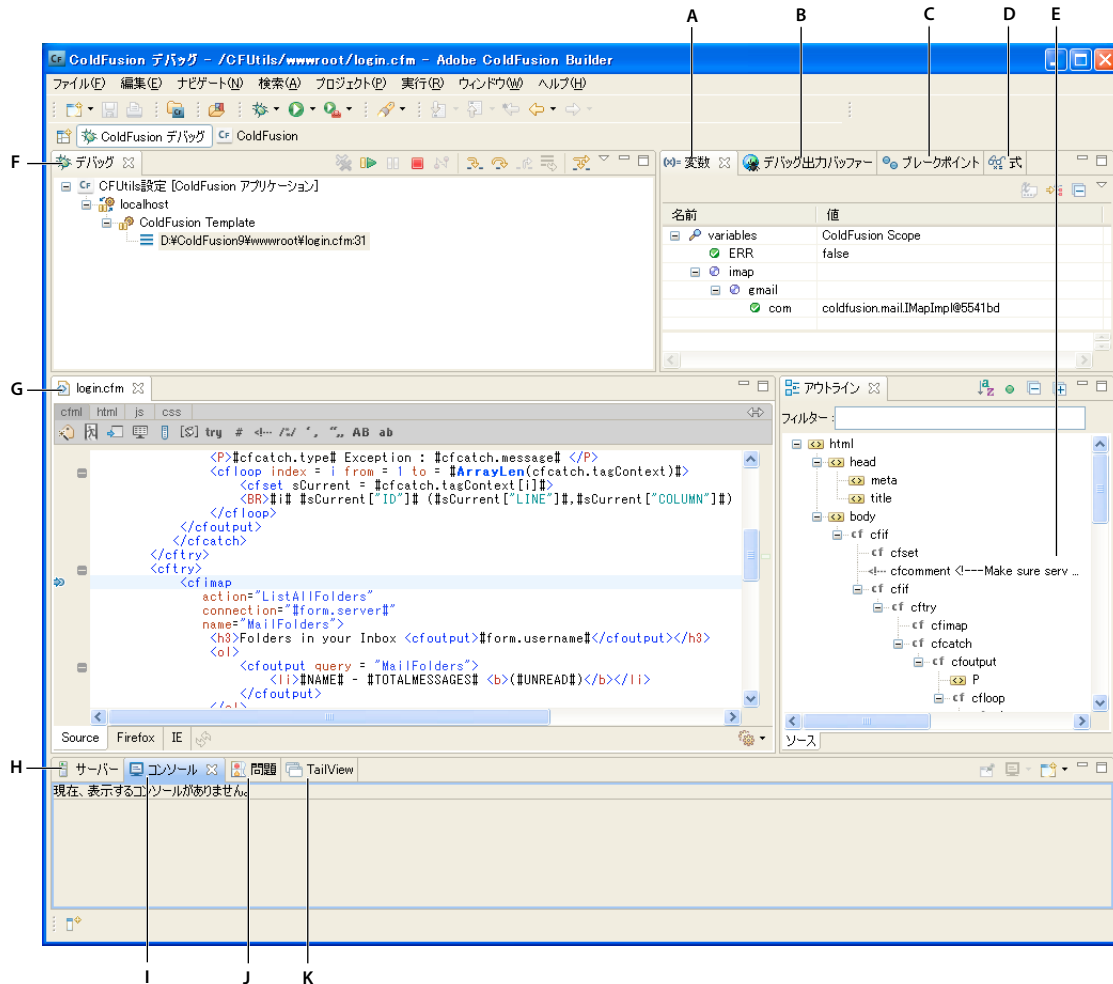
ユーザー定義スニペット変数

ユーザー定義変数では、変数名とデフォルト値の入力を求められます。例えば、次のユーザー定義変数では、作成者名の入力を求められます。

Author name: \$\$ {Author}

ColdFusion デバッグパースペクティブ

ColdFusion Debugging デバッグパースペクティブには、ColdFusion アプリケーションをデバッグするためのツールがあります。また、コードにブレークポイントを追加したり、コード内の関数にステップインまたはステップオーバーしたり、式を評価するための様々なビューが用意されています。エディターとデバッグツールを組み合わせると、修正が必要なコード行を検索して強調表示することができます。



ColdFusion デバッグパースペクティブ

A. 変数ビュー B. デバッグ出力バッファビュー C. ブレークポイントビュー D. 式ビュー E. アウトラインビュー F. デバッグビュー G. 編集ビュー H. サーバービュー I. コンソールビュー J. 問題ビュー K. TailView ビュー

ColdFusion デバッガーパースペクティブには次のビューがあります。

変数ビュー

変数ビューのツールバーには、(左から右の順に) 次のボタンが付いています。これらのボタンを使用すると、現在の変数 (変数スコープも含む) を表示できます。



変数ビューのツールバー

ボタン/コマンド	説明
型名の表示	変数の型を表示します。
論理構造の表示	このコマンドは、ColdFusion Builder ではサポートされていません。
すべてを折りたたむ	ビュー内の情報を折りたたんで、変数の型のみを表示します。

デバッグ出力バッファレビュー

デバッグ出力バッファには次の 2 のペインがあります。

ブラウザー アプリケーションの実行中にブラウザーに出力される内容を表示します。デバッグするページの URL を指定します。

注意：「ホーム」ボタンをクリックすると、環境設定ダイアログボックス（ウィンドウ／環境設定／ColdFusion／デバッグの設定）でホームページの URL として指定した URL が表示されます。デバッガーの設定を編集する方法については、85 ページの「[ColdFusion Builder でのデバッガー設定の指定](#)」を参照してください。

サーバー出力バッファ デバッガーの出力を 2 つのビュー（ソースビューと HTML ビュー）に表示します。

サーバー出力バッファペインには、デバッガーがブレークポイントで一時停止したときのみ出力が表示されます。表示される内容は、一時停止したブレークポイントに到達するまでの出力のみです。

注意：サーバー出力バッファには、外部ブラウザーで実行されたページの出力も表示されます。

ブレークポイントビュー

ブレークポイントビューのツールバーには、（左から右の順に）次のボタンがあります。これらのボタンを使用するとデバッグセッション中にブレークポイントを管理できます。

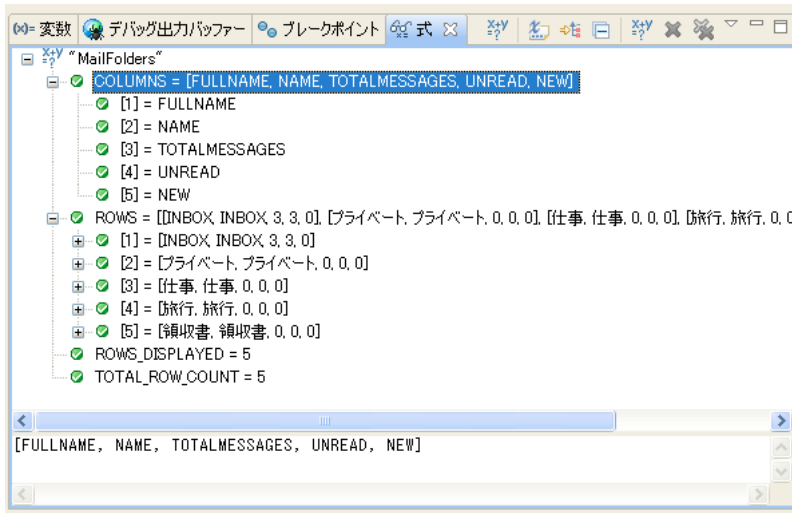


ブレークポイントビューのツールバー

ボタン / コマンド	説明
選択されたブレークポイントを削除	選択されたブレークポイントを削除します。
すべてのブレークポイントを削除	すべてのブレークポイントを削除します。
選択したターゲットでサポートされるブレークポイントを表示	現在デバッグ中のブレークポイントを表示します。
ブレークポイント用のファイルヘジャンプ	選択されたブレークポイントが設定されているファイルに移動します。
すべてのブレークポイントをスキップ	すべてのブレークポイントを無視します。
すべてを展開	ビュー内のブレークポイント情報をすべて展開します。
すべてを折りたたむ	ビュー内のブレークポイント情報をすべて折りたたみます。
デバッグビューにリンク	アプリケーションの実行が停止されたときに、選択されたブレークポイントをデバッグビューで強調表示します。
Java 例外ブレークポイントの追加	選択されたブレークポイントに達したときに返す Java 例外を指定できます。
メニュー	ブレークポイントビューに表示する情報の種類を指定できます。

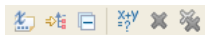
式ビュー

式ビューは、変数や関数を使用して式を作成し、式の検査、評価、監視を行うときに使用します。また、変数ビューで選択した変数の評価と監視を行うこともできます。



式ビュー

式ビューのツールバーには、(左から右の順に) 次のボタンがあります。これらのボタンを使用すると、式の作成、評価、監視を行えます。



式ビューのツールバー

ボタン / コマンド	説明
型名の表示	変数の型を表示します。
論理構造の表示	ビューに論理構造を表示します。
すべてを折りたたむ	ビュー内の式をすべて折りたたみます。
新規監視式を作成	監視式を追加します。
選択された式の削除	選択された変数または監視式を削除します。
すべての式を削除	式ビューに表示されている変数と監視式をすべて削除します。

アウトラインビュー

アウトラインビューには、現在のソースファイルの内容がアウトライン形式で表示されます。詳細については、23 ページの「[アウトラインビュー](#)」を参照してください。

デバッグビュー

デバッグビューには、各デバッグセッションの結果が表示されます。デバッグビューには、ブレークポイントでページの実行が中断したとき、またはコードにステップインまたはステップオーバーしたときのスタックトレースが表示されます。

デバッグツールバーには、(左から右の順に) 次のボタンがあります。



デバッグツールバー

ボタン/コマンド	説明
再開	デバッグセッションを再開します。
中断	デバッグセッションを一時停止します。
終了	デバッグセッションを停止します。
切断	リモートでデバッグしている場合に、選択したデバッグターゲットからデバッガーを切断します。
終了したすべての起動を削除	終了したすべてのデバッグターゲットをビューから消去します。
ステップイン	インクルードされたコード、UDF、CFC を含めて、コードを行単位で実行します。
ステップオーバー	インクルードされたコード、UDF、CFC を除いて、コードを行単位で実行します。
ステップリターン	インクルードされたコード、UDF、CFC の呼び出し元ページに戻ります。
フレームにドロップ	このコマンドは、ColdFusion Builder ではサポートされていません。
ステップ・フィルター/ステップ・デバッグの使用	すべてのステップ関数でステップフィルターを適用します。

編集ビュー

編集ビューにはソースのタブが並んで表示されます。各タブは、開いているソースファイルに対応します。

サーバービュー

サーバービューは、ColdFusion Builder からサーバーの起動、停止、管理を行う場合に使用します。また、サーバービューからは、ColdFusion サーバーモニターと ColdFusion Server Administrator も起動できます。



サーバービューのツールバー

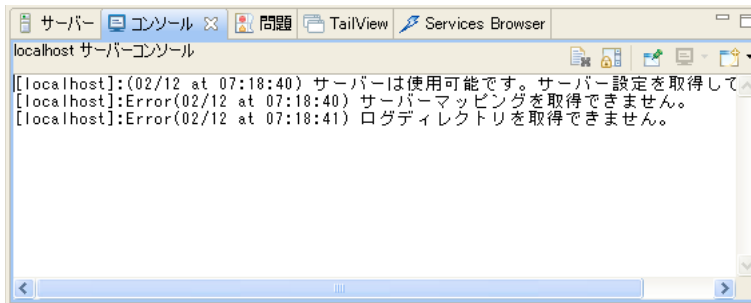
サーバービューのツールバーには、(左から右の順に) 次のボタンがあります。これらのボタンを使用すると、サーバーを管理できます。

ボタン/コマンド	説明
サーバーの追加	サーバーを追加します。
サーバーを起動	サーバーを起動します。
サーバーを再起動	サーバーを停止して起動します。
サーバーを一時停止	サーバーを一時停止します。
サーバーを停止	サーバーを停止します。
コンソール/シェルを開く	新しいコンソールを開き、他のサーバーを選択するためのポップアップメニューを表示します。
ログを開く	TailView ビューでサーバーログを開きます。

サーバーの管理方法とサーバービューの使用方法について詳しくは、72 ページの「[サーバービューの使用](#)」を参照してください。

コンソールビュー

コンソールビューには、CFML コードに含まれる trace ステートメントの出力と、デバッガー自体からのフィードバック（ステータス、警告、エラーなど）が表示されます。



コンソールビュー

コンソールビューのツールバーには、(左から右の順に) 次のボタンがあります。これらのボタンを使用すると、コンソールビューで trace ステートメントを表示して管理できます。

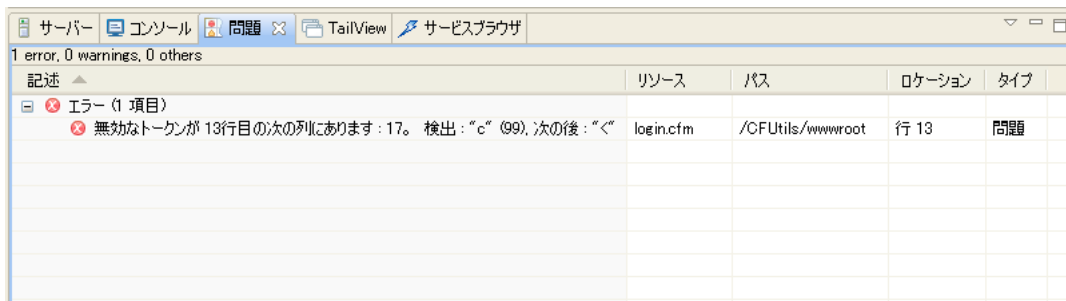


コンソールツールバー

ボタン / コマンド	説明
コンソールのクリア	コンソールビューの内容をすべて消去します。
スクロールロック	コンソールビューがスクロールしないように固定します。
コンソールのピン留め	他のプロセスが選択されたときに、コンソールの内容が更新されないようにします。
選択されたコンソールの表示	選択されたコンソールを表示します。
コンソールを開く	新しいコンソールを開き、他のコンソールビューを選択するためのポップアップメニューを表示します。

問題ビュー

コードを入力するときに、ColdFusion Builder コンパイラーによってシンタックスエラーや他のコンパイルエラーが検出されると、問題ビューにそれらのエラーが表示されます。問題ビューには、ColdFusion Builder で開かれている CFML ファイルに関連するエラーのみが表示されます。



問題ビュー

TailView ビュー

TailView ビューには、サーバーログが表示されます。詳細については、27 ページの「[TailView ビュー](#)」を参照してください。

ワークベンチへのビューの追加

ワークベンチには、ColdFusion Builder のデフォルトの開発パースペクティブとデバッグパースペクティブに関連付けられているビューの他にも、様々なビューを追加できます。これらのビューをワークベンチに追加すると、アプリケーション開発プロセスの効率化を図ることができます。

❖ ワークベンチにビューを追加するには、ウィンドウ/ビューの表示/その他/一般を選択します。

これらオプションのビューはタイプで分類され、ワークベンチの異なる機能または特定の Eclipse プラグインに関連付けられます。

タスク、ブックマーク、検索などのワークベンチビューは、ColdFusion Builder でのアプリケーション開発に役立つビューです。

ColdFusion Builder のエディター

エディターはリソースタイプに関連付けられているため、ワークベンチでリソースを開くと、該当するエディターが開かれます。ワークベンチは、アプリケーションを開発するためのドキュメント中心型（およびプロジェクト中心型）の環境です。

CFML エディターは、CFML ファイルを編集するときに使用します。詳細については、39 ページの「[ColdFusion Builder でのコード編集](#)」を参照してください。

HTML エディターは HTML コードを編集するときに、JavaScript エディターは JavaScript コードを編集するときにそれぞれ使用します。CSS エディターではカスケードリングスタイルシートを表示して編集することができ、アプリケーションの視覚的要素にスタイルを適用できます。

カーソルにリンク機能を使用すると、現在編集しているコードのタイプに応じてエディターを切り替えることができます。これらのエディターには、コードの色付け、コードアシスト、アウトラインビューなど、コードのナビゲーションを容易にし、コードの有効性を維持するための機能が数多く含まれています。

CFML、HTML、JavaScript、CSS コードを入力する際には、コードの完成を支援するためのヒントが表示されます。この機能をコードアシストといいます。詳細については、40 ページの「[コードアシスト](#)」を参照してください。

パースペクティブのカスタマイズ

パースペクティブのレイアウトはカスタマイズできます。つまり、パースペクティブ内で表示するビューとエディターは設定可能です。例えば、特定のパースペクティブではタスクビューを表示し、別のパースペクティブでは表示しないように設定できます。また、パースペクティブに関連する次の項目も設定できます。

- ファイル/新規のサブメニュー項目
- ウィンドウ/パースペクティブ/その他のサブメニュー項目
- ウィンドウ/ビューの表示/その他のサブメニュー項目
- ツールバーとメインメニューに表示するアクションセット（ボタンとオプション）

注意：プラグイン設定の ColdFusion Builder では、メニューの名前がやや異なります。

カスタマイズされたパースペクティブの作成

- 1 変更するパースペクティブを開きます。
- 2 ウィンドウ／パースペクティブのカスタマイズを選択します。
- 3 カスタマイズされたパースペクティブに追加する項目または削除する項目の種類に応じて、「ショートカット」タブまたは「コマンド」タブの「コマンドグループの可用性」タブを選択します。
- 4 選択したパースペクティブのメニューおよびツールバーに表示する要素のチェックボックスをオンにします。
- 5 「OK」をクリックします。
- 6 ウィンドウ／パースペクティブの別名保管を選択します。
- 7 パースペクティブの別名保管ダイアログボックスで、変更したパースペクティブの名前を入力し、「OK」をクリックします。

注意：変更したパースペクティブを保存すると、変更したパースペクティブの名前がウィンドウ／パースペクティブを開く／その他メニューに追加されます。

カスタマイズされたパースペクティブの削除

削除可能なパースペクティブは、独自に作成したもののみです。ワークベンチに最初から付属するパースペクティブは削除できません。

- 1 ColdFusion Builder でウィンドウ／環境設定を選択します。
- 2 環境設定ダイアログボックスの左側にツリービュー構造が表示されます。
- 3 ツリービュー構造で、一般／パースペクティブを選択します。
- 4 「選択可能なパースペクティブ」で、削除するパースペクティブを選択します。
- 5 「削除」をクリックし、確認ダイアログで「はい」をクリックします。
- 6 「OK」をクリックします。

カスタマイズされたパースペクティブのリセット

変更済みのパースペクティブは、初期のレイアウトに戻すことができます。

- 1 ColdFusion Builder でウィンドウ／環境設定を選択します。
- 2 環境設定ダイアログボックスの左側にツリービュー構造が表示されます。
- 3 ツリービュー構造で、一般／パースペクティブを選択します。
- 4 「選択可能なパースペクティブ」で、リセットするパースペクティブを選択します。
- 5 「リセット」をクリックし、「OK」をクリックします。

第 5 章：ColdFusion Builder でのコード編集

CFML エディター

ColdFusion Builder には、CFM ファイル、CFC ファイル、HTML ファイル、JavaScript ファイル、および CSS ファイルに対応する多彩なコード編集機能を備えた CFML エディターが組み込まれています。CFML エディターは、コード補完や効率的なコードナビゲーションなどの機能を提供することで、コードの記述を支援します。CFML エディターでは、様々な色やフォントを使用してワークスペース内のコードを表示できます。

開発環境をカスタマイズするには、エディターのプロファイルと設定を指定します。

エディタープロファイル

エディタープロファイルを使用すると、次のエディター設定を 1 つのプロファイルにグループ化して保存できます。

- コードアシスト
- カラー
- キー
- アウトライン
- シンタックスチェック
- 入力

エディタープロファイルは、開発のニーズに合わせて複数のエディター設定を使い分ける場合に便利です。例えば、コードアシスト、コードの色付け、キーボードショートカットのなどの設定を選択し、1 つのエディタープロファイルに保存できます。エディタープロファイルを選択することで、これらすべての設定を同時に適用できます。

エディタープロファイルを作成するには、次のようにします。

- 1 保存する個々のエディター設定を指定します。
- 2 ウィンドウメニューから「環境設定」を選択します。
- 3 ツリービューで、ColdFusion / プロファイルを選択し、「プロファイルの新規作成」をクリックします。
- 4 プロファイルの名前を入力します。設定した現在のエディター設定値が、このプロファイルに保存されます。

デフォルト設定を変更し、変更した設定を新しいプロファイルとして保存することもできます。プロファイルはインポートまたはエクスポートすることもできます。

ColdFusion Builder にはデフォルトで 3 つのエディタープロファイルが用意されています。

- デフォルト - ColdFusion Builder のデフォルトのエディター設定を適用します。
- Dreamweaver - Dreamweaver エディター設定に類似したエディター設定を適用します。
- CFEclipse - CFEclipse エディター設定に類似したエディター設定を適用します。

エディタープロファイルを選択するには、プロファイルダイアログボックスのアクティブなプロファイルドロップダウンリストをクリックし、プロファイルを選択します。

エディター設定

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディターを選択します。

作成する CFM ファイルおよび CFC ファイルのデフォルトのファイル名を指定できます。ファイル内の末尾の空白を削除するには、「ファイルを保存する前にスペースを削除」を選択します。

また、すべての CFM ファイルおよび CFC ファイルにデフォルトで挿入する任意のテキスト（ボイラープレートコード）も指定できます。例えば、自分が作成したすべての CFC ファイルに著作権情報を追加する必要があるとします。その場合は、新規ファイルの設定の「CFC」タブで、著作権のテキストを指定できます。

設定可能なエディター設定には、コードアシスト、コードの色付け、キーボードショートカット、アウトラインビュー、シNTAX チェック、入力の設定があります。

ColdFusion Builder のパフォーマンスの最適化

設定で選択するオプションは、ColdFusion Builder のパフォーマンスに影響を与えます。プリセットのエディター設定を適用すると、エディターのパフォーマンスを改善できます。プリセットの設定を適用すると、現在の設定は完全に失われます。現在の設定を保存するには、設定をエディタープロファイルとして保存します。

プリセットのエディター設定を適用するには、「エディター設定を最適化」をクリックします。次の設定が適用されます。

コードアシスト

次のオプションが選択解除されます。

- 入力時にコードアシストを自動表示
- 入力時に SQL コードアシストを自動表示
- 候補が 1 つだけの場合は自動的に挿入
- 終了タグを自動挿入

次のオプションが選択されます。

- 選択項目を挿入した後に空白を追加
- 等号 (=) を自動挿入
- 属性を自動的に引用符で囲む

入力

- 「自動挿入の有効化」が選択されます。
- 「次の文字に対応する文字を自動挿入」のオプションがすべて選択解除されます。
- 「タグの自動終了」が「なし」に設定されます。

コードアシスト

注意：コードアシストはコンテンツアシストとも呼ばれ、これらの用語は同じ意味で使用されています。

コードアシストは、コードの完成を支援する機能です。入力するコードに応じて、コードの完成を支援するヒントが表示されます。CFML エディターでコードを入力する際には、コードアシストにより、有効な CFML タグ、パラメーター、属性のリストが表示されます。これらの項目は、ポップアップメニューに表示されます。CFML コード内に HTML、JavaScript、または CSS コンテンツを記述する場合は、コードアシストにより、それらのコードを完成させるためのヒントも表示されます。コード完成ヒントを CFML エディターに挿入するには、ダブルクリックするか Enter キーを押します。

コードアシストは、スクリプトベースのシンタックスでも使用できます。例えば、スクリプトシンタックスの関数やコンポーネントのコードヒントも表示されます。ColdFusion プロジェクトに Ajax ライブラリをインポートするときは、JavaScript コードに関してもコードアシストを使用できます。

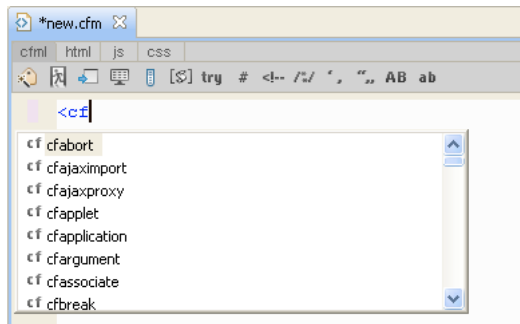
コードアシストの使用

コードヒントは、フレームワークまたは言語（CFML、HTML、JavaScript、および CSS）によって現在のコード式を完成させるためのオプションが提示されるたびに毎回表示されます。例えば、CFML タグの内容を入力する場合は、そのタグで指定できるすべての属性のリストが表示されます。

- 1 CFML ページでは、次のように入力することで CFML タグの入力を開始します。

```
<cf
```

関連するコードヒントが次のように表示されます。



コードヒント

- 2 コードヒントのリスト内を移動するには、上向き矢印（↑）キーまたは下向き矢印（↓）キーを使用します。
- 3 コードヒントを選択し、Enter キーを押します。選択したタグがエディターに追加されます。
- 4 コードの入力を続けると、追加のコードヒントが表示されます。コード行の入力中に Ctrl + スペースバーを押して、コードヒントを表示することもできます。

CFML エディターのスマートタグアシスト機能では、各タグで既に入力されているタグ属性が確認されます。既に入力されているタグ属性は、ヒントとして提示される属性のリストには表示されません。

CFM ページに関するコードアシスト

- CFM ファイルで <cfinvoke> タグを使用するときに、メソッドとコンポーネントのリストが表示されます。
- 「<cf」と入力して Ctrl + スペースバーを押すと、事前定義された属性と値のドロップダウンリストが表示されます。
- createobject() または <cfobject> タグを使用してロードできるコンポーネント（CFC）のリストが表示されます。
- CFC 内で作成されたメソッドが表示されます。これは CFM 内で作成されたコンポーネントオブジェクトを使用して呼び出すことができます。すべての拡張済み CFC ファイルのメソッドのリストも表示されます。
- すべてのビルトイン関数とユーザー定義関数のドロップダウンリストが表示されます。

- 構造体、配列、クエリなどの変数のリストが表示されます。これらの変数は、ページ内で宣言されています。これらの変数は、属性値のコードアシストとしても表示されます。
- `<cfquery>` や `queryNew()` を使用して作成されたすべてのクエリが表示されます。レコードセットを表示するには、`<cfoutput query="">` と入力して Enter キーを押します。または、**queryname.** を使用します（つまり、最後にドットがついたクエリ名を使用します）。
- `<cfinclude>` タグを使用して現在の CFM ページに取り込まれた、別の CFM ページの関数、変数、タグ、クエリも表示されます。
- CFM タグの属性の入力値として指定するファイルを参照して選択できます。入力値としてファイルを指定する必要がある CFM タグには、`<cfinclude>`、`<cfimage>`、`<cfdirectory>`、`<cfpdf>`、`<cffile>`、`<cfzip>`、`<cfspreadsheet>`、`<cfcollection>` などがあります。

CFC ページに関するコードアシスト

- `createobject()` を使用してロードできるコンポーネント（CFC）のリストが表示されます。
- `<cfobject>` タグを使用してロードできるコンポーネント（CFC）のリストが表示されます。
- CFC 内で作成されたメソッドが表示されます。これは CFM 内で作成されたコンポーネントオブジェクトを使用して呼び出すことができます。

例：

- ```
<cfset obj1 = createobject("component", "c1")>
<cfset x = obj1.method1()>
```
- ```
<cfset obj2 = createobject("component", "c2").init()>
```
- ```
<cfset obj3 = new "c3".method3()>
```
- ```
<cfinvoke component="c4" method="method4">
```
- すべての拡張済み CFC ファイルのメソッドのリストが表示されます。「**extends**」および「**implements**」キーワード（属性）に関しては、使用可能なコンポーネントとインターフェイスが表示されます。
- 現在の CFC ファイル内で拡張できるコンポーネントのリストが表示されます。
注意：コードアシストは、コンポーネントプロパティに追加されたコメントに対してはサポートされません。例えば、コードアシストは、数値型の `accountID` プロパティに追加された `/** @ -- code assist is not supported here --*/` のようなコメントに対してはサポートされません。
- `implement` キーワードとともに CFC で使用できるインターフェイスのリストが表示されます。
- CFM タグや `<cfif fileexists()>` などの関数の属性の入力値として、ファイルを参照して選択できます。
入力値としてファイルを指定する必要がある CFM タグには、`<cfinclude>`、`<cfimage>`、`<cfdirectory>`、`<cfpdf>`、`<cffile>`、`<cfzip>`、`<cfspreadsheet>`、`<cfcollection>` などがあります。

変数マッピングを使用したスコープ付き変数に対するコードアシスト

多くの ColdFusion フレームワークでは、アプリケーションの起動時に CFC が作成され、これらの CFC がスコープ変数として保存されます。このような CFC に関するコードアシストを利用するには、ColdFusion Builder がコンテンツとデータ型（CFC の完全修飾名）を識別できるようにする必要があります。

スコープ付き変数に格納された CFC に関連するコードアシストを有効にするには、変数名とそれに対応する CFC のタイプに関するプロジェクトレベルのマッピングを定義します。ColdFusion Builder では、このマッピング情報に基づいて、マッピングされた変数の CFC メソッドのリストが表示されます。ただし、そのデータ型のオブジェクトは作成されません。例えば、mycfc1 を com.adobe.mycfcs.cfc1 にマッピングするとします。mycfc1 と入力して Ctrl + スペースバーを押すと、com.adobe.mycfcs.cfc1 から利用できるすべてのメソッドが表示されます。

変数マッピングを設定するには、プロパティダイアログボックスで、マッピングする変数として mycfc1 を指定し、マッピング対象の値として com.adobe.mycfcs.cfc1 を指定します。詳細については、75 ページの「[変数マッピングの設定](#)」を参照してください。

ColdFusion ORM に関するコードアシスト

ORM (Object Relational Mapping : オブジェクトリレーショナルマッピング) は、オブジェクトモデルを使用してマッピング方針を定義できるプログラミング手法です。ORM を使用すると、リレーショナルデータベースのデータの格納および取得を行えます。

ColdFusion Builder では、次の ColdFusion ORM 設定に関してコードアシストが提供されます。

- entityLoad、entityNew、および entityFindByPK 関数内のエンティティ名。
- entityLoad、entityNew、および entityFindByPK 関数から返されるメソッドとプロパティ。
- エンティティ名とメソッドのコードハイパーリンク。
- application.cfc 内のデータソース。<cfset this.datasource = > など。
- cfcomponent タグのテーブル属性。

ColdFusion ORM の使用方法の詳細については、『ColdFusion アプリケーションの開発』の ColdFusion ORM を参照してください。

Application.cfm ファイルおよび Application.cfc ファイルに関するコードアシスト

コードアシストは、Application.cfm ファイルと Application.cfc ファイルに対してもサポートされています。

Application.cfm はインクルード対象のファイルと見なされ、そのファイル内にあるすべての変数と関数に対してコードアシストがサポートされます。

Application.cfc の場合、コードアシストは、onApplicationStart で宣言されているスコープ付き変数に対してのみサポートされます。

CFML エディターのコードアシストの設定

1 ウィンドウメニューから「環境設定」を選択します。

2 ツリービューで、ColdFusion / プロファイル / エディター / コードアシストを選択します。

- ここでは、コードの完成を支援するコードアシスト辞書のバージョンを選択できます。サポートされている辞書のバージョンは、ColdFusion 9、ColdFusion 8、および ColdFusion MX 7 です。
- 「入力時にコードアシストを自動表示」チェックボックスを選択すると、入力時にコードアシストが自動的に表示されます。

注意：サイズの大きいファイルを編集するときは、「入力時にコードアシストを自動表示」チェックボックスを選択解除すると、エディターのパフォーマンスが向上します。その場合でも、コード行の入力中に Ctrl + スペースバーを押すか、任意に設定したキーボードショートカットキーを押すことで、コードヒントを表示できます。

- 自動的に属性を引用符で囲むように設定したり、終了タグを選択することもできます。

CFC および UDF のコードハイパーリンク

Ctrl キーを押しながらマウスポインターを合わせると、コンポーネント名および UDF がハイパーリンクに変わります。このハイパーリンクをクリックすると、対応するコードが ColdFusion Builder で表示されます。コードハイパーリンクは次の文字列で使用できます。

- UDF : local、included、および cfc.udfName
- `<cfinclude template="">` で指定されているテンプレート
- `createobject()`、`<cfobject>`、`<cfinvoke>`、`new` キーワード、および `extends` 属性で指定されている CFC

サーバーマッピングを使用した CFC 名の解決

CFC のサーバーマッピングを定義するには、ColdFusion Administrator を使用します。サーバーマッピングを定義すると、`extend` キーワードまたは `implement` キーワードを使用するときに、コードアシストで CFC が表示されるようになります。

ColdFusion サーバーマッピングでは、`cfobject` タグと `cfinvoke` タグ、または `createObject` や `new` のような関数でページにアクセスして、ドキュメントルート以外にある ColdFusion コンポーネントを検索できます。

これらのタグで、マッピングの論理パスから始まるパスを指定すると、マッピングのディレクトリパスを使用して CFC が検索されます。CFC のサーバーマッピングは ColdFusion Administrator を使用して定義できます。マッピングを定義すると、`extend` または `implement` キーワード、および `component` または `method` 属性を使用するときに、コードアシストで CFC が表示されるようになります。コードアシストで提示される CFC は、それぞれの完全修飾名を使用して解決されます。

サーバーから CFC 名を解決するためのマッピングを作成すると、サーバー設定がキャッシュに保存され、サーバーが稼働していないときでも使用できるようになります。サーバー設定は、ColdFusion Builder からサーバーを起動または更新する際に収集されます。

サーバー設定の指定

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービュー構造で、ColdFusion /サーバーの設定を選択します。
- 3 「サーバー設定の構築」チェックボックスを選択し、サーバー設定の収集動作に関するチェックボックスを適切に選択します。例えば、ColdFusion Builder を起動するたびにサーバー設定を収集する場合は、「ColdFusion Builder 開始時」チェックボックスを選択します。

CFML 辞書

ColdFusion Builder には、CFML コードの入力を支援する辞書が組み込まれています。CFML 辞書は、ライブラリに含まれる各タグと関数に関する情報が記述された XML ファイルです。例えば、CF9 辞書ファイル (cf9.xml) には、CF9 で使用可能なすべてのタグと関数に関する情報が含まれています。

ColdFusion Builder は、ColdFusion 9、ColdFusion 8、および ColdFusion MX7 の各バージョンの辞書をサポートしています。

辞書のバージョンを選択するには、次のようにします。

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion /プロファイル/エディター/コードアシストを選択します。
- 3 コードアシスト辞書のバージョンドロップダウンリストを選択し、コードの入力を支援する辞書のバージョンを選択します。
- 4 カスタム辞書を含むすべての辞書をリロードするには、「辞書のリロード」をクリックします。

カスタム CFML 辞書の作成

カスタム CFML 辞書 (XML ファイル) を作成し、ColdFusion Builder 内のコードアシストやタグエディターでその辞書を使用することができます。

- 1 ColdFusion Builder のインストール先で、次の場所に移動します。

```
¥plugins¥com.adobe.ide.coldfusion.dictionary_XXX¥dictionary
```

- 2 Dictionary ディレクトリ内に「Custom」というフォルダーを作成します。

- 3 カスタムタグとカスタム関数に関する情報を含む XML ファイルを作成します。

次の例に示すように、XML ファイルで、各タグに関する情報を <tag></tag> 要素内に記述し、すべての tag 要素を <dictionary></dictionary> 要素で囲みます。

```
<dictionary>
  <tags>
    !--
    cfabort
    showError = "error_message"
    -->
    <tag endtagrequired="false" name="cfabort" single="true"
    xmlstyle="false">
    <parameter name="showerror" required="false" type="String">
    </parameter>
    </tag>
  </tags>
</dictionary>
```

カスタム CFML 辞書を作成するには、参考として cf9.xml ファイルまたは cf8.xml ファイルを参照してください。

必須属性の自動挿入

コードアシスト (ウィンドウ/環境設定/ColdFusion/プロファイル/エディター/コードアシスト) で、「必要なタグ属性を自動挿入」を選択すると、タグの必須属性が挿入されます。

注意: これはデフォルトの設定です。

例えば、cfpdf "action=write" を選択した場合、必須属性の destination、source および name のすべてが自動的に挿入されます。

この設定が選択されていない場合は、属性は挿入されず、単に提案候補として表示されます。

必須属性とオプション属性の分割リスト

コードアシストでは、次のようにタグ属性のリストが表示されます。

- 必須のタグ属性は、提案候補リストの上部にチェックマーク付きで表示され、その属性が必須であることが示されます。
- 必須の属性とオプションの属性は、線によって区切られます。

createObject の提案候補

コードアシストでは、createObject 関数でサポートされているすべてのオブジェクト (および対応する引数) の提案候補が表示されます。

コードアシストの提案候補の循環

コードアシスト（ウィンドウ／環境設定／ColdFusion／プロファイル／エディター／コードアシスト）で、「コードアシスト提案候補の循環」を選択すると、循環型のコードアシストが提供されます。

注意：これはデフォルトの設定です。

Ctrl + スペースバーを押した場合の流れを考えます。

- 変数、スコープおよび UDF が表示されます。
- このリストで、さらに Ctrl + スペースバーを押すと、すべてのビルトイン関数が提案候補として表示されます。
- このリストで、さらに Ctrl + スペースバーを押すと、すべての提案候補が表示されます。
- Ctrl + スペースバーを続けて使用すると、提案候補の循環を続行します。

注意：提案候補に表示したいテキストの一部を手動で指定した場合、循環型のコードアシストは表示されません。

テキストを含む提案候補のフィルタリング

コードアシスト（ウィンドウ／環境設定／ColdFusion／プロファイル／エディター／コードアシスト）で、「テキストを含むフィルター提案候補」を選択すると、指定したテキストに基づいて候補をフィルタリングできます。

注意：これはデフォルトの設定です。

指定したフィルターテキストを含むすべての提案候補が表示され、指定したテキストで始まる候補が選択状態になります。

コードアシストのデータ型認識

コードアシストの提案候補は、予想されるデータ型に基づいて表示されます。

例えば、Abs 関数内でコードアシストを読み込むと、型が数値、Any または複合データ型（配列、構造、コンポーネントなど）の関数と変数が表示されます。

関数の引数の自動挿入

関数を挿入すると、必須のすべての引数が自動的に挿入されます。デフォルトでは、最初の引数が選択されます。

関数コンテキストアシスト

- ColdFusion Builder では、関数のパラメーター内でコードアシストを読み込むと、毎回、関数コンテキストアシストが表示されます。

注意：ColdFusion Builder（以前のリリース）では、関数コンテキストアシストは、コードアシストの提案候補から関数を挿入した後にのみ表示されます。

- 現在編集中的のパラメーターは、太字で強調表示されます。
- オプションのパラメーターは、[] の中に表示されます。

接続属性のスマートコードアシスト

スマートコードアシストでは、cfexchangeconnection や cfftp などのタグ用に作成された接続変数が表示されます。

cfloop の使用法に基づいたコードアシスト

ループのタイプに基づいて、必須属性が自動的に挿入されます。例えば、配列をループする場合、index 属性が自動的に挿入されます。

任意のタグブロックの選択

Ctrl + Alt + B (Windows) または Command + Option + B (Mac) のショートカットを使用すると、タグブロックを選択できます。 caret 位置から囲まれるタグブロックが選択されます。

選択状態を維持したまま、ショートカットを繰り返すと、コードの選択を累積的に行うことができます。

例えば、次のスニペットで、Ctrl + Alt + B または Command + Option + B を押すと、cform タグブロックが選択されます。選択状態を維持したまま、Ctrl + Alt + B または Command + Option + B を押すと、コードの選択範囲が次の囲みのレベルまで広がります。つまり、コードブロック全体が選択されます。

```
<cffunction name="test">
<cform name="form1">
<!-- caret position --->
</cform>
```

対応するタグへのジャンプ

Ctrl + Alt + M (Windows) または Command + Option + M (Mac) のショートカットを使用すると、caret 位置をタグブロックの最初から最後に、または最後から最初に移動できます。

注意: caret 位置は、開始タグまたは終了タグの中であれば (ただし、タグ名の真上でなければ)、どこでもかまいません。

クイックフィックス

クイックフィックスは、コード内のメソッドやクラス、CFC/CFM ファイルの使用方法を認識して、これらの生成を支援します。

例えば、test() というユーザー定義関数を入力したときに、この関数とそのページやインクルードされているページで定義されていない場合、クイックフィックスによって、この関数の生成が支援されます。この関数呼び出しがファイルに挿入されます。

ファイルを開いたり編集したりすると、定義されていない関数呼び出し、CFC および CFM が自動的に識別されます。エディターの左余白に電球のアイコンが表示され、このアイコンをクリックすると、適切なクイックフィックスを実行するかどうかメッセージが表示されます。

注意: CTRL + 1 のショートカットを使用しても、同じ結果を得ることができます。

クイックフィックスをオフにするには、「クイックフィックスを使用可能にする」のチェックを解除します (ウィンドウ/環境設定/ ColdFusion / プロファイル/ エディター/ シンタックスチェック)。

次のような場合に、クイックフィックスによる支援が提供されます。

- **ローカル UDF の呼び出し** (すべての言語構造において)。例えば、cfset や cfscripct の代入、関数の引数、その他の式など。

UDF 内で引数を指定すると、クイックフィックスでは、その引数が作成されます。関数の呼び出し場所に応じて、コードが生成されます。

タグとスクリプトスタイルの両方のシンタックスが、コンテキストに応じてサポートされます。つまり、スクリプトベースのシンタックスでメソッドを呼び出した場合、クイックフィックスでは、スクリプトスタイルのメソッドが生成されません。

- **CFC 上のメソッド呼び出し**。例えば cfc1.function1() などです。CFC1 に function1 が定義されていない場合、クイックフィックスでは、このメソッドが作成されます。
- **createObject、new、cfobject および cfinvoke からの CFC の作成**。wwwroot (プロジェクトにサーバーが関連付けられている場合)、プロジェクト、またはユーザーが選択したフォルダーに対して CFC を作成するオプションが提供されます。
- **cfinclude および cfmodule からの CFM ページの作成**
- **extends および implement 属性からの CFC および CFM の作成**

例

```
<cfset o1 = new component1()>
<cfset st = structNew()>
<cfset result = getResults("abc", 10, st)>
<cfinclude template="test3.cfm" >
<cfscript >
    o2 = new component2();
    //Assume that component2 exists
    o2.someFunction();
</cfscript>
```

このスニペットでは、クイックフィックスによって 4 つの未解決の問題が検出され、アクションが要求されます。

コードの行	問題	指示メッセージ
<cfset o1 = new component1()>	コンポーネントが作成されていない	CFC の作成
<cfset result = getResults("abc", 10, st)>	getResults 関数が存在しない	getResults 関数の作成
<cfinclude template="test3.cfm" >	test3.cfm が存在しない	test3.cfm の作成
o2 = new component2();	コンポーネントが存在しない	component2 への someFunction の作成

その他のリソース

- [Quick Fix in ColdFusion Builder](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sagar Ganatra が、クイック補正の使用方法を実演します。

コードフォーマッター

コードフォーマッターを使用して、CFML コードの見た目を変更することができます。この機能は、インデントや行の長さ、タグおよび属性名の大文字 / 小文字表記を標準化するのに役立ちます。コードフォーマッターでは、タグベースとスクリプトベースの両方のシンタックスが整形されます。

次の操作が可能です。

- プロジェクト内の CFML ファイルを選択して、そのコードを整形
- エディターで開いている CFML ファイルのコードを整形
- CFML ファイル内の一部のコードを選択して整形

コードフォーマッターでは、CFM および CFC ファイルに加えて、CFM または CFC ファイル内の HTML、XML、CSS および JavaScript コードの整形もサポートします。

コードの整形

- 1 エディターで、CFM または CFC ファイルを開きます。
- 2 次のいずれかの操作を行います。
 - 右クリックして「フォーマット」を選択します。
 - Ctrl + Shift + F (Windows) または Command + Shift + F (Mac) のキーボードショートカットを使用します。

注意：コードの一部を整形する場合は、手順 2 の前に該当部分を選択します。

環境設定のカスタマイズ

コードフォーマッターには汎用的な環境設定があり、この設定はプロファイルに定義されます。環境設定をカスタマイズするには、プロファイルを作成するか、またはデフォルトのプロファイルに新しい名前を付けてからプロファイルを変更します。

- 1 ColdFusion Builder でウィンドウ／環境設定を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディター / フォーマッターを選択します。
- 3 「追加」をクリックして、新しいプロファイルを追加します。または、「アクティブなプロファイル」の既存のプロファイルを選択し、「編集」をクリックして、プロファイルを編集します。

注意：デフォルトのプロファイルは編集できません。

- 4 CFML 書式設定プロファイルで、次の表に記載されているルールを設定して、プロファイルを追加または変更します。コード整形の環境設定は、次のルールに基づきます。ダイアログボックス内の各タブが 1 つのルールを表します。

ルール	オプション
一般	<ul style="list-style-type: none">• タグおよび属性で現在使用している大文字 / 小文字表記をそのまま使用するか、大文字または小文字に変更するかを指定します。• タグの末尾に /> を追加します。例えば、<cfargument> を <cfargument /> に変更します。/> を追加するタグを指定して、「追加」をクリックします。• cfoutput タグの内容が複数の行にまたがる場合に限って、その終了タグ (</cfoutput>) を新しい行に配置します。
インデント	<ul style="list-style-type: none">• インデントの詳細と、インデントを適用しないタグの名前を指定します。

ルール	オプション
空白文字	<ul style="list-style-type: none">• 選択内容に基づいて、空白文字を追加します。• 選択内容に基づいて、空白行を追加します。• 保持する空白行の行数を指定します。
折り返し	<ul style="list-style-type: none">• タグ内の 1 行に含める属性の数を指定し、列の幅を設定します。• 折り返しに関する様々な構成を指定します。
中括弧	<ul style="list-style-type: none">• コンポーネント宣言、関数宣言、および switch、if、else、try ブロックの中括弧を同じ行に配置するか、新しい行に配置するかを指定します。

環境設定の共有

整形の環境設定は、XML 形式で共有することができます。

環境設定ダイアログボックス (ウィンドウ/環境設定/ ColdFusion /プロファイル/エディター/フォーマッター) の「フォーマッター」セクションの該当するボタンを使用して、環境設定を書き出したり、他のユーザーによって共有された環境設定を読み込んだりできます。

その他のリソース

- [CFML Code Formatter - An Introduction](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sandeep Paliwal が、CFML コードフォーマッターを使用してコーディングのガイドラインを適用する方法について説明します。

自動整形

ColdFusion Builder では、読みやすさを向上させるため、自動的にコードの行がインデントされて、終了タグが調整されます。

次の場合に自動整形が機能します。

コードの入力時

- カーソルが次の後ろにあるときに **Enter** を押すと、インデントされます。
 - 開始タグの >
 - 左中括弧 ({)
 - 終了タグの >
 - 終了タグ (タグが自動完了する場合は </ の入力)
- 右中括弧 () を閉じると、この括弧は、左中括弧 ({) に合わせて配置されます。

コードアシストの使用時

- 終了タグが自動挿入される場合、スラッシュ (/) を入力すると、終了タグは開始タグに合わせて適切な場所に配置されます。
- 終了タグが自動挿入される場合、終了タグは開始タグに合わせて配置されます。

コードの色付け

ColdFusion Builder では、シンタックスに含まれる CFML タグ名、属性、属性値、キーワード、コメントなどの要素を、別々の色で強調表示できます。コードの色付けは、スクリプトスタイルのコーディングでもサポートされています。カラー設定をカスタマイズすると、Eclipse エディターのデフォルトのカラー設定を上書きできます。

CFML エディターのカラー設定

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディター / 色を選択します。
 - カラー設定は、COL ファイルとしてインポートおよびエクスポートできます。
 - トークンのカラー設定を指定できます。トークンには、言語設定、エラー、識別子、キーワード、リテラル、区切り記号などがあります。
 - コードの特定の範囲のカラー設定を指定できます。コード要素の始まりから終わりまでの範囲を指定し、その範囲を識別するための色を定義します。

SQL エディターのカラー設定

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディター / 色を選択します。「トークン」で、SQL にスクロールし、コメント、キーワード、およびテキストのカラー設定を選択します。

SQL エディター

CFML エディターには SQL エディターが組み込まれており、SQL ステートメントを編集、作成、または実行できます。

SQL エディターを使用するには、ColdFusion Builder でサーバーを設定し、そのサーバーが稼動している必要があります。SQL エディターはオフラインデータベースをサポートしていません。

SQL エディターは、次の SQL ステートメントに対して、コードアシストとコードの色付けをサポートします。

- Select
- Insert
- Update
- Delete

注意：cfquery タグ内にある SQL ステートメントの場合、コードアシストは CFML エディター自体で使用できます。コード補完のヒントを表示するために SQL エディターを開く必要はありません。

SQL エディターの使用

- 1 次のいずれかの操作を行います。
 - CFML エディターを右クリックし、「SQL エディター」を選択します。
 - キーボードショートカット Ctrl + Alt + S キー（Windows の場合）または Command_Alt + S キー（Mac OS の場合）を使用します。

- 2 SQL エディターで、サーバドロップダウンリストからサーバを選択し、データソースドロップダウンリストからデータベースを選択します。
- 3 SQL ステートメントを入力し、次の操作を行います。
 - 「クエリの実行」をクリックすると、SQL ステートメントの結果が「クエリ結果」タブに表示されます。SQL エディターで実行できるのは、SELECT ステートメントのみです。
 - 「OK」をクリックすると、CFML エディターの現在のキャレット位置に SQL ステートメントが挿入されます。キャレットとは、次の文字が出現する位置を示す CFML エディター内のマーカーです。
- 4 コードブロックを選択し、Shift + Enter キーを押して、SQL エディターから CFML エディターに SQL コードブロックを直接コピーすることもできます。
- 5 CFML エディターで SQL コードを編集するには、コードブロックを選択し、右クリックして「SQL エディター」を選択します。選択したコードブロックが SQL エディターに表示されます。

基本的な SQL シンタックスと SQL ステートメントの作成の詳細については、『ColdFusion アプリケーションの開発』の SQL の使用を参照してください。

SQL コードアシスト

SQL コードアシストは、SQL ステートメント内のデータベーステーブル名とフィールド名に関してのみ使用できます。

ColdFusion Builder の SQL コードアシスト機能はデフォルトでオンになっているので、SQL ステートメントに関するコードヒントは CFML エディターで自動的に表示されます。SQL コードアシストの自動表示をオフにするには、次のようにします。

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディター / コードアシストを選択します。
- 3 「入力時に SQL コードアシストを自動表示」を選択解除します。

SQL コードアシストの自動表示がオフになっている場合でも、次のようにすると SQL ステートメント内でコードヒントを表示できます。

- 1 データベーステーブル名またはフィールド名を表示するには、SQL エディターに SQL ステートメントを入力し、Ctrl + スペースバーを押します。
- 2 上向き矢印 (↑) キーまたは下向き矢印 (↓) キーを使用して、テーブル名またはフィールド名のリスト内を移動します。
- 3 テーブル名またはフィールド名を選択し、Enter キーを押します。選択したテーブルまたはフィールドが SQL ステートメントに追加されます。

ColdFusion Builder には、<cfoutput> タグの query 属性のコードアシストが用意されています。例えば、<cfoutput query=""> と入力して、二重引用符 ("") の間にキャレットを置いて、Ctrl + スペースバーを押すとします。そのファイルやインクルードファイルで作成されたクエリのリストが表示されます。

式で使用されるクエリ名にも、コードアシストが用意されています。

例えば、次のようなクエリを作成したとします。

```
<cfquery datasource="dsn1" name="q1">
Select id, firstName, lastName from employee
</cfquery>
```

<cfset name = q1.> という式を入力して Ctrl + スペースバーを押すと、その SQL クエリで選択された列が ColdFusion Builder に表示されます。

SQL コードの色付け

CFML エディターで SQL コードの色付けがサポートされるのは、cfquery タグ内のみです。cfquery タグ内の SQL キーワード、コメント、およびテキストの色付けを指定できます。色付けは、次のタイプの SQL コメントに対してのみサポートされます。

- /* */
- --

SQL コードの色付けの設定については、51 ページの「[SQL エディターのカラー設定](#)」を参照してください。

コードの折りたたみと展開

CFML エディターには、コード内をナビゲートするためにショートカットが数多く用意されています。これらのショートカットには、コードブロックの折りたたみと展開、コード定義のソースのオープン、タイプの参照とオープンなどがあります。複数のコードブロックを折りたたんだり展開することで、複雑なコードドキュメントのナビゲート、表示、管理が容易になります。ColdFusion Builder では、複数行にまたがるコードステートメントを折りたたんだり展開することを、コードの折りたたみまたは展開と呼びます。この機能は、CFM、CFC、HTML ファイルで使用できます。

任意のコードブロックを非表示にしたり表示したりできます。定義したコードの折りたたみは保存されるため、それ以降のセッションでも使用できます。

- 1 CFML エディターで、左余白にある折りたたみアイコン (-) または展開アイコン (+) をクリックします。

```
+ <cffunction name="add" returnType="numeric">
- <p>
- <cfoutput>5 + 3 = #add(5,3)#</cfoutput>
- </p>
- <cffunction name="formatDateTime" output="no" verifycli
  <cfargument name="datetime" type="date" required="y
```

コードブロックを折りたたむと、最初のコード行を除くすべての行が非表示になります。

```
+ <cffunction name="add" returnType="numeric">
  <cfargument name="x" type="numeric">
  <cfargument name="y" type="numeric">
  <cfreturn x+y>
</cffunction>
- <cffunction name="formatDateTime" output="no" verifycli
  <cfargument name="datetime" type="date" required="y
```

展開アイコン (+) の上にマウスポインターを置くと、現在折りたたまれているコードがツールヒント内に表示されます。

```
- <cffunction name="add" returnType="numeric">
  <cfargument name="x" type="numeric">
  <cfargument name="y" type="numeric">
  <cfreturn x+y>
</cffunction>
- <p>
- <cfoutput>5 + 3 = #add(5,3)#</cfoutput>
- </p>
- <cffunction name="formatDateTime" output="no" verifycli
  <cfargument name="datetime" type="date" required="y
```

コードブロックを展開すると、コードが見えるようになります。

- 2 デフォルトでは、コードのすべての行が展開されます。最初のコード行を除くすべての行を折りたたむには、エディターの左余白を右クリックします。次に折りたたみ/すべてを折りたたむを選択します。

コードの折りたたみの追加

- 1 非表示にするコードのブロックを選択します。
- 2 次のいずれかの操作を行います。
 - Ctrl + Alt + F (Windows) または Command + Option + F (Mac) を押します。
 - 右クリックして、ソース/選択項目で折りたたみを切り替えを選択します。

コードの折りたたみの削除

コードのブロックの折りたたみを削除するには、

- ❖ コードの最初の行にカーソルを移動し、コードの折りたたみに使用した手順をもう一度行います。

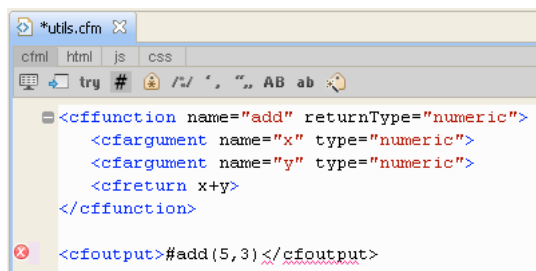
注意：コードの同じ行から、コードの折りたたみを複数回行うことはできません。

シンタックスのチェックと強調表示

CFM ファイル内のタグまたはコードブラケットを選択すると、そのタグまたはブラケットに対応するペアが自動的に強調表示されます。

有効な CFML コードとして認識されないコードを入力すると、次のような方法で通知されます。

- 次のように、該当するコード行の横に ×印が表示されます。

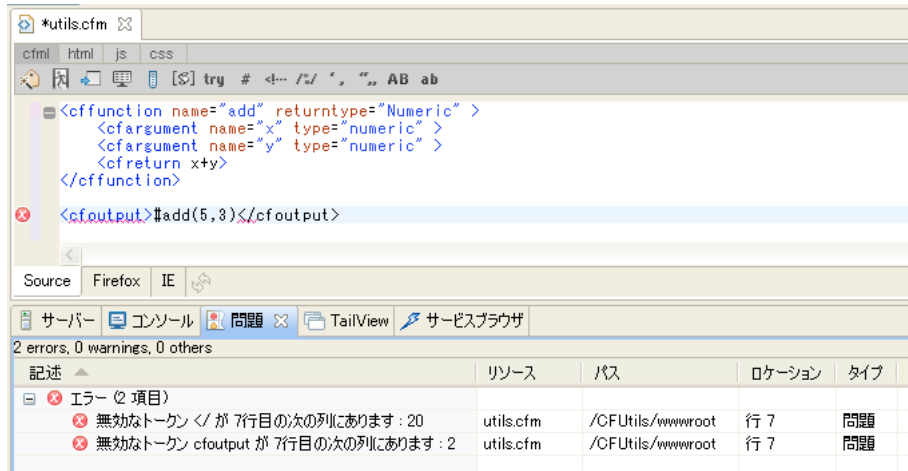


```
*utils.cfm
cfml html js css
try # /?/ ', "., AB ab
<cffunction name="add" returnType="numeric">
  <cfargument name="x" type="numeric">
  <cfargument name="y" type="numeric">
  <cfreturn x+y>
</cffunction>
<cfoutput>#add(5,3)</cfoutput>
```

強調表示されたシンタックスの問題

×印にマウスポインターを合わせると、エラーの詳細が表示されます。

- 次のように、問題ビューにエラーが一覧表示されます。



問題ビューに表示されたシンタックスの問題

各メッセージには、エラーの簡単な説明、エラーが発生したファイルとフォルダー、ファイル内の行番号が含まれます。エラーをダブルクリックすると、エディターでファイルが開かれ、該当するコード行が強調表示されます。

注意：表示される情報は、現在開かれている CFM ファイルのシンタックスエラーのみです。CFM ファイルを閉じると、そのファイルのエラーは表示されなくなります。

エラーの性質と重大度によっては、エラーを訂正するまでアプリケーションを適切に実行できない可能性があります。

シンタックスチェックの設定

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディター / シンタックスチェックを選択します。
- 3 シンタックスチェックは、デフォルトでオンになっています。シンタックスチェックをオフにするには、「シンタックスチェックの有効化」を選択解除します。
- 4 ファイルを開くときまたは保存するときのみシンタックスエラーを表示するには、「ファイル保存時にのみシンタックスエラーを表示」を選択します。ファイルの編集中は、ファイル内のすべてのエラーが問題ビューから削除されます。

コードのナビゲート

コードナビゲーション

Ctrl + Shift + 上向き矢印キーおよび Ctrl + Shift + 下向き矢印キー (Windows)、または Command + Shift + 上向き矢印キーおよび Command + Shift + 下向き矢印キー (Mac) のショートカットを使用して、コード内を簡単にナビゲートできます。

コードナビゲーションは、タグベースとスクリプトベースの両方のシンタックスで機能します。

次の節で、様々なケースでコードナビゲーションがどのように機能するかを説明します。

CFM ファイル内

キャレット位置が、ユーザー定義関数から、次または前のユーザー定義関数に移動します。

例えば、次のスニペットで、カーレット位置が func1 内にある場合に、Ctrl + Shift + 下向き矢印キーを押すと、カーレット位置が func2 に移動し、この関数名が選択されます。func2 内のいずれかの場所から、Ctrl + Shift + 上向き矢印キーを押すと、カーレット位置が func1 関数に移動し、この関数名が選択されます。

```
<cffunction name="func1">
</cffunction>
<cffunction name="func2">
</cffunction>
```

CFC ファイル内

カーレット位置は次のように移動します。

- ユーザー定義関数から、次または前のユーザー定義関数に移動します。
- CFC の最上位のメンバーに移動します。例えば、CFC 内の cfproperty まで一気に移動します。

例えば、次のスニペットで、行番号 1 (<cfcomponent>) から下方向にナビゲートすると、カーレット位置は最初に行 2 (<cfproperty name="prop1">) に移動し、続いて行 4 (<cffunction name="func1">)、行 6 (<cffunction name="func2">) に移動します。上方向に移動すると、カーレット位置は、同じルートを逆の順番で移動します。

カーレット位置が cfproperty に移動した場合、この名前が選択されます。

```
<cfcomponent>
<cfproperty name="prop1">
<cfset this.var1 = 100>
<cffunction name="func1">
</cffunction>
<cffunction name="func2">
</cffunction>
</cfcomponent>
```

フロー制御ステートメントの場合

CFScript で提供されるフロー制御ステートメントの場合 (例えば、if-else、switch-case、try-catch ステートメントの場合)、CFC ページと CFM ページの両方で、カーレット位置がこれらのステートメント内にあるときに、次または前のコードブロックにカーレット位置が移動します。

例えば、cfif タグ内で同じ組み合わせのキーを押すと、カーレット位置は、次の cfelseif または cfelse ステートメントに移動します。

次の例で下方向にナビゲートした場合、

- 行 1 (<cffunction name="func1">) からは、カーレット位置が行 10 (<cffunction name="func2">) まで一気に移動します。
- カーレット位置が 2 行目にある場合、カーレット位置は行 2 (<cfif a eq 10>) の cfif 内から行 4 (<cfelseif a gt 10>) に移動し、cfelseif が選択されます。
- 行 4 (<cfelseif a gt 10>) からは、カーレット位置が行 6 (<cfelse>) に移動し、cfelse が選択されます。
- 行 6 (<cfelse>) からは、カーレット位置が行 8 (</cfif>) に移動します。
- 行 8 (</cfif>) からは、カーレット位置が行 10 (<cffunction name="func2">) に移動し、func2 が選択されます。

逆方向にナビゲートした場合 (Ctrl + Shift + 上向き矢印キー)、

- 行 6 (<cfelse>) からは、カーレット位置が行 4 (<cfelseif a gt 10>) に移動し、cfelseif が選択されます。
- 行 10 からは、カーレット位置が行 1 (<cffunction name="func1">) に移動します。

```
<cffunction name="func1">
<cfif a eq 10>
<cfset b = 10>
<cfelseif a gt 10>
<cfset b = 20>
<cfelse>
<cfset b = 30>
</cfif>
</cffunction>
<cffunction name="func2">
</cffunction>
```

スマートタブナビゲーション

次または前の関数の引数やタグ属性にすばやくナビゲートできます。次にナビゲートするには、Ctrl +] (Windows) または Command +] (Mac) を、前にナビゲートするには、Ctrl + [(Windows) または Command + [(Mac) を使用します。

キャレット位置が関数呼び出しまたはタグ名にある場合に、下方向にナビゲートすると、キャレット位置は、最初のパラメーターまたは属性の値に移動します。

ネストされた関数呼び出しの場合、キャレット位置がすべてのパラメーターの最後または最初に達すると、次または前のパラメーターに移動します。

スマートタブナビゲーションは、関数定義の場合の cffunction の引数にも適用できます。例えば、

```
<cfscript>function abc(int a, int b) access="package" output="false"</cfscript>
```

この場合、キャレット位置が abc または function にあると、int a、int b、package および false にナビゲートされます。

スマートタブナビゲーションは、タグベースとスクリプトベースの両方のシンタックスに適用されます。

コードリファクタリング

コードリファクタリングとは、全体的な結果を変更せずにプログラムのソースコードを改善するプロセスです。一般に、コードリファクタリングを行うと、コードの可読性と保守性が向上します。

ColdFusion Builder では、CFC、CFM、UDF の名前の変更、検索、プレビューなど、様々なリファクタリング手法がプロジェクトレベルおよびワークスペースレベルでサポートされています。

CFC ファイル名または CFM ファイル名のリファクタリング

CFC または CFM の名前を変更する場合は、その CFM または CFC へのすべての参照を含め、CFM または CFC の有効なインスタンスをすべてリファクタリングできます。

- 1 ナビゲーターペインでファイルを右クリックします。
- 2 リファクタリング/名前変更を選択します。
- 3 新しい名前を入力し、変更結果をプレビューします。変更結果のプレビュー時には、選択されたインスタンスを確認して、名前を変更したくないインスタンスを選択解除できます。
- 4 ワークスペース内のすべてのプロジェクトにわたって CFC または CFM の名前を変更するには、「ワークスペース内のすべてのプロジェクトを更新」を選択します。

注意: HTTP または Web サーバーの URL に含まれる CFC または CFM の名前を変更するには、サーバーが登録済みで稼働している必要があります。

リファクタリングを取り消すには、キーボードショートカット Ctrl + Z キー（Windows の場合）または Command + Z（Mac の場合）を使用します。

CFC または CFM ファイル内の UDF のリファクタリング

- 1 CFM または CFC ファイルをエディターで開きます。
- 2 関数名をポイントし、右クリックしてリファクタリング/名前変更を選択します。
- 3 UDF 名を指定して、「OK」をクリックします。

注意：現在、関数名の変更は、メイン関数宣言内でのみサポートされており、関数呼び出しから関数名を変更することはできません。

メモおよび注意事項

- 固有の Eclipse エディターには「移動」オプションを使用してファイルを移動する機能がありますが、CFM および CFC ファイルの場所は変更できません。
- Eclipse ではフォルダー名を変更するオプションがありますが、CFML エディターではフォルダー名の変更をサポートしていません。

メソッドのリファクタリング

変数マッピングを作成することで、アプリケーション変数用に作成されたメソッドをリファクタリングできます。

例えば、次のように、CFC オブジェクトに対してアプリケーション変数が定義されているとします。

```
<cfset Application.appVar = createObject("component", "abc.AppCFM.a")>
```

また、次のように呼び出される CFC 関数 Method() があるとします。

```
<cfset Application.appVar.Method()>
```

Method() 関数をリファクタリングするには、次のようにアプリケーション変数 Application.appVar の変数マッピングを作成します。

- 1 ナビゲータービューでプロジェクトを右クリックし、「プロパティ」を選択します。
- 2 プロパティダイアログボックスで「ColdFusion 変数マッピング」を選択します。
- 3 「新規」をクリックし、マッピングの追加ダイアログボックスで次のように入力します。
 - 変数名：Application.appVar
 - マッピング対象：abc.AppCFM.a

参照検索

CFC、CFM、UDF の検索は、プロジェクトレベルまたはワークスペースレベルで行えます。

- 1 ナビゲーターペインでプロジェクトを右クリックします。
- 2 「参照」をクリックして、「プロジェクト」または「ワークスペース」を選択します。ColdFusion Builder の検索ウィンドウに結果が表示されます。

参照はプロジェクトレベルまたはワークスペースレベルで検索できるだけでなく、CFML コードエディターでの作業中に検索することもできます。ファイル参照を検索するには、CFML エディター内を右クリックし、参照/ファイルを選択します。CFML エディター内からプロジェクトレベルまたはワークスペースレベルで参照を検索することもできます。

タグエディター

タグエディターを使用すると、CFML に精通していなくても、タグや属性を簡単に追加できます。ColdFusion Builder でタグエディターを使用するには、次のようにします。

- 1 「CFML」タブが開いていない場合は、ソース領域で「CFML」タブをクリックします。
- 2 CFML ツールバーのタグエディターアイコンをクリックします。Ctrl + Shift + T キーを押してタグエディターを開くこともできます。
- 3 入力するタグがわからない場合は、タグエディターのダイアログボックスを使用してタグを検索します。
- 4 タグ属性の値を指定します。タグエディターには、各種のアクション属性に固有の値も表示されます。
- 5 「OK」をクリックすると、CFML ページにタグが追加されます。

注意: タグエディターは状況依存型のエディターです。タグの上にカーソルを移動して Ctrl + Shift + T キーを押すとタグの編集が可能になり、そのタグ用のタグエディターが表示され、タグ属性を指定または変更できます。

入力設定

CFML タグを入力する際には、入力設定をカスタマイズできます。次のような設定項目を指定できます。

- 終了タグおよび一致する文字（引用符や中括弧など）の自動挿入
- Enter キーを押したときの子タグの自動インデントおよびインデント文字による空白文字の置換
- 対応するタグが変更されたときの自動変更
- 属性を引用符で囲むときの一重引用符および二重引用符の自動提示

CFML エディターの入力設定

デフォルトでは、自動挿入が有効になっています。一重引用符、二重引用符、中括弧、および「#」記号に対応する文字を自動挿入できます。

自動挿入をオフにするには、次のようにします。

- 1 ウィンドウメニューから「環境設定」を選択します。
- 2 ツリービューで、ColdFusion / プロファイル / エディター / 入力を選択します。
- 3 「自動挿入の有効化」を選択解除します。

CFML エディターのキーボードショートカット

ColdFusion Builder では、次の操作を実行できます。

- クイックアシストを使用して、ウィザードやタグ、カスタムショートカットに関連するショートカットのカタログを表示
- カスタムキーボードショートカットの作成
- ショートカットの変更および削除
- ショートカットを検索し、フィルターテキストに基づいて特定のアクションを実行
- ショートカットを CSV 形式で書き出しおよび読み込み
- デフォルトのショートカット設定の復元

デフォルトのショートカット

デフォルトのショートカットは出荷時の設定で、環境設定ダイアログボックス（ウィンドウ／環境設定／ColdFusion／プロファイル／キー）に表示されます。

デフォルトのショートカットは、読み込んだり書き出したりできますが、名前や説明を変更することはできません。

デフォルトのショートカットを一時的に削除したり変更したりすることは可能です。環境設定ダイアログボックスの「キー」セクションの「デフォルトに戻す」をクリックすると、追加したすべてのショートカットと変更内容が失われます。また、削除したデフォルトのショートカットが復元されます。

ショートカットの規則

構成作業や使用を容易にするため、最良の方法は、自身のキーボードショートカットに合わせた標準形式に準拠することです。デフォルトのショートカットでは、ColdFusion Builder は次の形式に準拠しています。

- タグ用のショートカットには、Ctrl + T (Windows) または Command + T (Mac) 例えば、Ctrl + T、A (Windows) または Command + T、A (Mac) は cfabort のショートカットです。

注意：Ctrl + T および Command + T と、A の間の読点は、順序を示します。つまり、最初に Ctrl + T または Command + T を押し、一度キーを放してから A を押すと、ショートカットが機能します。

- ウィザード関連のショートカットには、Ctrl + Alt + W (Windows) または Command + Option + W (Mac)

同じように、ショートカットを直感的に使用できる規則に従います。

また、ショートカットが競合しないように、一意のキーの組み合わせを定義してください。

カスタムキーボードショートカットの作成

1 ColdFusion Builder で、ウィンドウ／環境設定／ColdFusion／プロファイル／キーを選択します。

2 「追加」をクリックします。

3 ショートカットの一意の名前と説明を指定します。

4 「キーバインディング」でショートカットを指定します。

一意のキーの組み合わせを指定しなかった場合は、競合が表示されます。

5 「テキストを挿入」ボックスに、ショートカットを使用したときにエディターに挿入されるテキストを指定します。

「キャレット位置の定義」の節で、挿入されるテキスト内でのキャレット位置を指定するマクロの使用方法について説明します。

6 「保存」をクリックし、「OK」をクリックして環境設定を閉じます。

キャレット位置の定義

マクロを使用して、(作成したショートカットの) テキストがエディターに表示されるときキャレット位置を定義できます。

環境設定ダイアログボックスの「キー」セクションでテキストを挿入したら、カーソルを表示させたい位置に `$${cp}` マクロを指定します。

例えば、

```
if($${cp}){ <cfoutput>#A#</cfoutput>}
```

Alt + T の組み合わせに上述のテキストを指定して、このショートカットを使用すると、テキストのブロックがエディターに挿入され、カーソルは if の後の丸括弧の間に移動します。

クイックアシスト

クイックアシストでは、クイックリファレンス用のショートカットのカタログが表示されます。

- Ctrl + T (Windows) または Command + T (Mac) を押すと、タグ関連のショートカットとカスタムキーボードショートカットのクイックアシストを起動できます。
- Ctrl + Alt + T (Windows) または Command + Option + T (Mac) を押すと、ウィザード関連のショートカットのクイックアシストを起動できます。
- カスタムキーボードショートカットの読点の前までのキーを押します。例えば、Alt + T、A と Alt + T、B の組み合わせを設定した場合を考えます。この場合、Alt + T を押すと、クイックアシストで両方のショートカットが表示されます。

その他のリソース

- [A cheat sheet of keyboard shortcuts in ColdFusion Builder](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sagar Ganatra が、ColdFusion Builder のすべてのショートカットについて説明します。

- [ColdFusion Search vs Eclipse Search](#)

ColdFusion Builder エンジニアリングチームのメンバーである Kiran Sakhare が、Eclipse の検索機能と ColdFusion Builder の検索機能を比較します。

- [Keyboard shortcuts in ColdFusion Builder](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sagar Ganatra が、ColdFusion Builder 2 でのキーボードショートカット拡張機能について説明します。

ColdFusion Builder の検索

ColdFusion Builder の検索では、テキストやタグを検索するために、専用の検索と置換の機能が用意されています。現在のドキュメントを検索するだけでなく、開いているすべてのドキュメントや、ナビゲーターペインで選択したリソース、プロジェクト、作業セット、ワークスペース、ローカルディレクトリ、FTP の場所、RDS の場所を検索の範囲に設定できます。この機能では、正規表現と複数行検索もサポートしています。

時間を有効に活用するために、バックグラウンドでリモート検索を実行して、他の作業を進めることもできます。検索ビューに、結果と、検索した各ファイル内の一致の詳細が表示されます。

ColdFusion Builder の検索は、CFC、CFM、HTML、XML、CSS および JS のファイルタイプのみ適用されます。

検索の実行

- 1 次のいずれかのオプションを使用して、ColdFusion 検索ダイアログボックスを実行します。

- 開いているドキュメントで、Ctrl + F (Win) または Command + F (Mac) のショートカットを使用します。
- ウィンドウメニューから、検索 / ColdFusion 検索を選択します。

注意：その他のコンテキストで（例えば TXT ファイルで）このショートカットを使用すると、Eclipse の検索 / 置換ダイアログボックスが代わりに表示されます。ただし、検索 / ColdFusion 検索のオプションを使用して、意図的に ColdFusion 検索ダイアログボックスを開くこともできます。

- 2 ColdFusion 検索ダイアログボックスで、検索の範囲を指定します。範囲に応じて、検索オプションは変わります。

例えば、「サブフォルダーを再帰的に処理」オプション（検索対象にサブフォルダーを含める場合に使用）は、範囲が FTP の場所、RDS の場所、またはローカルディレクトリの場合にのみ適用できます。

- 3 (ファイルを開いていない場合に) 検索対象を絞り込むには、ファイル名の拡張子を指定します。
- 4 次のいずれかの操作を行います。
 - 検索するテキストを指定します。必要に応じて、置換するテキストも指定します。
 - 検索するタグを指定し、次の操作を行います。
 - 1 タグを選択し、検索の条件を指定します。例えば、source 属性に cf.jpg が含まれる cfimage タグを検索します。
 - 2 「追加」をクリックして、リストに条件を追加します。
 - 3 タグが見つかった場合に実行するアクション (タグの削除や置換など) を選択します。

適用する場合は、アクションの実行に必要な追加情報を指定します。例えば、cfimage の thickness に新しい属性を設定します。

注意: アクションは、「置換」または「すべて置換」を使用した場合にのみ適用されます。
- 4 次のオプションを使用して、検索の範囲を広げたり、制限したりします。
 - **大文字と小文字の一致:** 検索するテキストに大文字と小文字が完全に一致するテキストに検索を制限します。
 - **単語全体的一致:** 1 つ以上の単語全体が一致するテキストに検索を制限します。
 - **前を検索**
 - **インクリメンタル検索:** 連続的な検索を実行して、テキストによるフィルタリングを実行できます。検索基準を指定すると、入力文字の最初の該当箇所が強調表示されます。
 - **正規表現を使用:** 検索文字列内の特定の文字や短い文字列 (?、*、\w、\b など) が、正規表現の演算子として解釈されるようになります。

例えば、「\b\w*\b dog」で検索すると、「black dog」と「barking dog」の両方が一致します。
 - **空白を無視:** 一致結果に含まれるように、すべての空白文字を 1 つの空白文字と見なして処理します。例えば、このオプションを選択した「this text」では、「this text」と「this text」は一致しますが、「thistext」は一致しません。「正規表現を使用」オプションを選択した場合、このオプションは使用できません。この場合は、正規表現で、空白を無視するように明示的に記述する必要があります。

注意: <p> と
 のタグは空白とはみなされません。
 - **循環検索:** 現在のカーソル位置から、エンドツーエンド検索が実行されます。

注意: 検索範囲によっては、使用できないオプションもあります。
- 5 検索を実行する (アクションは実行しない) 場合は、「検索」または「すべて検索」をクリックします。
 - 「検索」では、現在のドキュメント (のみ) の次の検索テキストの場所に移動して、該当箇所を選択します。

タグが選択された場合は、タグブロックが強調表示されます。
 - 「すべて検索」では、検索ビューにファイルの一覧と各ファイルでの一致件数が表示されます。
- 6 見つかったテキストを置換したり、タグに対してアクションを実行する場合は、「置換」または「すべて置換」をクリックします。

検索オプションは、ColdFusion Builder の終了時に保存されます。

ColdFusion Builder の検索ビュー

ColdFusion Builder の検索ビューには、最新の検索結果が表示され、以前の検索履歴も保持されます。

ColdFusion Builder の検索ビュー

- 「すべて検索」をクリックした場合のみ、結果が表示されます。
- 結果はフォルダー階層で表示されます。ファイルをクリックすると、検索インスタンスを表示できます。
- 検索基準に一致する場合、コードの行番号が表示されます。行をダブルクリックすると、ファイルを開いて、該当する場所を表示できます。
- 一致が見つからなかったかどうかが表示されます。

その他のリソース

- [ColdFusion Builder Search\Replace](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sagar Ganatra が、ColdFusion Builder の検索機能について説明します。

- [Tag search](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sagar Ganatra が、検索基準に一致するタグを検索する方法について説明します。

タスクビューへのタスクの追加

タスクを CFML コメントとして定義すると、タスクビュー（ウィンドウ/ビューの表示/タスク）に、設定した優先度に基づいてタスクが表示されます。タスクは、デフォルトの TODO および FIXME タグ、またはカスタムタスクタグの接頭辞に従って表示されます。

タスクの追加

- 1 CFM または CFC ファイルをエディターで開きます。
- 2 次の例に示す形式でタスクを追加します。

```
<!--- TODO: Fix the bug 268451 --->
```

または cfscript 内の

```
//TODO: Fix the bug 268451
```

および

```
/*TODO: Fix the bug 268451*/
```

タスクの表示

タスクビュー（ウィンドウ/ビューの表示/タスク）には、CFML タスクコメントの最初の行が表示されます。

優先度の列は、タスクの優先度が高、標準、低のいずれであるかを示します。高優先度のタスクは赤色で表示されます。

開いているすべての CFM ファイルのタスクが表示されます。タスクは保存されるので、それ以降のセッションでファイルを開いたときにも表示されます。

カスタムタスクタグの作成

- 1 ウィンドウ/環境設定/ColdFusion/プロファイル/エディター/タスクタグに移動します。
- 2 「新規」をクリックします。

3 タスクタグを追加ダイアログボックスで、カスタムタスクタグの一意の名前を指定し、優先度を選択します。
コメントとして作成するタスク内でカスタムタグを使用します。

タスクの削除

ソースコードからタスクを手動で削除します。

これは、タスクビューで手動で作成したタスクと異なります。タスクビューで作成したタスクは、ビューで削除できます。

第6章：サーバーの管理

ColdFusion Builder の包括的なサーバー管理機能を使用すると、1つのサーバービューから、ColdFusion サーバーの起動や停止、再起動を実行したり、ColdFusion Administrator やサーバーモニターにアクセスしたりできます。サーバービューの使用方法について詳しくは、72 ページの「[サーバービューの使用](#)」を参照してください。

プロジェクト内のファイルをデバッグしたりプレビューしたりするには、サーバーを追加してプロジェクトに関連付けます。また、ColdFusion Builder に付属する拡張機能をインストールして実行する場合も、サーバーをプロジェクトに関連付けます。サーバーの追加方法について詳しくは、66 ページの「[ColdFusion サーバーの追加](#)」を参照してください。

Web サーバーの用語について

サーバーをセットアップして管理する前に、以下のトピックに目を通して、マニュアルで使用されている Web サーバーの用語と概念について理解しておいてください。

Web サーバーの詳細と、Web サーバーを ColdFusion で使用できるように設定する方法については、『ColdFusion 設定と管理』の Web サーバーの管理を参照してください。

ドキュメントルート

ドキュメントルートという用語は、Web サーバーから Web ページを提供するときに使用されるファイルシステムディレクトリを指します。この用語はサーバーの種類によって異なる場合がありますが、大部分の Web サーバーには同じ概念が当てはまります。

URL 接頭辞

URL 接頭辞は、ローカルファイルシステムのリソースを URL にマップします。

ColdFusion Builder では、Web ルートまたはドキュメントルートの外にあるプロジェクトをプレビューまたはデバッグする場合に、URL 接頭辞を使用します。

URL 接頭辞は、サーバーの作成時に指定することも、既存サーバーの設定を編集して指定することもできます。詳細については、67 ページの「[URL 接頭辞](#)」を参照してください。

既存のプロジェクトまたはフォルダーに対する URL 接頭辞を指定することもできます。詳細については、78 ページの「[URL 接頭辞の設定](#)」を参照してください。

使用例

「Project1」という名前のプロジェクトがあります。Project1 は、server1 に対して設定されています。server1 のドキュメントルートは C:\server1\MyDocs にあり、URL は http://www.example1.com です。Project1 の中には「xyz」という名前のリンクフォルダーがあります。フォルダー xyz は server2 のドキュメントルート C:\server2\MyDocs にリンクされており、このフォルダーにアクセスするための URL は http://www.example2.com です。ColdFusion Builder で xyz リンクフォルダー内のファイルをすべてプレビューする必要があるとします。

この場合、xyz リンクフォルダー内のファイルをプレビューするには、URL 接頭辞を作成します。URL 接頭辞を作成するには、次の情報を指定します。

- 絶対パス：C:\server2\MyDocs
- xyz フォルダーにアクセスするための URL：http://www.example2.com

仮想ホスト

仮想ホストという用語は、単一の Web サーバー上で複数の Web サイト（ドメイン名）をホストする方法を指します。複数の Web サイトは、識別可能なホスト名によって区別されます。例えば、単一の IP アドレスで `www.example1.com`、`www.example2.com`、および `www.example3.com` という Web サイトを運用できます。

ColdFusion Builder の場合は、単一の ColdFusion サーバーを使用して、IIS や Apache などの外部 Web サーバー上の仮想ホストとして設定された複数の Web サイトを運用できます。

Web サーバーで仮想ホストを設定する方法については、各 Web サーバーのマニュアルを参照してください。

プロジェクトを仮想ホストに関連付けると、ColdFusion Builder の機能（プレビュー、デバッグ、コンテンツアシスト、拡張機能の作成など）が、プロジェクト内のすべてのフォルダーとサブフォルダーに適用されます。

仮想ディレクトリ

仮想ディレクトリという用語は、ドキュメントルートに物理的には含まれていないが、サーバー URL を通じてアクセス可能なフォルダーを指します。仮想ディレクトリを作成するには、フォルダーのパスのエイリアスを URL で指定します。エイリアス名は、フォルダー内のリソースにアクセスするために使用されます。


Web サイト (`www.example.com`) のドキュメントルートが `c:\xyz\docs` であり、Web サイトにコンテンツを提供するフォルダーが `d:\abc\content` にあるとします。その場合は、このフォルダーのエイリアスを `content` という名前で定義します。これにより、`http://www.example.com/content/` という URL を使用して、Web サイトにアクセスできるようになります。

ColdFusion サーバーの追加

ColdFusion Builder ワークスペース内に作成したプロジェクトやアプリケーションをテストするには、ColdFusion サーバーを追加します。この ColdFusion サーバーは、本稼動サーバーにデプロイする前にアプリケーションをテストして実行するための開発サーバーとして使用できます。ローカルとリモート両方の ColdFusion サーバーを追加できます。

ローカルサーバーの追加

サーバービューで、次のいずれかの操作を行います。

- 右クリックして「サーバーの追加」を選択します。
-  をクリックします。

新規 ColdFusion サーバーセットアップウィザードで次の情報を入力します。

一般設定

- サーバー名：ColdFusion サーバーの名前。
- 説明：（オプション）サーバーの説明。
- アプリケーションサーバー：ドロップダウンリストを選択して「JRun」を選択するか、「その他」を選択して Jrun 以外のサーバーを設定します。

注意： ColdFusion Builder 内で JRun 以外のサーバーを起動、停止、または再起動することはできません。

- ホスト名：ColdFusion サーバーホストの名前。例えば、`localhost` や `127.0.0.1` のように入力します。
- 「ローカル」を選択します。

その他の設定

- **Webserver ポート**：設定する ColdFusion サーバーインスタンスのポート番号を指定します。ColdFusion サーバーのデフォルトポート番号は 8500 です。
- **コンテキストルート**：(JRun サーバーを使用する J2EE 設定のみに適用) コンテキストルートを入力します。J2EE 環境では、1 つのサーバーインスタンス上で、独立した複数の Web アプリケーションを実行できます。したがって、サーバー上で実行される J2EE Web アプリケーションのルートは、それぞれ固有のベース URL に設定されます。これが、コンテキストルート (またはコンテキストパス) と呼ばれるものです。
- **アプリケーションサーバー名**：(JRun サーバーを使用する J2EE 設定のみに適用) ColdFusion をデプロイする JRun サーバーの名前。
- **RDS ユーザー名**：(オプション) RDS を使用している場合は、RDS ユーザー名を指定します。
- **RDS パスワード**：(オプション) RDS パスワードを指定します。

注意：RDS パスワードの設定は ColdFusion Administrator で行います。同じく ColdFusion Administrator で管理される ColdFusion Administrator パスワードと RDS パスワードを混同しないでください。

- ColdFusion Builder で SSL のサポートを有効にするには、「SSL の有効化」を選択します。Server Manager に登録されているサーバーとの通信には SSL を使用できます。
- ColdFusion Builder を起動または終了するたびに ColdFusion サーバーが自動的に起動または停止されるようにするには、「自動起動」および「自動停止」を選択します。

「次へ」をクリックします。

ローカルサーバーの設定

「ローカルサーバーの設定」タブを選択し、必要に応じて次のローカルサーバー設定を指定します。

- **サーバーホーム**：(JRun サーバーのみに適用) ColdFusion サーバーのホームディレクトリを参照して選択します。例えば、スタンドアローンの ColdFusion サーバーインスタンスの場合は C:\ColdFusion9 です。マルチサーバーインスタンスの場合、サーバーホームはアプリケーションサーバーのホームディレクトリです。例えば、JRun-J2EE またはマルチサーバー設定の場合は C:\JRun4 です。
- **ドキュメントルート**：Web ルートの場所を参照して選択します。ColdFusion を IIS などの Web サーバーとともに設定する場合は、Web サーバーのドキュメントルート (c:\inetpub\wwwroot など) を選択します。
この設定は、ColdFusion Builder でプレビュー、デバッグ、および CFC 名の解決を行うために必要です。
- **バージョン**：(Windows で動作する JRun サーバーのみに適用) バージョンドロップダウンリストから ColdFusion サーバーのバージョンを選択します。
- **Windows サービス**：(Windows で動作する JRun サーバーのみに適用) 「Windows サービス」オプションは、スタンドアローン設定およびマルチサーバー設定の場合にのみ使用でき、J2EE 設定では使用できません。Windows Service を使用して ColdFusion サーバーの起動と停止を行う場合は、「Windows のサービスを使用してサーバーを起動または停止する」を選択します。

URL 接頭辞

(オプション)

注意：必ずしも、サーバーの作成時に URL を指定する必要はありません。サーバーを作成した後でも、サーバービューでサーバー設定を編集することで、URL 接頭辞を指定できます。既存のプロジェクトまたはフォルダーに対する URL 接頭辞を指定することもできます。詳細については、78 ページの「[URL 接頭辞の設定](#)」を参照してください。

「URL 接頭辞」タブを選択して、次の情報を入力します。

- 1 **ローカルパス**：ローカルファイルシステムリソースのパスを参照するか、手動で入力します。

- 2 URL 接頭辞：URL 接頭辞を入力します。
- 3 「追加」をクリックします。

仮想ホストの設定 (オプション)

注意：ColdFusion Builder で仮想ホストまたは仮想ディレクトリを指定する場合は、設定済みの Web サーバーで、対応する設定を指定する必要があります。ColdFusion Builder ではこれらの設定は検証されません。そのため、Web サーバーの設定が ColdFusion Builder で指定した内容と異なっていても、ColdFusion Builder はエラーを生成しません。

仮想ホストを設定するには、「仮想ホストの設定」タブを選択して、次のようにします。

- 1 「新規」をクリックし、「仮想ホストの設定」セクションで次の情報を入力します。
 - a 名前：仮想ホストの名前を指定します。(vh1 など)
注意：任意の名前を指定できます。必ずしも Web サーバーで指定した名前でなくてもかまいません。
 - b ホスト名：IIS または Apache Web サーバーの設定で指定したとおりの仮想ホスト名。(www.example1.com など)
ColdFusion Builder で仮想ホストを作成した場合、その仮想ホストでは **server name-virtual host name** というネーミング規則が使用されます。プロジェクトプロパティのサーバードロップダウンリストには、このネーミング規則を使用した仮想ホスト名が表示されます。例えば、ColdFusion サーバー (localhost) で「vh1」という名前の仮想ホストを作成した場合、仮想ホストを識別するために ColdFusion Builder で使用されるネーミング規則は、「localhost-vh1」です。
 - c ポート：Web サーバーの仮想ホストに割り当てられたポート。
 - d タイプ：ドロップダウンリストから「HTTP」または「HTTPS」を選択します。
 - e ドキュメントルート：仮想ホストのホームディレクトリの場所を参照するか、手動で入力します。
例えば、Web サイト www.example.com が Apache Web サーバー上の C:\abc ディレクトリにマップされている場合は、ホームディレクトリとして「C:\abc」と入力します。
- 2 (オプション) 仮想ディレクトリを作成するには、「仮想ディレクトリ」をクリックして次の情報を入力します。
 - a エイリアス：フォルダーパスのエイリアスを指定します。
 - b 場所：エイリアスを指定するフォルダーパスを参照するか、手動で入力します。
例えば、Web サイト (www.example.com) のドキュメントルートが c:\abc であり、d:\xyz\images でアクセス可能なフォルダー内のイメージを使用する必要があるとします。その場合は、フォルダーパス「d:\xyz\images」に対して「images」という名前のエイリアスを定義します。仮想ディレクトリの詳細については、66 ページの「[仮想ディレクトリ](#)」を参照してください。
 - c 「追加」をクリックします。
 - d 「OK」をクリックすると、仮想ディレクトリが仮想ディレクトリの設定テーブルに追加されます。
- 3 「適用」をクリックします。仮想ホストが仮想ホストテーブルに追加されます。

仮想ホストの設定を変更するには、仮想ホストテーブルから仮想ホストを選択し、設定を変更して、「更新」をクリックします。

仮想ホストの詳細と、ColdFusion Builder での有用性については、66 ページの「[仮想ホスト](#)」を参照してください。

拡張機能のインストール

ColdFusion Builder に付属する拡張機能をインストールするには、「拡張機能のインストール」を選択します。

- 1 ColdFusion Web ルートの場所を参照して選択します。

- 2 拡張機能をインストールする Web ルート内の場所を参照します。拡張機能は、選択した場所の Extensions ディレクトリにインストールされます。

「終了」をクリックすると、ローカルの ColdFusion サーバーインスタンスが作成されます。これらの拡張機能の使用方法については、121 ページの「[拡張機能の使用](#)」を参照してください。

リモートサーバーの追加

ColdFusion Builder でリモートサーバーを追加する前に、次の作業を行います。

リモート ColdFusion サーバーでの Admin サーバーインスタンスの実行

リモートサーバーのバージョンと設定に応じて、次の作業を行います。

スタンドアローンのリモートサーバー (バージョン 7.0.2、8.0.1)

- 1 AdminServerComponents.zip ファイルを {CFHome} に解凍します。

次のファイルが {cfhome}/runtime/bin にコピーされます。

- adminstart.bat
- admin_jvm.config
- adminstart.sh

Admin Server インスタンスが {cfhome}/runtime/servers にコピーされます。

- 2 {cfhome}/runtime/bin に移動し、adminstart.bat (Windows の場合) または adminstart.sh (Mac OS の場合) を実行します。

マルチサーバーまたは J2EE のリモートサーバー (バージョン 7.0.2、8.0.1)

- 1 AdminServerComponents.zip ファイルから、次のファイルを JRun_Home/bin に解凍します。

- admin_jvm.config
- adminstart.bat (Windows の場合) または adminstart.sh (Mac OS の場合)

- 2 JRun_Home/bin に移動し、adminstart.bat (Windows の場合) または adminstart.sh (Mac OS の場合) を実行します。

スタンドアローン、マルチサーバー、J2EE のリモートサーバー (バージョン 9)

リモートサーバーがバージョン 9 の場合は、Admin Server コンポーネントがデフォルトで含まれています。


- スタンドアローンサーバーの場合は、{cfhome}/runtime/bin に移動して、adminstart スクリプトファイルを実行します。
- マルチサーバーまたは J2EE サーバーの場合は、JRun_Home/bin に移動して、adminstart スクリプトファイルを実行します。

リモート ColdFusion サーバーのセキュリティプロパティの更新

- 1 {CFHome}/runtime/lib/security.properties に移動します。
- 2 jrun.subnet.restriction および jrun.trusted.hosts の値を、ColdFusion Builder がインストールされているコンピューターの IP アドレスに変更します。また、IP アドレスの値としてアスタリスクワイルドカード (*) を使用すると、制約なしでサーバーを起動および停止できるようになります。

ColdFusion Builder でのリモートサーバー設定の指定

サーバービューで、次のいずれかの操作を行います。

- 右クリックして「サーバーの追加」を選択します。
-  をクリックします。

新規 ColdFusion サーバーセットアップウィザードで次の情報を入力します。

一般設定

- サーバー名：ColdFusion サーバーの名前。
- 説明：(オプション) サーバーの説明。
- アプリケーションサーバー：ドロップダウンリストを選択して「JRun」を選択するか、「その他」を選択して Jrun 以外のサーバーを設定します。

注意：ColdFusion Builder 内で JRun 以外のサーバーを起動、停止、または再起動することはできません。

- ホスト名：リモートサーバーホストの名前。
- 「リモート」を選択します。

注意：localhost または 127.0.0.1 以外のホスト名を入力した場合、「リモート」は自動的に選択されます。

その他の設定

- Webserver ポート：設定する ColdFusion サーバーインスタンスのポート番号を指定します。ColdFusion サーバーのデフォルトポート番号は 8500 です。
- コンテキストルート：(JRun サーバーを使用する J2EE 設定のみに適用) コンテキストルートを入力します。
J2EE 環境では、1 つのサーバーインスタンス上で、独立した複数の Web アプリケーションを実行できます。したがって、サーバー上で実行される J2EE Web アプリケーションのルートは、それぞれ固有のベース URL に設定されます。これが、コンテキストルート (またはコンテキストパス) と呼ばれるものです。
- アプリケーションサーバー名：(JRun サーバーを使用する J2EE 設定のみに適用) ColdFusion をデプロイする JRun サーバーの名前。
- RDS ユーザー名：(オプション) RDS を使用している場合は、RDS ユーザー名を指定します。
- RDS パスワード：(オプション) RDS パスワードを指定します。

注意：RDS パスワードの設定は ColdFusion Administrator で行います。同じく ColdFusion Administrator で管理される ColdFusion Administrator パスワードと RDS パスワードを混同しないでください。

- ColdFusion Builder で SSL のサポートを有効にするには、「SSL の有効化」を選択します。Server Manager に登録されているサーバーとの通信には SSL を使用できます。
- ColdFusion Builder を起動するたびに ColdFusion サーバーを自動的に起動させるには、「自動起動」を選択します。

注意：リモートサーバーの場合、「自動停止」は使用できません。

「次へ」をクリックします。

リモートサーバーの設定

- 1 ネーミングポート：リモートサーバーで実行されている administrator サーバーインスタンスのネーミングポートを指定します。デフォルトのネーミングポート値は 2910 です。ネーミングポート値は {servers}/admin/SERVER-INF/jndi.properties で指定されています。プロパティ java.naming.provider.url のポート値は、ネーミングポートまたは jndiport です。
- 2 ユーザー名：jrun-users.xml ファイルに記載されている jmc ユーザー名を指定します。

- 3 パスワード：jmc パスワードを指定します。
- 4 ドキュメントルート：Web ルートの場所を参照して選択します。ColdFusion を IIS などの Web サーバーとともに設定する場合は、Web サーバーのドキュメントルート (c:\inetpub\wwwroot など) を選択します。
- 5 「マッピング」タブを選択し、次の情報を入力します。マッピングの情報は、リモートサーバーのファイルをプレビューするときやデバッグするときを使用されます。

- ローカルパス：リモート ColdFusion サーバー上でプロジェクトまたはフォルダーを探すときに ColdFusion Builder で使用するパス。
- リモートパス：(URL 接頭辞を指定する場合はオプション) リモート ColdFusion サーバー上のプロジェクトのパス。
- URL 接頭辞：(リモートパスを指定する場合はオプション) URL 接頭辞を入力します。URL 接頭辞の詳細と、URL 接頭辞の使用例については、65 ページの「[URL 接頭辞](#)」を参照してください。

注意：必ずしも、サーバーの作成時に URL を指定する必要はありません。サーバーを作成した後でも、サーバービューでサーバー設定を編集することで、URL 接頭辞を指定できます。既存のプロジェクトまたはフォルダーに対する URL 接頭辞を指定することもできます。詳細については、78 ページの「[URL 接頭辞の設定](#)」を参照してください。

- 「追加」をクリックしてこのマッピングを追加します。

- 6 (オプション) 仮想ホストを設定するには、「仮想ホストの設定」タブを選択して、次のようにします。

注意：ColdFusion Builder で仮想ホストまたは仮想ディレクトリを指定する場合は、設定済みの Web サーバーで、対応する設定を指定する必要があります。ColdFusion Builder ではこれらの設定は検証されません。そのため、Web サーバーの設定が ColdFusion Builder で指定した内容と異なっても、ColdFusion Builder はエラーを生成しません。

- 「新規」をクリックし、「仮想ホストの設定」セクションで次の情報を入力します。

- 1 名前：仮想ホストの名前を指定します。(vh1 など)

注意：任意の名前を指定できます。必ずしも Web サーバーで指定した名前でもなくともかまいません。

- 2 ホスト名：IIS または Apache Web サーバーの設定で指定したとおりの仮想ホスト名。(www.example1.com など)

ColdFusion Builder で仮想ホストを作成した場合、その仮想ホストでは **server name-virtual host name** というネーミング規則が使用されます。プロジェクトプロパティのサーバードロップダウンリストには、このネーミング規則を使用した仮想ホスト名が表示されます。例えば、ColdFusion サーバー (localhost) で「vh1」という名前の仮想ホストを作成した場合、仮想ホストを識別するために ColdFusion Builder で使用されるネーミング規則は、「localhost-vh1」です。

- 3 ポート：Web サーバーの仮想ホストに割り当てられたポート。

- 4 タイプ：ドロップダウンリストから「HTTP」または「HTTPS」を選択します。

- 5 ドキュメントルート：仮想ホストのホームディレクトリの場所を参照するか、手動で入力します。

例えば、Web サイト www.example.com が Apache Web サーバー上の C:\abc ディレクトリにマップされている場合は、ホームディレクトリとして「C:\abc」と入力します。

- (オプション) 仮想ディレクトリを作成するには、「仮想ディレクトリ」をクリックして次の情報を入力します。

- 1 エイリアス：フォルダーパスのエイリアスを指定します。

- 2 場所：エイリアスを指定するフォルダーパスを参照するか、手動で入力します。

例えば、Web サイト (www.example.com) のドキュメントルートが c:\¥abc であり、d:\¥xyz¥images でアクセス可能なフォルダー内のイメージを使用する必要があるとします。その場合は、フォルダーパス「d:\¥xyz¥images」に対して「images」という名前のエイリアスを定義します。仮想ディレクトリの詳細については、66 ページの「[仮想ディレクトリ](#)」を参照してください。

3 「追加」をクリックします。

4 「OK」をクリックすると、仮想ディレクトリが仮想ディレクトリの設定テーブルに追加されます。

- 「適用」をクリックします。仮想ホストが仮想ホストテーブルに追加されます。

仮想ホストの設定を変更するには、仮想ホストテーブルから仮想ホストを選択し、設定を変更して、「更新」をクリックします。

仮想ホストの詳細と、ColdFusion Builder での有用性については、66 ページの「[仮想ホスト](#)」を参照してください。

「次へ」をクリックします。

拡張機能のインストール

ColdFusion Builder に付属する拡張機能をインストールするには、「拡張機能のインストール」を選択します。

1 ColdFusion Web ルートの場所を参照して選択します。

2 リモート ColdFusion サーバー上の ColdFusion Web ルートの場所を参照して選択します。

3 拡張機能をインストールする Web ルート内の場所を参照します。拡張機能は、選択した場所の Extensions ディレクトリにインストールされます。

「終了」をクリックすると、リモート ColdFusion サーバーインスタンスが追加されます。これらの拡張機能の使用方法については、121 ページの「[拡張機能の使用](#)」を参照してください。

リモートサーバーへの接続に成功すると、ColdFusion Builder に表示されるサーバーステータスが実行中に変ります。リモートサーバーへの接続に成功しなかった場合は、サーバーステータスが不明に変ります。エラーの詳細については、36 ページの「[コンソールビュー](#)」を参照してください。

サーバービューの使用

サーバービューは、ColdFusion Builder からサーバーの起動、停止、再起動を行う場合に使用します。また、サーバービューからは、ColdFusion サーバーモニターと ColdFusion Server Administrator も起動できます。

サーバービューがワークベンチに表示されていない場合は、ウィンドウ／ビューの表示／その他を選択して、サーバービューを追加します。ビューの表示ダイアログボックスで、ColdFusion /サーバービューを選択します。

サーバーの起動、停止、再起動、削除

1 サーバービューでサーバーを右クリックします。

2 「サーバーを起動」、「サーバーを停止」、「サーバーを再起動」、「サーバーを削除」のいずれかを選択します。「サーバーを再起動」を選択すると、サーバーが停止した後に起動されます。

ColdFusion サーバーモニターの起動

1 サーバービュー内を右クリックします。

2 「サーバーモニターの起動」を選択して ColdFusion Administrator を開きます。

3 パスワードを指定します。

- 4 「OK」をクリックしてサーバーモニターを表示します。

ColdFusion Server Administrator の起動

- 1 サーバービュー内を右クリックします。
- 2 「ColdFusion Administrator を起動」を選択します。
- 3 ユーザー名とパスワードを指定します。
- 4 「OK」をクリックして、ColdFusion Administrator を表示します。

RDS サーバー設定のインポート

RDS サーバーが既に設定されている場合は、RDS サーバーの設定を直接インポートして、RDS サーバーをサーバービューに追加できます。

- 1 サーバービューを右クリックして、「サーバーの追加」を選択します。
- 2 「RDS サーバーから設定をインポート」を選択し、サーバービューに追加する RDS サーバーをドロップダウンリストから選択します。
- 3 「OK」をクリックします。
- 4 RDS 設定を変更するには、ColdFusion サーバーセットアップの変更ダイアログボックスで変更を指定して、「終了」をクリックします。

これで、サーバーをプロジェクトに関連付けて、プロジェクトのデバッグ、プレビュー、テストを行えるようになります。

第7章：プロジェクトの管理

プロジェクトについて

プロジェクトには、ColdFusion アプリケーションの開発に使用される ColdFusion コンポーネント、インターフェイス、HTML ページ、CFML ページなどのリソースが格納されています。

各プロジェクトはデフォルトのワークスペースに保存されます。ワークスペースには、プロジェクトと他のメタデータが保存されます。推奨されるワークスペースの場所は、ColdFusion のドキュメントルートです。

プロジェクトを ColdFusion サーバーに関連付けることができます。そのようにすれば、最終的なデプロイの前にプロジェクトをテストできます。

注意：プロジェクトに含まれるファイルを Web ブラウザーでプレビューするには、プロジェクトの場所が ColdFusion のドキュメントルートに設定されていることを確認します。

ColdFusion Builder で CFML を開発するには、最初に ColdFusion プロジェクトを作成します。ColdFusion Builder でプロジェクトを作成する方法については、74 ページの「[ColdFusion プロジェクトの作成](#)」を参照してください。

ColdFusion プロジェクトの作成

プロジェクトビルダーウィザードの案内に従って、ColdFusion プロジェクトを作成します。

プロジェクト情報の入力

- 1 ナビゲーター領域内を右クリックし、新規 / ColdFusion プロジェクトをクリックします。
- 2 プロジェクトビルダーウィザードでプロジェクト名を指定します。
- 3 デフォルトのプロジェクトの場所を変更するには、「デフォルトの場所を使用」を選択解除します。
- 4 「次へ」をクリックして ColdFusion サーバーの詳細を指定します。

サーバーの詳細の入力

- 1 サーバーポップアップメニューからサーバーを選択します。まだ ColdFusion サーバーが設定されていない場合は、「サーバーの追加」をクリックしてサーバーを追加します。ColdFusion サーバーをセットアップする方法については、66 ページの「[ColdFusion サーバーの追加](#)」を参照してください。

注意：プロジェクトがサーバーの Web ルートにある場合は、「サンプル URL」ボックスにサーバーの URL が自動的に入力されます。例えば、`http://127.0.0.1:8500/eval` のように入力されます（127.0.0.1 はサーバーホスト、8500 はポート番号、eval はプロジェクト名です）。

- 2 コンピューターにインストールされている Web ブラウザーを選択して、外部 Web ブラウザーを使用するようにプレビュー設定を指定します。デフォルトでは、内部ブラウザが選択されています。
- 3 「次へ」をクリックします。

既存のソースの追加

1 この手順では、次の操作を行えます。

- 既存のリソースフォルダーをプロジェクトにリンクする。
- 以前に設定したアプリケーションを現在のプロジェクトで選択する。

2 「追加」をクリックして、プロジェクトにリンクするフォルダーを選択します。

3 「終了」をクリックして、新しい ColdFusion プロジェクトを構築します。

リソースをリンクする方法については、80 ページの「[ワークスペース外のリソースへのリンク](#)」を参照してください。

プロジェクトのプロパティおよびサーバーの設定

ColdFusion Builder の プロパティダイアログボックスを使用すると、プロジェクトおよびサーバーのプロパティを設定できます。

このダイアログボックスを開くには、ナビゲータービューでプロジェクトを右クリックして「プロパティ」を選択します。

プロジェクトのプロパティの設定

❖ プロパティダイアログボックスの左ペインで「ColdFusion プロジェクト」を選択します。ここで、外部プロジェクトをリンクとして追加および削除できます。

変数マッピングの設定

編集中のファイルやインクルードファイルで定義されていないコンポーネントの引数にコードアシストを用意するには、変数マッピングを使用します。

- 1 プロパティダイアログボックスの左側にツリービュー構造が表示されます。「ColdFusion 変数マッピング」を選択します。
- 2 「新規」をクリックします。
- 3 「変数名」フィールドで、マッピングする変数の名前（例えば `application.cfc1`）を指定します。
- 4 「マッピング対象」フィールドに、プロジェクト変数の完全修飾名（例えば `com.adobe.mycfcs.cfc1`）を入力します。

サーバー設定の指定

- プロパティダイアログボックスの左ペインで「ColdFusion サーバーの設定」を選択すると、ColdFusion サーバーの設定ページが開きます。
- ColdFusion サーバーのインスタンスを作成するには、サーバーポップアップメニューを使用します。

設定済みのサーバーインスタンスがある場合は、そのインスタンスを ColdFusion プロジェクトに割り当てることができます。サーバーインスタンスを作成する方法については、66 ページの「[ColdFusion サーバーの追加](#)」を参照してください。

ColdFusion ページ、インターフェイス、およびコンポーネントの追加

ColdFusion Builder には、ColdFusion ページ、インターフェイス、およびコンポーネントを作成するためのウィザードが用意されています。

ColdFusion ページの作成

- 1 ナビゲーターで ColdFusion プロジェクトを右クリックします。
- 2 新規 / ColdFusion ページをクリックします。
- 3 新規 ColdFusion ページウィザードで、ColdFusion (CFM) ページの名前を指定します。
- 4 「終了」をクリックします。

ColdFusion コンポーネントの作成

- 1 ナビゲーターで ColdFusion プロジェクトを右クリックします。
- 2 新規 / ColdFusion コンポーネントをクリックします。
- 3 新規 ColdFusion コンポーネントウィザードで、次の項目を指定します。
 - コンポーネント名
 - コンポーネントを識別するためのヒント (オプション)
 - 拡張するコンポーネント
 - 実装するインターフェイス
- 4 出力の型を true または false に設定します。
- 5 「次へ」をクリックします。
- 6 「追加」をクリックして、コンポーネントのプロパティの詳細 (プロパティ名、表示名、ヒント、デフォルト値など) を指定します。
- 7 タイプドロップダウンリストからプロパティの型を選択するか入力します。
- 8 (オプション) 「getter の追加」を選択し、アクセスドロップダウンリストからアクセスタイプを選択します。
- 9 (オプション) 「setter の追加」を選択し、アクセスドロップダウンリストからアクセスタイプを選択します。
- 10 「OK」をクリックして、新規 ColdFusion コンポーネントウィザードに戻ります。
- 11 ColdFusion コンポーネントに関数を追加するには、「関数の追加」をクリックします。
- 12 新規関数ウィザードで、関数名、表示名、ヒント、アクセスタイプ、戻り値の型、役割、出力の型などの関数の詳細を指定します。
- 13 「OK」をクリックします。
- 14 選択した関数に引数を追加するには、「引数の追加」をクリックします。
- 15 新規関数引数ダイアログボックスで、引数名、表示名、ヒント、引数の型、および引数のデフォルト値を指定します。

注意: 選択できる引数の型は、ColdFusion インターフェイスの引数の型と同じです。
- 16 「OK」をクリックします。
- 17 「終了」をクリックして ColdFusion コンポーネント (CFC ファイル) を作成します。

注意: アウトラインビューを使用すると、ColdFusion ページまたは ColdFusion コンポーネントに追加した関数やタグを参照できます。

ColdFusion インターフェイスの作成

- 1 ナビゲーターで ColdFusion プロジェクトを右クリックします。
- 2 新規 / ColdFusion インターフェイスをクリックします。
- 3 新規 ColdFusion インターフェイスウィザードで、インターフェイス名、ヒント、および表示名を指定します。
- 4 別のインターフェイスを拡張するには、CFC の名前を参照または指定します。
- 5 関数を追加するには、「関数の追加」をクリックし、関数名、関数のヒント、戻り値の型、役割、出力 (true または false) などの詳細を指定します。
- 6 「引数の追加」をクリックして、選択した関数の引数を指定します。タイプドロップダウンリストから引数の型を選択するか入力します。
- 7 「終了」をクリックして ColdFusion インターフェイスを作成します。

その他のファイルの追加

ColdFusion プロジェクトには次のファイルも追加できます。

- .html
- .js
- .css
- .lxr (レクサーファイル)
- .col (色付けファイル)
- .sdoc (ScriptDoc ファイル)
- 表題なしの上記ファイル

また、汎用的なテキストファイル、フォルダー、プロジェクトも作成できます。これらのファイルをプロジェクトに追加する手順は、ColdFusion ページや ColdFusion コンポーネントを追加する手順とほぼ同じです。これらのファイルをプロジェクトに追加するには、次のようにします。

- 1 プロジェクトを右クリックします。
- 2 新規 / その他を選択します。
- 3 「プロジェクトファイル」を展開してファイル名を指定するか、「表題なしのファイル」を展開して表題なしのファイルを作成します。
- 4 作成するファイルを選択します。
 - 表題なしのファイルを作成する場合は、「終了」をクリックします。
 - 「プロジェクトファイル」を選択した場合は、「次へ」をクリックしてファイル名を指定します。「終了」をクリックします。

CFM または CFC ファイルのワークスペース外への作成

ファイルビューを使用して、CFM または CFC ファイルをワークスペースの外に作成することができます。

- 1 ファイルビューで、CFM または CFC ファイルを作成するローカルフォルダーを右クリックします。
- 2 新規 / ColdFusion ページ / ColdFusion コンポーネント / ColdFusion インターフェイスを選択し、詳細を指定します。

起動ページの設定

特定のファイルをプロジェクトの起動ページとして指定することができます。プロジェクト内のファイルを実行したりデバッグしたりすると、このページが読み込まれます。

- 1 ナビゲータービューで、開始ページとして設定するページを選択します。
- 2 ページを右クリックして、「開始ページとして設定」を選択します。

ページの起動ページ指定を無効にするには、

- 1 プロジェクトビューで、プロジェクトを右クリックし、プロパティ / ColdFusion プロジェクトを選択します。
- 2 「開始ページの設定」の「開始ページを使用」を選択解除します。

拡張機能を使用してフレームワークアプリケーションの開始ページの URL を動的に生成

119 ページの「[拡張機能による起動ページ設定のサポート](#)」を参照してください。

URL 接頭辞の設定

- 1 ナビゲータービューで、URL 接頭辞を指定するプロジェクトまたはフォルダーを選択します。
- 2 選択したプロジェクトまたはフォルダーを右クリックし、「URL 接頭辞の設定」を選択します。
- 3 選択したリソースの URL を入力します。

プロジェクトのインポート、エクスポート、および削除

プロジェクトのインポート

ColdFusion Builder には、ColdFusion プロジェクトだけでなく、ColdFusion 以外のプロジェクトもインポートできます。

- 1 ナビゲータービュー内を右クリックし、「インポート」を選択します。
- 2 インポートウィザードで、ColdFusion / 既存プロジェクトのインポートを選択します。
- 3 ColdFusion プロジェクトのインポートダイアログボックスで、プロジェクトを参照して選択します。すべてのプロジェクトのリストが表示されます。
- 4 「更新」をクリックするとプロジェクトのリストが更新されます。
- 5 同じ場所にある ColdFusion 以外のプロジェクトや ColdFusion Builder 以外のプロジェクトもすべて表示するには、「すべてのプロジェクトを表示」チェックボックスを選択します。

6 プレビュー、編集、デバッグなど ColdFusion Builder 固有の機能を ColdFusion Builder 以外のプロジェクトに適用するには、「ColdFusion Nature を非 ColdFusion プロジェクトに追加」を選択します。

7 適切なプロジェクトを選択して、「終了」をクリックします。

注意：プロジェクトのインポートでは、サーバーの詳細、起動設定、ブラウザーの詳細、アプリケーションの詳細、リンク先フォルダーなどのプロジェクトプロパティが保持されます。

プロジェクトのエクスポート

ColdFusion Builder プロジェクトまたは ColdFusion Builder 以外のプロジェクトから指定の場所にファイルとフォルダーをエクスポートできます。ただし、ColdFusion Builder プロジェクトの場合は、サーバー設定や環境設定など、ColdFusion Builder 固有の機能はエクスポートされません。

- 1 ナビゲータービュー内を右クリックし、「エクスポート」を選択します。
- 2 エクスポートウィザードで、ColdFusion /エクスポートを選択します。
- 3 プロジェクトのエクスポートダイアログボックスで、プロジェクトをエクスポートする場所を参照して選択します。
- 4 「すべてを選択」をクリックして、プロジェクト内のファイルとフォルダーをすべてエクスポートするか、「タイプをフィルター」をクリックして、エクスポートするファイルを指定します。
- 5 「終了」をクリックすると、指定した場所にプロジェクトがエクスポートされます。

プロジェクトの削除

プロジェクトを削除する場合は、現在のワークスペースからプロジェクトを削除します。同時にコンピューターのファイルシステムからプロジェクトを削除することもできます。

- 1 ナビゲータービュー内を右クリックし、「削除」を選択します。
- 2 ワークスペースとファイルシステムの両方からプロジェクトを削除するには、「ディスク上からプロジェクト・コンテンツを削除」を選択します。ファイルシステムからプロジェクトを完全に削除した後に、元に戻すことはできません。

プロジェクトおよびファイルの除外

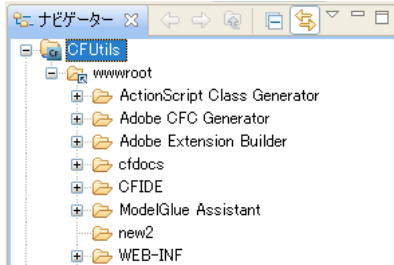
ColdFusion Builder では、特定のファイル、ファイルタイプ、およびフォルダーを、同期、アップロード、ダウンロード操作の対象から除外する（隠す）ことができます。この機能は、同期やアップロードの対象から除外するファイルを大量に含むディレクトリがある場合に便利です。

ファイルやフォルダーを除外する方法には、次の 2 つがあります。

- ナビゲータービューでファイルまたはフォルダーを右クリックして、同期 / このファイルタイプを除外を選択します。ファイルタイプの除外を解除するには、ファイルまたはフォルダーを右クリックして、同期 / このファイルタイプの除外を解除を選択します。
- 次の手順に従い、ワイルドカードや正規表現を使用して特定のファイルタイプを除外するように設定をセットアップします。
 - 1 ナビゲータービューでファイルまたはフォルダーを右クリックして、同期 / 拡張除外設定を選択します。
 - 2 環境設定ダイアログボックスで (+) アイコンをクリックします。
 - 3 ファイル / フォルダーを無視ダイアログボックスで、ファイル名またはフォルダー名を指定します。
 - 4 「OK」をクリックします。除外されるファイルのリストに、指定したファイルまたはフォルダーが表示されます。

ワークスペース外のリソースへのリンク

プロジェクトやワークスペースの外部にあるリソースにもリンクできます。この機能は、共用の場所にあるリソースを使用する場合に役に立ちます。リンク先のリソース、またはリンク先のリソースを含んでいるフォルダーは、次のように表示されます。



ワークスペース外のリソースへのリンク

- 1 ColdFusion Builder のナビゲータービューで、リンク対象のリソースを追加するプロジェクトを右クリックします。
- 2 リンクするリソースを選択します。例えば、フォルダーにリンクする場合は、新規/フォルダーを選択します。
- 3 リンクするリソースの名前を入力します。ここで名前を入力しなかった場合は、リソースのデフォルト名が割り当てられます。
- 4 「拡張」をクリックします。
- 5 「ファイル・システム内のフォルダーにリンク」を選択します。リソースの場所を参照して選択するか、手動で入力します。
- 6 「終了」をクリックして、リソースをプロジェクトにリンクします。

パス変数を使用したリソースへのリンク

リソースにリンクするときは、リソースが保存されている場所の絶対パスを指定する代わりに、パス変数を定義することもできます。パス変数を定義した後に、リンクするリソースのパスを設定できます。

- 1 ColdFusion Builder のナビゲータービューで、リンク対象のリソースを追加するプロジェクトを右クリックします。
- 2 リンクするリソースを選択します。例えば、フォルダーにリンクする場合は、新規/フォルダーを選択します。
- 3 リンクするリソースの名前を入力します。ここで名前を入力しなかった場合は、リソースのデフォルト名が割り当てられます。
- 4 「拡張」をクリックします。
- 5 「ファイル・システム内のフォルダーにリンク」を選択します。リソースの場所を参照して選択するか、手動で入力します。
- 6 「変数」をクリックします。
- 7 既存のパス変数を選択するか、「新規」をクリックしてパス変数を作成します。

注意：既存のリソース変数のリストは、メインメニューのウィンドウ/環境設定を選択してから、一般/ワークスペース/リンクされたリソースを選択して表示することもできます。リンクされたリソースダイアログボックスでは、リンクされたリソース変数の編集と作成を行えます。

- 8 パス変数を作成するには、リソースの名前を入力し、リソースの場所を参照して選択するか、絶対パスを入力します。「OK」をクリックして、パス変数を追加します。
- 9 「終了」をクリックして、リソースをプロジェクトにリンクします。

FTP および Secure FTP 接続によるプロジェクトのデプロイ

ColdFusion Builder では、ファイル転送プロトコル (FTP) を使用して ColdFusion プロジェクトを Web サーバーにデプロイできます。

また、ColdFusion Builder は、Secure File Transfer Protocol (SFTP) 接続および File Transfer Protocol Secure (FTPS) 接続を使用したファイル転送もサポートしています。SFTP および FTPS は、セキュリティ機能を備えた FTP であり、リモートサーバーとの間で送受信されるファイルが暗号化されます。

ColdFusion Builder での FTP 接続または Secure FTP 接続の作成

- 1 ナビゲータービューでプロジェクトを右クリックします。
- 2 同期/同期接続を新規作成をクリックします。
- 3 接続の名前を指定します。
- 4 ローカルドロップダウンリストを選択し、ファイルまたはプロジェクトが含まれる場所を選択して、デプロイする ColdFusion プロジェクトまたはファイルを選択します。
- 5 リモートサーバーを選択し、新規接続ダイアログボックスで次の情報を入力します。
 - a サイト名：新しいリモートサイトの名前。
 - b 接続タイプを選択します。
 - c サーバー：接続先サイトの URL (admin.myWebSite.com など)。サイトの IP アドレスを入力することもできます。
 - d ユーザー名/パスワード：サーバーのユーザー名およびパスワード。
 - e 公開鍵認証を使用：このオプションは、接続タイプとして SFTP を選択した場合にのみ使用できます。公開鍵認証では、コンピューター上の鍵ペアを選択して、公開鍵をサーバーにコピーできます。
 - f リモートパス：リモートサイトのルートの下にあるサブフォルダーに接続する場合は、フォルダーパスを指定します。
 - g 「拡張オプション」をクリックして、次の項目を指定します。
 - ポート：22 番以外のポートを使用する場合は、そのポートを指定します。
 - FTP 接続セッションをバックグラウンドで実行する場合は、「パッシブモードを使用」を選択します。
 - チーム環境で作業している場合は、ファイルをアップロードすると、他のユーザーが加えた変更が上書きされてしまう可能性があります。そのような場合は、予防策として「サーバーとクライアントの時差を自動的に計算」を選択してください。
- 6 「OK」をクリックして FTP 接続を追加します。

プロジェクト、ファイル、フォルダーのアップロード、ダウンロード、または同期

- 1 ナビゲータービューでプロジェクトまたはファイルを右クリックし、同期/アップロードまたはダウンロードを選択します。
- 2 サイトを選択ダイアログボックスで接続を選択します。
- 3 「OK」をクリックします。

Ajax ライブラリのインポート

- 1 ナビゲータービューで、Ajax ライブラリをインポートするプロジェクトを右クリックし、「インポート」をクリックします。

- 2 インポートウィザードで、ColdFusion / Ajax ライブラリインポートウィザードを選択し、「次へ」をクリックします。
- 3 インポートする Ajax ライブラリを選択し、「終了」をクリックします。

Flash Builder プロジェクトの使用

ColdFusion Builder で Flash Builder プロジェクトを使用するには、Flash Builder プロジェクトを ColdFusion Builder にインポートします。その後、ナビゲータービューで Flash Builder プロジェクトを選択して ColdFusion Builder の機能を適用し、右クリックして「CF Nature の適用」を選択します。

ColdFusion Builder の機能を Flash Builder プロジェクトに適用すると、ColdFusion Builder 内から Flash Builder プロジェクトのプレビュー、編集、デバッグを行えるようになります。


AIR アプリケーションの開発

Adobe ColdFusion Builder には、Adobe® AIR® プロジェクトの作成、Adobe AIR アプリケーションのデバッグ、パッケージ化、および電子署名の追加を行うためのツールが用意されています。

AIR プロジェクトの作成

- 1 ファイル / 新規 / プロジェクトを選択します。
- 2 新規プロジェクトウィザードで、HTML プロジェクト / Adobe AIR プロジェクトを選択し、「次へ」をクリックします。
- 3 プロジェクト名と、プロジェクト内のファイルの場所を入力します。
- 4 サンドボックスアプリケーションコードを含む HTML スタートページを指定します。アプリケーション以外のサンドボックスコードを選択することもできます。「次へ」をクリックします。
- 5 アプリケーション XML ファイルに関する次の情報を指定し、「次へ」をクリックします。
 - ID：アプリケーションの識別文字列として使用する一意のアプリケーション識別子。
 - 名前：アプリケーションの名前。
 - ファイル名：ファイルの名前とアプリケーションがインストールされるフォルダー。
 - バージョン：アプリケーションのバージョン。
 - アイコン：アプリケーションを表すアイコンファイル。アイコンを指定しない場合は、オペレーティングシステムのデフォルトアイコンが使用されます。
- 6 ウィンドウのスタイル、寸法、サイズ変更オプションを指定し、「次へ」をクリックします。
- 7 プロジェクトにインポートする AIR フレームワークを選択し、「次へ」をクリックします。
- 8 インポートする Ajax JavaScript ライブラリを選択し、「終了」をクリックして AIR プロジェクトを作成します。

AIR アプリケーションの実行とデバッグ

- 1 ナビゲータービューから、アプリケーションのソースファイル (Application.XML) を開きます。
- 2 メインツールバーの「実行」をクリックして AIR アプリケーションを起動します。
- 3 アプリケーションをデバッグするには、ワークベンチツールバーで  をクリックします。

アプリケーションが ADL アプリケーション (AIR Debugger Launcher) 内で起動して実行されます。ColdFusion デバッガーによってブレークポイントまたは実行時エラーが検出され、他の ColdFusion アプリケーションと同様にアプリケーションをデバッグできます。詳細については、84 ページの「[アプリケーションのデバッグ](#)」を参照してください。

AIR アプリケーションのパッケージ化と電子署名の追加

アプリケーションが完成して配布する準備ができたなら、AIR ファイルにパッケージ化します。パッケージ化の手順は次のとおりです。

- 1 ナビゲータービューで、パッケージ化する AIR プロジェクトを右クリックし、「エクスポート」をクリックします。
 - 2 エクスポートウィザードで、Adobe AIR / Adobe AIR パッケージを選択し、「次へ」をクリックします。
 - 3 AIR プロジェクトおよびアプリケーション記述ファイルを選択します。
 - 4 デフォルトの AIR SDK を選択するか、別の Adobe AIR SDK を設定し、「次へ」をクリックします。別の AIR SDK を設定するには、「Adobe AIR SDK を設定」をクリックし、AIR SDK を選択します。また、AIR SDK の追加、編集、削除も可能です。
 - 5 既存の電子証明書を選択するか、自己署名証明書を作成して、AIR アプリケーションに電子署名を追加できます。AIR アプリケーションに電子署名を追加すると、アプリケーションが信頼できる証明書で署名されていることがユーザーに保証され、発行者 ID が表示されます。
 - a 「エクスポートされた AIR アプリケーションに電子署名を追加」を選択します。
 - b 既存の電子証明書を選択するか、「Configure Certificates」をクリックして別の電子証明書を選択します。「追加」をクリックして自己署名証明書を作成します。
 - c 電子証明書のパスワードを指定します。
 - d 「TimeStamp AIR Package」を選択した場合は、インストールパッケージに署名する際に、ADT (AIR Developer Tool) がタイムスタンプ局と自動的に通信して時刻を確認します。タイムスタンプ情報は AIR ファイルに記録されます。確認済みのタイムスタンプを含む AIR ファイルは、いつでもインストールできます。
 - e AIR アプリケーションを新しい証明書に移行するには、「Migrate AIR Application」を選択します。その場合は、新しい証明書と古い旧証明書の両方を使用して AIR ファイルに署名します。移行前の証明書と、そのパスワードを指定します。
- 電子署名なしで AIR アプリケーションをパッケージ化することもできます。その場合は、「エクスポートされた AIR アプリケーションに電子署名を追加」を選択解除します。電子署名なしで AIR アプリケーションをパッケージ化すると、中間 AIR ファイル (.airi) が作成されます。中間 AIR ファイルをデプロイまたはインストールすることはできません。このファイルは一般に、(開発者によって) テスト目的で使用され、AIR ADT コマンドラインツールで起動できます。
- 6 「次へ」をクリックして、エクスポートする AIR ファイルから除外するファイルを必要に応じて選択します。デフォルトでは、すべてのファイルがパッケージに含まれます。
 - 7 「終了」をクリックして AIR ファイルを生成します。

第 8 章：アプリケーションのデバッグ

デバッグでは、アプリケーションを調べてトラブルシューティングできます。デバッグ中には、コード内の特定のポイントでアプリケーションを停止するかどうかを制御できます。また、重要な変数を監視したり、コードをテストすることもできます。デバッグでは、アプリケーションの起動方法を制御するための設定が使用されます。アプリケーションをデバッグするときは、デバッグ版のアプリケーションファイルを実行します。

ColdFusion デバッガーの使用

ColdFusion デバッガーを使用する前に、次の作業を行ってください。

デバッガーを使用するための ColdFusion の設定

デバッガーを使用する前に、次のことを確認します。

- デバッグ対象のプロジェクト、またはデバッグ対象のファイルを含むプロジェクトにサーバーが関連付けられているかどうかを確認します。
 - 1 ナビゲータービューでプロジェクトを右クリックし、「プロパティ」を選択します。
 - 2 プロパティダイアログボックスで「ColdFusion サーバーの設定」を選択します。
 - 3 「サーバーの選択」で、サーバーが選択されていることを確認します。サーバーが選択されていない場合は、サーバードロップダウンリストからサーバーを選択するか、「サーバーの追加」を選択して新しいサーバーを設定します。
- ColdFusion サーバーで RDS を有効にし、ColdFusion Builder で RDS の設定情報が正しく指定されていることを確認します。
- ColdFusion Administrator でデバッグが有効になっていることを確認します。
 - 1 ColdFusion Administrator で、デバッグとロギング/デバッガーの設定を選択します。
 - 2 「ラインデバッグの許可」を選択します。
 - 3 デバッグに使用するポートを指定します。デフォルト値は 5005 です。
 - 4 同時デバッグセッションの最大数を指定します。デフォルト値は 5 です。
 - 5 「変更の送信」をクリックします。
 - 6 リクエストのタイムアウト時間を長くするには、次のようにします。
 - a サーバーの設定/設定を選択します。
 - b 「リクエストタイムアウト」(秒)を選択して、必要なタイムアウト値を入力します。例えば、300 のように入力します。
 - c 「変更の送信」をクリックします。
 - 7 デバッガーサーバーは、手順 3 で指定したものと別個のポートで、ColdFusion Builder からのコマンドをリスンします。ColdFusion では、デフォルトで、使用可能なポートをランダムに使用してデバッガーサーバーを起動します。そのため、ColdFusion (つまりデバッガーサーバー) がファイアウォールの後ろにある場合は、問題が発生する可能性があります。デバッガーがリスンしているランダムなポートをファイアウォールがブロックするためです。

この問題を回避するには、デバッガーサーバーが使用するポート番号を固定的に指定し、このポートの使用をファイアウォールで許可します。デバッガーサーバーのポート番号を固定的に設定するには、ColdFusion Administrator の Java と JVM ページ（または J2EE アプリケーションサーバーの適切なセクション）で、次の JVM 引数を指定します。portNumber の部分は、実際に使用するポート番号に置き換えてください。

```
-DDEBUGGER_SERVER_PORT=portNumber
```

- 8 ColdFusion を再起動します。J2EE 設定の ColdFusion を実行している場合は、指定のデバッグポートを使用してデバッグモードでサーバーを再起動します。

J2EE 設定の ColdFusion におけるデバッグのセットアップ

- 1 サーバー設定の ColdFusion を実行していない場合は、実行するアプリケーションサーバーの設定ファイルまたはスタートアップスクリプトで Java デバッグパラメーターを指定します。パラメーターは次のようになります。

```
-Xdebug -Xrunjdw:transport=dt_socket,server=y,suspend=n,address=<port_number>
```

ColdFusion Administrator のデバッガーの設定ページで指定したのと同じポート番号を指定する必要があります。

サーバー設定の ColdFusion を実行している場合は、ColdFusion Administrator のデバッガーの設定ページを使用すると、ColdFusion によってこれらのデバッグパラメーターが jvm.config ファイルに書き込まれます。

- 2 サーバー設定の ColdFusion を実行しておらず、アプリケーションサーバーが JRE 1.6 で実行されていない場合は、アプリケーションサーバーで実行している JDK バージョンの tools.jar ファイルを、ColdFusion の ¥lib フォルダにコピーします。例えば、JRE 1.4 で JRun を実行している場合は、JDK 1.4 の tools.jar ファイルを ColdFusion の ¥lib フォルダにコピーする必要があります。
- 3 サーバーバージョンの ColdFusion を実行しており、JRE 1.6 以外のバージョンの JRE を jvm.config ファイルで指定する場合は、jvm.config ファイルで指定されている JDK バージョンの tools.jar ファイルを ColdFusion の ¥lib フォルダにコピーします。

マルチサーバー設定の ColdFusion におけるデバッグのセットアップ

マルチサーバー設定で実行している ColdFusion アプリケーションをデバッグするには、次のコマンドを使用してコマンドラインから ColdFusion サーバーを開始します。

```
jrun -config <path_to_jvm_config> -start <server_name>
```

ColdFusion Builder でのデバッガー設定の指定

- 1 ColdFusion Builder でウィンドウ／環境設定を選択します。
- 2 ツリービューで、ColdFusion / デバッグの設定を選択します。
- 3 「ホーム」ボタンをクリックしたときにデバッガーの「デバッグ出力バッファー」に表示するホームページの URL を指定します。
- 4 デバッグ可能なファイルタイプの拡張子と、デバッガーに認識させるデバッガースコープを指定します。サイズの大きなファイルをデバッグする際のパフォーマンスを向上させるために、情報が不要なスコープのチェックはすべて選択解除してください。
- 5 ColdFusion エラーが発生した行でデバッガーを停止する場合は、「CFML ランタイム例外発生時にブレーク」を選択します。
- 6 警告ダイアログボックスを表示せずに TailView ビューでサーバーログを確認する場合は、「警告ダイアログボックスを表示せず Eclipse エラーログに例外を記録」を選択します。

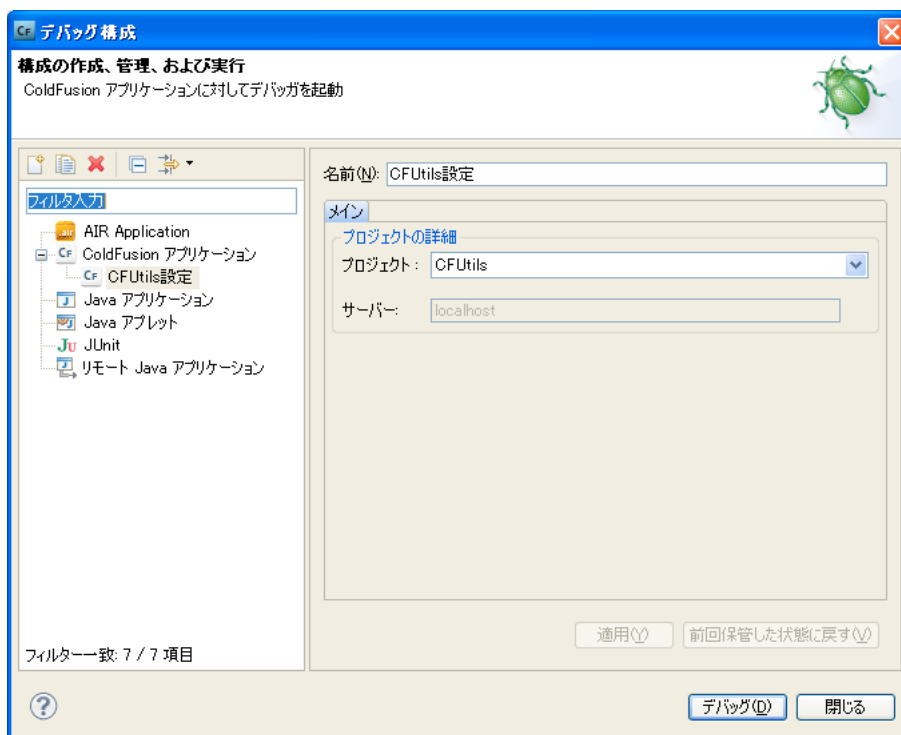
アプリケーションのデバッグ

ColdFusion Administrator でデバッガーを有効にし、ColdFusion Builder でデバッガーを設定すると、ColdFusion Builder でプロジェクトをデバッグできるようになります。

起動設定の作成と編集


ColdFusion Builder でアプリケーションのプロジェクトを初めてデバッグすると、そのプロジェクトに固有の起動設定が自動的に作成されます。起動設定では、プロジェクト名（デバッグ対象のプロジェクトに基づく名前）、メインのアプリケーションファイル、およびアプリケーションをデバッグするためのパスが自動的に定義されます。

起動設定の管理は、構成の作成、管理、および実行ダイアログボックスで行います。



デバッグ設定

ColdFusion Builder によって作成されたデフォルトの起動設定は編集できます。

- 1 デバッグするプロジェクトをナビゲータービューで選択します。
- 2 起動設定には、次のいずれかの方法でアクセスできます。
 - 実行/デバッグまたは実行/デバッグの構成を選択します。
 - ワークベンチツールバーで  を選択します。
 - プロジェクトを右クリックし、デバッグ/デバッグの構成を選択します。
- 3 デバッグの構成ダイアログボックスで、編集する起動設定を選択します。
起動設定を新規に作成することもできます。また、既存の設定に基づいて起動設定を作成することもできます。
- 4 必要に応じて、起動設定の内容を変更するための「パースペクティブ」リンクをクリックします。

アプリケーションの実行

ブレークポイントを設定する前に、デバッグするアプリケーションを実行します。デバッグする前にアプリケーションを実行するとアプリケーションがコンパイルされるため、デバッグ時のパフォーマンスが向上します。

- 1 実行するプロジェクトをナビゲータービューで選択します。
- 2 実行構成ダイアログボックスで、実行設定を選択または変更します。このダイアログボックスは、次のいずれかの方法で表示できます。
 - 実行/実行構成を選択します。
 - プロジェクトを右クリックし、実行/実行構成を選択します。
- 3 必要に応じて、実行するか変更する設定を選択します。
- 4 「実行」をクリックします。

コンピューターのデフォルトブラウザでアプリケーションが実行されます。別の外部ブラウザを指定するには、次のようにします。

- (Windows の場合) プロジェクトを右クリックして、「プロパティ」を選択します。「ColdFusion サーバーの設定」に移動して、コンピューターにインストールされている Web ブラウザーを選択します。
- (Mac の場合) 環境設定ダイアログボックスで、一般/ Web ブラウザーを選択して、コンピューターにインストールされている Web ブラウザーを選択します。

ブレークポイントの設定と削除

ブレークポイントを使用すると、アプリケーションの実行の制御できます。これにより、コードの詳細を検査して、アプリケーションをデバッグできます。ブレークポイントを追加するにはコードエディターを使用します。追加したブレークポイントはブレークポイントビューで管理できます。ブレークポイントは、コードを記述するときや、デバッグするときにも設定できます。

ブレークポイントビューでブレークポイントを管理する方法については、33 ページの「[ブレークポイントビュー](#)」を参照してください。

ColdFusion ブレークポイント

ColdFusion デバッガーの ColdFusion ブレークポイントには、次の 4 つの状態があります。

使用可能で有効 この状態は、ブレークポイントが有効な位置にあることを示します。ブレークポイントは、CFML エディターの左マージンにべた塗りの青い円として表示されます。このブレークポイントに到達するとコードの実行が停止します。

未解決 ColdFusion のメモリにロードされたページにはブレークポイントが設定されます。そのページを変更した後に実行していない場合は、サーバー上のページとソースの間で同期が失われます。この状況では、ブレークポイントを設定する行が有効であるかどうかを ColdFusion が判断できない場合があります。疑問符 (?) は、このタイプのブレークポイントを表します。

無効 ColdFusion Builder で編集されている CFML がメモリ内の CFML と同一であり、無効な行にブレークポイントが設定されていると ColdFusion で判断された場合、そのブレークポイントは赤い X 印で表されます。

使用不可 この状態は、ブレークポイントが使用不可であることを示します。

コードエディターでのブレークポイントの設定

- 1 ブレークポイントを設定するコードを含んでいる ColdFusion Builder プロジェクトを開きます。
- 2 ブレークポイントを設定するコード行を探し、次のいずれかの操作を行います。
 - エディターの左端にあるマーカーバーをダブルクリックします。

- 実行/ブレークポイントを切り替えまたは実行/行ブレークポイントを切り替えを選択します。

注意: ColdFusion Builder では、メソッドブレークポイントの切り替えと監視ポイントの切り替えはサポートされていません。

- 3 CFML ファイル内にブレークポイントを設定すると、特定の場所でページの実行を停止できます。特定の行にブレークポイントを設定すると、その行の直前で CFML の実行が停止します。例えば、次の CFML ページの 3 行目にブレークポイントを設定した場合、<cfset myName = "Wilson"> の前で実行が停止します。

```
<cfset yourName = "Tuckerman">
<cfoutput>Your name is #yourName#.</cfoutput>
<cfset myName = "Wilson"
```

ブレークポイントが設定された行の頭に青のドットが表示されます。

ColdFusion Builder デバッグパースペクティブのブレークポイントビューで、現在のプロジェクトに設定されているブレークポイントのリストを確認することもできます。

コードエディターでのすべてのブレークポイントのスキップ

コード内にブレークポイントを設定した後で、デバッグ時に、それらのブレークポイントをすべて無視することもできます。

- 1 ブレークポイントを設定したコードを含んでいる ColdFusion Builder プロジェクトを開きます。
- 2 ブレークポイントビューのツールバーで、「すべてのブレークポイントをスキップ」を選択します。


コードエディターでのブレークポイントの削除

❖ エディターの左端にあるマーカーバーで、既存のブレークポイントをダブルクリックします。

選択したブレークポイントが、マーカーバーと、ColdFusion Builder デバッグパースペクティブのブレークポイントビューから削除されます。

ファイルからすべてのブレークポイントを削除するには、メインツールバーのメニューから実行/すべてのブレークポイントを削除を選択します。

デバッグセッションの開始

- 1 ナビゲータービューで、デバッグするプロジェクトまたはファイルを選択します。
- 2 次のいずれかの方法で、デバッグセッションを開始できます。
 - 実行/デバッグを選択します。
 - ワークベンチツールバーで  をクリックします。
 - プロジェクトを右クリックし、デバッグ/ ColdFusion アプリケーションを選択します。

デバッグの起動設定が自動的に作成されて起動されます。

注意: ページのデバッグ中にそのページを閲覧しようとしたり更新しようとしたりすると、デバッガーが予期しない動作をする可能性があります。

デバッグセッションの管理

デバッグビューを使用すると、アプリケーションのデバッグの制御、アプリケーションの一時停止、再開、終了、コードへのステップイン、ステップオーバーなどを行えます。

ColdFusion デバッグパースペクティブの各種ビューの詳細については、31 ページの「[ColdFusion デバッグパースペクティブ](#)」を参照してください。

行単位でのコードの実行

「ステップイン」、「ステップオーバー」、「ステップリターン」ボタンを使用して、行単位で CFML アプリケーションを実行することができます。

ステップを正しく機能させるには、コンパイル済みクラスのキャッシュをクリアします。これを行うには、以前のバージョンの ColdFusion でコンパイルした CFML ページをすべて再コンパイルします。

サイズの大きいファイルでは、ステップやブレークポイントの動作が遅くなる場合があります。パフォーマンスを上げるには、次のようにします。

- 1 ColdFusion Builder でウィンドウ／環境設定を選択します。
- 2 ツリービューで、ColdFusion / デバッグの設定を選択します。
- 3 情報を収集する必要がないスコープを選択解除します。

ステップイン

ステップインは、UDF、CFC、カスタムタグ、およびプロジェクトに含まれるファイルに対して使用します。

cfset タグなどの CFML タグに対してはステップインを使用しないでください。ステップインを使用すると、ステップオーバーよりもパフォーマンスに負荷がかかります。

関数、タグ、ファイルにステップインする場合は、開いているプロジェクトのいずれかでファイルが表示されている必要があります。ステップイン対象のファイルは開いているプロジェクト内にある必要があります。

ステップオーバー

ステップオーバーは、プロジェクトに含まれるファイル（UDF や CFC など）をスキップして CFML アプリケーションの処理を進めるときに使用します。

ステップリターン

ステップリターンは、プロジェクトに含まれるファイル（UDF や CFC など）から呼び出し元のページに戻るときに使用します。

変数の状態の確認

CFML コードを実行するときには、変数ビューに変数の値とスコープを表示できます。変数ビューに表示されるのは、設定ダイアログボックスで選択したスコープに属している変数のみです。変数ビューの使用方法について詳しくは、32 ページの「[変数ビュー](#)」を参照してください。

監視式

別の関数にステップインしたときに、スコープから外れる可能性がある重要な変数を監視するときは、監視式を使用すると便利です。独自の式を作成して、監視や評価を行うことができます。デバッグセッション中に、式を変更することもできます。

式ビューでは、次の操作を行うことができます。

- 監視式を作成するには、右クリックメニューから「監視式を追加」を選択します。監視式を追加ダイアログボックスが開いたら、式を入力できます。
- 追加した監視式を無視するには、式を右クリックし、「使用不可にする」を選択します。
- 監視式を編集するには、式を右クリックし、「監視式を編集」を選択します。その後、式を変更できるようになります。

式ビューの使用方法について詳しくは、33 ページの「[式ビュー](#)」を参照してください。

リモートアプリケーションのデバッグ

ColdFusion をリモートサーバーで実行している場合は、リモートサーバー接続を設定し、ColdFusion と ColdFusion Builder の間のマッピングを指定します。マッピングを指定すると、ColdFusion Builder と ColdFusion の間で同じプロジェクトまたはファイルのコピーが処理されるようになります。

- 1 リモートサーバーの設定については、69 ページの「[リモートサーバーの追加](#)」を参照してください。必要に応じて、RDS 設定情報を入力します。
- 2 サーバービューでサーバーを右クリックし、「サーバーの編集」を選択します。サーバーウィザードのマッピング画面で、次のマッピング情報を入力します。
 - ローカルパス：リモート ColdFusion サーバー上でプロジェクトまたはフォルダーを探すときに ColdFusion Builder で使用するパス。
 - リモートパス：リモート ColdFusion サーバー上にあるプロジェクトまたはフォルダーのパス。

注意：リモート ColdFusion サーバーと ColdFusion Builder の間では、複数のマッピングを指定できます。


例えば、D:\MyCoolApp を参照する ColdFusion プロジェクト内のファイルを編集していて、それに対応するファイルがリモートサーバーの D:\Shared\websites\MyCoolSite にあるとします。この場合は、ローカルパスとして D:\MyCool App を指定し、リモートパスとして D:\Shared\websites\MyCoolSite を指定してマッピングを作成します。

ColdFusion サーバーにファイルをデプロイするときは W:\websites\MyCoolSite\ にファイルをコピーします。ColdFusion サーバーはこのディレクトリを D:\Shared\websites\MyCoolSite として認識します。

ColdFusion Builder のマッピングで、ColdFusion Builder ディレクトリは D:\MyCoolApp であり、サーバーは D:\Shared\websites\MyCoolSite であると指定します。これにより、ColdFusion Builder は、ファイルパス (D:\MyCoolApp\index.cfm) を、ColdFusion サーバーが認識するパス (D:\Shared\websites\MyCoolSite\index.cfm) に変換します。

クライアントとサーバー間の通信の詳細を確認するには、ColdFusion Administrator の JVM 引数に次を追加します。

```
-DDEBUGGER_TRACE=true
```

- 3 ナビゲータービューで、デバッグするプロジェクトまたはファイルを選択し、ワークベンチツールバーで  をクリックします。

第9章：ColdFusion Builder 拡張機能

拡張機能について

ColdFusion Builder IDE (Integrated Development Environment: 統合開発環境) の機能は、各種の ColdFusion フレームワークをサポートしたり、コード生成に必要な条件を満たすように拡張することができます。ColdFusion Builder 拡張機能を開発すると、コードを生成したり、動的なユーザーインターフェイスを設計したり、データベースに対する基本的な CRUD (作成、読み込み、更新、削除) 操作を実行できます。ColdFusion Builder 拡張機能は、ColdFusion Builder IDE にコンテキストメニューを追加し、それらのメニューに対するイベントを処理する構造化コンポーネントです。ColdFusion 開発者は、独自の ColdFusion Builder 拡張機能を開発することも、既存の拡張機能をインストールすることもできます。

独自の ColdFusion Builder 拡張機能を開発するには、設定ファイルおよびハンドラーファイルを作成します。コンテキストメニューとイベントは、設定ファイル (IDE_Config.xml) で定義します。詳細については、91 ページの「[拡張機能の開発](#)」を参照してください。

ColdFusion Builder では、ユーザーインターフェイスを作成するなど、様々なレベルで IDE を拡張できます。詳細については、106 ページの「[拡張機能のユーザーインターフェイスの作成](#)」を参照してください。

ColdFusion Builder に出荷時から付属する拡張機能には、Adobe CFC Generator と AS Class Generator があります。これらの拡張機能はいつでもインストールして、ColdFusion Builder に統合できます。詳しくは、121 ページの「[拡張機能の使用](#)」を参照してください。

ColdFusion Builder には、拡張機能の作成とパッケージ化に必要な手順をガイドする ColdFusion Builder Extension Creator が用意されています。詳しくは、116 ページの「[ColdFusion Builder Extension Creator を使用した拡張機能の作成とパッケージ化](#)」を参照してください。

ColdFusion Builder 2 の拡張機能の新機能

拡張機能の開発

ColdFusion Builder 拡張機能を開発するには、次の要素を作成します。

- 設定ファイル (IDE_config.xml)
- ハンドラーファイル (CFM ファイル)

これらの要素を作成するには、手動でコードを記述するか、ColdFusion Builder Extension Creator ウィザードを使用します。ColdFusion Builder Extension Creator ウィザードの使用方法について詳しくは、116 ページの「[ColdFusion Builder Extension Creator を使用した拡張機能の作成とパッケージ化](#)」を参照してください。

設定ファイル

設定ファイル (IDE_config.xml) の作成は、ColdFusion Builder 拡張機能を開発する上で重要な手順となります。設定ファイルの要素の定義はすべて、application タグ内で行います。

メタデータ要素の指定

拡張機能を作成し、拡張機能の情報（名前、作成者、バージョン、説明など）を指定するには、メタデータ要素を使用します。

設定ファイルでアプリケーションメタデータを指定するには、次の要素を使用します。

要素	説明
name	拡張機能の名前。
author	作成者の名前。
version	ファイルのバージョン。
email	電子メールアドレスを指定します。
description	アプリケーションの簡単な説明。 この説明はプレーンテキストで記述できます。また、アプリケーションの説明が記載された HTML ファイルのパスを指定することもできます。HTML ファイルのパスを指定する場合は、拡張機能内の Install ディレクトリにその HTML ファイルを保存します。
license	拡張機能をインストールするときに表示されるライセンス契約の条項。 ライセンス契約の条項はプレーンテキストで表示できます。また、ライセンス契約の条項が記載された HTML ファイルのパスを指定することもできます。HTML ファイルのパスを指定する場合は、拡張機能内の Install ディレクトリにその HTML ファイルを保存します。

例

```
<application>
  <name>ORM CFC Generator</name>
  <author>Adobe</author>
  <version>1.0</version>
  <email>user@xyz.com</email>
  <description>ORM CFC code Generator</description>
  <license>license.html</license>
</application>
```

ColdFusion Builder 拡張機能のインストールウィザードへのページの追加

設定ファイルを定義する際に、ColdFusion Builder 拡張機能のインストールウィザードに画面を追加するコードを指定できます。これらの画面を使用して、ユーザー入力を取得できます。通常、インストール後の設定タスクを実行するには、ユーザーの入力が必要です。

入力の詳細を指定するには input タグを使用します。input タグを指定する方法については、99 ページの「[入力型の指定](#)」を参照してください。これらの入力詳細に基づき、wizardtag の handlerid 属性で指定されているハンドラーが呼び出されます。また、wizard タグの height 属性と width 属性を使用して、インストールウィザードの高さと幅を指定することもできます。このウィザードの各ページのタイトルを指定するには、page タグの title 属性を使用します。

シンタックス

```
<application>
<name>Name of the ColdFusion Builder extension</name>
<install>
<wizard height="" width="" handlerid="handlerID" >
<page title="Wizard page title" >
<input name="Input" ... />
<input name="Input" ... />
</page>
</wizard>
</install>
</application>
```

例

次の例では、拡張機能のインストールウィザードによる拡張機能のインストールが終了した後に、指定された入力詳細に基づいてハンドラー ID `postinstallhandler` が呼び出されます。

```
<application>
<name>ORM CFC Generator</name>
<install>
<wizard height="" width="" handlerid="postinstallhandler" >
<page title="Install settings" >
<input name="Mapping" ... />
<input name="Datasource" ... />
</page>
</wizard>
</install>
</application>
```

IDE の拡張

ColdFusion Builder IDE は次のレベルで拡張できます。

- コンテキストメニューの追加：コンテキストメニューは次のビューに追加できます。
 - リソースナビゲーター
 - RDS データビュー
 - アウトラインビュー
 - CFML エディター
- ワークスペースイベントの処理：現時点でサポートされているイベントは `onprojectcreate` のみです。94 ページの「[ColdFusion Builder 2.0.1 で追加された新しいイベント](#)」も参照してください。
- 108 ページの「[ビューの作成](#)」
- 118 ページの「[拡張機能からコードアシストへのコントリビュート](#)」

CRUD またはコード生成操作を実行するコンテキストメニューおよびイベントのハンドラーを指定できます。

ハンドラーの指定

ColdFusion Builder は、CFM ハンドラーをサポートしています。ColdFusion Builder の場合、ハンドラーとはイベントやアクションに対する応答として実行されるコードを含んだファイルのことです。ハンドラーの指定は `handlers` タグ内で行います。ハンドラーをイベントまたはアクションに関連付けるには、`handlerid` 属性を使用します。ハンドラーの詳細はすべて `handlers` タグ内で指定します。イベントおよびアクションを指定する方法の詳細については、94 ページの「[イベントの指定](#)」および 97 ページの「[コンテキストメニューの指定](#)」を参照してください。ハンドラーファイルはすべて、`Handlers` フォルダーに保存する必要があります。

シンタックス

```
<handlers>
  <handler id ="cfm" type="cfm" filename="filename" />
</handlers>
```

属性	説明
id	ハンドラー ID
type	ハンドラータイプを指定します。 指定できるハンドラータイプは「CFM」です。
filename	CFM ファイルの名前

注意：属性値として使用できる文字は英数字のみです。特殊文字は指定できません。

例

```
<handlers>
  <handler id="cfgenerator" type="CFM"
    filename="ormCFGenerator.cfm" />
  <handler id="gridgenerator" type="CFM"
    filename="cfgridGenerator.cfm" />
</handlers>
```

イベントの指定

イベントを指定するには `events` タグを使用します。現在 ColdFusion Builder でサポートされているイベントは、プロジェクト作成に関するイベントのみです。サポートされているイベントのタイプは `onprojectcreate` です。`onprojectcreate` イベントのハンドラー ID は必須です。イベントが発生すると、ハンドラー ID に関連付けられているハンドラーが呼び出されます。

`onprojectcreate` イベントが指定されているアプリケーションは、74 ページの「[ColdFusion プロジェクトの作成](#)」にあるアプリケーションのリストにすべて記載されています。リストに記載されているアプリケーションのいずれかをユーザーが選択すると、そのアプリケーションに関連付けられているハンドラーが呼び出されます。`onprojectcreate` イベントは、特定の ColdFusion Builder 拡張機能の基本的なプロジェクト構造を作成するときに役立ちます。

シンタックス

```
<events>
  <event
    type="onprojectcreate"
    handlerid="handler id" />
</events>
```

属性	説明
type	ハンドラーを実行するイベントを指定します。 指定できるイベントタイプは <code>type="onprojectcreate"</code> です。
handlerid	渡すハンドラー ID を指定します。

注意：属性値として使用できる文字は英数字のみです。特殊文字は指定できません。

例

```
<events>
  <event type="onprojectcreate"
    handlerid="projectCreationHandler" />
</events>
```

ColdFusion Builder 2.0.1 で追加された新しいイベント

onfilechangeineditor：拡張機能はエディターでファイルが変更されたときのイベントを listen できます

エディターのタブをクリックして、またはエディターでドキュメントを開いて、異なるドキュメントに移動するものとします。拡張機能でこのことを認識できるようになりました。

このために、新しいイベント `onfilechangeineditor` が導入されています。このイベントは次のようにして登録できます。

```
<event type="onfilechangeineditor" handlerid="any_handler_id"/>
```

イベントのメッセージでは次のような情報が送信されます。

- type：イベントのタイプは `onfilechangeineditor` です。

- `project_path` : エディターで開かれたファイルが属しているプロジェクトの絶対パスです。
- `file_path` : 現在エディターで開かれている、または切り替えられたファイルの絶対パスです。
- `editor_name` : ファイルを開いたエディターの名前です。CFML ファイルの場合、この名前は Adobe CFML Editor です。

出力の例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <ide version="2.0">
    <callbackurl>
      http://[ip_address:port]/index.cfm?extension=<Extension_Name>
    </callbackurl>
    <eventinfo type="onfilechangeineditor">
      <project>
        Extensions
      </project>
      <project_path>
        [project_path]
      </project_path>
      <file_path>
        [file_path]
      </file_path>
    </eventinfo>
    onfilechangeineditor
  </ide>
  <editor_name>
    Adobe CFML Editor
  </editor_name>
  </eventinfo>
</event>
```

onRDSDataViewSelectionChange : 拡張機能は、RDS データビューで選択範囲が変更されたときのイベントを listen できます

ColdFusion Builder は、RDS データビューでの選択に応じて、異なるイベント情報メッセージをハンドラーに送信します。

このために、新しいイベント `onRDSDataViewSelectionChange` が導入されました。このイベントは次のようにして登録できます。

```
<event type="onfilechangeineditor" handlerid="any_handler_id">
```

選択されたノード	送信される情報	例
サーバー	<ul style="list-style-type: none"> type : イベントのタイプ onRDSDataViewSelectionChange name : サーバー名 node_type : ノードの種類 - サーバー 	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0" > <callbackurl>http://[ip_address:port]/index.cfm? extension=<Extension_Name></callbackurl> <eventinfo type="onRDSDataViewSelectionChange"> <event>onRDSDataViewSelectionChange</event> <name>LocalCF9</name> <node_type>server</node_type> </eventinfo> </ide> <user></user> </event></pre>
データソース	<ul style="list-style-type: none"> type : イベントのタイプ onRDSDataViewSelectionChange name : データソースの名前 parent_node : 親ノードの名前。つまり、データソースが属しているサーバーの名前。 node_type : ノードの種類 - データソース 	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0" > <callbackurl>http://[ip_address:port]/index.cfm? extension=<Extension_Name></callbackurl> <eventinfo type="onRDSDataViewSelectionChange"> <event>onRDSDataViewSelectionChange</event> <name>AUTHORS</name> <parent>Tables</parent> <node_type>table</node_type> </eventinfo> </ide> <user></user> </event></pre>
テーブルグループ	<ul style="list-style-type: none"> type : イベントのタイプ onRDSDataViewSelectionChange name : テーブルグループの名前 parent_node : 親ノードの名前。つまり、このテーブルグループが属しているデータソースの名前。 node_type : ノードの種類 - グループ 	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0" > <callbackurl>http://[ip_address:port]/index.cfm? extension=<Extension_Name></callbackurl> <eventinfo type="onRDSDataViewSelectionChange"> <event>onRDSDataViewSelectionChange</event> <name>Tables</name> <parent>cfbookclub</parent> <node_type>group</node_type> </eventinfo> </ide> <user></user> </event></pre>
テーブル	<ul style="list-style-type: none"> type : イベントのタイプ onRDSDataViewSelectionChange name : テーブルの名前 parent_node : 親ノードの名前。つまり、テーブルが属しているデータソース。 node_type : ノードの種類 - テーブル 	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0" > <callbackurl>http://[ip_address:port]/index.cfm? extension=<Extension_Name></callbackurl> <eventinfo type="onRDSDataViewSelectionChange"> <event>onRDSDataViewSelectionChange</event> <name>AUTHORS</name> <parent>Tables</parent> <node_type>table</node_type> </eventinfo> </ide> <user></user> </event></pre>
列	<ul style="list-style-type: none"> type : イベントのタイプ onRDSDataViewSelectionChange name : 列の名前 parent_node : 親ノードの名前。つまり、列が属しているテーブルの名前。 node_type : ノードの種類 - 列 	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0" > <callbackurl>http://[ip_address:port]/index.cfm? extension=<Extension_Name></callbackurl> <eventinfo type="onRDSDataViewSelectionChange"> <event>onRDSDataViewSelectionChange</event> <name>AUTHORID</name> <parent>APP.AUTHORS</parent> <node_type>field</node_type> </eventinfo> </ide> <user></user> </event></pre>

OnFileSaved : ファイルが保存されるときイベント通知を追加します

ColdFusion Builder でファイルが保存されると、イベント通知がハンドラーの CFM ファイルに送信されます。

このために、新しいイベント onFileSaved が導入されています。このイベントは次のようにして登録できます。

```
<event type="onFileSaved" handlerid="OnFileSaved"/>
```

ハンドラーには次のような情報が送信されます。

- `type`：イベントのタイプは `onFileSaved` です。
- `project_path`：保存されたファイルが属しているプロジェクトの絶対パスです。
- `file_path`：保存されたファイルの絶対パスです。
- `editor_name`：ファイルを開いたエディターの名前です。CFML ファイルの場合、この名前は Adobe CFML Editor です。

出力の例を次に示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <ide version="2.0">
    <callbackurl>
      http://[ip_address:port]/index.cfm? extension=<Extension_Name>
    </callbackurl>
    <eventinfo type="onfilechangeineditor">
      <project>
        Extensions
      </project>
      <project_path>
        [project_path]
      </project_path>
      <file_path>
        [file_path]
      </file_path>
      <event>
        onfilesaved
      </event>
      <editor_name>
        Adobe CFML Editor
      </editor_name>
    </eventinfo>
  </ide>
  <user>
  </user>
</event>
```

コンテキストメニューの指定

コンテキストメニューを指定するには `menucontributions` タグを使用します。コンテキストメニューとは、右クリックイベントで表示されるポップアップメニューのことです。

各種ビューのメニューコントリビューションを指定するには、`contribution` タグを使用します。メニューを表示するビュー（ターゲットビュー）を指定するには、`contribution` タグ内で `target` 属性を使用します。追加するメニューを指定するには、`contribution` タグ内で `menu` タグを使用します。メニューアイテムを指定するには、`menu` タグ内で `action` タグを使用します。

シンタックス

```
<menucontributions>
  <contribution target="rdsview|projectview|outlineview">
    <menu name="name of the menu item">
      <action name="action name">
      </action>
    </menu>
  </contribution>
</menucontributions>
```

contribution

属性	説明
target	ターゲットビューとして指定できる値は次のいずれかになります。 <ul style="list-style-type: none"> • rdsview • projectview • outlineview • editor

注意：属性値として使用できる文字は英数字のみです。特殊文字は指定できません。

menu

属性	説明
name	メニューの名前を指定します。

注意：属性値として使用できる文字は英数字のみです。特殊文字は指定できません。

action

属性	説明
name	アクションの名前を指定します。
handlerid	関連付けられているハンドラーの ID を指定します。
showresponse	ハンドラーが受信した HTML レスポンスをユーザーに表示するかどうかに応じて、次のブール値を指定します。 <ul style="list-style-type: none"> • yes • no デフォルト値は false です。

注意：属性値として使用できる文字は英数字のみです。特殊文字は指定できません。

メニューフィルターの指定

メニューフィルターを使用すると、メニューやメニューアイテムをどこに表示するかを簡単に制御できます。メニューフィルターを指定するには、`filters` タグを使用します。filters タグ内では、様々なタイプのフィルターを指定できます。メニューやメニューアイテムをどこに表示するかを指定するには、`type` 属性と `pattern` 属性を使用します。指定したフィルターのいずれかに一致すると、指定したメニューまたはメニューアイテムが表示されます。

フィルターはメニューとアクションの両方に対して指定できます。menu タグ内で filter タグを指定すると、そのフィルターによってメニューの表示が制御されます。例えば、次のコードでは、ユーザーが `modelglue.xml` ファイルを右クリックした場合にのみ、ORM Code Generator のメニューが表示されます。

```
<menu name="ORM Code Generator">
<filters>
<filter type="file" pattern="modelGlue.xml" />
</filters>
</menu>
```

ナビゲータービューに対するフィルターの指定

ナビゲータービューのプロジェクト、フォルダー、またはファイルに対してフィルターを指定できます。

シンタックス

```
<filter type="folder|project|file"  
pattern="regular expression to match folder, project, or filename" />
```

例

フォルダーのコンテキストでのみ ORM Code Generator のメニューが表示されるようにするには、次のようなコードを使用します。

```
<menu name="ORM Code Generator">  
<filters>  
<filter type="folder" />  
</filters>  
</menu>
```

アウトラインビューに対するフィルターの指定

アウトラインビューの様々なノードタイプに対してフィルターを指定できます。指定したノード名がフィルターとして機能します。アウトラインビューでは、ファイルに対してもフィルターを指定できます。CFML エディターでフィルターと一致するファイルを開くと、コントリビュートされたメニューがアウトラインビューに表示されます。

シンタックス

```
<menu name="menu name">  
<filters>  
<filter type="node name" />  
</filters>  
</menu>
```

例

アウトラインビューの CFfunction ノードでのみ ORM Code Generator のメニューが表示されるようにするには、次のようなコードを使用します。

```
<menu name="ORM Code Generator">  
<filters>  
<filter type="cffunction" />  
</filters>  
</menu>
```

入力型の指定

ハンドラーを呼び出す前に input タグを使用すると、ユーザーに入力を求めることができます。ユーザーの入力は、関連付けられているハンドラーによって処理されます。アクションごとに input タグを指定し、input タグを action タグ内に含める必要があります。

入力ダイアログの高さと幅を制御するには、dialog タグ内で input タグを指定します。例えば、次のようなコードを指定できます。

```
<dialog height="400" width="600" title="title" image="path to the image file">  
<input name="Mapping" ... />  
<input name="Datasource" ... />  
</dialog>
```

属性	説明
height	ダイアログボックスの高さを指定します。
width	ダイアログボックスの幅を指定します。
title	ダイアログボックスのタイトルを指定します。
image	タイトルバーに表示されるイメージのパスを指定します。ここでは、Extension フォルダーからの相対パスを指定する必要があります。

シンタックス

```
<action name="action name">
  <input name="input variable name"
    label="label for the input dialog box"
    tooltip="tool tip"
    type="dir|string|boolean|file|password|list"/>
</action>
```

属性	説明
name	入力変数の名前
label	入力ダイアログボックスのラベル
tooltip	ユーザーがマウスを入力ダイアログボックスの上に移動したときに表示するツールヒント。
type	<p>入力変数として受け入れられるデータ型は次のいずれかになります。</p> <ul style="list-style-type: none"> • dir • string • Boolean • file • password • list • projectdir • projectfile
required	<p>必須の入力フィールドである場合に指定します。</p> <p>入力フィールドを必須フィールドとして指定した場合、ユーザーがその必須フィールドに値を入力するまで、「OK」は有効になりません。</p>
pattern	<p>ユーザー入力の検証に使用する正規表現を指定します。</p> <p>検証エラーの場合に表示するエラーメッセージも指定できます。エラーメッセージを指定するには、<code>errormessage</code> 属性を使用します。</p>
errormessage	特定のパターンで検証が失敗したときに表示するエラーメッセージ。
helpmessage	特定の入力フィールドに関してダイアログボックスのタイトル領域に表示するヘルプヒント。
default	特定の入力型のデフォルト値を指定します。入力型が <code>Boolean</code> の場合、デフォルト値は指定できません。リストの場合、デフォルト値はあらかじめ選択されています。
checked	<p>チェックボックスフィールドをデフォルトでオンにするかオフにするかを指定するブール値。</p> <ul style="list-style-type: none"> • <code>true</code> - チェックボックスをオンにします（これがデフォルト値です）。 • <code>false</code> - チェックボックスをオフにします。

注意：属性値として使用できる文字は英数字のみです。特殊文字は指定できません。

各データ型は、次の表に示す入力型に対応しています。各入力型のシンタックスも示します。

データ型	入力型	シンタックス
dir	ディレクトリ選択フィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="dir"/>
string	テキストフィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="string"/>
boolean	チェックボックス	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="boolean"/>
file	ファイル選択フィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="file"/>
password	パスワードフィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="password"/>
list	リストフィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="list"> <option value=" オプション 1"> <option value=" オプション 2"> </input>
projectdir	プロジェクトディレクトリ選択フィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="projectdir"/>
projectfile	プロジェクトファイル選択フィールド	<input name=" 入力変数名 " label=" ラベル " tooltip=" ツールヒント " type="projectfile"/>

例

```
<menucontributions >
  <contribution target="rdsview" >
    <menu name="ORM Code Generator">
      <action name="Generate ORM CFC" handlerid="cfm1" >
        <input name="Location" label="Enter location"
        tooltip="Location where generated CFCs will be stored" type="dir"/>
        <input name="generateAppCFC" label="Generate
        Application CFC" tooltip="Generate Application CFC along with ORM CFC"
        type="boolean"/>
        <input name="generateView" label="Generate View"
        tooltip="Generate View template along with ORM CFC" type="boolean"/>
      </action>
      <action name="Generate Ajax Grid" handlerid="cfm2" >
        <input name="Location" label="Enter location"
        tooltip="Location where generated View will be stored" type="dir"/>
      </action>
    </menu>
  </contribution>
  <contribution target="projectview" >
    <menu name="ORM Code Generator">
      <action name="Generate ORM CFC" handlerid="cfm1" />
      <action name="Generate Ajax Grid" handlerid="cfm2" />
    </menu>
  </contribution>
  <contribution target="outlineview" >
    <menu name="ORM Code Generator">
      <action name="Generate ORM CFC" handlerid="cfm1" />
      <action name="Generate Ajax Grid" handlerid="cfm2" />
    </menu>
  </contribution>
</menucontributions>
```

キーワードのサポート

ColdFusion Builder では、入力ダイアログボックスにデフォルト値を追加するためのキーワードがサポートされています。これらのキーワードは実行時に、実際の値に置き換えられます。

次の表に、サポートされているキーワードおよび対応する実際の値を示します。

キーワード	実際の値
projectlocation	プロジェクトの場所の絶対パス
projectname	プロジェクトの名前
serverhome	サーバーのインストール場所の絶対パス
wwwroot	サーバーの Web ルート

例えば、projectname キーワードは次のように指定できます。

```
<input name="projectname" label="projectname" default="{ $projectname }" type="dir"/>
```

拡張機能のインポート、リロード、パッケージ化、およびデバッグの方法については、115 ページの「[拡張機能ビューの使用](#)」を参照してください。

複数ノードでのメニューコントリビューションのサポート

注意： ColdFusion Builder 2.0.1 で導入されました

RDS データビューでは、テーブルノードに加えて、サーバー、データベース、ファイルの各ノードにメニューコントリビューションを追加できます。

サーバーノードを選択した場合、サーバー名がハンドラーに送信されます。属性 selected が true の場合は、次に示すハンドラーに送信される応答の例のように、サーバー、データベース、テーブル、またはフィールドが選択範囲の一部かどうか示されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <ide version="2.0">
    <callbackurl>
      url
    </callbackurl>
    <rdsview>
      <servers>
        <server name="server1">
          <server name="server2">
        </servers>
        <database name="db1" server="server1" selected="true">
          <table name="table1" selected="false">
            <fields>
              <field
                name="field1" .... selected="true">
            </fields>
          </table>
        </database>
      </rdsview>
    </ide>
  </event>
```

フィルター

メニューコントリビューションを RDS データビューの複数のノードに追加できるので、次の例に示すように、フィルターを指定してアクティブにするノードを指定できます。

有効なフィルタータイプは、server、database、table、field です。

```
<contribution target="rdsview">
<menu name="menu1">
<action name="action1">
</action>
</filters>
<filter type="database" />
<filter type="table" />
</filters>
</menu>
</contribution>-
```

ColdFusion Builder とハンドラーの通信について

ハンドラーと ColdFusion Builder IDE の間の通信には XML が使用されます。イベントが発生すると、ColdFusion Builder からそのイベントに関連付けられているハンドラーへ、イベントの詳細が XML 形式で送信されます。この XML は `ideEventInfo` という名前の FORM スコープ変数としてハンドラーに送信されます。FORM スコープ変数から XML を取得するには、ハンドラーで `cfparam` タグを使用します。

シンタックス

```
<cfparam name="ideeventinfo">
```

または

```
<cfset ideData= form.ideEventInfo >
```

XML の構造について

ここでは実際の XML コードを例に取り、XML の構造について説明します。

コードを見る前に、次のポイントに注意してください。

- ハンドラーに送信される情報はすべて `event` タグ内にあります。
- イベントおよびアクションの情報はすべて `ide` タグ内にあります。
- ユーザー入力はすべて `user` タグ内にあります。

次の表で、各種の XML コードについて説明します。XML コードの内容は、イベントによって異なります。

コード	Context	説明
<pre><rdsview > <database name=""> <table name=""> <fields> <field name="" type="" length="" nullallowed="" primarykey=""> </fields> </table> </database> </rdsview></pre>	rdsview	RDS ビューのメニューアイテムをクリックすると、関連付けられているハンドラーにイベントの詳細が送信されます。
<pre><projectview projectname="" projectlocation=""> <resource name="" path="" type="file/folder/project" /> </projectview></pre>	projectview	プロジェクトビューのメニューアイテムをクリックすると、関連付けられているハンドラーにイベントの詳細が送信されます。
<pre><outlineview projectname="" projectlocation="" > <source filename="" path="" > <node type="function/other"> <function name="" returntype=""> <argument name="" type="" /> </function> </node> </source> </outlineview></pre>	outlineview	アウトラインビューのメニューアイテムをクリックすると、関連付けられているハンドラーにイベントの詳細が送信されます。

コード	Context	説明
<code><eventinfo projectname="" projectlocation="" eventtype="" > <resource name="" path="" type="file/folder/project" /> </eventinfo></code>	プロジェクト作成時	指定したイベントが発生すると、関連付けられているハンドラーにイベントの詳細が送信されます。
<code><user> <input name="" value="" /> <input name="" value="" /> <input name="" value="" /> </user></code>	入力ダイアログボックス	ユーザー入力はすべて user タグ内で指定します。user タグ内の各 input タグで、具体的なユーザー入力を指定します。入力名は、IDE_Config.xml ファイルで指定されている入力名と同じです。
<code><user> <page index="" > <input name="" value="" /> </page> </user></code>	インストールウィザード	user タグ内で指定されたすべてのユーザー入力が、インストールウィザードに関連付けられているハンドラーに送信されます。個々のユーザー入力は user タグ内の各 input タグに格納されます。入力名は、IDE_Config.xml ファイルで指定されている入力名と同じです。

ColdFusion Builder からハンドラー CFM ファイルに送信される ideeventinfo.xml には、次の情報が含まれます。

- ColdFusion Builder のバージョン情報。次のように記述されます。

```
<event>
  <ide version="2.0" >
    ....
  </ide>
</event>
```

- サーバーの詳細（ホスト名やポート番号など）。次のように記述されます。

```
<eventinfo projectname="mg1"
projectlocation="C:/Documents and Settings/sandeep/runtime-EclipseApplication/mg1"
eventtype="onprojectcreate">
<server name="local1" hostname="localhost" port="8501" wwwroot="C:\ColdFusion9\wwwroot"/>
</eventinfo>
```

拡張機能間でのメッセージの送信

注意： ColdFusion Builder 2.0.1 で導入されました

拡張機能では、別の拡張機能にメッセージを送信して、アクションを実行したり、イベントを通知したりできます。メッセージを受信した拡張機能は、アクションを実行することも、メッセージを無視することもできます。

次のシナリオでは、この機能の利点について説明します。

エディターでの異なるファイルへの切り替えや、RDS データビューでの選択範囲の変更など、イベントを listen できる拡張機能を作成できます。この機能を使用すると、必要に応じて、ビューの内容をこの情報（例えば、新しい選択範囲に関する情報）で更新できます。

イベント変更通知を受け取るハンドラーの CFM ファイルは、次の情報を使用して ColdFusion Builder に対するコールバックコマンド sendMessageToExtension を実行します。

- メッセージ情報
- メッセージを送信する必要がある拡張機能の名前
- 実行する必要がある拡張機能内のハンドラー CFM ファイル

次の例では、ColdFusion Builder に対して、New Extension Enhancements in CFB Twister 拡張機能にメッセージを送信し、そのハンドラー NewFeaturesTestViewHandler を呼び出し、イベントタイプや他のイベント関連情報などの情報を渡すように要求します。

```
<cfsavecontent variable="command" > <cfoutput> <response>
<ide> <commands> <command type="sendMessageToExtension">
<params> <param
key="extension_name" value="New Extension Enhancements in CFB Twister" />
<param key="handlerid" value="NewFeaturesTestViewHandler" />
<param key="event_type" value="#type#" /> <cfloop
list="#structKeyList(eventinfo[1])#" index="key"> <param
key="#key#" value="#eventinfo[1][key].xmlText#" />
</cfloop> </params> </command>
</commands> </ide> </response> </cfoutput> </cfsavecontent>
<cftry>
    <cfhttp url="#callbackURL#" method="post" result="httpResult">
        <cfhttpparam type="body" value="#command#">
    </cfhttp>
<cfcatch>
    <cfset httpError = cfcatch.Message>
</cfcatch>
</cftry>
```

ハンドラーの通信

ColdFusion Builder はイベントの詳細をハンドラー NewFeaturesTestViewHandler に、次のような XML (ideEventInfo という FORM スコープ変数) で送信します。

```
<event>
    <ide version="2.0">
        <callbackurl>
            http://[ip_address:port]/index.cfm?extension=New Extension Enhancements in CFB 2.0.1
        </callbackurl>
        <extension_message>
            <from_extension_name>
                New Extension Enhancements in CFB 2.0.1
            </from_extension_name>
            <params>
                <param key="project" value="Extensions"/>
                <param key="project_path" value="[project_path]"/>
                <param key="file_path" value="[file_path]"/>
                <param key="event" value="onfilechangeineditor"/>
                <param key="event_type" value="onfilechangeineditor"/>
                <param key="editor_name" value="Adobe CFML Editor"/>
            </params>
            <from_extension_server>
                LocalCF9
            </from_extension_server>
        </extension_message>
    </ide>
    <user>
    </user>
</event>
```

extension_message ノードでは、次のような情報が渡されます。

情報	説明
from_extension_name	メッセージを送信した拡張機能。
params	メッセージのパラメーター。

拡張機能のユーザーインターフェイスの作成

入力ダイアログボックスの作成

入力ダイアログボックスは、設定ファイル (IDE_Config.xml) と CFM ページのいずれかを使用して作成できます。

注意: XML レスポンスを使用してユーザーインターフェイスを作成する場合は、該当するコードで cfsetting showdebugoutput="no" を指定してデバッグ出力を無効にしてください。

設定ファイルを使用する場合

設定ファイル (IDE_config.xml) を作成し、必要な項目を指定することによって入力ダイアログボックスを作成します。詳細については、91 ページの「[設定ファイル](#)」を参照してください。

例えば、次のようなコードを指定できます。

```
<action name="action name">
<dialog>
<input name="input variable name" label="label for the input dialog"
tooltip="tool tip" type="dir"/>
</dialog>
</action>
```

CFM ページを使用する場合

- 次のように、IDE_config.xml ファイルで、Action タグの showResponse 属性を yes に設定します。

```
<action name="generate CFC" handlerid="cfcgenerator" showResponse="yes" />
```

- CFM ハンドラー側で次の作業を行います。

- 1 次のように、cfheader タグの content-type の値を text/xml に設定します。

```
<cfheader name="Content-Type" value="text/xml">
```

- 2 次のような XML を作成して入力ダイアログボックスを作成します。

```
<cfoutput>
<response>
<ide>
<dialog >
<input name="location" Label="Enter Location" type="dir" />
</dialog>
</ide>
</response>
</cfoutput>
```

HTML ユーザーインターフェイスの作成

HTML ユーザーインターフェイスは、HTML レスポンスと XML レスポンスのいずれかを使用して作成できます。

HTML レスポンスを使用する場合

次のように、IDE_config.xml ファイルで、Action タグの showResponse 属性を yes に設定します。

```
<action name="generate CFC" handlerid="cfcgenerator" showResponse="yes" />
```

CFM ページの HTML コンテンツがダイアログボックスに表示されます。HTML コンテンツでは、Ajax と Javascript はサポートされません。HTML コンテンツでリンクを使用する場合は、相対 URL ではなく絶対 URL を使用する必要があります。

XML レスポンスを使用する場合

CFM ハンドラー側で次の作業を行います。

- 1 次のように、`cfheader` タグの `content-type` の値を `text/xml` に設定します。

```
<cfheader name="Content-Type" value="text/xml">
```

- 2 次のような XML を作成して HTML ユーザーインターフェイスを作成します。

```
<cfoutput>
<response showresponse="true">
<ide url="http://localhost:8500/local/Dynamic%20UI%20Test/handlers/main.swf" >
<dialog width="455" height="470" />
</ide>
</response>
</cfoutput>
```

指定した URL の HTML コンテンツがダイアログボックスに表示されます。

次のように、XML レスポンスを使用しても HTML ユーザーインターフェイスを表示できます。

```
<cfoutput>
<response showresponse="true">
<ide >
<dialog width="100" height="400" />
<body>
<![CDATA[
Any HTML content
]]>
</body>
</ide>
</response>
</cfoutput>
```

注意： 拡張機能では、CSS および JavaScript の相対パスはサポートされていません。CDATA セクションで指定されている Ajax コードおよび JavaScript コードは動作しません。CDATA セクションで URL を指定する場合は、絶対 URL を使用する必要があります。

XML レスポンスの構造について

Response

XML レスポンスは、次の属性を持つ `response` タグ内で定義します。

属性	説明
<code>showresponse</code>	レスポンスをユーザーに表示するかどうかに応じて、次のブール値を指定します。 <ul style="list-style-type: none">• true• false
<code>status</code>	成功またはエラーのダイアログボックスを表示します。 ステータス（成功かエラーか）に応じて、対応するアイコンがダイアログボックスに表示されます。IDE タグで <code>message</code> 属性が指定されている場合のみ適用されます。 この属性は、IDE タグ内で <code>message</code> 属性が指定されている場合のみ適用されます。

IDE

IDE タグには次の属性があります。

属性	説明
message	成功またはエラーのダイアログボックスに、指定されたメッセージを表示します。メッセージをエラーメッセージとして表示するかどうかを指定するには、response タグの status="success/error" 属性を使用します。
url	ユーザーに表示するページの URL。
handlerfile	ユーザーがメッセージダイアログボックスで「OK」をクリックしたときに呼び出すハンドラーファイルのパス。 この属性は、XML レスポンスから入力ダイアログボックスが作成される場合にのみ適用されます。

Body

body タグは、HTML コンテンツを指定する場合に使用します。body タグ内の HTML コンテンツは、CDATA セクション内で指定する必要があります。

Dialog

dialog タグは、入力ダイアログボックスを作成する場合に使用します。このタグには次の属性があります。

属性	説明
height	ダイアログボックスの高さを指定します。
width	ダイアログボックスの幅を指定します。
title	ダイアログボックスのタイトルを指定します。
image	タイトルバーに表示されるイメージのパスを指定します。
dialogclosehandler	ユーザーがメッセージダイアログボックスで「閉じる」または「キャンセル」をクリックしたときに呼び出すハンドラーファイルのパス。 この属性は、XML レスポンスから入力ダイアログボックスが作成される場合にのみ適用されます。

入力ダイアログの高さと幅を制御するには、dialog タグ内で input タグを使用します。

属性	説明
height	ダイアログボックスの高さを指定します。
width	ダイアログボックスの幅を指定します。
title	ダイアログボックスのタイトルを指定します。
image	タイトルバーに表示されるイメージのパスを指定します。ここでは、Extension フォルダーからの相対パスを指定する必要があります。

ビューの作成

ColdFusion Builder では、拡張機能を使用してビューを作成できます。また、作成するビューに、ビューアイコンやツールバーのアイテムを指定できます。

この機能は、データを表示したまま IDE の機能を使用する場合に便利です。

次のいずれかの方法でビューを作成します。

- ide_config.xml にコントリビューションを追加します。

- ハンドラーのレスポンスから動的に作成します。

IDE_config.xml へのコントリビューションの追加

IDE_config.xml の設定ファイルに詳細を指定します。次のコードに例を示します。ide_config.xml にビューのコントリビューションを追加すると、ビューの表示ダイアログボックス（ウィンドウ/ビューの表示/その他）の「ColdFusion」カテゴリの下のビューリストにビューが追加されます。

```
<viewcontributions>
  <view id="ID" title="title" icon="relative_path_to_icon" handlerid="handler_id" >
    <toolbarcontributions>
      <toolbaritem icon="relative_path_to_icon" handlerid="handler_id" />
      <toolbaritem icon="relative_path_to_icon" handlerid="handler_id" />
    </toolbarcontributions>
  </view>
</viewcontributions>
```

属性	説明
id	ビューを識別します。拡張機能によってコントリビュートするビューは、一意の ID を持つ必要があります。id は、ハンドラーのレスポンスからビューの内容を更新する場合に使用できます。
title	ビューのタイトル。
toolbaritem	ビューツールバーにアイテムを追加します。
icon	拡張機能のインストール場所に存在するイメージファイルへの相対パス。例えば、（拡張機能のインストール場所の images フォルダ内の）images/icon.png を指定すると、icon.png がビューアイコンとして表示されます。 標準サイズは、16x16 ピクセルまたは 20x20 ピクセルです。 一般的なイメージ形式はすべてサポートされています。
handlerid	関連付けるハンドラーの ID。
keepFocus (ColdFusion Builder 2.0.1 で追加)	true の場合、拡張機能ビューが動的に生成されたときにフォーカスを移動しません。 ファイルを編集していて、アクションの結果が拡張機能ビューの更新または作成になったものとします。このようなとき、エディターからビューにフォーカスを移動したくない場合があります。属性 keepFocus を true に設定すると、IDE_config.xml の <view> タグを使用して拡張機能ビューが動的に生成されるときに、フォーカスが移動しません。デフォルトでは、動的に作成された拡張機能ビューはフォーカスを受け取りません。メニューオプションのウィンドウ/ビューの表示から表示される拡張機能ビューは常に、フォーカスを受け取ります。

ビューの表示

- 1 ビューをコントリビュートする拡張機能をインストールまたは読み込みます。
- 2 ビューの表示ダイアログボックスを開きます（ウィンドウ/ビューの表示/その他）。
- 3 フォルダーリストの「ColdFusion」をダブルクリックします。
- 4 （コントリビュートする）ビューを選択し、「OK」をクリックします。

ビューが表示され、対応するハンドラー CFM ファイルが呼び出されます。

ハンドラーの使用（動的ビュー用）

CFM ハンドラーで、

- 1 次のように、cfheader タグの content-type の値を（text/xml に）設定します。

```
<cfheader name="Content-Type" value="text/xml">
```

- 2 次のような XML を作成して、HTML コンテンツを表示するビューを作成します。

```
<cfoutput>
<response showresponse="true">
  <ide url="URL">
    <view id="ID" title="title" icon="icon_path" />
  </ide>
</response>
</cfoutput>
```

次のように、XML レスポンスを使用しても HTML コンテンツを表示できます。

```
<cfoutput>
<response showresponse="true">
  <ide>
    <view id="ID" title="title" icon="icon_path" handlerid="ID" />
  <body>
    [![CDATA[any html content]]]
  </body>
</ide>
</response>
</cfoutput>
```

注意：現在のところ、ビューは、ColdFusion Builder セッション間では保持されません。

ハンドラーからのコールバックコマンドの指定

ColdFusion Builder（以前のリリース）では、ハンドラー CFM ファイルから、次のコマンドの実行を要求できます。
refreshproject、refereshfolder、refreshfile、inserttext および openfile

これらすべての操作とその他のいくつかの操作を、（実行終了時ではなく）実行フェーズ中に完了することができるようになりました。ハンドラー CFM ファイルから、コールバック URL を使用して、ColdFusion Builder への呼び出しを送信できます。ColdFusion Builder では、ideeventinfo の XML で、次のようにコールバック URL が提供されます。

```
<event>
  <ide version="2.0">
    <callbackurl>
      callbackURL
    </callbackurl>
  </ide>
</event>
```

エディター関連のコンテキスト情報を取得するコマンドだけでなく、サーバー固有の情報（データソースやテーブルの詳細など）を取得することもできます。

次の表に、実行フェーズで様々な操作を実行する便利なコマンドの一覧を示します。データを要求するすべてのコマンドでは、コマンドの結果が XML 形式で返されます。各コマンドで返される結果についても表に記載します。

名前	アクション	入力	結果
refreshFile	絶対パスまたは相対パスを送信して、指定したファイルを更新します。相対パスを指定する場合は、プロジェクト名を指定します。 projectName はオプションです。対象ファイルがプロジェクトに含まれる場合に指定できます。	<command type="refreshFile"> <params> <param key="filename" value="filePath" /> <param key="projectname" value="#arguments.projectName#" /> </params> </command>	該当なし
refreshFolder	指定したフォルダーを更新します。 projectName はオプションです。対象フォルダーがプロジェクトに含まれる場合に指定できます。	<command type="refreshFolder"> <params> <param key="foldername" value="folderPath" /> <param key="projectname" value="projectName" /> </params> </command>	該当なし
refreshProject	指定したプロジェクトを更新します。	<command type="refreshProject"> <params> <param key="projectname" value="projectName" /> </params> </command>	該当なし
openFile	絶対パスまたは相対パスを送信して、指定したファイルを編集するために開きます。相対パスを指定する場合は、プロジェクト名を指定します。 projectName はオプションです。対象フォルダーがプロジェクトに含まれる場合に指定できます。 lineNumber は、オプションのパラメーターです (ColdFusion Builder 2.0 で追加)。	<command type="openFile"> <params> <param key="filename" value="filePath" /> <param key="projectname" value="projectName" /> <param key="linenumber" value="10" /> </params> </command>	該当なし
insertText	指定したテキストをアクティブなエディターに挿入します。このコマンドを実行すると、エディターで選択されているテキストが置換されます。	<command type="inserttext"> <params> <param key="text"> <![CDATA[text_to_insert]]> </param> <param key="insertmode" value="replace/insert" /> </params> </command>	該当なし
getServers	Server Manager に追加されているサーバーを一覧表示します。	<command type="getservers" > </command>	<?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0"> <callbackurl> [call back URL] </callbackurl> <command_results> <command_result type="getservers"> <servers> <server name=""/> ... </servers> </command_result> </command_results> </ide> </event>

名前	アクション	入力	結果
getDatabases	<p>特定のサーバーのデータソースを一覧表示します。</p> <p>出力変数は、データソース名のカンマ区切りリストです。</p> <p>サーバー名が指定されなかった場合は、デフォルトで、拡張機能が実行されるサーバーになります。</p>	<pre><command type="getdatabases" > <params> <param key="server" value="comma_seperated_server_names" /> </params> </command></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0"> <callbackurl> [callback url] </callbackurl> <command_results> <command_result type="getdatabases"> <databases> <datasource name=" " server=" " /> ... </databases> </command_result> </command_results> </ide> </event></pre>
getTables	<p>特定のデータソースのテーブルの詳細を取得します。</p> <p>サーバー名が指定されなかった場合は、デフォルトで、拡張機能が実行されるサーバーになります。</p>	<pre><command type="gettables"> <params> <param key="server" value="servers" /> <param key="datasource" value="datasources" /> </params> </command></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0"> <callbackurl> [callback url] </callbackurl> <command_results> <command_result type="gettables"> <tables> <table datasource=" " name="table_Name" server=" " > <field cfsqltype=" " cftype=" " javatype=" " name=" " nullallowed=" " primarykey="true false" type=" " /> [list of other columns] </table> ... [list of other tables] </tables> </command_result> </command_results> </ide> </event></pre>
getTable	<p>特定のデータソースの特定のテーブルの詳細を取得します。</p> <p>サーバー名が指定されなかった場合は、デフォルトで、拡張機能が実行されるサーバーになります。</p>	<pre><command type="gettable"> <params> <param key="server" value="servername" /> <param key="datasource" value="dsn" /> <param key="table" value="tablename" /> </params> </command></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0"> <callbackurl> [callback url] </callbackurl> <command_results> <command_result type="gettable"> <table datasource=" " name=" " server=" " > <field cfsqltype=" " cftype=" " javatype=" " name=" " nullallowed=" " primarykey="true false" type=" " /> ... </table> </command_result> </command_results> </ide> </event></pre>
searchFile	<p>テキストコンテンツファイルを検索して、その内容を返します。</p> <p>getcontent は、テキストファイルの場合にのみ使用します。</p> <p>searchDirection では、親フォルダー (up) とサブフォルダー (down) のいずれに対して検索を実行するかを指定します。</p>	<pre><command type="searchfile" > <params> <param key="fileName" value="ide_config.xml" /> <param key="from" value="search_from_this_file_or_folder" /> <param key="getcontent" value="true/false" /> <param key="searchDirection" value="up/down" /> <param key="matchfolder" value="true/false" /> </params> </command></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0"> <callbackurl> [callback url] </callbackurl> <command_results> <command_result type="searchfile"> <files> <file path=" " type="file" /> </files> </command_result> </command_results> </ide> </event></pre>

名前	アクション	入力	結果
getfunctionsandvariables (拡張機能の詳細については、「ColdFusion Builder 2.0.1 の既存のコールバックコマンドに対する拡張機能」を参照してください)	指定したファイル内の関数と変数を一覧表示します。	<pre><command type="getfunctionsandvariables" > <params> <param key="filePath" value="path"/> </params> </command></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <event> <ide version="2.0"> <callbackurl> [callback url] </callbackurl> <command_results> <command_result type="getfunctionsandvariables"> <cfmlfile type="component interface cfml"> <variables> <variable function="name of function if variables is inside function" name="" type="" /> ... </variables> <functions> <function file="" name="" /> ... </functions> </cfmlfile> </command_result> </command_results> </ide> </event></pre>
getProjects (ColdFusion Builder 2.0.1 で追加)	<p>ColdFusion Builder プロジェクトのリストを取得します。</p> <p>次の詳細が含まれます。</p> <ul style="list-style-type: none"> name : プロジェクトの名前 path : プロジェクトの絶対パス isopen : プロジェクトが開かれたか、閉じられたか。開かれた場合は値は true です。 iscfproject : ColdFusion プロジェクトでない場合は false。ColdFusion プロジェクトの場合は、サーバー名などの追加情報が返されます。 	<pre><commands> <command type="getprojects" > </command> </commands></pre>	<pre><?xml version="1.0" encoding="UTF-8"?><event> <ide version="2.0"> <callbackurl> http://<ip_address:port>/index.cfm?extension=untitled </callbackurl> <command_results> <command_result type="getprojects"> <projects> <project name="TestProject" path="<Project_Path">"> <cfproject> <server name="local_server"/> </cfproject> </project> </projects> </command_result> </command_results> </ide></event></pre>
reloadExtensions (ColdFusion Builder 2.0.1 で追加)	拡張機能を使用して拡張機能をリロードします。	<pre><commands> <command type="reloadExtensions" > </command> </commands></pre>	該当なし
sendMessageToExtension (ColdFusion Builder 2.0.1 で追加)	詳細については、104 ページの「 拡張機能間でのメッセージの送信 」を参照してください。		

ColdFusion Builder 2.0.1 の既存のコールバックコマンドに対する拡張機能

コールバックコマンド getfunctionsandvariables によって返される応答には、変数スコープ関連の情報が含まれるようになっています。例えば変数が関数で var スコープになっている場合、スコープを特定する拡張機能を作成できます。

次の3つの変数が識別されます。

- 引数スコープ
- Var スコープ
- タグで作成された変数（例えば、cfquery で作成された変数）

ハンドラー CFM ファイルのその他のスコープは、this や application などの変数名の接頭辞から特定できます。

次に示すのは、スコープ関連の情報を含む getfunctionsandvariables の出力の例です。

```
<event>
  <ide version="2.0">
    <callbackurl>
      http://ip_address:port/index.cfm?extension=untitled
    </callbackurl>
    <command_results>
      <command_result type="getfunctionsandvariables">
        <cfmlfile type="cfml">
          <variables>
            <variable name="i" type="numeric"/>
          </variables>
          <functions>
            <function name="testFunc" file="C:\ColdFusion9\wwwroot\MyExtension\handlers\test.cfm">
              <variables>
                <variable name="j" type="numeric" function="testFunc" scope="varscope"/>
                <variable name="arg1" function="testFunc" scope="argument"/>
                <variable name="variables.1" type="numeric" function="testFunc"/>
              </variables>
            </function>
          </functions>
        </cfmlfile>
      </command_result>
    </command_results>
  </ide>
</event>
```

例：コールバック URL を使用したコマンドの実行

ステップ 1：ideeventinfo の XML を取得

```
<cfparam name="ideeventinfo" >
<cfset xmldoc=xmlParse (ideeventinfo)>
```

ステップ 2：ideeventinfo の XML からコールバック URL を取得

```
<cfset callbackurl= xmldoc.event.ide.callbackurl.xmlText>
```

ステップ 3：ColdFusion Builder にコマンドを送信する XML を作成

```
<cfsavecontent variable="commandxml" >
<cfoutput>
  <response>
    <ide>
      <commands>
        [Any command xml to be executed]
      </commands>
    </ide>
  </response>
</cfoutput>
</cfsavecontent>
```

ステップ 4: コールバックコマンドを実行して、レスポンスを取得

```
<cfhttp method="post" url="#callbackurl#" result="commandresponse" >
  <cfhttpparam type="body" value="#commandxml#" >
</cfhttp>
```

コールバックコマンド使用中のエラー処理

データを返すコマンドの実行中にエラーが発生した場合、コマンドのレスポンス XML によって、エラーノードと考えられるエラー原因の情報が提供されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <ide version="2.0">
    <callbackurl>
      [callback url]
    </callbackurl>
    <command_results>
      <error>
        [error message]
      </error>
    </command_results>
  </ide>
</event>
```

その他のリソース

- [ColdFusion Builder 2 Extensions - Understanding Callback Commands](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sandeep Paliwal が、ColdFusion Builder 2 拡張機能でのコールバックコマンドの可能性について説明します。

拡張機能ビューの使用

拡張機能ビューでは、拡張機能のインストール、アンインストール、インポート、リロードを行えます。

拡張機能ビューがまだワークベンチに表示されていない場合、拡張機能ビューを追加するには、ウィンドウ/ビューの表示/その他を選択します。次に、ビューの表示ダイアログボックスで、ColdFusion/拡張機能ビューを選択します。

拡張機能のインストールとアンインストール

- 1 拡張機能ビューで  をクリックし、インストールするアーカイブファイルを選択します。

注意: ColdFusion Builder に付属する拡張機能をインストールする場合は、ColdFusion Builder のインストール先の Extensions ディレクトリにアーカイブファイルが保存されています。

- 2 拡張機能のインストールウィザードに従ってインストール作業を進めます。

- 3 サーバーを選択します。サーバーがまだ 1 台も設定されていない場合は、「サーバーの追加」をクリックして新しいサーバーを追加します。サーバーを追加する方法については、66 ページの「[ColdFusion サーバーの追加](#)」を参照してください。


複数のサーバーに同じ拡張機能をインストールすることもできます。例えば、CF9 ローカルサーバーと CF8 ローカルサーバーを選択できます。ただし、アクティブにできるサーバーは、拡張機能によって使用されるサーバーのみです。

- 4 サーバーを選択するときにパスが自動的に挿入されない場合は、ColdFusion の Web ルートのパスを入力します。例えば、C:\ColdFusion9\wwwroot のように入力します。

5 拡張機能のインストール先となる Web ルート内のフォルダーを選択します。アーカイブファイルは、このインストール先に解凍されます。

6 「インストール」をクリックします。


インストールされている拡張機能の簡単な説明を表示するには、拡張機能の名前をクリックします。複数のサーバーに同じ拡張機能をインストールした場合、アクティブにできるサーバーは 1 つのみです。これを行うには、アクティブにするサーバーをアクティブサーバードロップダウンリストから選択します。

拡張機能をアンインストールするには、拡張機能ビューで拡張機能を選択し、 をクリックします。拡張機能は、IDE からのみアンインストールすることも、IDE とコンピューターのファイルシステムから同時に削除することもできます。ファイルシステムから拡張機能を完全に削除した後に、元に戻すことはできません。


注意：複数のサーバーにインストールした拡張機能をアンインストールする場合は、各サーバーで同じアンインストール手順を繰り返します。

拡張機能のインポートとリロード

拡張機能ビューでは、拡張機能のインポートとリロードを行えます。

- 既存のディレクトリから ColdFusion Builder に拡張機能をインポートできます。拡張機能ビューで  をクリックし、拡張機能ファイルが保存されているディレクトリを選択します。

注意：既存の拡張機能を使用して新しい拡張機能を開発するときは、拡張機能をインポートすると便利です。

- インストール済みの拡張機能の設定ファイルを変更した場合は、拡張機能をリロードします。設定ファイルの変更は、拡張機能をリロードしないと反映されません。拡張機能ビューで  をクリックすると、インストール済みの拡張機能がすべてリロードされます。

拡張機能のデバッグとパッケージ化

設定ファイルから拡張機能をロードする際にエラーが発生した場合は、Eclipse ログにエラーが記録されます。エラーをデバッグするには、ウィンドウ/ビューの表示/その他/一般/エラー・ログを選択して、ColdFusion Builder エラーログを表示します。ColdFusion Builder のエラーログがエラーログビューに表示されます。

拡張機能をパッケージ化するには、次のコンテンツを ZIP ファイルに追加します。

- 設定ファイル (IDE_config.xml)
- すべてのハンドラー CFM ファイルを含むハンドラーディレクトリ

注意：設定ファイルとハンドラーディレクトリは、ZIP ファイルのルートの直下に追加してください。

ColdFusion Builder Extension Creator を使用した拡張機能の作成とパッケージ化

ColdFusion Builder Extension Creator ウィザードは、拡張機能の作成とパッケージ化の手順を案内します。このウィザードでは、設定ファイル (IDE_Config.xml) とハンドラーファイル (CFM) の作成、メタデータ要素の指定、コンテキストメニューの追加、ユーザーインターフェイスの作成を行います。

注意：ColdFusion Builder Extension Creator ウィザードは、ColdFusion サーバーの設定時に「拡張機能のインストール」を選択した場合にのみインストールされます。インストール済みの拡張機能のリストは、拡張機能ビューで確認できません。

- 1 ナビゲータービューで、拡張機能を作成するプロジェクトを右クリックし、**Adobe Extension Builder / New** を選択します。
- 2 拡張機能を保存する場所を参照して選択するか、手動で入力します。拡張機能と同じ名前のフォルダーに拡張機能ファイルを保存するには「**Create Folder With Extension Name**」を選択します。
- 3 「**Extension Details**」タブを選択して、拡張機能の名前と詳細を入力します。拡張機能のインストール時に表示するライセンス契約のテキストを指定することもできます。

メニューコントリビューションの指定

コンテキストメニューを指定するには、「**Menu contributions**」タブを選択して「**Add Menu**」をクリックします。コンテキストメニューとは、右クリックアクションで表示されるポップアップメニューのことです。

- メニューの名前と、メニューを表示するターゲットビューを入力します。ターゲットビューとして指定できるのは、ナビゲータービュー、プロジェクトビュー、RDS ビュー、アウトラインビュー、CFML エディターのいずれかです。「**Save Menu**」をクリックします。
- メニューアイテムを指定するには、「**Add Action**」をクリックします。
 - 1 アクションにはハンドラーを関連付けることができます。ハンドラーは、サーバーから取得したコード生成操作を表示します。ハンドラー ID とハンドラー名を指定して「**Show Response**」を選択し、サーバーのレスポンスを表示します。「**保存**」をクリックします。
 - 2 ハンドラーが呼び出される前に、ユーザー入力を取得できます。これを行うには、「**Add Input**」をクリックします。
 - 3 入力変数名、入力ダイアログボックスのラベル、ツールヒント、および入力変数タイプを指定します。
 - 4 ユーザー入力の検証に使用するパターンと、検証の失敗時に表示するエラーメッセージを指定することもできます。
 - 5 入力フィールドを必須フィールドとして指定するには、「**Required**」を選択します。デフォルトでチェックボックスをオンにするには、「**Checked**」を選択します。「**保存**」をクリックします。
- メニューの表示を制御するためのフィルターを指定するには、「**Add Filter**」をクリックします。フィルターは、プロジェクトレベル、フォルダーレベル、およびファイルレベルで指定できます。

ハンドラーの指定

インストール済みのハンドラーに関する詳細を表示するには、「**Handlers**」タブを選択します。

- 1 新しいハンドラーを指定するには、「**Add Handler**」をクリックし、必要な詳細を入力します。
- 2 ハンドラーを呼び出す前にユーザー入力を取得するには、「**Generates Response**」を選択します。
- 3 入力ダイアログボックスのタイトルを指定し、高さと幅をピクセル単位で指定します。「**Save Handler**」をクリックします。

拡張機能のインストールウィザードの作成

拡張機能のインストール手順を案内するインストールウィザードを作成するには、「**Install Wizards**」タブを選択します。

- 1 ウィザードの高さと幅をピクセル単位で指定します。ウィザードにハンドラーを関連付けることもできます。「**保存**」をクリックします。
- 2 ウィザードのタイトルを指定して、「**Save**」をクリックします。
- 3 入力変数名、入力ダイアログボックスのラベル、ツールヒント、および入力変数タイプを指定します。ユーザー入力の検証に使用するパターンと、検証の失敗時に表示するエラーメッセージを指定することもできます。

拡張機能のパッケージ化

- 1 ナビゲータービューで、`ide_config.xml` ファイルと、すべてのハンドラー CFM ファイルを含む Handler フォルダを右クリックし、Adobe Extension Builder / Package を選択します。これらのファイルが ZIP ファイルに追加されます。
- 2 ZIP ファイルを保存する場所を参照して選択します。

拡張機能からコードアシストへのコントリビュート

拡張機能から、コードアシストに提案候補を追加できます。

- 1 IDE_config.xml で、次の作業を行います。

- 関数パラメーターのコードアシストに提案候補を追加するには、次のコードを追加します。

```
<codeassistcontribution>
  <functions>
    <function name="linkTo" variableName="event" componentName="component_name"
      handlerId="CodeAssistHandler">
      <parameter index="1" />
    </function>
  </functions>
</codeassistcontribution>
```

- 変数のコードアシストに提案候補を追加するには、次のコードを追加します。

```
<codeassistcontribution>
  <variables>
    <variable name="event" componentName="component_name" handlerId="CodeAssistHandler">
      <parameter index="1" />
    </variable>
  </variables>
</codeassistcontribution>
```

属性	説明
name	表示される関数の名前。
variableName	変数の名前。指定すると、関数を呼び出す変数名が、この変数名と一致する場合にのみ、ハンドラーが呼び出されます。 変数の場合、variablename を指定すると、エディターにその変数のコントリビューションが表示されます。
componentName	指定すると、関数を呼び出す変数のタイプが、指定したコンポーネント名と一致する場合にのみ、ハンドラーが呼び出されます。
handlerID	関連付けるハンドラーの ID。
index	ハンドラーがコードアシストを提供可能なパラメーターのインデックスを指定します。指定しなかった場合は、関数内のすべてのパラメーターで、ハンドラーが呼び出されます。

- 2 ハンドラー CFM ファイルで、次の形式で、コードアシストの提案候補を追加するレスポンスを指定します。

```
<codeassist_response>
<proposal display="display_value" insert="insert_this_value" inquotes="true/false"/>
</codeassist_response>
```

属性	説明
display	ここに指定した値が提案候補ウィンドウに表示されます。
insert	その提案候補が選択された場合、ここに指定した値がエディターに挿入されます。
inquotes	true に設定すると、エディターに挿入されるテキストが二重引用符で囲まれます。

- 3 拡張機能をインストールまたは読み込みます。ColdFusion Builder のコードアシストを起動すると、提案候補ウィンドウに関数が表示されます。

その他のリソース

- [How to contribute to code assist from extensions](#)

ColdFusion Builder エンジニアリングチームのメンバーである Sandeep Paliwal が、拡張機能を使用したコードアシストへの寄与に関してよくある質問に答えます。

拡張機能による起動ページ設定のサポート

拡張機能を使用して、フレームワークアプリケーションの開始ページの URL を動的に生成することができます。

ColdFusion Builder の機能（ColdFusion デバッガー、外部ブラウザの使用、ColdFusion アプリケーションとしての実行など）では、この URL が開始ページの URL として使用されます。

生成される URL には、URL に追加されるクエリパラメーターも含まれます。

注意：ColdFusion Builder では、プロジェクトレベルで 78 ページの「[起動ページの設定](#)」を行えます。この設定は、単純なアプリケーションの場合に便利です。この機能では、フレームワークに基づいて、プロジェクトの起動ページを設定するオプションが提供されます。

IDE_config.xml への開始ページのコントリビューションの追加

IDE_config.xml の設定ファイルに詳細を指定します。次のコードに例を示します。

```
<startpagecontribution>
<urlgenerator handlerid="ID" />
</startpagecontribution>
```

ハンドラーの通信

ColdFusion Builder はイベントの詳細をハンドラーに、次のような XML（ideEventInfo という FORM スcope変数）で送信します。

```
<event>
  <ide version="2.0">
    <callbackurl>
      ...
    </callbackurl>
    <startpage_request>
      <project name="" projectlocation="">
        <resource name="" path="" type="file|Folder|Project"/>
        <server name="" hostname="" port="" wwwroot="" />
      </project>
    </startpage_request>
  </ide>
</event>
```

この情報は、URL を生成する場合に使用できます。

拡張機能を使用した開始ページの URL の生成

- 1 プロジェクトビューで、プロジェクトを右クリックし、プロパティ / ColdFusion プロジェクトを選択します。
- 2 「開始ページの設定」の拡張機能を使用リストで使用可能なオプションを選択します。
- 3 「OK」をクリックします。

次回、アプリケーションを実行すると、IDE_config.xml で指定した ID で特定されるハンドラーが呼び出されます。プロジェクトまたは CFM に関連する情報が表示されます。

例

```
<cfheader name="Content-Type" value="text/xml">
<cfoutput>
<startpage_response>
  <url>Any URL</url>
</startpage_response>
</cfoutput>
```

第 10 章：拡張機能の使用

サーバーの作成時に「拡張機能のインストール」オプションを選択した場合は、ColdFusion Builder とともに次の拡張機能が自動的にインストールされます。詳しくは、66 ページの「[ColdFusion サーバーの追加](#)」を参照してください。

- Adobe CFC Generator
- ActionScript Class Generator

ここでは、これらの拡張機能の使用方法について説明します。拡張機能の開発方法については説明しません。

注意：拡張機能のユーザーインターフェイスが、コンピューターのデフォルトブラウザに表示されます。Mozilla をブラウザとして指定する場合は、環境設定ダイアログボックスに移動して、ColdFusion / 拡張機能を選択します。「拡張機能に Mozilla ブラウザーを使用します」を選択します。

Adobe CFC Generator

Adobe CFC Generator ジェネレーターは、データベーステーブルから ORM CFC (ColdFusion コンポーネント) を生成します。Adobe CFC Generator 拡張機能を使用すると、従来の CFC または ORM CFC を一連のテーブルから作成できます。

ORM (Object Relational Mapping : オブジェクトリレーショナルマッピング) は、オブジェクトモデルを使用してリレーショナルデータベースのデータを格納および取得するためのマッピング方針を定義するプログラミング手法です。Adobe CFC Generator では、関係のマッピングを定義し、結合条件を指定し、CFC のサービスの生成できます。ORM の詳細については、ColdFusion ORM を参照してください。

Adobe CFC Generator のインストール

サーバーの作成時に「拡張機能のインストール」オプションを選択した場合は、ColdFusion Builder とともに Adobe CFC Generator 拡張機能がインストールされます。

サーバーの作成時に拡張機能をインストールしなかった場合は、115 ページの「[拡張機能のインストールとアンインストール](#)」の説明に従って拡張機能をインストールできます。Adobe CFC Generator 拡張機能のアーカイブファイル (AdobeCfcGenerator.zip) は、ColdFusion Builder に出荷時から付属しています。このアーカイブファイルは、ColdFusion Builder のインストール先の Extensions ディレクトリにあります。

Adobe CFC Generator の使用 (ColdFusion Builder 2.0 の場合)

- 1 ColdFusion Builder パースペクティブで、「RDS データビュー」を選択します。
- 2 RDS ユーザー名とパスワードの設定が既に完了している場合は、その RDS ユーザー名とパスワードを指定します。
注意：RDS パスワードの設定は ColdFusion Administrator で行います。同じく ColdFusion Administrator で管理される ColdFusion Administrator パスワードと RDS パスワードを混同しないでください。
- 3 設定したサーバーで利用可能なデータソースのリストが表示されます。
- 4 CFC を生成するデータソースおよびテーブルを選択します。Shift キーを押しながらテーブル名を選択すると、一度に複数のテーブルを選択できます。
- 5 選択したテーブルを右クリックし、「Adobe CFC Generator」を選択して、次のいずれかのオプションを選択します。

注意：「Adobe CFC Generator」オプションは、テーブル名を右クリックした場合にのみ表示されます。データソース自体またはデータソース内のレコードをクリックしても、これらのオプションは表示されません。

Create CFC ColdFusion コンポーネントを生成します。

Create ORM CFC ORM CFC を生成してデータベースにマッピングします。

Create CFC

- 1 選択したデータソース内のテーブル（APP.EMPLOYEES テーブルなど）を右クリックし、Adobe CFC Generator / Create CFC を選択します。
- 2 Adobe CFC Generator ダイアログボックスで次の操作を行います。
 - 生成された CFC を保存する場所を指定します。
 - 「Generate Service」を選択すると、メソッドが自動的に生成されます。

追加の機能を利用できるように CFC のソースコードを変更することもできます。

Create ORM CFC

- 1 選択したデータソース内のテーブル（APP.EMPLOYEES テーブルなど）を右クリックし、Adobe CFC Generator / Create ORM CFC を選択します。
- 2 生成された CFC を保存する場所を指定します。
- 3 Adobe CFC Generator ダイアログボックスのダイナミックデータグリッドに、選択したデータベーステーブルとフィールドが表示されます。このグリッドではデータの編集やソートが可能です。

注意：このデータグリッドは Ajax のリッチユーザーインターフェイスであるため、ロードに時間がかかる場合があります。

ORM CFC は、次のようにカスタマイズできます。

- CFC 名の変更：自動生成された CFC 名は、「CFC Name」フィールドで編集できます。
 - プロパティ名の変更：プロパティ名は、「Property Name」フィールドで編集できます。
 - プロパティタイプの変更：プロパティタイプを編集するには、Property Type 列に表示されるプロパティタイプをクリックし、ドロップダウンリストからプロパティタイプを選択します。
 - プライマリキーの指定：必要な列で「True」または「False」を選択することにより、プライマリキーの設定を編集できます。
 - それぞれの列名の横にあるチェックボックスをオンまたはオフにすることにより、CFC で列を生成するか除外するかを選択できます。
 - データのソート：データグリッドに表示されているテーブルと列をソートできます。ソートの方向は、昇順または降順です。テーブル名をソートするには、テーブル列をクリックして、ソート順を選択します。列データをソートするには、列見出しをクリックして、ソート順を選択します。
- 4 基本的な ORM CFC を生成するには、「Generate Code」リンクをクリックします。

ORM CFC の関係および結合条件を定義するには、次のようにします。

- a Relationships データグリッドで「Insert」をクリックします。新しい行が挿入されます。
- b 「Relation Name」フィールドをクリックして、関係の名前を入力します。例えば、Art テーブルと Artists テーブルの関係を定義できます。
- c Target Table ドロップダウンリストからターゲットテーブルを選択します。
- d Multiplicity ドロップダウンリストを選択して、関係タイプを指定します。

- e 必要に応じてリンクテーブルを指定します。
 - f 「Save」をクリックします。
 - g 関係を定義するときは、結合条件を定義します。指定した関係の結合条件を定義するには、手順 3 で選択したターゲットテーブルをクリックします。
 - h Join Conditions データグリッドで「Insert」をクリックします。
 - i ソースドロップダウンリストからソースフィールドを選択します。
 - j ターゲットドロップダウンリストからターゲットフィールドを選択します。
 - k 「Save」をクリックします。
 - l 「Generate Code」リンクをクリックします。
- 5 ナビゲータービューで、CFC を含むフォルダーを右クリックし、「更新」を選択します。追加の機能を利用できるように CFC のソースコードを変更することもできます。

Adobe CFC Generator の使用 (ColdFusion Builder 2.0.1 の場合)

Adobe CFC Generator は ColdFusion Builder 2.0.1 で拡張されて、ユーザーエクスペリエンスが向上しました。

- 1 ColdFusion Builder で RDS データビューを選択します。サーバービューで ColdFusion サーバーを追加してあるものとします。
- 2 データソースから CFC を生成するテーブルを選択します。
- 3 データソースで選択したテーブルを右クリックし、**Adobe CFC Generator** / **Create ORM CFC** を選択します。
- 4 適切なオプションを選択することで、サービスおよびスクリプト CFC の生成を選択できます。「Generate Services」オプションを選択すると、選択した各テーブルに対して CRUD（作成、読み取り、更新、削除）操作が作成されます。スクリプト CFC では、タグではなくスクリプトスタイルでコードを生成できます。
- 5 生成された CFC を保存する場所を指定します。
- 6 Adobe CFC Generator に選択したテーブルとそのフィールドが表示されます。

Adobe CFC Generator ダイアログボックス

テーブルの移動

Adobe CFC Generator でテーブルを移動するには、テーブルヘッダーをクリックしてドラッグします。

テーブルの編集

- 1 テーブルヘッダーをダブルクリックしてテーブルエディターを表示します。
- 2 次のことを行うことができます。
 - 必要に応じて、CFC の名前を変更できます。
 - プロパティ名をクリックして編集します。
 - 変更するプロパティタイプをクリックし、ドロップダウンリストから選択します。
 - チェックボックスを使用してテーブルのフィールドを個別に選択または選択解除します（グリッドの最初の列）。選択しないと、そのフィールドにはプロパティが作成されません。テーブルのフィールドを何も選択しないと、そのテーブルの CFC は生成されません。

関係の作成、編集、削除

- 異なるテーブルの2つのフィールド間に関係を作成するには、一方のフィールドを他方にドラッグします。接続された2つのフィールドの間にコネクタが表示されます。
- Edit Relationship 画面ダイアログボックスで関係を編集するには、コネクタをダブルクリックします。
 - プライマリキーと外部キーの両方のプロパティの作成を選択します。
 - 関係の各端のカーディナリティを設定し、関係の名前を指定します。
 - (多対多関係の場合) ドロップダウンリストからリンクテーブルを選択します。
 - 関係の条件を編集するには、編集するフィールドをクリックします。
 - 新しい条件を追加するには、テーブルの最後にある空の行をクリックします。
 - 関係を削除するには、テーブルの最後の列にある「X」マークをクリックします。
- 関係を削除するには、リンクしている線の間にある円をクリックします。

AS Class Generator

AS Class Generator は ActionScript クラスを生成します。この ActionScript クラスでは、オフラインメタデータを含めることも除外することもできます。オフラインメタデータは、ColdFusion 9 の ColdFusion-AIR オフライン統合機能の一部です。ActionScript クラスは CFC から生成することも、データベーステーブルから直接生成することもできます。

AS Class Generator のインストール

サーバーの作成時に「拡張機能のインストール」オプションを選択した場合は、ColdFusion Builder とともに AS Class Generator 拡張機能がインストールされます。

サーバーの作成時に AS Class Generator 拡張機能をインストールしなかった場合は、AS Class Generator 拡張機能のアーカイブファイル (ASClassGenerator.zip) を使用して後からインストールできます。AS Class Generator 拡張機能のアーカイブファイルは、ColdFusion Builder に出荷時から付属し、ColdFusion Builder のインストール先の Extensions ディレクトリ内にあります。拡張機能のインストールの詳細については、115 ページの「[拡張機能のインストールとアンインストール](#)」を参照してください。

AS Class Generator の使用

AS Class Generator アプリケーションは次のレベルで使用できます。

- RDS データビュービュー
- プロジェクトナビゲータービュー

AS クラスの生成は、次のシナリオでサポートされています。

- データベーステーブルから生成
- CFC から生成

AS クラスは、ORM CFC または従来の CFC から生成できます。

データベーステーブルからの AS クラスの生成

- ColdFusion Builder パースペクティブで、「RDS データビュー」を選択します。
- RDS ユーザー名とパスワードの設定が既に完了している場合は、その RDS ユーザー名とパスワードを指定します。

注意: RDS パスワードの設定は ColdFusion Administrator で行います。同じく ColdFusion Administrator で管理される ColdFusion Administrator パスワードと RDS パスワードを混同しないでください。

- 3 設定したサーバーで利用可能なデータソースのリストが表示されます。
- 4 データソースを選択します。
- 5 データソース内のテーブルを右クリックし、Generate AS Class from DB Table / Generate AS Class を選択します。

注意: 「Generate AS Class from DB Table」オプションは、テーブル名を右クリックした場合にのみ表示されます。データソース自体またはデータソース内のレコードをクリックしても、これらのオプションは表示されません。

- 6 Generate AS Class ダイアログボックスで、次の詳細を入力します。

- ActionScript クラスファイルの保存先を指定または選択します。
- 次のメタデータタグを含めるには、「Include AS Metadata Tags」チェックボックスを選択します。

[RemoteClass] ColdFusion サーバー上の ColdFusion CFC または Java オブジェクトを、生成される ActionScript クラスにマッピングします。

[Entity] 生成される ActionScript クラスを、AIR SQLite データベース内のテーブルとして永続的に保存する必要があることを示します。このタグは ColdFusion-AIR オフライン統合タグです。

[Id] ActionScript クラスを AIR SQLite データベース内のテーブルとして永続的に保存した後、AS クラスプロパティをプライマリキーとして指定します。

- Web ルートを基準として、リモート CFC クラスの相対パスをドット表記法で入力します。例えば、CFC の絶対パスが C:\ColdFusion9\wwwroot\CFProj\xyz.cfc である場合、ドット表記法によるパスは CFProj.xyz となります。

注意: このエントリは、「AS Metadata Tags」チェックボックスを選択しなかった場合には無視されます。

- パッケージの名前を入力します。ここで使用するネーミング規則は、ActionScript のネーミング規則と同じです。

- 7 databasetablename.as というネーミング規則で ActionScript クラスファイルが作成されます。

生成された ActionScript クラスファイルの確認

次のコード例は Employees.as という名前の ActionScript クラスファイルを示しています。このクラスファイルは cfdocexamples データソース内の Employees テーブルから生成されます。

コード	説明
package package1	「package1」という名前でパッケージが作成されます。
[RemoteClass(alias="cfproj.xyz")]	ActionScript クラスがサーバー上の cfproj/xyz.cfc にマッピングされます。
[Entity] public class EMPLOYEES { [Id] public var EMP_ID:int; public var FIRSTNAME:String; public var LASTNAME:String; public var EMAIL:String; public var PHONE:String; public var DEPARTMENT:String; public var LOCATION:String; public var IM_ID:String; public function EMPLOYEES() {} }	クラス名はデータベーステーブル名と同じになります。このクラスの各プロパティは、Employees データベーステーブル内の列に対応しています。各プロパティのデータ型は、列のデータ型のイントロスペクションが完了した後に自動的にマッピングされます。

CFC からの AS クラスの生成

ORM CFC から生成する場合

- 1 ORM CFC を作成します。

注意: ActionScript クラスの生成対象となる CFC は、ColdFusion Web ルート内に存在する必要があります。

- 2 プロジェクトナビゲーターで、ORM CFC を右クリックし、Generate AS Class from CFC / Generate AS Class を選択します。

3 Generate AS Class ダイアログボックスで、次の詳細を入力します。

- ActionScript クラスファイルの保存先を指定または選択します。
- パッケージの名前を入力します。ここで使用するネーミング規則は、ActionScript のネーミング規則と同じです。
- [RemoteClass] タグを含めるには、「Include AS Metadata tags」チェックボックスを選択します。
- 次のタグを含めるには、「Include CF-AIR Offline Metadata」チェックボックスを選択します。
 - [Entity]
 - [Id]
- 各プロパティのプライマリキーやデータ型など、必要な情報が ORM CFC に既に含まれている場合は、ColdFusion データソースを入力する必要はありません。

従来の CFC から生成する場合

1 ナビゲータービューで CFC を右クリックし、Generate AS Class from CFC / Generate AS Class を選択します。

注意：ActionScript クラスの生成対象となる CFC は、ColdFusion Web ルート内に存在している必要があります。

2 Generate AS Class ダイアログボックスで、次の詳細を入力します。

- ActionScript クラスファイルの保存先を指定または選択します。
- パッケージの名前を入力します。ここで使用するネーミング規則は、ActionScript のネーミング規則と同じです。
- [RemoteClass] タグを含めるには、「Include AS Metadata tags」チェックボックスを選択します。
- 次のタグを含めるには、「Include CF-AIR Offline Metadata」チェックボックスを選択します。
 - [Entity]
 - [Id]
- 各プロパティのプライマリキーとデータ型が CFC で指定されていない場合は、ColdFusion データソース名を入力します。

注意事項

- CF-AIR オフライン統合を使用する場合、列固有のメタデータタグ [Column] の追加は現在のところサポートされていません。ただし、ActionScript クラスファイルを生成した後に、手動で列固有のメタデータタグを追加することは可能です。
- RDS は、データソース内のテーブルのプライマリキーに関する情報を提供しません。プライマリキーを指定するには cfdbinfo タグを使用します。cfdbinfo タグでは、Microsoft Access などの ODBC ドライバーがサポートされていません。そのため、サポートされていないデータソースを使用する場合は、生成された ActionScript クラスファイル内で Id 列を手動で定義する必要があります。
- ActionScript クラスファイルの生成対象となる CFC は、ColdFusion Web ルート内に存在している必要があります。
- ORM CFC の場合、datatype/type は cfproperty タグのオプション属性です。この属性が CFC に存在していない場合、ActionScript クラスファイルは生成されません。ActionScript クラスファイルが生成されるようにするには、ColdFusion データソースを指定します。これを指定すると、データ型情報がデータベーステーブルから取得されて、ActionScript クラスファイルが作成されます。