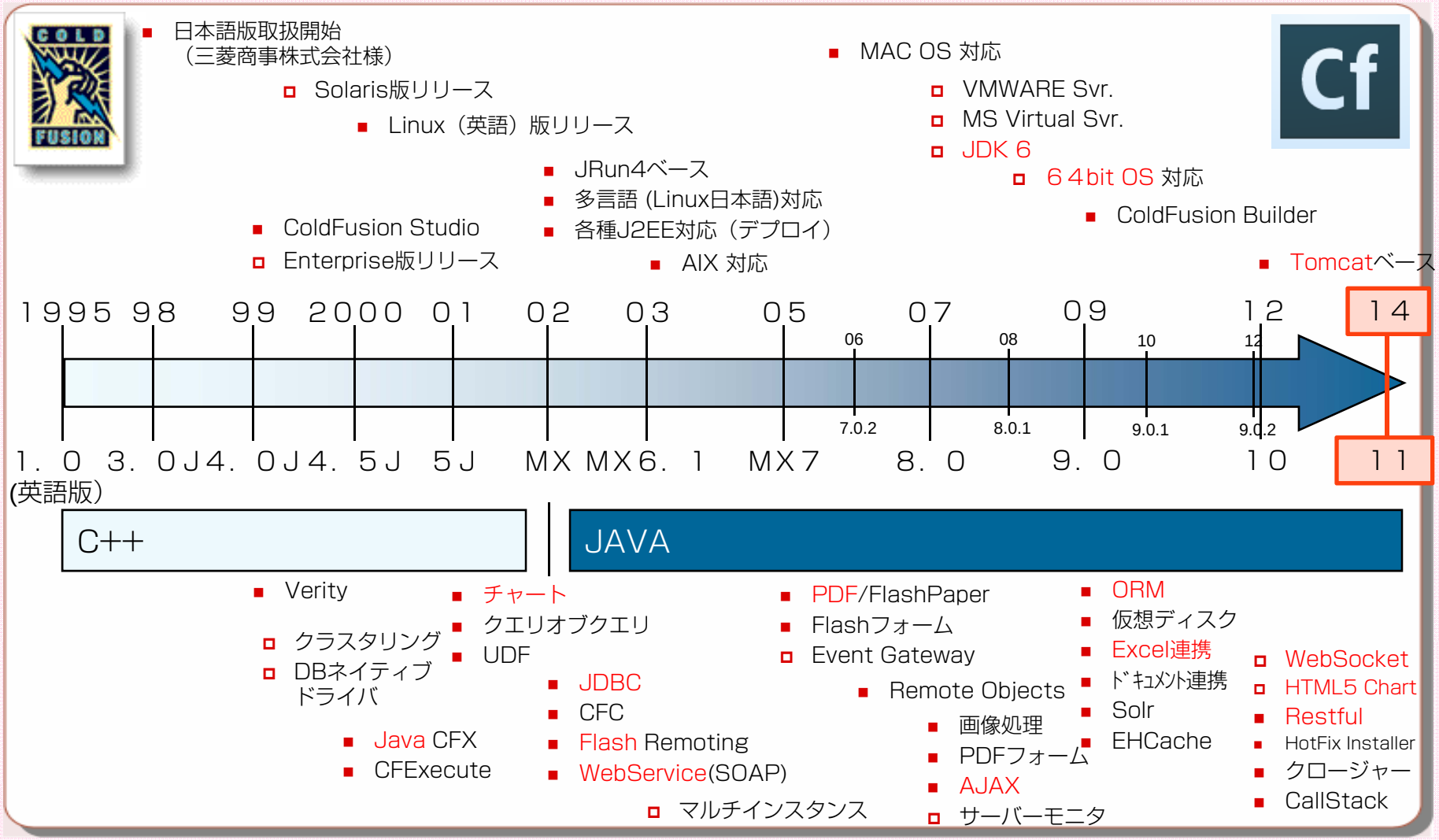
Adobe ColdFusion11
最新機能のご紹介

2014.5

株式会社サムライズ

Webのトレンドを取り入れ20年近い進化



Enterprise版のみ
一部バージョンに限るものを含む

ColdFusion 11とは？

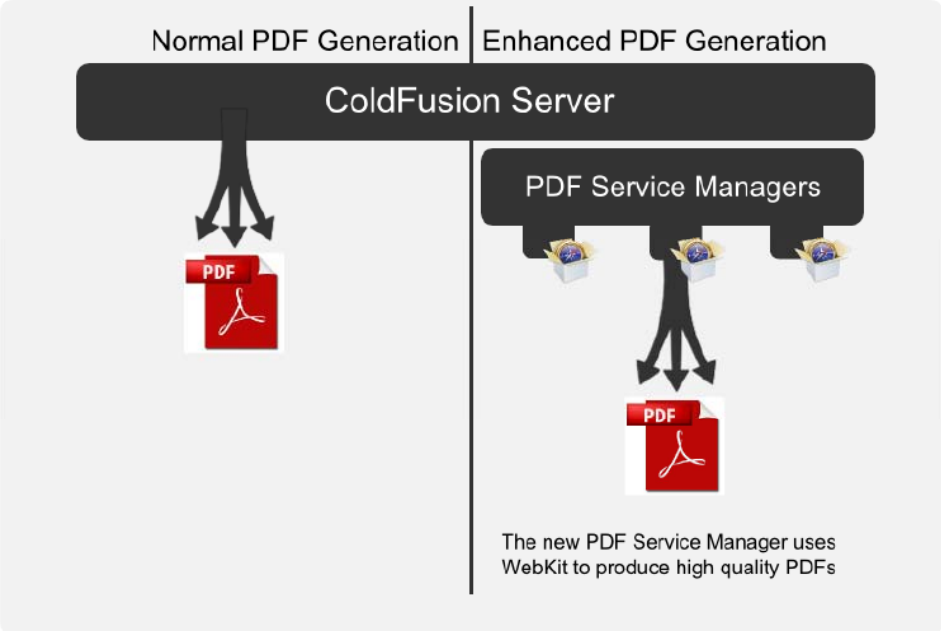
- 米国時間 2014年4月29日にリリースされたメジャーバージョンアップ
  - Apache Tomcat を内部エンジンとして利用する Javaベースのアプリケーション
  - International English版と日本語版が提供されている
- 専用エディタのColdFusion Builder3も同時リリース
- ColdFusion 11のポイント
  - PDF生成＆操作
  - モバイルアプリケーション機能
  - JSON機能強化
  - プログラミング強化
  - バンドルされたライブラリの強化
  - 他

PDF生成&操作

ColdFusion 11
新しいPDF生成エンジン

- 従来（CF MX7～）
  - IceBrowser & iText
    - <CFDocument>
    - <CFDocumentitem>
    - <CFDocumentsection>

- 新しいPDF生成エンジン
  - Webkit エンジンを利用した高クオリティのHTML→PDF変換
    - <CFHtmlToPDF>
    - <CFHtmlToPDFItem>



※PDF Service Managerのクラスタ化はEnterprise版のみ

ColdFusion 11
PDFファイルの操作

- PDFアーカイブ
  - PDFファイルをPDF/Aに変換
    - Preserve PDF for long time as a self contained document.
    - <CFPDF action="archive" source=".." destination="..">

- 電子署名(Enterprise版のみ)
  - PDFに電子署名を付与等の処理
    - <cfpdf action="sign" source="…" destination=".." keystore="cert.jks" keystorepassword="password" signaturefieldname="signField"/>
    - 追加されたアクション
      - SIGN, UNSIGN, VALIDATESIGNATURE, READSIGNATUREFIELDS

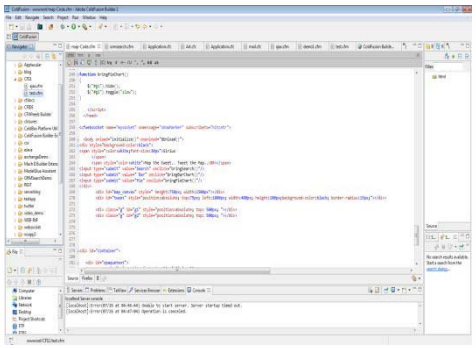
モバイルアプリケーション機能

Mobile & ColdFusion

ColdFusion 11
総合的なモバイルアプリケーション機能

1. 作成

<CFML> と JS を使用



2. ビルド & デプロイ

phoneGap Build を利用してiOSやAndroidアプリを作成

3. デバッグ

On Device Step デバッグ

4. テスト

Inspect & Debug across devices

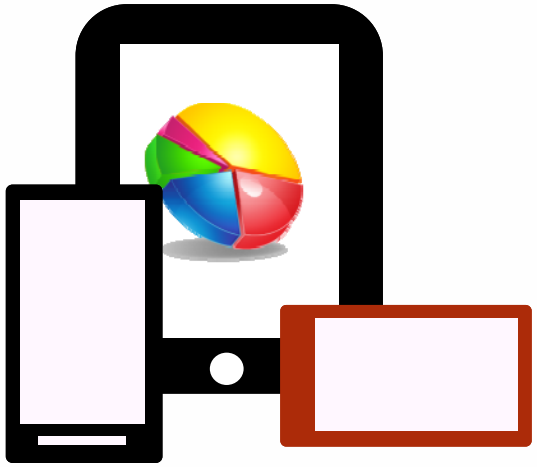
ColdFusion 11

CFML→JavaScript変換

- <cfclient>: CFML → Client Side
  - CFMLはコンパイル時にJavaScriptに変換
  - CFSET, CFOUTPUT, CFSCRIPT
  - CFInclude - CFM, JS & CSS
  - CFC(サーバーサイド、クライアントサイド)
  - CFML カスタムタグの記述形式に対応
- JavaScriptとの相互運用性
- お好みのJSフレームワークを使用可能

ColdFusion 11
モバイルアプリケーション化

- デバイス機能へのアクセス
  - ColdFusion 11 + ColdFusion Builder 3 と、PhoneGap Buildサービスを連携し、モバイルアプリケーションを生成
- モバイルアプリケーション化の利点
  - PhoneGapに用意されているデバイス機能に対応する処理を利用可能
    - デバイスのプロパティや特性を識別可能
    - カメラ、GPS、ファイルアクセス、オーディオ、サウンド等を使用するアプリケーションを作成可能



JSON機能強化

ColdFusion 11
構造体のキーの大文字小文字の維持

```
<cfscript>
  stTest=StructNew();
  stTest.aaa="aaa";
  stTest["bbb"]="bbb";
</cfscript>
```

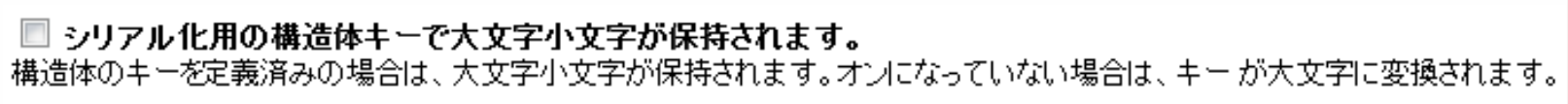
- ColdFusion 10以前では、ドット表記のキー名は大文字に変換される： {"AAA":"aaa", "bbb":"bbb"}
- ColdFusion 11では、ドット表記のキーの大文字小文字の維持が可能： {"aaa":"aaa", "bbb":"bbb"}

ColdFusion 11
構造体のキーの大文字小文字の維持（続き）

- 構造体のキーの大文字小文字の維持の有効・無効
  - アプリケーションレベル
    - Application.cfc

```
<cfscript>
  this.name="cfdemo_cf11";
  this.serialization.preserveCaseForStructKey=true;
</cfscript>
```

  - サーバーレベル
    - ColdFusion Administrator の[設定]



ColdFusion 11
クエリオブジェクトのJSON変換の強化

- クエリオブジェクトに対するserializeJSON変換方法を強化
  - struct

    serializeJSON(objToSerialize [, queryFormat] [,boolean secure])
    - QueryFormat : " true ¦ false " または " column ¦ row ¦ struct"

  - サンプル

query

	COLOR	ID
1	red	1
2	green	2
3	blue	3

```
//Serialized Output as row
{"COLUMNS":["ID","COLOR"],"DATA":[["1","red"],["2","green"],["3","blue"]]}

//Serialized Output as column
{"ROWCOUNT":3,"COLUMNS":["ID","COLOR"],"DATA":{"ID":["1","2","3"],"COLOR":["red","green","blue"]}}

//Serialized Output as struct
[{"ID":"1","COLOR":"red"},{"ID":"2","COLOR":"green"},{"ID":"3","COLOR":"blue"}]
```

ColdFusion 11
serializeJSON関数の機能強化（続き）

- queryFormatをアプリケーションレベルで指定
  - Application.cfc

```
component
{
    this.name="implicitMetaDataDemo";
    this.serialization.serializeQueryAs = "struct | column | row";
}
```

ColdFusion 11
その他のシリアライズ強化

- ４つのシリアライゼーション関数
  - XML シリアライゼーション関数
    - *serializeXML(objToSerialize, useCustomSerializer)*
    - *deserializeXML(stringToDeserialize, useCustomSerializer)*
  - XML, JSON, カスタム用シリアライゼーション関数
    - *serialize(objToSerialize, type, useCustomSerializer)*
    - *deserialize(strToDeserialize, type, useCustomSerializer)*
- 複合型のカスタムシリアライザ/デシリアライザ
  - Application.cfcでカスタム･シリアライザ/デシリアライザのパスを指定

    this.cutomSerializer = {pathToSerializerCFC};
  - シリアライザCFCには、下記のメソッドの指定が必要:

    boolean canSerialize(type);

    boolean canDeSerialize(type);

    string serializeData(objToBeSerialized, type);

    Object deserializeData(strToBeDeserialized, type);

ColdFusion 11
ソーシャル機能

- OAuth 2 認証プロバイダーを使ったログイン認証
  - Google, Facebook の認証プロバイダーに対応

```
<cflogin>
  <cfoauth type="Facebook" clientid="1509307672624465"
    secretkey="d1085efa4499b811a9d04294654d6e64"
    result="res"
    redirecturi = "http://xxxxxxx/cfdemo/1405/cfoauth.cfm" >
  <cfloginuser name = "#res.other.first_name#" password =
   "#res.access_token#" roles = "fblogin"/>
</cflogin>
<cflocation url="http://xxxxxxx/cfdemo/1405/next.cfm" >
```

  - ソーシャルの強化

    「いいね！」ボタン、ツイートボタン、Facebook のコメントボックス
    Google+ ボタン、Facebook の購読ボタン、「いいね！」ボックス
    アクティビティフィード、フォロー

プログラミング強化

ColdFusion 11
ハイブリッドコーディング

- 一般的なスクリプトのシンタックスに合わせた記述に対応

```
<cfscript>
  親（ベース）タグ(属性1=値1, 属性2=値2 …)
  {
     サブタグ (属性1=値1,属性2=値2 …)
     {
          サブタグ …
     }
  }
</cfscript>
```

- <CFHTTP>,<CFHTTPPARAM>

```
<cfscript>
 cfhttp(method="post",url="http://www.samuraiz.co.jp/", charset="MS932")
  {
     cfhttpparam(name="username", type="FormField", value= "aaa");
     cfhttpparam(name="password", type="FormField", value= "bbb");
   }
</cfscript>
```

ColdFusion 11
メンバー関数のサポート

- ネイティブデータ構造/オブジェクトに対するオブジェクト指向型メソッド呼び出し。
  - ArrayLen(arr)に相当：arr.len()
- Array
- Struct
- String
- List
- Date
- Query
- Image
- Spreadsheet
- XML

ColdFusion 11
メンバー関数のサポート（続き）

- これまでは関数としては、ColdFusionで以前から定められてる方法だけだった
  - ArrayAppend(arrayObj, objToAppend)
- メンバー関数の導入により、オブジェクト指向モデルと整合するコーディング記述が行える
  - arrayObj.append(objToAppend);
- シンタックス
  - obj.MemberFunction([argument(s)])

ColdFusion 11
メンバー関数のサポート＜例＞

■記述例

```
<cfscript>
  aFruit=ArrayNew(1);
  ArrayAppend(aFruit,"リンゴ");
  ArrayAppend(aFruit,"みかん");
  ArrayAppend(aFruit,"メロン");
  writeOutput(ArrayLen(aFruit));

  //新しい記述方法
  aFruit.add("パパイヤ"); // Java API
  aFruit.append("キウイ"); // CF API
  writeOutput(aFruit.Len());
</cfscript>
```

ColdFusion 11
メンバー関数のサポート（補足）

- リスト関数はStringオブジェクトとして利用できる

- 関数は、両方のデータ型で存在する

List	String
listFind	find
listFindNoCase	findNoCase
listLen	len

- 文字列関数は、リテラル文字列は呼び出すことはできない
  - "xyz".len()　//は動作しない

ColdFusion 11
QueryExecute関数

- QueryExecute
  - QueryExecute(クエリ文 [,クエリパラメーター] [,クエリオプション])

```
<cfscript>
  qEmp1=QueryExecute("SELECT * FROM
  Employee","",{datasource="cfdocexamples"});
  qEmp2=QueryExecute("SELECT * FROM Employee where
  DEPT_ID=1","",{datasource="cfdocexamples"});
  qEmp3=QueryExecute("SELECT * FROM Employee where
  DEPT_ID=?",[1],{datasource="cfdocexamples"});
  qEmp4=QueryExecute("SELECT * FROM Employee where
  DEPT_ID=? AND EMP_ID=?",[1,1],{datasource="cfdocexamples"});
  qEmp5=QueryExecute("SELECT * FROM Employee where
  DEPT_ID=:deptid AND EMP_ID=:empid",
  {deptid:{type:"integer",value:1}, empid:7},
  {datasource="cfdocexamples"});
</cfscript>
```

ColdFusion 11
QueryGetRow関数

- QueryExecute
  - 特定のクエリ行を取得
    - QueryGetRow(queryObj, rowIndex)
  - クエリオブジェクトに対しては下記の方法でも同様に対応
    - row = queryObj.getRow(rowIndex)

```
<cfscript>
  qEmp=QueryExecute("SELECT * FROM
  Employee","",{datasource="cfdocexamples"});
  stEmpData=QueryGetRow(qEmp,1);
  writeDump(stEmpData);
</cfscript>
```

struct	
CONTRACT	Y
CONTRACT_FILE	/opt/coldfusionmx/wwwroot/vw_files/contracts/frueh.txt
DEPT_ID	1
EMP_ID	1
FIRSTNAME	Benerwrwerrw
LASTNAME	Frueh
PROJECT_DOCS	/opt/coldfusionmx/wwwroot/vw_files/frueh
SALARY	100000
STARTDATE	{ts '1987-01-19 00:00:00'}

ColdFusion 11
エルビス演算子

- 今までの記述例

```
If(IsDefined("userName")){
  displayName=userName;
} else {
  displayName="未ログイン";
};
```

- エルビス演算子での記述例
  - x = a ? a : b　→　x = a ?: b

```
displayName = userName ?: "未ログイン";
```

ColdFusion 11
ZIPファイルのパスワードサポート

- CFZIPタグで、パスワード付きのファイルをサポート
  - 追加された属性
    - Password
    - encryptionAlgorithm
  - 暗号化アルゴリズム
    - Standard (zip2.0)
    - AES-128
    - AES-256 (default)
  - <cfzip action="zip" encryptionAlgorithm="AES-128¦AES-256¦Standard" password = "passwd" … >
  - <cfzip action="unzip" password="passwd" …>

ColdFusion 11
CFLOGIN, セキュリティ関連

- <cflogin>
  - MX～9 同じユーザー名での複数同時ログイン可
  - 10 同じユーザー名での複数同時ログイン不可
  - 11 allowconcurrent="true ¦ false" 設定可（Administrator設定有）
- ColdFusion Administrator でセキュアプロファイルの有効/無効切り替え
  - 10 インストール時に有効/無効を切り替え
- Adminコンポーネントに対するIP アドレス制限の追加
  - MX～9 IPアドレス制限なし
  - 10 Administrator に対するIPアドレス制限を追加
  - IPアドレス制限に adminapi, componentexplorer を追加

ColdFusion 11
ライブラリのバージョンアップ

- ColdFusionに同梱された各種ライブラリのバージョンアップ
  - Ajax(JavaScript)フレームワーク：Ext JS
    - ColdFusion 8 : 1.0 / 1.1
    - ColdFusion 9 : 3.0 / 3.1
    - ColdFusion 10 : 3.1
    - ColdFusion 11 : 4.1(sencha)
  - Excel操作：POI
    - ColdFusion 9・10 : 3.6
    - ColdFusion 11 : 3.9
  - ORM：Hibernate
    - ColdFusion 11 : 4.1.10

※独自に Ajax のフレームワーク機能を使用していた場合などは
ColdFusion11移行時にマイグレーションが必要

ColdFusion 11
サーバーサイドチャートの変更

- ColdFusion MX6～
  - WebCharts3D
- ColdFusion 10
  - サーバーチャート：WebCharts3D（変更なし）
  - クライアントチャート（Enterpriseのみ）：ZingChart
- ColdFusion 11
  - サーバー＆クライアントチャート：ZingChartに統一

※グラフの見た目が変わるため、旧バージョンで使用している場合は見た目の調整が必要となる

お問い合わせ先

株式会社サムライズ

アドビソフトウェア事業部 ColdFusion ビジネスユニット

E-mail: adobe_software@samuraiz.co.jp

http://www.samuraiz.co.jp/

※サムライズのホームページでColdFusion情報を公開中

http://www.samuraiz.co.jp/coldfusion

(ColdFusion カフェテリア) http://forum.samuraiz.co.jp

(ColdFusion Associate) http://cfassociates.samuraiz.co.jp