

webサービス ~セールスフォース・ドット・コム

By Jonathan Le, Technical Support, 4D, Inc.
Technical Note 05-03

(原題: 4D and salesforce.com)

概要

salesforceソリューションは、パッケージソフトウェアの代わりにカスタマイズ可能なAPIを提供しているユニークなソリューションである。デベロッパは自由に機能を追加することができ、自らが開発したアプリケーションにsalesforceを融合させることができる。核となっているのは、プラットフォームや言語に依存しないsalesforceのWebサービスAPIである。

salesforceにはwebサイト<http://www.salesforce.com>からアクセスすることができ、ブラウザの操作によって売上管理、顧客管理などの業務ができるようになっている。マーケティング担当はここでキャンペーンを展開し、経営者はすべてを把握できるという仕組みである。

The screenshot displays the Salesforce Developer Edition interface. At the top, the browser address bar shows the URL <https://na1.salesforce.com/home/home.jsp>. The page header features the Salesforce logo and navigation tabs for 'ホーム', 'キャンペーン', 'リード', '取引先', '取引先担当者', '商談', '売上予測', '契約', 'ケース', 'ソリューション', '商品', 'レポート', 'ドキュメント', and 'ダッシュボード'. Below the header, there is a navigation bar with links for '新規キャンペーン', '新規リード', '新規取引先', '新規取引先担当者', '新規商談', '新規売上予測', '新規契約', '新規ケース', '新規ソリューション', '新規ドキュメント', '新規ToDo', '新規商品', and 'ごみ箱'. The main content area is divided into several sections: a user profile for 'miyako keisuke (4d-japan)' with a 'Winter '05 新機能ご紹介' button; a 'Calendar' section for April 2005 showing the current date as 2005/04/19; a 'ToDo' section with a table of tasks; and a 'Site Search' section. The 'ToDo' table is as follows:

完了	期限	件名	取引先担当者/リード	関連先
X		Download your IDE toolkit		
X		Visit the sforce Developer Forum		

The footer of the page contains the text: Copyright © 2000-2005 salesforce.com, inc. All rights reserved.

4Dとsalesforce

4Dにはwebサービスが実装されているので、.NET、Java、Visual Studioなどで開発されたアプリケーションとまったく同じようにsalesforceにアクセスすることができる。バージョン2004ではwebサービス関連のコマンドが強化された結果、追加プラグインを使用しなくても様々なエンコードに対応できるようになった。

webサービスとは？

簡単にいって、サーバにリクエストを送信し、結果を受け取るという処理をするための技術で、一般にSOAPやXML-Restのようなプロトコルを使用する。オープンな規格であるため、ランゲージやプラットフォームを問わないのが特徴である。

4Dにはwebサービスウィザードが実装されているため、アプリケーションにwebサービスを統合することは非常に簡単である。SOAP、WSDL、XMLを使用して、salesforceともデータを共有することができる。

SOAP (Simple object access protocol)

SOAPとは、ソフトウェア同士でオブジェクトをやり取りするためのプロトコルのことで、salesforce.comは主にこの方法で通信を行なっている。

HTTP

SOAPはどのようなネットワークレイヤーでも使用することができるが、通常はHTTP上で使用される。

WSDL (Web Services Description Language)

WSDLとは、公開されているwebサービスの情報、たとえばデータタイプやサービスの種類などの説明を記述するための言語である。

バインド方法

webサービスにはRPC、DOCという2種類のバインドタイプがあるが、salesforceはDOCタイプを採用している。

salesforce API

salesforceでは、オブジェクト指向と呼ばれるデザインをとられており、すべてのものがオブジェクトとして管理されている。この場合、オブジェクトにはデータがどのように管理されているかの情報のみが納められており、データストラクチャに相当する。実際のデータが納められているのはインスタンスである。4Dの用語に置き換えるならば、オブジェクト=テーブル、インスタンス=フィールドと考えても構わない。

カスタムオブジェクト

ユーザはカスタムオブジェクトを定義することができ、それには自由にデータを納めることができる。

APIコール

salesforceのAPIには、次のような基本操作が用意されている。

メソッド	説明
create	オブジェクトを追加する
delete	オブジェクトを削除する
describeGlobal	オブジェクトリストを取得する
describeObject	オブジェクトのメタデータ（フィールドリスト/オブジェクトプロパティ）を取得する
getDeleted	ある時刻以降に削除されたオブジェクトのIDリストを取得する
getServerTimestamp	システム時刻（グリニッジ標準時刻）を取得する
getUpdated	ある時刻以降に更新されたオブジェクトのIDリストを取得する
getUserInfo	カレントセッションのユーザ情報を取得する
login	クライアントセッションを開始する
query	特定のオブジェクトに対してクエリを実行して結果を取得する
queryMore	クエリ結果の続きを取得する
resetPassword	新たなパスワードをサーバに発行させる
retrieve	オブジェクトを検索する
search	テキスト検索を実行する
setPassword	パスワードを設定する
update	オブジェクトを更新する

salesforceと4Dを融合する

4D 2004の新機能を活用すれば、従来のようにInternet Commandsプラグインを使用しなくても簡単にsalesforceと4Dを融合することが可能である。

webサービスウィザード

webサービスウィザードによって、プロキシメソッドと呼ばれる"webサービスをコールするためのメソッド"が自動的に作成される。ウィザードは、インターネット上で公開されているWSDL、あるいはローカルにファイルとして存在するWSDLを解析し、サービスの内容や必要な引数のタイプを特定し、次のようなコードを出力する。

```
$root:=DOM Create XML Ref("create";$namespace)

$subelem:=DOM Create XML element($root;"" )
DOM SET XML ELEMENT VALUE($subelem;$1)

SET WEB SERVICE PARAMETER("XMLIn";$root)
```

CALL WEB

```
SERVICE("http://aspen.salesforce.com/services/Soap/u/4.0";"";"create";"urn:subject.partner.soap.sforce.com";Web Service Manual )
```

Webサービスコマンド

4Dでは主に次のwebサービスコマンドを使用する。

```
SET WEB SERVICES PARAMETER
SOAPリクエストに渡すパラメータを定義する
CALL WEB SERVICE
URLで特定したwebサービスを呼び出す
SET WEB SERVICE OPTION
SOAPにヘッダのような補足情報を追加する
GET WEB SERVICE RESULT
webサービスコールに対するSOAPレスポンスを取得する
```

webサービスの結果は次のようなコードで取得する。

```
If (OK=1)
C_BLOB($blob)
C_STRING(16;$resroot)
C_STRING(16;$ressubelem)
GET WEB SERVICE RESULT($blob;"XMLOut";*)
$resroot:=DOM Parse XML variable($blob)

$ressubelem:=DOM Find XML element($resroot;""")
DOM GET XML ELEMENT VALUE($ressubelem;$0)
DOM CLOSE XML($resroot)
End if
```

XML解析

4D 2004にはXMLの読み書きに必要なコマンドが揃っているので、XML形式であるSOAPメッセージの処理に活用することができる。

- DOM Create XML Ref
- DOM Create XML element
- SOAPリクエストに挿入するXML要素の作成に使用できる
- DOM Parse XML Variable
- SOAPリクエストを解析し、必要な情報を取得するために使用できる
- DOM Find XML element
- XMLに含まれる属性を取得するために使用できる

webサービス実践編

実際にsalesforceのwebサービスを利用してみることにする。まずはAPIメソッド "login"でsalesforce.comにログインする。パラメータとしてユーザ名とパスワードを送ると、次の3アイテムが返される。

サーバURL
セッションID
ユーザID

ログイン以外のSOAPリクエストには、セッションIDが求められる。(2時間有効)

ログインSOAPリクエストの本体は次のような内容になる。

```
POST https://www.salesforce.com/services/Soap/u/4.0 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client
Protocol 1.1.4322.573)
Content-Type: text/xml; charset=utf-8
SOAPAction: ""Content-Length: 364
Expect: 100-continue
Proxy-Connection: Keep-Alive
Host: www.salesforce.com

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<soap:Body>
<login xmlns="urn:partner.soap.sforce.com">
<username>user@domain.com</username>
<password>
secret
</password>
</login>
</soap:Body>
</soap:Envelope>
```

このうち<soap:Body>と</soap:Body>の挟まれた部分以外は、4Dが自動的に出力するので、実際に作成する必要があるのは次の部分だけである。

```
<login xmlns="urn:partner.soap.sforce.com">
<username>
user@domain.com
</username>
<password>
secret
</password>
</login>
```

この部分も、プロキシメソッドによっておおかた設定されているので、それを必要に応じて修正するだけでよい。たとえば、デモDB"salesforce Toolkit"のメソッド"proxy_login"には、次のように記述されている。

```
$username:=$1
$pass:=$2

$namespace:="urn:partner.soap.sforce.com"
$login:=DOM Create XML Ref("login";$namespace)

$subelem:=DOM Create XML element($login;"username")
DOM SET XML ELEMENT VALUE($subelem;$username)

$subelem:=DOM Create XML element($login;"password")
DOM SET XML ELEMENT VALUE($subelem;$pass)

SET WEB SERVICE PARAMETER("XMLIn";$login)

ON ERR CALL("SF_ErrorHandler")
CALL WEB
SERVICE("https://www.salesforce.com/services/Soap/u/4.0";"";"login";"urn:
partner.soap.sforce.com"; Web Service Manual )
ON ERR CALL("")
```

このメソッドは<login>部分を設定し、引数\$1、\$2を使用して<username><password>部分を作成する。作成されたXMLはメモリに置かれ、コマンドSET WEB SERVICE PARAMETERおよびCALL WEB SERVICEが実行されるときにSOAPリクエストの<soapenv:body>部分に挿入される。

結果として送られるSOAPレスポンスは次のようなものである。

```
HTTP/1.1 200 OK
Server: Resin/3.0.s040331
Content-Type: text/xml; charset=utf-8
Connection: close
Transfer-Encoding: chunked
Date: Wed, 07 Jul 2004 16:26:26 GMT

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<loginResponse xmlns="urn:partner.soap.sforce.com">
<result>
<serverUrl>
https://na1.salesforce.com/services/Soap/u/4.0
</serverUrl>
<sessionId>
MkogrC4M2WUuwaKjLUIPmLaKyTTif
</sessionId>
<userId>
005300000000vgHMAAY
</userId>
</result>
</loginResponse>
</soapenv:Body>
</soapenv:Envelope>
```

この場合も、4Dは自動的に<soapenv:Body>と</soapenv:Body>の間の部分だけを取り出してBLOBに納めるので (GET WEB SERVICE RESULT) 、XMLコマンドを使用して内容を解析すればよい (DOM Find XML element、DOM GET XML ELEMENT VALUE) 。

```
C_BLOB($response)
C_STRING(16;$response;$result;$child)

GET WEB SERVICE RESULT($resultBlob;"XMLOut";*)

$response:=DOM Parse XML variable($resultBlob)
$result:=DOM Get First XML element($response)

` here we set the process variable serverUrl
$child:=DOM Find XML element($result;"result/serverUrl")
DOM GET XML ELEMENT VALUE($child;serverURL)

` session ID we need for updating session headers in future web service
calls
$child:=DOM Find XML element($result;"/result/sessionId")
DOM GET XML ELEMENT VALUE($child;sessionId)

` the user id (of user we logged in under)
$child:=DOM Find XML element($result;"/result/userId")
DOM GET XML ELEMENT VALUE($child;userId)

DOM CLOSE XML($response)
```

最後にコマンドDOM CLOSE XMLでメモリを解放する。

sfrc Toolkit

サンプルDBには、salesforceのwebサービスを4Dで使用するための様々なツールが用意されている。APIに対応したプロキシメソッドを使用して、login、create、update、delete、< queryなどのwebサービスがコールできる他、salesforceのオブジェクトを4Dのテーブルに変換するツールも提供されている。

salesforceと融合するアプリケーションを短時間で開発し、既存のアプリケーションにsalesforceを追加できるよう、テンプレート、コンポーネント、オブジェクトライブラリ、ソースコードも添付した。

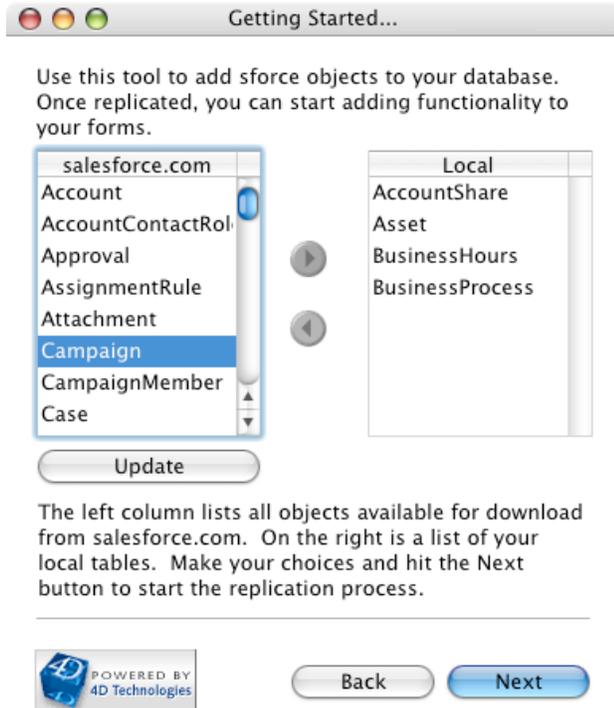
(.4DB、RSRファイルを置き換えて使用。PluginsフォルダにはMacOS/Windows適切な4D packが必要。)

はじめにセールスフォースのホームページ (<http://www.salesforce.com/>) でログインIDを取得してから、サンプルDBを起動する。

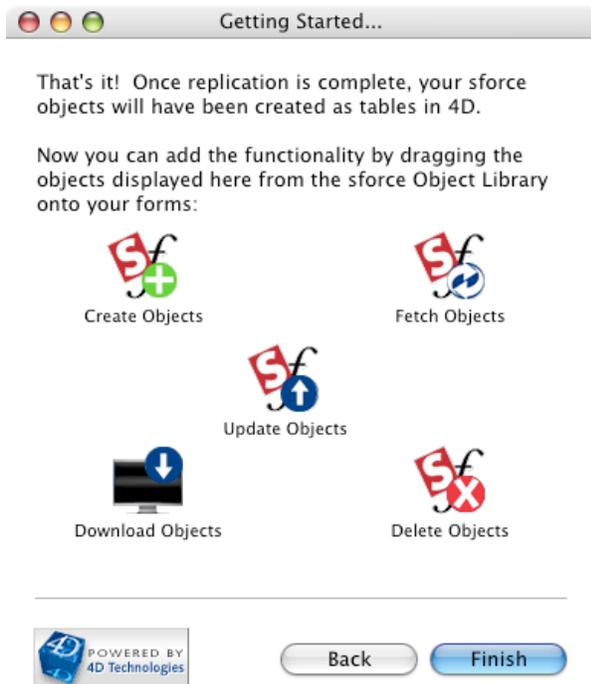
ログインに成功すると右の画面が表示される。
(アカウント作成後、webアクセスが有効になってからwebサービスが有効になるまで、若干の時間がかかる場合がある。)

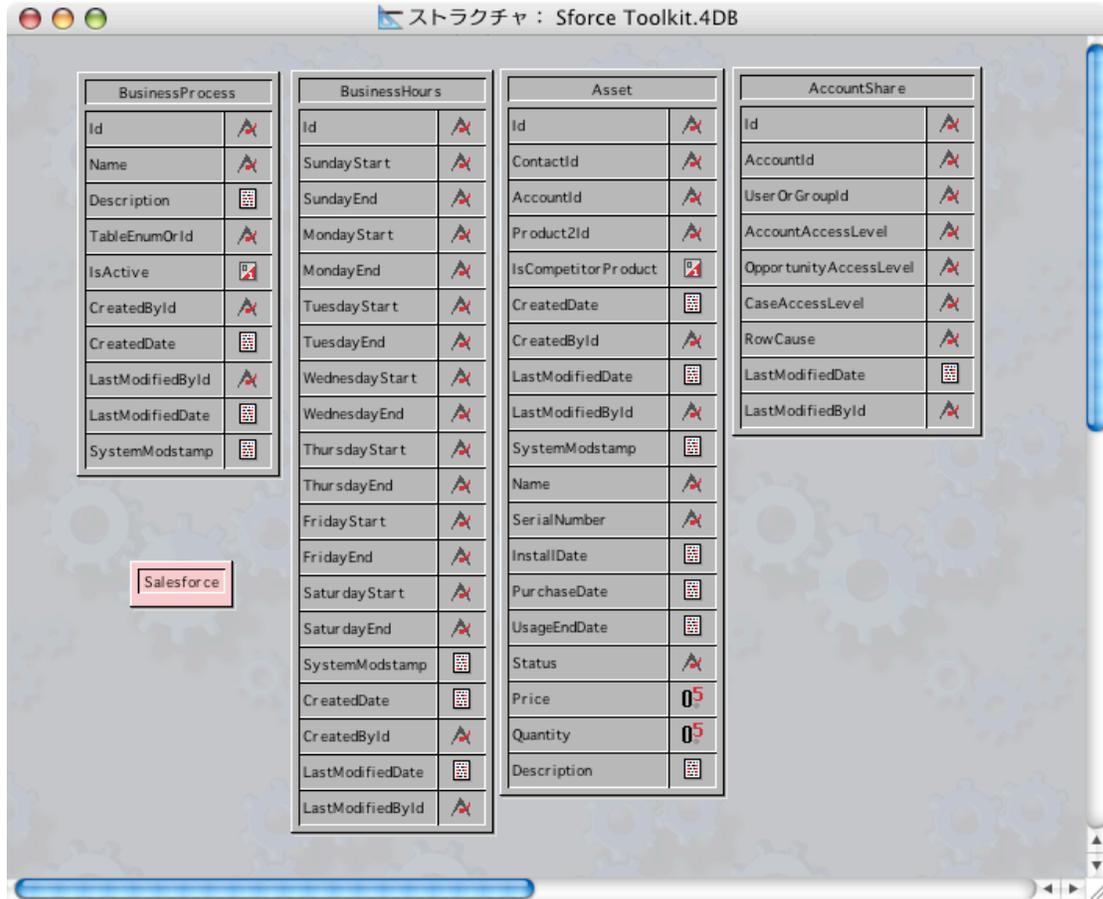


ウィザードを使用して、salesforceのデータを4Dに複製することができる。再現された4Dのテーブル/フィールドとsalesforceオブジェクトのバインド情報は、外部XMLファイル（Preferencesフォルダ）に記録される。

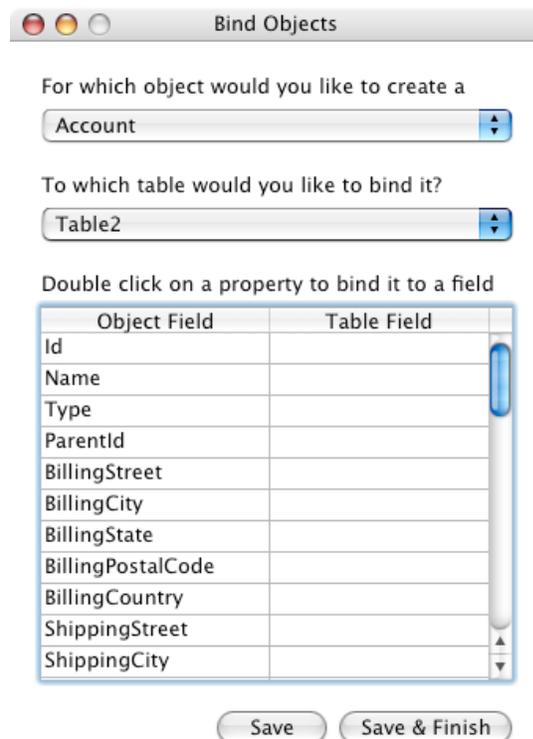


通信に成功する、右の画面が表示され、カスタムモードが終了する。デザインモードで新規テーブル（salesforceのミラー）が作成されたことを確認する。テーブルは当初、折り重なった状態で最小化されている。

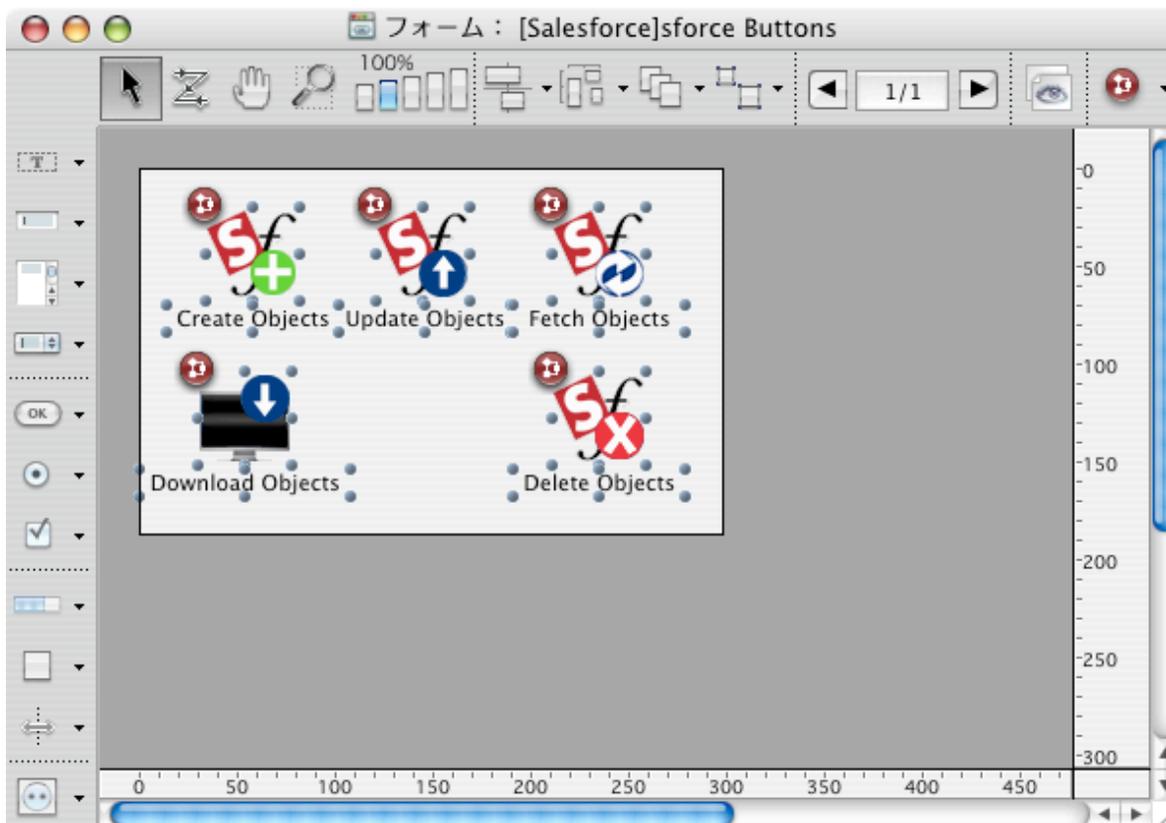




新規ストラクチャを構築するのではなく、既存の4Dストラクチャとsalesforceをバインドさせたいときには、コンポーネントのBind Toolを使用してマッピングする。



salesforceとバインドされた4Dストラクチャが作成できたら、オブジェクトライブラリのボタンを使用して、salesforceアクションをフォームに追加することができる。



オブジェクトライブラリには、salesforceアクションを実行することのできるオブジェクトが揃っており、ドラッグ&ドロップによってフォームに追加することができる。オブジェクトメソッドでは特定のコマンドが使用されているので、コンポーネントをインストールするか、4D Packプラグインが利用できる状態にしておく必要がある。



コンポーネントを使用する場合は、salesforceのオブジェクトと4Dのテーブルが正しくバインドされている必要がある。さらに、メソッドはIdフィールドを参照しているので、たとえばsalesforceのカスタムオブジェクトのIdフィールドは、同名の4DテーブルのフィールドIdにバインドされている必要がある。

以上の条件さえ満たされていれば、ドラッグ&ドロップによってフォームに追加されたオブジェクトはすぐにユーザ/カスタムモードで使用することができる。