

Skype API Book Vol.1

Japanese Skype Developers Forum API Book Project 監修

岩田真一、r 埜 /xai、池嶋 俊、大谷弘喜、須崎雅道、寺田 亮、谷萩毅之、山本達也 共著



Skype、SkypeIn、SkypeOut、Skype Me は、Skype Limited の登録商標です。Microsoft、Visual Basic、Visual C++、Visual C#、Visual Studio、Win32、Windows は、Microsoft Corporation の登録商標です。その他、本書に記載されている商品名・社名等は、一般に各企業の商標または登録商標です。なお、本文中では、™・©・® を明記していません。

まえがき

私が Skype Technologies 社へ入社した 2005 年 8 月当時、日本語で得られるまともな Skype API の情報はほぼ皆無でした。

Skype API 開発者支援を使命としていた私がまず実行したのは、日本国内で英語の資料を頼りに細々と Skype API プログラミングを続けられているの方々のためのコミュニケーションの場、Japanese Skype Developers Forum を開設することでした。同フォーラムは、入社してから 14 日後にオープンすることができ、現在では 30 名近い開発者の方々に賑わっています。

日本人ソフトウェアエンジニアの優れた能力を、英語という壁によって埋もれさせたくない、という思いが常にあり、エンジニアの方々には英語の資料を読むこと、また英語で発信することを促しながらも、日本語での迅速な情報、議論の場の提供を心がけてきました。

既存の Skype API 開発者支援に一区切りついたとき、次にやらなければならないことは明白でした。これから SKYPE API を使おうとするプログラマの方々の支援です。Skype は支援してくださる開発者によって支えられています。つまり、その開発者が増えれば増えるほど、Skype にとっても、そして Skype のユーザーにとってもよりよい状態となるわけです。

そんな思いから本書はスタートしました。本書には、Japanese Skype Developers Forum のメンバーによって蓄積された、マニュアルには載っていない Tips をふんだんに盛り込みました。また、新しい API を扱うときにありがちな、「最初の壁」をいかに越えやすくするかということにも重点を置き、API の呼び出しから Skype への接続部分など「ごく基本的であるが、ここを越えないとその先に広がるすばらしい Skype API の世界が見えない」という壁を突破するためのヒントがたっぷりつまっています。

Japanese Skype Developers Forum の優秀なプログラマ自らが、自分たちの体験を元に「こんな本があれば、Skype API の初心者もつまづかないだろう」という共通の目的のために書き下ろした渾身のトピック集です。Skype API の入門から、その応用までをカバーしています。

また、本書の企画そのものも非常にオープンなものにしました。企画の発表から、執筆者の募集、企画会議の議事録までをすべて Japanese Skype Developers Forum 内で行い、公開しています。まさに、開発者たちのための開発者たちによる書籍です。本書のコンセプトは、

フォーラムにたまった Tips を書籍化するというものであるため、今後も Tips がたまりしだい、vol.2、vol.3 と続けていくつもりです。

本書のもう 1 つの特徴は、サポートしている OS と言語の種類の数です（下表参照）。まだ世界を見渡しても、Skype API の使い方についてここまで詳細に、多言語、多プラットフォームに渡って解説されたものはありません。このような書籍が、まず日本語で世に出ることを非常に嬉しく思っています（Windows では、それぞれ COM ラッパーを使った場合、使わない場合に分けて解説しています）。

	Windows	GNU/Linux	Mac OS X
C/C++	○	○	-
Visual Basic	○	-	-
C#	○	-	-
Python	○	○	-
Ruby	○	-	-
Java	○	-	-
ObjectiveC	-	-	○
AppleScript	-	-	○

※本書は Skype API の入門から応用までを取り扱っていますが、プログラミング言語の入門書ではありませんので、各プラットフォーム、言語の知識があることを前提としています。

本書を読んで Skype API を学習し、Skype 連携アプリケーションを開発しようとされている方は、もうつまらないところで壁にぶつかることはありません。存分にアイデアを形にすることに集中してください。

本書の出版にあたり、企画からずっと共に議論を続け、執筆してきた Japanese Skype Developers Forum の執筆者の方々に心から敬意と感謝の意を表します。本当にありがとうございました。フォーラムを立ち上げた当時は、このような書籍を出版できるとは夢にも思っていませんでした。皆様のおかげです。また、編集を担当していただいた鈴木さんにも心から感謝いたします。この書籍の意義を理解していただき、世に出したいという思いが重ならなければ、実現は不可能でした。ありがとうございました。

本書によって、1 人でも多くのプログラマ、開発者の方が Skype API の楽しさに触れ、そして 1 つでも多くの Skype 連携ソフトウェアが生まれることを祈りつつ、まえがきといたします。

Japanese Skype Developers Forum メンバーを代表して
Skype Technologies S.A.

日本マーケット開発マネージャ 兼 Developer Relations
岩田 真一

目次

まえがき	3
第 1 章 Skype API 概要	11
1-1 Skype と Skype Public API	11
1-2 Skype API を使ったプログラミング	15
1-3 A2A - ネットワーク上のインフラへ	16
1-4 Extras Gallery	18
1-5 Skype API 利用ガイドライン	19
1-6 Japanese Skype Developers Forum	21
1-7 Skype エコシステム	22
1-8 Skype 認定プログラム	23
第 2 章 さまざまな開発環境	27
2-1 ActiveS とは	27
2-2 紹介する開発環境の概要	28
2-3 例題	30
第 3 章 Windows Visual C++	31
3-1 概要	31
3-2 開発環境の準備	31
3-3 例題のコーディングと実行	36
3-4 ソースコードの解説	42
3-5 sample ソースコード	44
第 4 章 Windows Visual C++/ActiveS	51
4-1 概要	51
4-2 プログラム作成	53

4-3	実行	57
4-4	Sample ソースコード	58
4-5	参考文献	59
第 5 章	Windows Visual Basic	61
5-1	Visual Basic 2005 Express Edition のダウンロードとインストール	61
5-2	プログラム作成	62
5-3	ソースコード記述	64
5-4	プログラムの流れ	70
5-5	ビルド	70
5-6	実行	71
5-7	Form1.vb ソースコード	72
第 6 章	Windows Visual Basic/ActiveS	79
6-1	まずは準備	79
6-2	プログラム作成	79
6-3	ソースコード記述	82
6-4	プログラムの流れ	84
6-5	ビルド	84
6-6	実行	84
6-7	Form1.vb ソースコード	86
第 7 章	Windows Visual C#/ActiveS	89
7-1	サンプルプログラムについて	89
7-2	Visual C# 2005 Express Edition のダウンロードとインストール	89
7-3	Visual C# 2005 Express Edition の起動	90
7-4	プロジェクトの新規作成	90
7-5	ActiveS の参照の追加	91
7-6	サンプルコード解説	92
7-7	ビルド	93
7-8	実行	93
7-9	Program.cs ソースコード全文	95
第 8 章	Windows Python	97
8-1	インストール	97
8-2	プログラムの作成	98

8-3	実行結果	99
8-4	skypeapibook ソースコード	100
第 9 章	Windows Python/ActiveS	101
9-1	インストール	101
9-2	プログラム作成	101
9-3	実行結果	102
9-4	pythonactives.py ソースコードの全文	103
第 10 章	Windows Ruby/ActiveS	105
10-1	Ruby のダウンロードとインストール	105
10-2	実験	105
10-3	サンプルソースコード	106
10-4	まとめ	107
第 11 章	Windows Java/ActiveS	109
11-1	ソフトウェアのダウンロードとインストール	109
11-2	プログラム解説	111
11-3	実行	112
11-4	サンプルソースコード	113
第 12 章	Mac OS X ObjectiveC	115
12-1	はじめに	115
12-2	Xcode で開発するときの注意事項	115
12-3	まずは準備	116
12-4	プログラム作成	117
12-5	ソース解説	123
12-6	ビルド / 実行	124
12-7	サンプルプログラムの入手と利用方法	126
12-8	ソースコード全文	128
第 13 章	Mac OS X AppleScript	131
13-1	AppleScript とは	131
13-2	スクリプトの記述	131
13-3	スクリプト解説	132
13-4	スクリプト全文	133
第 14 章	Linux C	135
14-1	dbus-0.23.x のインストール	135
14-2	DBUS システムバスの準備	136

14-3	ソースコード解説	136
14-4	ビルドと実行	138
14-5	サンプルソースコード全文	139
第 15 章	Linux Python	143
15-1	ソフトウェアのインストール	143
15-2	プログラム解説	144
15-3	実行結果	146
15-4	ソースコード全文	146
第 16 章	Skype の音声処理	149
16-1	電話としての Skype	149
16-2	サンプルについて	150
16-3	サンプルの準備	152
16-4	DirectSound による再生・録音	153
16-5	ミキサーを使用した留守番電話	168
16-6	ミキサークラスの作成	172
16-7	オーディオミキサーを使用した留守番電話の実装	193
16-8	VAC (Virtual Audio Cable) を使用した留守番電話	201
第 17 章	A2A	223
17-1	はじめに	223
17-2	API 解説	225
17-3	テスト方法	230
17-4	サンプルアプリケーション SkypePad	231
17-5	接続形態	233
17-6	SkypePad ソースコード	233
第 18 章	SKSweets For Mac	253
18-1	機能解説	253
18-2	プログラム解説	258
18-3	ビルド・実行について	265
18-4	SKSweets プログラムソース全文	265
18-5	SKSweets のダウンロード	278
第 19 章	Skype Web Gateway	279
19-1	はじめに	279
19-2	プログラムの動作	280
19-3	プログラムの流れ	282

19-4	Main.cpp ソースコード全文	289
19-5	Web サービスを使用する場合の注意	304
	サンプルソースコードの入手とサポートについて	305
	サンプルソースコードのダウンロード	305
	サポートについて	305
	免責について	306
	著者紹介	307
	索引	309

第 1 章 Skype API 概要

Skype Technologies S.A. 岩田真一

1-1 Skype と Skype Public API

1-1-1 Skype

本書を手にとった方には、もはや説明不要かもしれないが Skype について一度振り返っておこう。Skype は 2006 年 5 月現在、全世界で約 1 億人の登録ユーザーを持つインターネット・コミュニケーションツールである。Skype を開発、提供する Skype Technologies 社は 2003 年 8 月にルクセンブルグに設立されたばかりのベンチャー企業である。その特徴はなんといっても、P2P テクノロジーを利用した高音質通話を無料で楽しめることにある。さらにソフトウェアの導入が非常に簡単で、特別なネットワーク設定が不要だったため、Skype 社が特に積極的なマーケティングを行わなかったにもかかわらず、口コミで全世界に広まった。簡単・無料・高音質で一気に普及したのだ。

また Skype は Skype クライアントに電話番号を入力することで、一般公衆回線に通話を発信することもできる。これは「SkypeOut (スカイプアウト)」というサービスで、プリペイド方式になっている。またその逆に Skype の ID に電話番号を結びつけることで、一般公衆回線から Skype への着信も可能にする「SkypeIn (スカイプイン)」というサービスもある。これらの 2 つの有料サービスが Skype 社の主な収入源である。

もちろん Skype が備えているのは単なる音声通話だけではない。「コミュニケーションツール」と謳っているように、最大 5 人までの会議通話に加え、Windows 版の Skype 2.0 では 1 対 1 のビデオ通話 (テレビ電話) が搭載された。その他にもテキストチャット、グループチャット (最大 50 人まで)、ファイル転送など、およそコミュニケーションに必要となる機能はすべて備えている。

Skype の対応プラットフォームは次の通りだ。

- Windows (2000、XP)
- GNU/Linux
- Mac OS X
- Windows Mobile for PocketPC (2003 以降)

将来的には、ここに Symbian や Windows Mobile for Smartphone などが加わる予定である。これらのプラットフォームの中で Skype Public API がサポートされているのは

- Windows
- GNU/Linux
- Mac OS X

である (対応 API のバージョンによる違いはあるが)。

1-1-2 Skype Public API とは

このような機能の豊富さと音声通話のクオリティの高さに加え、もう 1 つ Skype の魅力をあげるとするならば、それはまちがいない、ごく初期から公開されている Skype Public API (以下 Skype API) であろう。この API の存在が、多くのデベロッパーからも Skype が評価されている理由であり、そして Skype をプラットフォームたらしめている要因である。

Skype API の役目を一言でいうと、他のプログラムから Skype の機能、そして Skype のネットワークを使うための仕組みだ。Skype 社では、この Skype API を早くから公開することにより (2004 年 11 月)、Skype をより便利に、より楽しく、さまざまな用途で使ってもらえるようなサードパーティ製ツールの登場を促進してきた。Skype 社はコアとなる最大公約数的なアプリケーション (Skype) を開発提供する (これは同時に Skype のネットワークを提供することにもなる)。そしてあらゆる方面からの要求にすべて応える代わりに、だれでも好きな機能を追加できるようにするための API を用意しているというわけだ。

たとえば、Skype を企業で使うための製品が欲しいとしよう。これを Skype 社自身がヨーロッパの習慣に合わせてビジネス用 Skype を提供したらどうだろうか。それをそのまま日本に持ち込んだ場合、他の多くの外資系ビジネスソフトウェアと同様、きっと日本では細かい文化の違いで受け入れられないものになる可能性が高い。そこで、Skype は代わりに Skype をビジネス対応させるのに必要な API を提供する。そして各地域の開発者 (この場合は Sier やソフトハウスになるだろうか) が、その地域の商習慣に合わせた連携製品を作って提供するので。これが Skype の取った立場である。このとき、その製品および製品の保守サポートが有償で

あれば、各地域に独自のビジネスが発生する。そう、Skype API はビジネスチャンスをも生み出している。これは、後述する「Skype エコシステム」の根幹をなす部分である。

そのようなビジネスチャンスの話は置いておくとしても、現在の Skype と Skype をとりまくムーブメントは Skype API なくしては語れない。たとえば Skype に録音機能を付け加える製品や、不在時に自動応答してくれるソフトウェア、などが API 公開後すぐに登場している。また、Skype が Windows 版のバージョン 2.0 でビデオ通話をサポートする以前に、Skype と連携してビデオ通話を実現するようなソフトウェアがいくつも発表された。このように世界中の開発者が Skype ユーザーに対して（あるいは自分が開発者だった場合は単に自分のために）追加機能を提供することが可能なのだ。

Skype 社は API の提供、普及、利用を非常に重視している。筆者の肩書き「Developer Relations」は、Skype API を多くの方々へ紹介し、その可能性を啓発していく役割を現している。開発者の方には Skype API を使って Skype と連携する有益で楽しいソフトウェアをどんどん開発（そしてビジネス）していただく、そして Skype 社はそれらのバリエーション豊かな連携製品が揃うことで、多くのユーザーを満足させることができる、まさに Win-Win の状態が生まれる。どこにも無理がないし、開発者は世界中にいるためスケラブルだ。無理がないので理論上、今後も提供が続けられるのだ。

1-1-3 入手方法

Skype の API を別途入手する必要はない。Skype をインストールすれば自動的に API が使えるようになる（逆に、Skype API なしで Skype をインストールすることもできない）。入手する必要があるのは、Skype のバージョンに合った API のドキュメント、仕様書だけだ。

さらに、Skype の API を利用するときには何かライブラリのようなものをリンクする必要もない。現在プログラミングができる環境があるのなら、何かを新しくインストールする必要すらない。これは Skype API の特徴で、後述の「1-2 Skype API を使ったプログラミング」でその理由がわかるだろう。

ドキュメント類一式は、以下のサイトから入手可能だ。

[Skype Developer Zone (英語)]

<https://developer.skype.com/>

1-1-4 Skype API の分類

Skype API はその役割によって、3つに分類することができる。それが、

第 2 章 さまざまな開発環境

Skype API はテキストベースのメッセージ交換方式を採用しているので、アプリケーションの作成は、一般に入手できるさまざまな開発言語・開発環境を自由に選んで行える。

また、プログラミングを容易にするコンポーネントの類も提供されているので、それを活用して Skype API 利用の面倒な部分や、開発言語によっては苦手とする部分を隠蔽しながら簡単に目的とするアプリケーションを作成することができる。

ここでは、OS、言語、コンポーネントごとのプログラミング環境の紹介と共に、それぞれの環境で Skype API を使用する同じ処理の記述方法を具体的に説明するので、自分に合った開発方法を習得できるだろう。

2-1 ActiveS とは

ActiveS とは、KhaosLabs (<http://www.khaoslabs.com/>) が無償で公開している Microsoft Windows の COM インターフェイスを用いて、Skype API を利用するためのライブラリである。

COM インターフェイスでやり取りするため、Windows 上のいろいろなプログラミング言語から、それを利用することができる。

まず、Skype 社から提供されている Skype API は、Windows で一番レガシーな通信方法 (Window Messaging) を用いている。

ただし、この通信方法だと、Skype API を利用するアプリケーションすべてが、それぞれ個別にウィンドウメッセージを扱う処理を記述しなくてはならなくなってしまう。これが、Skype API を用いたアプリケーション開発の敷居を高くしていた。

この問題を解決し、開発者が簡単に Skype API を用いたアプリケーションを開発できるように Window Messaging 処理部分をライブラリ内に隠蔽して、それらを COM インターフェイスで扱えるようにしたのが ActiveS なのである。この ActiveS を利用することにより、コード記述量がどのくらい少なくなるかは、第 3 章以降の Windows 各言語のプログラムを見ていた

できれば、一目瞭然である。

ActiveS の Web サイトは以下の通りである。

<http://www.khaoslabs.com/actives.php>

上記 Web サイトから Microsoft Installer 形式のインストールキットをダウンロードして、クリックすることで簡単にインストールできる。

インストールの途中で ActiveS を使用できるユーザーについての問い合わせがあるが、システム全体で使用できるように指定することをお勧めする。

また、Window Messaging を他 (Winsock など) との通信のために使用しなくてはならないようなプログラムで、ActiveS を利用する際には注意が必要になる。ただし、実は ActiveS のインストールキットには、それ自身のソースコードが添付されている。ActiveS が内部でどのように Skype と通信しているのか等、それを見れば明らかである。

2-2 紹介する開発環境の概要

本書では、以下の OS、言語、コンポーネントにおけるプログラミング例を紹介し、それぞれの環境における特徴を示す。

Windows	C/C++	Skype API 使用の基本技法となる。Skype と連携するアプリケーションにおけるプログラミング上の構成要素を理解できるので、他の開発環境を使用する方も一通り目を通すことをお勧めする。
	C++/ActiveS	Windows 9x/NT/2000 との互換性を考慮して、Win32 ネイティブベースの開発を行うために VC++ を使用する開発者は多いと思う。そのような開発者のために、.NET Framework を必要としない Win32 ネイティブベースでの ActiveS の使用を試みる。
	Visual Basic Visual Basic/ ActiveS	Quickly & Easy Programming 環境として、日曜プログラマなどに親しまれてきた Visual Basic。Win32 API を呼び出すテクニックやノウハウも長年にわたり蓄積されている。C や Win32 API とのデータのやり取りについて、旧来はテクニックを要したが、近年では Marshial クラスでの型変換で大分、見通しのよいものとなってきている。Visual Basic 未経験の方も、一度、垣間みてほしい開発環境である。

	C#/ActiveS	C、C++、Java、Visual Basic のよいところを集めた、Microsoft の開発した言語である。 .NET ベースのアプリケーションを作成するうえで C# を利用する開発者向けに、ActiveS を利用した Skype 連携の方法を解説する。
	Python	海外では人気のあるスクリプト言語である。 インターラクティブに操作できるので、API の確認が簡単にでき、プロトタイプの実成にも便利である。
	Python/ActiveS	
	Ruby/ActiveS	国産でオブジェクト指向スクリプト言語である Ruby は、理解しやすく、また、プログラミングの入門にも向いている。 今回は、Ruby を対話的に使用し、ActiveS を使うスクリプトを開発している例を示す。
	Java/ActiveS	その剛健性のため、システム開発でよく使われている Java 言語。 今回は、Java から COM を使う JCOM を用いて、ActiveS 経由で Skype を操作する例を紹介する。
Linux	C/C++	Skype API を Linux で使用する際の基本技法となる。 Linux、特に DBUS でアプリケーション間通信を行う上での基礎を垣間みることができる。 必見だ。
	Python	Python を使用することで、DBUS の扱いが簡潔になる。 また、インターラクティブに操作を確認しながら開発が進められる。
Mac OS X	ObjectiveC	Skype API を Mac OS X で使用する際の基本技法となる。 ObjectiveC は C に比べて少々記述方法やメモリ管理方法が異なるが、Apple が OS に標準で提供している開発環境でもあり、Mac OS X で開発する方は避けて通ることはできないだろう。 UI 構築に関しては、Next Step からの一貫したポリシーがあり、それを体験するだけでも価値があると思う。
	AppleScript	古くから Macintosh 標準搭載の AppleScript。 今回は、AppleScript から Skype に Tel するスクリプトを紹介しよう。

2-3 例題

Skype API を利用したプログラミングの例題として、Skype の利用者が Skype に登録しているコンタクトの一覧を取得してみる。この目的を実現するプログラミング上の一連の処理を、それぞれの開発環境ごとに紹介する。

コンタクトのリストを取得するには、Skype と接続した後に以下のコマンドを送信する。

```
SEARCH FRIENDS
```

このコマンドを送信すると、Skype は以下の形式で処理結果（コンタクトのリスト）をクライアントに送信する。

```
USERS (ログインしているユーザーのコンタクトのリスト。スペースで区切られた文字列として返信される)
```

例：ログイン中のユーザーが、"yamada" "tanaka" "sato" の3人のコンタクトを登録していた場合

```
(Skype へコマンドを送信) --> SEARCH FRIENDS
```

```
(Skype から処理結果を受信) <-- USERS yamada tanaka sato
```

一般的には、取得した文字列（ここでは、"USERS yamada tanaka sato"）を解析して、画面に表示したり、取得した情報を元に再度 Skype にコマンドを送信して補足的な情報を取得したりすることになるだろう（例：コンタクト ID ごとに、性別、ステータスを取得するなど）。

プログラム例では、以下の処理をそれぞれの環境に応じた形で紹介するので、プログラミング方法を比べてみて欲しい。

- Skype と通信するための準備
- Skype への接続
- リストを取得するコマンドの送信
- 結果の受信と表示

第 3 章 Windows Visual C++

この章では、Visual C++ のみを使用したサンプルプログラムの作成方法について解説する。

- 使用言語：Visual C++ 2005 Express Edition
- ライブラリ：Microsoft Platform SDK

3-1 概要

ここでは、Windows 環境で、他にコンポーネント類を使用せず C/C++ のみで Skype API を利用する方法を、例題の作成を通して学ぶことにする。

C/C++ で作成する方法は、Skype と通信するアプリケーションを作成する上での基本手法であり、Skype API を使用するプログラミングに関係する要素がほぼ明確になるので、他の開発環境を使用する方も一通り目を通しておくとよいだろう。

以下の内容を順に解説する。

- 開発環境の準備
- 例題のコーディングと実行
- 解説
- 全ソースコード

3-2 開発環境の準備

Microsoft は、Windows 上の開発ツールである Visual Studio シリーズのうち、Express Edition を無償で提供している。ここでは、この開発ツールのなかで C/C++ を対象とする Visual C++ 2005 Express Edition を利用することとし、サンプルアプリケーションを作成す

るための環境を準備する方法を説明する（以下、Visual C++ 2005 Express Edition を、Visual C++ 2005、または VC と表現する場合がある）。

以下の手順で環境を構築する。

1. Visual C++ 2005 Express Edition のダウンロードとインストール
2. Microsoft Platform SDK のダウンロードとインストール
(Visual C++ 2005 Express Edition は、.NET Framework アプリケーションを構築するための開発ツールであるため、単体では Win32 アプリケーションを作成することができない。Win32 アプリケーションを作成するには、Microsoft Platform SDK を追加してインストールする必要がある。)
3. Visual C++ 2005 Express Edition の環境設定

3-2-1 Visual C++ 2005 Express Edition のダウンロードとインストール

はじめに、C/C++ による .NET アプリケーションの開発環境である、Visual C++ 2005 Express Edition をインストールする。

1. Microsoft のサイト (<http://www.microsoft.com/japan/msdn/vstudio/express/visualc/>、2006 年 5 月 21 日現在) をブラウザで開く。
表示されたページの「CD イメージのダウンロードおよびインストール方法」を選択し、指示に従って "Visual C++ 2005 Express Edition" を選択して CD イメージ形式のファイル (vc.iso) をダウンロードする。
2. サイトに記載されている説明にしたがって、ダウンロードしたプログラムをインストールする。
(ダウンロードされるファイルは、ISO イメージである。この ISO イメージから CD を作成して CD からインストールする。あるいは ISO イメージからファイルを展開し、setup.exe を実行してインストールする。いずれの方法も、Microsoft のサイトで詳しく説明されているので事前に確認すること。)

3-2-2 Microsoft Platform SDK のダウンロードとインストール

続いて、Visual C++ 2005 で Win32 アプリケーションを作成する場合に必要な Microsoft Platform SDK をインストールする。

SDK は、先にインストールした Visual C++ 2005 を実行すると表示されるウィンドウで、SDK に関する項目を選択することで入手できる。

1. Windows のスタートメニューから、[すべてのプログラム] - [Visual C++ 2005 Express Edition] - [Microsoft Visual C++ 2005 Express Edition] を選択すると VC が起動する。
2. [スタートページ] タブの、[作業の開始] ボックスから、「PSDK のダウンロード」を選択する (図 3-1)。



図 3-1 Visual C++ 2005 [スタートページ]

「Microsoft Document Explorer」ウィンドウが開き、該当ページが表示される。

3. 「Step2」の、「Install the Microsoft Platform SDK」に表示されているリンク、「Platform SDK」をクリックする (図 3-2)。

第 4 章 Windows Visual C++/ActiveS

この章では、Visual C++ で ActiveS を使用したサンプルプログラムの作成方法について解説する。

- 使用言語 : Visual C++ 2005 Express Edition
- ライブラリ : ActiveS COM Wrapper Ver1.02.03

4-1 概要

Visual Studio 2005 では .NET Framework を使用した開発が主であるが、Visual C++ 2005 においては Windows 9x/NT/2000 との互換性を考慮して Win32 ネイティブベースの開発を行う開発者も多いと思う (Skype は Windows 9x/NT では動作しないため、実際には Windows 2000 との互換性を重視した場合ということになるだろう)。

本章ではそのような開発者のために、.NET Framework を必要としない Win32 ネイティブベースでの ActiveS の使用を試みる。

.NET Framework ベースで ActiveS を使用する場合は、Visual C# の場合とほぼ同様であるため、そちらを参考にしてほしい。

4-1-1 Visual C++ 2005 Express Edition における Win32 の扱いについて

Visual Studio 2005 では、Win32 ネイティブベースでの開発は上位エディションに位置づけられ、Visual C++ 2005 Express Edition では MFC も ATL もサポートされていない。プロジェクトの新規作成ウィザードにおいても、Win32 ネイティブの Windows フォームアプリケーションの開発は行えず、唯一 Win32 コンソールアプリケーションのみが作成可能である。

MFC や ATL を使用するためには Visual Studio 2005 Standard Edition 以上が必要であり、

本格的な Win32 ネイティブベースの開発を行う場合は、本来ならそちらを使用すべきである。しかし、学習や小規模なツールの作成の場合、Express Edition において Win32 コンソールアプリケーションとして開発を行うことも有効であろう。

4-1-2 Visual C++ のネイティブ COM サポート

今回は、Win32 コンソール アプリケーションとしてサンプルを作成する。ActiveS の呼び出しについては、MFC および ATL を使用せず、Visual C++ のネイティブ COM サポートにより実装する。

この方法は MFC や ATL を使用せずに済むだけでなく、COM クライアントの実装としてサンプルで柔軟なものである。実装方法の詳細については、サンプルコードの解説で行う。

4-1-3 Microsoft Platform SDK の導入

Visual C++ のネイティブ COM サポートでは MFC も ATL も使用しないが、COM のベースとなる OLE 関連のリソース（ヘッダ、ライブラリ）は必要となる。Visual C++ 2005 Express Edition では OLE 関連のリソースも不足しているため、Microsoft Platform SDK を導入してリソースを補う必要がある。

Microsoft Platform SDK のインストールについては、第3章の「3-2-2 Microsoft Platform SDK のダウンロードとインストール」を参照してほしい。

TIPS

実は、Platform SDK には MFC や ATL のリソースも含まれている。しかし、パスを設定するだけでは Visual C++ 2005 Express Edition で MFC や ATL を使用したソースを正常にビルドできない。しかし、以下の Microsoft 社のサイトを参考にすれば、WTL (Windows Template Library) を使用することは可能である。

Using Visual C++ 2005 Express Edition with the Microsoft Platform SDK

<http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/default.aspx>

WTL は ATL の GUI 拡張といえるものなので、設定次第では Express Edition で ATL を使用することは可能であろう。MFC については、上位エディションを使用するしかなさそうである。

4-2 プログラム作成

今回作成するのは、Skype に登録しているコンタクトリストを簡単に出力するプログラムである。起動中かつログイン済みの Skype に接続し、コンタクトリストを参照して、リスト上の Skype 名をコンソール画面（標準出力）に出力する。

4-2-1 プロジェクトの作成

[プロジェクトの種類] で「Win32」、[Visual Studio にインストールされたテンプレート] で「Win32 コンソールアプリケーション」を選択して、新規プロジェクトを作成する。

例として、プロジェクト名は「Sample」とする。



図 4-1 Visual C++ 2005 新規プロジェクト作成画面

Win32 アプリケーションウィザードが起動する。通常、ここでは何も設定せずに [完了] としても構わないが、念のため、設定を確認しておこう。図 4-2 のように設定されていれば OK である。

第 5 章 Windows Visual Basic

この章では、Visual Basic で Skype API を使用した簡単なサンプルプログラムの作成方法について解説する。

- 使用言語：Visual Basic 2005 Express Edition
- OS：Windows XP Professional SP2

5-1 Visual Basic 2005 Express Edition のダウンロードとインストール

Microsoft 社のサイト(以下の URL)から Visual Basic 2005 Express Edition インストールキットの CD イメージをダウンロードする。

<http://www.microsoft.com/japan/msdn/vstudio/express/vbasic/>

CD イメージとは、CD 上のデータをそのまま書き出したファイルで、そのままでは使えない。そのイメージを CD-R に焼いて CD を作成するか、DAEMON Tools などの仮想 CD ソフトウェアでそのまま CD として仮想ドライブにマウントする必要がある。DAEMON Tools は、以下の URL からダウンロードすることができる。

<http://www.daemon-tools.cc/dtcc/download.php?mode=ViewCategory&catid=5>

Visual Basic 2005 Express Edition のインストール方法については、前出の Microsoft 社のページを参照していただきたい。

基本的には、ダウンロードした CD イメージをマウントするとインストーラが自動再生され

るので、そのダイアログに従ってインストールすればよい。環境にもよるが、ダウンロード、インストールともに予想外に時間がかかる。覚悟しておこう。また、1.5GBのディスク領域が必要なので、事前にディスクの空き容量を確認しておいたほうが無難だ。

インストールまで無事に終わったなら、Windowsのスタートメニューの[すべてのプログラム]に図5-1のようにVisual Basic 2005 Express Editionが登録されているはずだ。



図5-1 スタートメニューに登録された Visual Basic 2005 Express Edition

5-2 プログラム作成

今回作成するプログラムは、Skypeに登録しているコンタクトリストを出力するプログラムである。Visual Basicでは、フォーム上に貼り付けたラベル (Label1) にコンタクトリストを出力する。

プログラムの作成手順は以下の通り。

1. Visual Basic 2005 Express Edition の起動
2. プロジェクトの新規作成
3. フォームの作成
4. ソースコード記述
5. ビルド
6. 実行

以下、この手順に従って説明していく。

5-2-1 Visual Basic 2005 Express Edition の起動

WindowsのスタートメニューからMicrosoft Visual Basic 2005 Express Editionを起動する。起動すると、図5-2の画面が表示される。



図 5-2 Microsoft Visual Basic 2005 Express Edition スタートページ

5-2-2 プロジェクトの新規作成

[ファイル] - [新しいプロジェクト] を開き、テンプレートに「Windows アプリケーション」を指定して、新規プロジェクトを作成する。今回、プロジェクト名は「SkypeAPI_Sample」とした (図 5-3)。



図 5-3 プロジェクト新規作成時の設定画面

第 6 章 Windows Visual Basic/ActiveS

この章では、Visual Basic で Skype API を使用した簡単なサンプルプログラムの作成方法について解説する。

- 使用言語：Visual Basic 2005 Express Edition
- OS：Windows XP Professional SP2
- ライブラリ：ActiveS COM Wrapper

6-1 まずは準備

Visual Basic 2005 Express Edition のインストールについては第 5 章「5-1 Visual Basic 2005 Express Edition のダウンロードとインストール」を、ActiveS のインストールについては第 2 章「2-2 ActiveS とは」を参照していただきたい。

6-2 プログラム作成

今回作成するプログラムは、Skype に登録しているコンタクトリストを出力するプログラムである。Visual Basic では、フォーム上に貼り付けたラベル (Label1) にコンタクトリストを出力する。

プログラムの作成手順は以下の通り。

1. Visual Basic 2005 Express Edition の起動
2. プロジェクトの新規作成
3. フォームの作成
4. ActiveS 参照の追加

5. ソースコード記述
6. ビルド
7. 実行

以下、この手順に従って説明していく。

6-2-1 Visual Basic 2005 Express Edition の起動

Windows のスタートメニューから Microsoft Visual Basic 2005 Express Edition を起動する。



図 6-1 Microsoft Visual Basic 2005 Express Edition スタートページ

6-2-2 プロジェクトの新規作成

[ファイル] - [新しいプロジェクト] を開き、テンプレートに「Windows アプリケーション」を指定して、新規プロジェクトを作成する。今回、プロジェクト名は「SkypeAPI_Sample」とした。



図 6-2 プロジェクト新規作成時の設定画面

6-2-3 フォーム作成

新規作成した Windows アプリケーションプロジェクトには、Form1.vb がすでに作成されている。その Form1 上にツールボックスから「Label」を選択して、フォーム上にラベルを配置する（ツールボックスが表示されていない場合には、メニュー [表示] - [ツールボックス] を選択し、ツールボックスを表示して欲しい）。

このとき、そのラベル名は「Label1」とする（図 6-3）。



図 6-3 Form1.vb デザイン画面

第 7 章 Windows Visual C#/ActiveS

この章では、Visual C# で、Skype API を使用した簡単なサンプルプログラムの作成方法について解説する。

- 使用言語：Visual C# 2005 Express Edition
- OS：Windows XP Professional SP2
- ライブラリ：ActiveS COM Wrapper

7-1 サンプルプログラムについて

この章で作成するサンプルプログラムは、Skype に登録しているコンタクトリストを出力するものである。シンプルなプログラム構成にするため、コマンドプロンプトから実行するコンソールアプリケーションの形とした。

7-2 Visual C# 2005 Express Edition のダウンロードとインストール

Microsoft 社の Visual C# 2005 Express Edition を開発環境として利用する。Visual C# 2005 Express Edition は、以下の URL から CD イメージをダウンロードすることができる。

<http://www.microsoft.com/japan/msdn/vstudio/express/vcsharp/>

CD イメージファイル (.iso) としてパッケージ化されているので、インストールするには以下のようなステップが必要となる。

- 市販のCD 作成ソフトウェアを利用してCDを作成し、そのCDからインストールする。
- CD イメージファイルを展開できるソフトウェアを利用し、ハードディスクに展開してインストールする。
- CD イメージファイルを仮想CD ドライブソフトウェアでマウントし、インストールする。

インストールについての詳しい情報は、以下の URL の Microsoft 社の Visual Studio 2005 Express Edition のインストール方法のページを参照していただきたい。

<http://www.microsoft.com/japan/msdn/vstudio/express/maninstall/>

7-3 Visual C# 2005 Express Edition の起動

Windows のスタートメニューから、「Microsoft Visual C# 2005 Express Edition」を選択する。正しく起動できれば、図 7-1 のようなスタートページ画面が表示される。



図 7-1 Visual C# 2005 Express Edition のスタートページ

7-4 プロジェクトの新規作成

サンプルプログラムを作るために、まずプロジェクトを作成する。[ファイル] - [新しいプロジェクト] を選択し、テンプレートに「コンソールアプリケーション」を指定する。今回、

プロジェクト名は「SkypeAPI_Sample」とした（図 7-2）。



図 7-2 プロジェクトの作成画面

7-5 ActiveS の参照の追加

このサンプルコードにおいて ActiveS を使用するためには、ActiveS ライブラリを参照する必要がある。

7-5-1 参照の追加方法

メニューから [プロジェクト] - [参照の追加] を選択する。参照の追加ダイアログが表示されるので、「COM」タブを選択する。コンポーネント名「ActiveS 1.0 Type Library」を選択する（図 7-3）。



図 7-3 参照の追加ダイアログ

第 8 章 Windows Python

ここでは、Windows 環境で Python から Skype API を利用する方法を解説する。Python から Skype API を使用するためには、ActivePython 2.4 の他に追加のモジュールをインストールする必要がある。

- 使用言語：ActivePython 2.4
- OS：Windows XP
- ライブラリ：CTypes、Python Skype Wrapper

8-1 インストール

Python から Skype API を利用するためには、Python の実行形式をあらかじめインストールしておく必要がある。Windows での Python の実行形式ファイルはいくつか提供されているが、Skype API を使用するためには Win32 API や COM API があらかじめ組み込まれている ActiveState の ActivePython 2.4 を使用するのがよい。ActiveState のサイト (<http://www.activestate.com/>) から ActivePython 2.4 の最新版をダウンロードしてインストールする。

Skype のメッセージは、ウィンドウメッセージでやりとりされる。Windows のメッセージにセットされている構造体を扱うために、Python モジュールの CTypes をインストールする必要がある。CTypes は、<http://starship.python.net/crew/theller/ctypes/> で msi のインストーラとして配布されているので、Python 2.4 用のモジュールをダウンロードしてインストールする。

最後に Python の Skype Wrapper を <http://www.liris.org/program/skypewrapper.en/> からダウンロードして、zip ファイルを展開する。展開したディレクトリに移動して、コマンドラインから次のコマンドを実行してインストールする。

```
python setup.py install
```

8-2 プログラムの作成

Python の開発環境は、コマンドラインとエディタである。まず、作業用のディレクトリを作成する。ここでは、`c:\tmp\pyskype` ディレクトリを作成する。`c:\tmp\pyskype` に `skypeapibook.py` ファイルを作成して、テキストエディタで開く。

■ 1 行目：skypeserver モジュールのインポート

まず、Skype Python Wrapper の `skypeserver` モジュールをインポートする。

```
0001 import skypeserver
```

■ 5～7 行目：ハンドラクラスの作成

`skypeserver` のインポートが終了したら、Skype からのメッセージを処理するためのハンドラクラスを作成する。Skype のメッセージを `skypeserver` モジュールが受け取ると、各メッセージに対応するハンドラクラスの `do_` で始まるメソッドがコールされる。対応するメソッドが存在しない場合は、単にそのメッセージは無視される。Skype のコンタクトリストをハンドルするためには、`do_USERS` メソッドを実装する。2 番目の引数の `args` には、コンタクトのリストがセットされている。

```
0005 class myhandler:
0006     def do_USERS(self, *args):
0007         print "%n".join(args)
```

■ 9、10 行目：skypeserver の初期化とハンドラクラスの登録

次に `skypeserver` を初期化して、ハンドラクラスのオブジェクトを登録する。そして、Skype とメッセージの送受信を開始する。

```
0009 myskype = skypeserver.SkypeRawServer(myhandler())
0010 myskype.connect()
```

■ 2、18 行目：イベントループを回す

`SkypeRawServer` は、独自のイベントループを持たないため、アプリケーションの作成者が Windows のイベントループを回す必要がある。このため、`win32gui` モジュールをインポートして、Windows のイベントループを回すようにする。

```
0002 import win32gui
      :
0018 win32gui.PumpMessages()
```

第 9 章 Windows Python/ActiveS

この章では、Python から ActiveS を使用した簡単なサンプルプログラムの作成方法について解説する。

- 使用言語：ActivePython 2.4
- OS：Windows XP Professional
- ライブラリ：ActiveS COM Wrapper

9-1 インストール

ActivePython 2.4 のインストールについては第 8 章「8-1 インストール」を、ActiveS のインストールについては第 2 章「2-2 ActiveS とは」を参照していただきたい。

9-2 プログラム作成

今回作成するプログラムは、Skype に登録しているコンタクトリストをコンソールに出力するプログラムである。

Python の開発環境は、コマンドラインとエディタである。エディタは、自分の好みにあったエディタを使うのがよい。ActivePython には、「Pythonwin IDE」という優れた開発環境も付属するので、デバッグ環境を統合したい場合は、お勧めする。

まず、作業用のディレクトリを作成する。ここでは、`c:\tmp\pyskype` ディレクトリを作成する。`c:\tmp\pyskype` に `pythonactives.py` というファイルを作成して、テキストエディタで開く。

■ 1 行目：COM クライアント機能を有効にする

Python で COM を利用できるようにするためには、厳密にいうと（COM サーバを使う）

COM クライアントとしての機能を有効にする必要がある。そのためには、次の1行が必要である。

```
0001 import win32com.client
```

■ 2行目：SKYPEAPI.Access オブジェクトの取得

COM が使えるようになると、次に Skype オブジェクトを呼び出す。Skype API の COM コンポーネントは「Skypeapi」という名前である。大文字、小文字を区別するので注意すること。その中で、Skype と通信するメインのクラスは「Access」クラスである。以下のように記述する。

```
0002 oSkype = win32com.client.Dispatch("Skypeapi.Access")
```

■ 3行目：Skype との接続

これで、SKYPEAPI.Access オブジェクトを oSkype に取得できた。これからは、oSkype を通じて Skype とやり取りを行うことになる。実は、この時点では、まだ Skype との接続は確立していない。そこで、Access クラスの「Connect」というメソッドをコールして Skype と接続する。

```
0003 oSkype.connect()
```

■ 5行目：コンタクトリストの取得

Skype のコンタクトリストを取得する API は、GetFriendList() である。oSkype に対して GetFriendlist() をコールすると、コンタクトリストが取得できる。

```
0005 friends = oSkype.GetFriendList()
```

■ 6、7行目：ハンドル名の出力

取得した friends オブジェクトには、ユーザーのリストが格納されている。for ループで各ユーザーの情報を取得して、ハンドル名を出力する。

```
0006 for user in friends:  
0007     print user.Handle
```

9-3 実行結果

ソースコードが完成したら保存して、コンソール画面で c:\tmp\pyskype ディレクトリに移動して、pythonactives.py を実行する

```
c:¥> cd tmp¥pyskype  
c:¥tmp¥pyskype> python pythonactives.py
```

第 10 章 Windows Ruby/ActiveS

オブジェクト指向スクリプト言語である Ruby を用いて Skype API を使う方法を解説する。なお、ここでは、ターゲットを Windows とし、ActiveS を利用する。

10-1 Ruby のダウンロードとインストール

Ruby は、オブジェクト指向スクリプト言語である。Ruby は、まつもとゆきひろ氏によって開発された純国産言語で、本格的なオブジェクト指向開発を手軽に行うことが可能である。

Windows 環境上で動作する Ruby のバイナリは数多くあるが、ここでは、OneClick Ruby Installer を利用する。RubyForge のプロジェクトページ (<http://rubyinstaller.rubyforge.org/wiki/wiki.pl>) からインストーラをダウンロードし、インストールを行う。

今回は、Ruby 1.8.2 (ファイル名:ruby182-15.exe) を利用した。また、ActiveS も必要である。こちらは、第 2 章「2-2 ActiveS とは」に従ってインストールを行って欲しい。

10-2 実験

まず、インタラクティブな Ruby 環境である irb を用いて実験を行う。irb を使うことで、対話的に Ruby のコードを実行することができる。

[スタート] - [ファイル名を指定して実行] をクリックし、「irb」と入力することで、irb を起動できる。irb を起動すると、

```
irb(main):001:0>
```

のようなプロンプトが表示される。ここに Ruby のコードを入力するとすぐに実行される。

まず、ActiveS を使うために、Ruby-COM ブリッジである win32ole モジュールをロードする。次のように入力を行う。

```
require "win32ole"
```

これにより win32ole モジュールがロードされる。次に、ActiveS の初期化を行う。

```
skype = WIN32OLE.new("SKYPEAPI.Access")
```

これにより ActiveS が初期化され、skype という変数にバインディングされる。このオブジェクト skype に対して、ActiveS のメソッドを呼び出すことで Skype にアクセスできる。たとえば、次のようなコードを実行する。

```
skype.getfriendlist().item(1).Handle
```

ActiveS のマニュアルによると、skype.getfriendlist() の戻り値は UserCollection 型である。このコードでは、UserCollection に対して item(n) で n 番目の User オブジェクトを取得でき、さらにその User オブジェクトの Handle プロパティを取得している。

```

C:\Ruby\bin> irb
irb(main):001:0> require "win32ole"
=> true
irb(main):002:0> skype = WIN32OLE.new("SKYPEAPI.Access")
=> #<WIN32OLE:0x2e105-0>
irb(main):003:0> skype.getfriendlist().item(1).Handle
=> 11611_standard
irb(main):004:0>

```

図 10-1 irb によるコードの実行

10-3 サンプルソースコード

次に irb 上ではなく、単一のスクリプトを作成する。ここでは、コンタクトリストに登録されている Skype 名の一覧を取得するスクリプトを作成する。

次のコードをテキストエディタで作成し、getfriendlist.rb というファイル名で保存する。

第 11 章 Windows Java/ActiveS

プログラミング言語 Java による、Skype API を使用したアプリケーションの作り方を説明する。ここでは、Java-COM ブリッジである JCOM と Skype API COM ラッパーである ActiveS を利用する。

- 使用言語：Java (J2SE 1.5.0)
- OS：Windows XP
- ライブラリ：JCOM、ActiveS COM Wrapper

11-1 ソフトウェアのダウンロードとインストール

11-1-1 Java

Java を利用するためには、JRE または JDK をインストールする必要がある。ここでは、Java の開発が行える JDK をインストールする。JDK は、下記の Sun のサイトからダウンロードできる。

<http://java.sun.com/j2se/1.5.0/ja/download.html>

今回は、Java の最新版である Java1.5.0 Update6 (ファイル名：jdk-1_5_0_06-windows-i586-p.exe) を利用した。

インストーラが付属しているため、インストールは簡単である。ここでは、標準のフォルダ (C:\Program Files\Java\jdk1.5.0_06) にインストールしたとして話を進める。

11-1-2 JCOM

JCOM は、Java から COM オブジェクトを利用するためのライブラリで、渡辺義則氏の手によって作られた。

<http://www.nexb.org/open-source-it-asset-management/Wiki.jsp?page=Jcom>

ここでは、JCOM のプロジェクトページから最新版をダウンロードして利用する。

<http://sourceforge.net/projects/jcom>

今回は、JCOM-2.2.4（ファイル名：jcom224.zip）を使用する。

JCOM にはインストーラが付属していないため、自分でインストールを行う必要がある。ただし、インストールは非常に簡単で、パッケージ jcom224.zip に含まれる以下のファイルを下記のディレクトリに配置するだけである。

jcom.jar

<JDK インストールフォルダ>%jre%lib%ext

ここでは、C:%Program Files%Java%jdk1.5.0_06%jre%lib%ext

jcom.dll

<JDK インストールフォルダ>%jre%bin

ここでは、C:%Program Files%Java%jdk1.5.0_06%jre%bin

11-1-3 ActiveS

ActiveS は、Skype API を簡単に利用できるようにする COM ラッパーである。ActiveS のインストールに関しては第 2 章「2-2 ActiveS とは」を参考にしてほしい。

11-2 プログラム解説

では、サンプルプログラムについて解説しよう。ここでは、GetFriendList.java としてプログラムを保存している。

■ 1 行目：JCOM を使うための宣言

```
0001 import jp.ne.so_net.ga2.no_ji.jcom.*;
```

■ 5 行目：JCOM が COM を管理するためのメモリマネージャを作成

```
0005         ReleaseManager rm = new ReleaseManager();
```

■ 7 行目：ActiveS を初期化

```
0007         IDispatch app = new IDispatch(rm, "SKYPEAPI.Access");
```

■ 8 行目：コンタクトリスト一覧の UserCollection の取得

```
0008         IDispatch ret = (IDispatch)app.method("getfriendlist",
            new Object[]{});
```

■ 9 行目：UserCollection の分だけ for 文を繰り返す

```
0009         for (int i = 1; i <=
            ((Integer)ret.get("count")).intValue(); i++) {
```

■ 11 行目：i 番目を取り出す

```
0011             IUnknown kn = (IUnknown)ret.method("Item",
                new Object[]{new Integer(i)});
```

■ 12 行目：i 番目の型をプロパティ取得可能な IDispatch に変換する

```
0012             IDispatch user =
                (IDispatch)kn.queryInterface(
                    IDispatch.class, IDispatch.IID);
```

■ 14 行目：i 番目の人の Handle (Skype 名) を取り出し表示

```
0014                 System.out.println(user.get("Handle"));
```

第 12 章 Mac OS X ObjectiveC

12-1 はじめに

ここにきて、Mac OS X で Skype API 1.4 が使用できるようになった。この Skype API 1.4 には、A2A 関連、MOOD TEXT 関連の API などが含まれる。今まで、Windows の Skype ではできていたことが、だんだんと Mac でもできるようになってきたのだ。さあ、これからだ！ ぜひ、これを読んでいるあなたのアイデアで、Mac での Skype コミュニケーションに華を添えていただきたい。

Be happy with Mac and Skype!!

12-2 Xcode で開発するときの注意事項

Xcode は Apple から無料で、あなたに提供されている。とても喜ばしいことだ。ただし、Windows や Unix で Text ベースで開発することに慣れてしまっている開発者は十二分に注意しなくてはならない。

12-2-1 なぜなら

Xcode において、UI 関連の設定は GUI で設定することが多い。だが、確認する際には、また GUI 上でその設定を確認しなくてはならない。要するに、操作を誤って設定してしまった内容を、またその GUI で確認しなくてはならない。

他の環境であれば、テキスト検索などで、その存在を検索することもできるが、Xcode ではなかなかできない。画面が複雑化し、UI 周りの設定が多くなってきた際には、設定が多くなる分、注意が必要だ。

また、作成したプログラムを他に公開する場合には、それを Zero-Link してはいけない。筆者も SKSweets For Mac を最初にリリースしたバージョンで、この過ちを犯してしまった。

Zero-Link したプログラムは、その Mac でしか動作しないのだ。

必ず、公開する前に、Deployment モードで、Zero-Link でない設定でコンパイル・リンクしたプログラムであることを確認して欲しい。

また、プロジェクト構成・設定項目が非常に柔軟で、非常に多く、初めての人は戸惑ってしまうかもしれない。くれぐれも、無意識に設定を変更することがないように注意しなくてはならない。

12-2-2 これから

Mac OS X もサーババージョンが出てきたこともあり、これから先、ますます、開発機会が増えること間違いなしだ。まずは、この章を読んで、チャレンジしてみて欲しい。

12-3 まずは準備

この章では、Mac OS X で Skype API を使用した簡単なサンプルプログラムの作成方法について解説する。解説にあたって、Xcode 環境に関してある程度の経験・知識があることを前提としている。

Xcode は Apple が無料で配布している開発環境でもあり、NextStep からの斬新な UI や思想さえ感じられる、たいへんおもしろい開発環境である。

これから、Mac OS X で動作するソフトウェアを開発していくのであれば、習得しておいて損のない環境だと思う。

- 使用言語：Xcode 1.5
- OS：Mac OS X 10.3.9

12-3-1 Xcode とサンプルコードのインストール

■ Xcode

Mac 本体に添付の CD もしくは Apple の Developer HP から Xcode のキットをダウンロードする。

<http://developer.apple.com/tools/xcode/>

インストール方法については、Apple のページを参照していただきたい。基本的には、ダウンロードしたインストーラを起動し、そのダイアログに従ってインストールすればよい。

■ Mac OS X 用 Skype API サンプルコード

Mac OS X で Skype API を使用しやすいように、Skype 社がサンプルコードを用意している。そのなかに、Skype API を使用するために必要な Framework が用意されているので、今回は、それを使用して解説する。

http://download.skype.com/share/devzone/example_macosx.dmg

ダウンロード後、ディスク上の任意の場所に展開しておく。

12-4 プログラム作成

今回作成するのは、Skype に登録しているコンタクトリストを画面に出力するプログラムである。本章では、プログラムのメインウインドウに貼り付けたラベルへコンタクトリストを表示する。

12-4-1 新規プロジェクトの作成

Xcode を起動し、サンプルプログラムのプロジェクトを作成する。Mac OS X のシステムディスクのディレクトリ「Developer¥Applications」配下に Xcode のアイコンがある。それをダブルクリックして Xcode を起動する。



図 12-1 Xcode のアイコン

第 13 章 Mac OS X AppleScript

本章では、AppleScript を用いて Skype を制御する方法について説明する。

13-1 AppleScript とは

AppleScript は、Mac OS X に標準で搭載されているスクリプティング環境である。文法は最近のプログラミング言語に比べるといささか貧弱なきらいがあるが、他の言語にない点として、大量のソフトウェアとの連携があげられる。この連携機能は作業工数が増えがちな GUI の中で自動化のための 1 つの答えであり、Mac OS X の価値を高めるのに一躍かっている。

Skype も AppleScript との連携機能を持っている。そのため、Skype がインストールされていれば、何の準備もなしに AppleScript から Skype が利用できる。

13-2 スクリプトの記述

AppleScript を記述するためのスクリプトエディタは標準でインストールされている。「アプリケーション」フォルダ内の「AppleScript」フォルダ内にある「スクリプトエディタ」がそれである。



図 13-1 スクリプトエディタのアイコン

これをダブルクリックすると、白紙のスクリプトエディタウィンドウが開く。上部の欄にコードを書く。スクリプトを記述し、ツールバーの実行ボタンを押せばスクリプトが実行される。今回のスクリプトでは、コンタクトリストのユーザーの数だけダイアログボックスが表示される。



図 13-2 スクリプトエディタ

13-3 スクリプト解説

次に、今回使用したスクリプトの解説を行う。本スクリプトは4つの部分からなる。

■ 1～3行目：Skype 通信部

まず、Skypeに"SEARCH FRIENDS"コマンドを送り、結果を取得する。結果は"USER hoge, fuga, hani"といった形で返却される。

```

0001 tell application "Skype"
0002 set ret to send command "SEARCH FRIENDS" script name "My Script"
0003 end tell
  
```

第 14 章 Linux C

この章では、Linux 上で gcc を用いて Skype API を使用した簡単なサンプルプログラムの作成方法について解説する。

- 使用言語：gcc 4.0
- OS：KNOPPIX v4.0.2jp (Debian 系 Linux、<http://unit.aist.go.jp/itri/knoppix/index.html>)
- ライブラリ：dbus version 0.23.x (DBUS ライブラリ、Skype との通信に使用)

14-1 dbus-0.23.x のインストール

Linux において、Skype は DBUS を用いて外部プログラムと通信する。ここでは、その DBUS のインストールについて解説する。なお、以下の説明は、インターネットに接続されていることを前提としている。

Linux ディストリビューションにおける、DBUS パッケージをインストールする。筆者の Debian 系 (KNOPPIX v4.0.2JP) の場合は、以下のパッケージを apt-get install コマンドでインストールしておく。

- dbus-1
- dbus-1-dev
- dbus-qt-1

14-2 DBUS システムバスの準備

1. /etc/dbus-1/system.d 配下に、以下の内容の設定ファイル skype.conf を置く。

```
<!DOCTYPE busconfig PUBLIC "-//freedesktop//DTD D-BUS Bus Configuration 1.0//
  EN" "http://www.freedesktop.org/standards/dbus/1.0/busconfig.dtd">
<busconfig>

  <!-- ../system.conf have denied everything, so we just punch some holes
  -->

  <policy context="default">
    <allow own="com.Skype.API"/>

    <allow send_destination="com.Skype.API"/>
    <allow receive_sender="com.Skype.API"/>

    <allow send_path="/com/Skype"/>
  </policy>

</busconfig>
```

2. dbus サービスの開始

次のコマンドで dbus 用のサービスを起動する。

```
/etc/init.d/dbus-1 start
```

14-3 ソースコード解説

今回作成するのは、Skype に登録しているコンタクトリストを簡単に出力するプログラムである。Linux 上においては、ターミナルで起動し、標準出力にコンタクトリストを出力する。

■ 93 行目：DBUS への接続

Skype へ接続するためには、まず、DBUS へ接続しなくてはならない。本プログラムでは、DBUS ライブラリの関数 `dbus_bus_get` を使用して DBUS へ接続している。

```
0093     dbus_cn = dbus_bus_get(DBUS_BUS_SYSTEM, &dbus_err);
```

■ 17 行目：Skype へのコマンド送信

Skype へコマンドを送信する際、Linux では DBUS（アプリケーションバス）を使用する。本プログラムでは、関数 `sendCommandToSkype` 内で Skype へのコマンド送信を行っている。

```
0017 static char *sendCommandToSkype(DBusConnection *dbus_cn, char *cmd)
```

実際の送信方法について、以下に順を追って解説しておく。

■ 32 行目：送信用メッセージの作成

Skype へコマンドを送信するために、DBUS へ渡すメッセージを DBUS の関数 `dbus_message_new_method_call` で作成する。

```
0032     dbus_msg = dbus_message_new_method_call( "com.Skype.API",
                                                "/com/Skype", "com.Skype.API", "Invoke" );
```

■ 37 行目：メッセージへ送信する Skype API コマンドの設定

作成したメッセージに Skype API コマンドを、DBUS の関数 `dbus_message_append_args` で設定する。

```
0037     if( ! dbus_message_append_args( dbus_msg, DBUS_TYPE_STRING, cmd,
                                       DBUS_TYPE_INVALID ) ){
```

■ 43 行目：メッセージの送信およびレスポンスの受信

メッセージを DBUS の関数 `dbus_connection_send_with_reply_and_block` で送信する。この関数を使用すると、同時にレスポンスを待ち、それを呼び出した側に返してくれる。

```
0043     dbus_res = dbus_connection_send_with_reply_and_block( dbus_cn,
                                                            dbus_msg, -1 /* タイムアウトしない */, &dbus_err );
```

■ 53 行目：レスポンスの取得

受信したレスポンスを DBUS の関数 `dbus_message_get_args` で取得する。

```
0053     dbus_message_get_args( dbus_res, &dbus_err, DBUS_TYPE_STRING,
                            &str, DBUS_TYPE_INVALID );
```

作成したメッセージおよび取得したレスポンスは、不要になったら必ず、DBUS の関数 `dbus_message_unref` で開放しなくてはならない。

```
例： dbus_message_unref( dbus_msg );
```

■ 104 行目：Skype への接続

Skype へ接続するためには、DBUS を通じて Skype へ NAME コマンドを送信しなくてはならない。さらに、NAME コマンドの引数には、アプリケーション固有の文字列を渡さなくて

第 15 章 Linux Python

この章では、Linux 上で Skype API を使用した Python での簡単なサンプルプログラムの作成方法について解説する。

- 使用言語：Python 2.4
- OS：Gentoo Linux (<http://www.gentoo.org>)
- ライブラリ：dbus version 0.23.x (DBUS ライブラリ、Skype との通信に使用)

15-1 ソフトウェアのインストール

Linux では、Skype は DBUS を使用して外部プログラムと通信する。Skype と通信するためには、DBUS のバージョンは 0.23.x を使用しなければならない。

15-1-1 dbus-0.23.x のインストール

ここでは、Gentoo Linux のパッケージ管理システムを使ってインストールする。デフォルトの状態では、dbus の最新バージョンがインストールされるため、0.23 より大きいバージョンの dbus はマスクして、インストールされないようにする。

```
$ echo ">=sys-apps/dbus-0.24" >> /etc/portage/package.mask
```

次に dbus をインストールする。

```
$ emerge sys-apps/dbus
```

15-1-2 Skype のインストール

Gentoo Linux のパッケージで Skype をインストールする。

```
$ emerge net-im/skype
```

Gentoo Linux のパッケージ管理システムでインストールすると、/etc/dbus-1/system.d 配下に skype.conf もインストールされるため、設定ファイルの作成、変更は必要ない。

15-1-3 dbus サービスの開始

最後に dbus サービスを開始する。

```
$ /etc/init.d/dbus start
```

15-2 プログラム解説

今回作成するのは、Skype に登録しているコンタクトリストを簡単に出力するプログラムである。Linux 上においては、ターミナルで起動し、標準出力にコンタクトリストを出力する。プログラムのファイル名は、SkypeApiClient.py とする。

■ 1～13 行目：初期化

まず、dbus を初期化して、Skype と接続する。Skype と接続するためには、まず、クライアント（自分のプログラム）を dbus に登録する。dbus は、この登録した名前を元にプログラムを識別して、プロセス間のメッセージの送受信を行う。メッセージは、クライアントから Skype へ送信する経路である /com/Skype と、Skype からクライアントへ送信される /com/Skype/Client の 2 つの経路を通る。/com/Skype/Client の経路は、Skype からの通知として使用される。今回のサンプルでは、Skype からの通知メッセージは無視する。

次の手順で dbus を初期化して、Skype と接続する。

```
0001 import dbus
0002
0003 # Dbus の初期化
0004 remote_bus = dbus.SystemBus()
0005 skype_service = remote_bus.get_service('com.Skype.API')
0006 skype_api_object = skype_service.get_object('/com/Skype',
                                             'com.Skype.API')
```

```

0007
0008 # Skype に登録
0009 authQueryStr = 'NAME SkypeApiPythonTestClient'
0010 authQuery = skype_api_object.Invoke(authQueryStr)
0011
0012 if authQuery != 'OK':
0013     sys.exit('Could not bind to Skype client')

```

■ 15～20、30～34 行目：Skype API でのコンタクトリストの取得

Skype との接続が確立できたら、Skype API が使用可能になる。最初に「PROTOCOL 3」を送信して、Skype との通信で使用するプロトコルをネゴシエーションする。Skype にこのコマンドを送信すると、使用可能なプロトコル情報が返される。

```

0015 # プロトコル・ハンドシェーク
0016 protocolQueryStr = 'PROTOCOL 3'
0017 protocolQuery = skype_api_object.Invoke(protocolQueryStr)
0018
0019 if protocolQuery != 'PROTOCOL 3':
0020     sys.exit('This test program only supports Skype API protocol
              version 3')

0030 # Skype API によるコンタクトリストの取得
0031 contactListQuery = 'SEARCH FRIENDS'
0032 userList = skype_api_object.Invoke(contactListQuery)
0033 if userList[:6] == 'USERS ':
0034     print "Friend List=>%s" % (userList[6:])

```

■ 22～28 行目：Skype からの通知

今回のサンプルでは使用しないが、Skype からの通知イベントを処理するためには、Skype に通知イベントのコールバックを設定する。手順は次のようになる。

```

0022 # Skype からのコールバック処理
0023 # 今回のサンプルでは使用しないが、参考として掲載
0024 def notify_signal_handler(interface, signal_name, service, path,
                          message):
0025     print message
0026
0027 skype_callback_object
    = skype_service.get_object("/com/Skype/Client", "com.Skype.API")
0028 skype_callback_object.connect_to_signal('Notify',
    notify_signal_handler)

```

コールバック関数の引数の interface には、remote_bus が設定されている。signal_name に

第 16 章 Skype の音声処理

本章では、Skype で音声データを扱うプログラムの作成方法について解説する。

16-1 電話としての Skype

Skype は本質的には IM (インスタントメッセージング) であり、そこでやり取りされるデータはテキスト (チャット)、音声、ビデオ、A2A (アプリケーション間通信)、と幅広い。しかし、Skype が急速に普及した要因は、インターネット電話としての使いやすさ、音質のよさによるものである。

Skype を電話として見た場合、通常の電話機あるいは電話サービスと比べると機能面で不足している部分もある。しかし、PC 上のソフトウェアであるがゆえに、通常の電話機あるいは電話サービスよりも柔軟に他のアプリケーションと連携することができる。そのため、通常の電話サービスと同様の、あるいは新しい視点による音声・電話サービスを構築することも可能であろう。

本章では、そのような音声サービスを開発するための基礎的なテクニックについて解説する。

開発環境としては、以下を使用した。他の言語や ActiveS を使用しないで Skype API を使用する方法については、第 3 章から第 15 章を参考にしてほしい。基本的な手法は変わらないはずだ。

- 使用言語：Visual C# 2005 Express Edition
- ライブラリ：ActiveS COM Wrapper Ver1.02.03
- ライブラリ：Microsoft DirectX SDK Ver9.0c

16-1-1 Skype における音声処理

残念ながら、Skype には音声データを扱うための API が存在しない。つまり、音声ファイル

を指定して再生したり、通話を録音したりすることは、Skype API を使用して実現することはできない。音声関連の API が存在しない理由は不明だが、おそらくプライバシーの問題を考慮した結果の判断ではないかと、筆者は推測している。通話の録音がプライバシーの問題に関わることは容易に想像がつく（コラム「その声は誰のもの？」参照）。

Skype に音声関連の API が存在しないため、音声処理は OS に依存した音声処理用の API を使用することになる。ここでは、Windows を対象としている。

16-1-2 Windows における音声処理

Windows における音声処理には、大別して 2 つの手法がある。

1. Win32 ベースのマルチメディア API
2. DirectX

Win32 ベースのマルチメディア API には、高レベル API としてメディアコントロールインターフェイス (MCI) API があり、低レベル API としてウェーブフォーム API やオーディオミキサー API などがある。MCI は機能が単純で手軽に実装できるが、複雑な音声処理はできない。ウェーブフォーム API 等の低レベル API はプログラミングが複雑になるが、自由度の高い音声処理が可能となる。

DirectX は、Win32 および .NET Framework の両方で使用可能である。DirectX は機能別の API 群で構成されるが、音声処理については DirectSound、DirectMusic で処理する。DirectSound が低レベル、DirectMusic が高レベルといえるが、DirectMusic は高度な再生機能を実現するものであり、WAVE 音声の録音はできない。

今回は、実装が比較的容易で自由度も高いことから、主に DirectSound を使用して音声処理を行う。

16-2 サンプルについて

サンプルとして、単純な留守番電話の機能を実装する。サンプルでは Skype 通話時において、以下の基本的な音声処理を行うものとする。

- 音声ファイルを通話先へ再生
- 通話相手の音声を録音

16-2-1 サンプルの動作

先に作成するサンプルの仕様を決めておく。

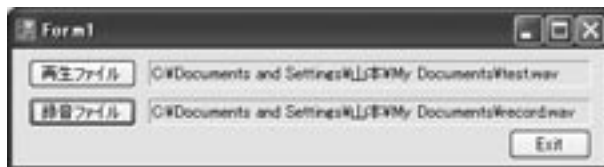


図 16-1 サンプル実行画面

- [再生ファイル] ボタン：応答音声ファイルを指定
- [録音ファイル] ボタン：保存音声ファイルを指定

プログラムは以下のように動作する。

1. Skype による着信があった場合、自動的に応答する。
2. 指定済みの再生ファイルを通話相手に再生する。
3. 通話相手の声を録音する。
4. 通話が終了したときに録音を終了し、録音ファイル名で保存する。

このサンプルは、最低限の機能のみを実装した留守番電話である。実用的な用途に用いるためには、通話毎に個別の録音ファイルを生成したり、着信時間や発信者の Skype 名を記録する機能などが必要であろう。

また、処理の流れをわかりやすくするために、サンプル中のエラー制御も最小限にしている。

16-2-2 2つの実装方法

以下の2つの方法で実装する。両者とも DirectSound を使用する点は同じである。

- 16-5 ～ 16-7：Windows オーディオミキサーを使用した実装
- 16-8：VAC（Virtual Audio Cable）を使用した実装

Windows オーディオミキサーを使用した実装は、PC が使用しているサウンドデバイスの仕様に依存する。

VAC (Virtual Audio Cable) は仮想サウンドデバイスといえるものだが、有料の製品であり、使用するためにはライセンスが必要となる。しかし、ミキサーを使用した場合のように PC のサウンドデバイスに依存することはない。

残念ながら、どちらも一長一短あり、決定版といえる解はない。実務においては、導入対象となる PC が確定している場合はミキサーによる実装を検討してもよいが、汎用的なアプリケーションを開発する場合には VAC を検討すべきであろう。

16-3 サンプルの準備

サンプル用にプロジェクトを作成しておく。以下の設定は、2 つの実装方法に共通のものとなる。

16-3-1 ActiveS の導入

第 2 章「2-2 ActiveS とは」の項目を参照。

16-3-2 プロジェクトの作成

[Visual Studio にインストールされたテンプレート] - [Windows アプリケーション] として、新規プロジェクトを作成する。例として、プロジェクト名は「Sample」とする。



図 16-2 新規プロジェクト画面

Form1 フォームに、以下のコントロールを追加する。

第 17 章 A2A

17-1 はじめに

ここまで紹介した API は、すべてローカル（1 台の PC）の内部でアプリケーションと Skype が連携するための API であるが、これから紹介する A2A（Application to Application）はリモート（他の PC）との通信を行う API という点で他の API とは大きく一線を画している。また、ここでは Skype API の基本的な利用方法は理解されているものとして A2A にクローズアップして解説を進める。

A2A とは、簡単にいえば Skype ネットワーク上でデータ通信を提供する API である。以前はこの機能がなかったため、Skype を通じてデータのやり取りを行おうとする場合チャットを利用するしかなかった。チャットでもデータの送受信は行えるが、チャットのウィンドウにデータが表示されてしまい、ユーザーの目に触れてしまうという欠点があった。A2A の登場により、ユーザーの目に触れることなくデータ通信が利用できる。ただし、通信を行う各ノードに A2A を利用したアプリケーションが配置されていることが必要となる。

17-1-1 A2A の魅力

■ Skype のネットワークを利用できる

- Skype は高い接続性を誇る。インターネットに直接接続している場合はいうまでもなく、NAT や HTTPS-Proxy 等を利用している場合も利用できる。その高い接続性を A2A は利用できる点で魅力である。もちろん、VirtualEther 等を利用して VPN を構築していれば Skype より高い接続性を確保できるが、インストール・設定という点で Skype さえインストールしていれば高い接続性が得られるという点は魅力である。
- A2A は、Skype の通話と同様にノード間の通信はすべて暗号化されている。

■音声と同時に使用できる（音声+チャット+A2A）ことで新しいコミュニケーションが可能となる

- Skype は、元々音声通話のアプリケーションである。A2A を利用したアプリケーションを併用することで、単に他のユーザーとデータ通信が可能だけでなく音声による会話を行いながらデータ通信ができる。これは電話というコミュニケーション手段を独自に拡張できるという魅力がある。たとえば通販カタログアプリケーションで A2A を使えば、音声通話とともに商品をナビゲートしながら顧客に発注を促すことができる。もちろん、従来のクライアントサーバ方式でノード間をサーバが取り持てば同様のことは可能であるが、通話しているノード間で基本的にサーバを介さず通信が実施できること（P2P）は新しい魅力であろう。

17-1-2 A2A の注意点

■セキュリティ

- ネットワーク：A2A を使うことで高い接続性と暗号化が得られることは前述の通りだが、逆にそのことが諸刃の剣となる。昨今の個人情報保護法施行以来、データの持ち出しに関しては各企業ともかなり注意を払っていると思うが、Skype のファイル転送を禁止（レジストリ）したとしても A2A を使ってファイル転送を行うプログラムを作成すればファイルの転送が可能になる。Skype には API を禁止するレジストリも存在するが、API を禁止されてしまうような事態になっては元も子もない。各人モラルを持ってアプリケーションを作成するよう注意願いたい。
- 端末内：Skype により端末間の通信は暗号化されているが、端末内でクライアントプログラムと Skype の連携に使用するメッセージは暗号化されていない。Windows のメッセージをのぞき見するようなツールや Bot であれば、アプリケーションと Skype とのやりとりをのぞき見することができる。いくら Skype 間が暗号化されているとはいえ、パスワード等の重要なデータを A2A でやりとりする場合はアプリケーション側で暗号化することをお勧めする。

■A2A は通信手段でしかない

- プロトコルの設計が必要：A2A はノード間でデータを送受信する機能を提供するのみで、各ノード間でデータの同期を取ったりはしてくれない。そのため、ノード間で送受信するデータに関してはきちんと設計することが必要である。通信系のアプリケーションを作成しようと思えば当たり前のことではあるが、Skype が何でもやってくれるわけではない点は注意が必要である。
- Skype への依存：A2A を基本の通信手段にしてしまうと、Skype がないと話にならない

ソフトウェアになってしまう。この点に脅威を感じる趣は、別の通信手段を用意するかプラグインなどとして万が一の場合に足回りの通信経路を別の方法に差し替えられるように設計しておくべきであろう。

- 利用できる Skype のバージョン：1.4 以降の Windows 版 Skype（Macintosh 版は準備中の模様）

■執筆時点での問題点（Skype の不備）

- データの先頭に 0x20（半角スペース）が含まれている場合切り捨てられる。
- データの末尾に 0x20 以下のデータがある場合切り捨てられる。
- 送受信通知のデータ長が異なり、送信通知のバイト数が +2 されている。

17-2 API 解説

A2A で実際にプログラムを作成する前に、A2A 特有の用語を説明する。

● APPLICATION（アプリケーションオブジェクト）

A2A を利用するアプリケーションでは、Skype に対しアプリケーションオブジェクトを作成する必要がある。アプリケーションオブジェクトとは、A2A を利用する際のプログラム識別子のようなもので、以下の STREAM、DATAGRAM はこのアプリケーションオブジェクトに対する変更という形でコマンドを発行する。

● STREAM（ストリーム）

IP 通信を行う際の TCP のようなもの。コネクションの作成、データの送受信、コネクションの切断という手順をふむ必要があるが、データの送受信が保障される。

● DATAGRAM（データグラム）

IP 通信を行う際の UDP のようなもの。データの送受信に関して保障されないため、ユーザー側で送受信の確認や再送のプロトコルを実装する必要がある。A2A では、Skype に対してまずアプリケーションオブジェクトを作成し、作成したアプリケーションオブジェクトを指定して STREAM または DATAGRAM を利用してデータの送受信を行う。

第 18 章 SKSweets For Mac

ここで紹介するプログラムは、2005 年 3 月より公開している Windows 版 SKSweets から、通知系機能を Mac OS X へ移植したプログラムである。SKSweets For Mac は、イベントをメッセージやメールで通知するツールである。

18-1 機能解説

SKSweets For Mac では、以下の機能を提供している。

●チャットメッセージ受信通知機能

チャットを受信した際に、他の Skype アカウントへその内容を通知する。

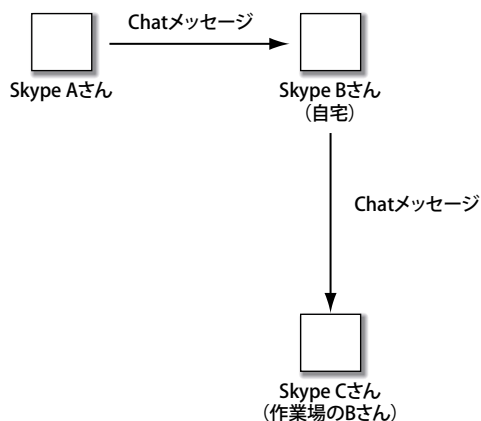


図 18-1 SKSweets チャットメッセージ受信通知機能

●着信通知機能

着信した際に、他の Skype アカウントへ着信があったことを通知する機能

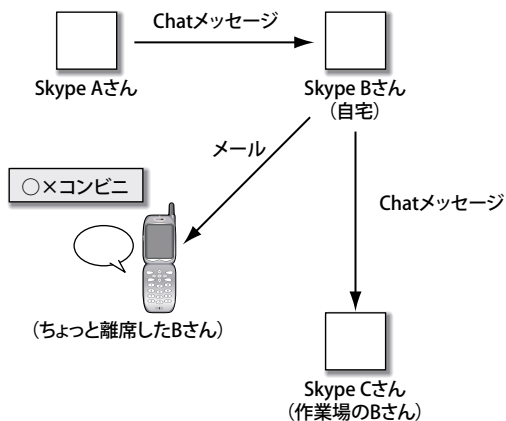


図 18-2 SKSweets 着信通知機能

●不在案内機能

着信した際に、その発信側に不在案内をチャットで通知する機能



図 18-3 SKSweets 不在案内機能

着信をメールで通知する機能などは、PC からの中座の際に着信を逃さないために便利なようだ。

18-1-1 機能フロー

まず、Skype との接続までの処理については、「第 12 章 Mac OS X ObjectiveC」を参照していただきたい。ここでは、このアプリケーション特有の部分について説明する。

■チャットメッセージ受信イベントの処理

Skype がチャットメッセージを受信すると、接続しているアプリケーションへ「MESSAGE

<msgID> RECEIVED」を通知する（<msgID> はメッセージを唯一に特定するための ID である）。

その通知を受信した SKSweets は、そのメッセージ ID（<msgID>）を記憶し、その送信者およびメッセージ本文を取得するために、Skype へコマンドを送信する。

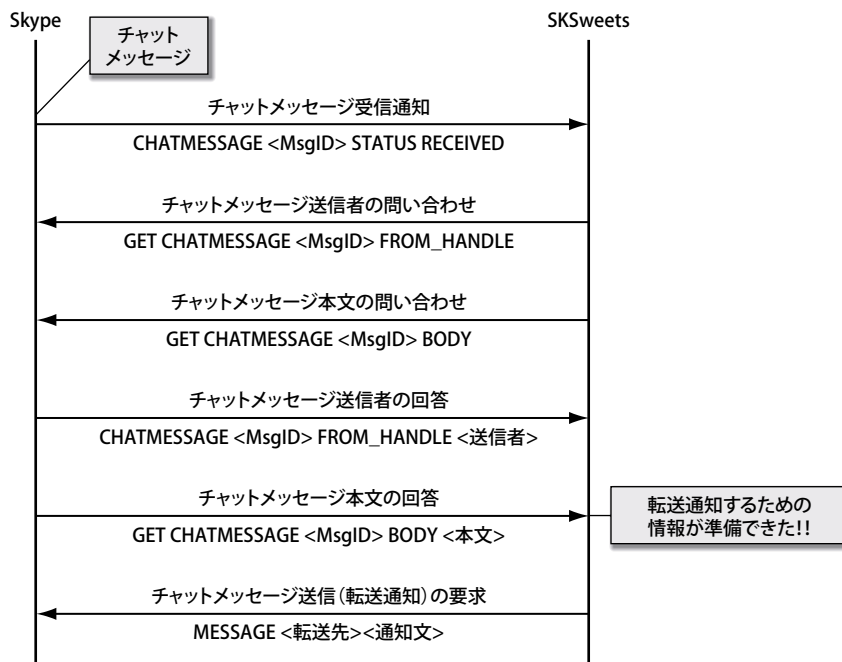


図 18-4 SKSweets チャットメッセージ受信イベントの処理

メッセージの送信者・本文の内容通知を受信した際、その情報を蓄積する。受信するつど、送信者・本文双方の情報がそろったメッセージをチェックし、そろっているメッセージに関しては、その内容をメールやチャットメッセージで、他へ送信する。

第 19 章 Skype Web Gateway

Skype Web Gateway とは、Skype の機能と Web サービス機能を結びつけるゲートウェイプログラムのことである。この Skype Web Gateway をサンプルとして、Skype API のメッセージ処理の基本について解説する。

●使用言語：Visual C++ 2005 Express Edition

19-1 はじめに

最初に、Skype Web Gateway が生まれた経緯について説明しておこう。Skype Web Gateway の原型は、本書執筆者でもある池嶋氏によって 2005 年 11 月に提示された。池嶋氏は、ポータルサイトの翻訳機能を Web サービスとして利用し、Skype のチャットに日本語 / 英語翻訳機能を付加する簡潔な方法を示した。このプログラムは、池嶋氏の「public Skype forums for the world」(<http://ikejisoft.com/skype/publicforum>) において「Kiroku-kun」(secretary services) の機能として実装されている。池嶋氏のアイデアは、Skype 開発者コミュニティの有志によるディスカッションにより「Skype Web Gateway」というコンセプトに整理された。

このコンセプトに基づき、oyajin 氏により改めて Windows 用のサンプルソースコードが記述され、Japanese Skype Developers Forum (<http://forum.skype.com/>) において公開された。このプログラムは、<http://www.oyajin.jp/skype/sample/Translator.lzh> にてダウンロード可能である。また、Japanese Skype Developers Forum 上では、本書執筆者でもある大谷氏による Python 版 Skype Web Gateway も公開されている (<http://www.liris.org/program/skypewrapper.en/skygw.zip>)。

今回、Skype Web Gateway をサンプルとして取り上げる理由は 2 つある。

1. Windows による Skype のメッセージ処理の簡潔な実装例であること。

2. Skype と Web サービスの簡単なマッシュアップ方法として、幅広い応用例が考えられること。

掲載するプログラムは、oyajin 氏のソースコードをベースに、筆者による若干の変更を加えたものである。変更之际は、本書執筆者でもある xai 氏によるプログラムを参考にさせていただいた。

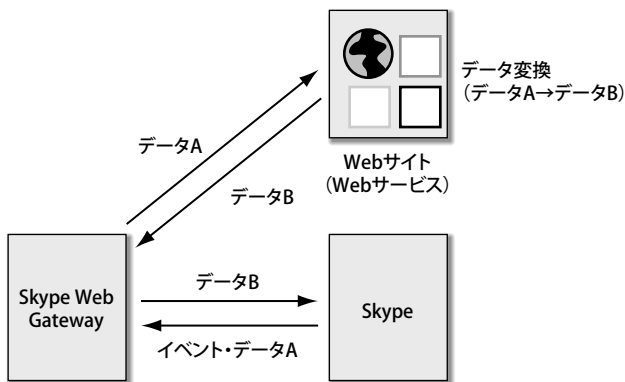


図 19-1 Skype Web Gateway の概念

19-2 プログラムの動作

Skype Web Gateway は Web サービスの内容と Skype のイベントの組み合わせにより、さまざまに応用できるものだが、oyajin 氏のプログラムは池嶋氏の原型と同じく日本語 / 英語翻訳プログラム「Translator」として実装されている。プログラムは Skype のチャットウィンドウを監視し、先頭に ## がついた行を見つけると、そのテキストを翻訳用の Web サーバに投げる。Web サーバから戻ってきたテキストを整形して、再度 Skype のチャットウィンドウに戻している。

まずは動かしてみよう。

1. oyajin 氏のソースコードを以下よりダウンロードする。

<http://www.oyajin.jp/skype/sample/Translator.lzh>

2. ソリューションファイル Translator.sln を Visual C++ 2005 で読み込む。プログラムは Visual C++ 2003 で作成されたものなので、Visual C++ 2005 ではプロジェクトの変換が

必要になる。Visual C++ 2005 Express Edition でも問題なく変換できるはずだ。

- リソースでMFCのヘッダを使用しているが、Visual C++ 2005 Express EditionにはMFCが含まれていないため、Microsoft Platform SDKを導入する。Microsoft Platform SDKの導入方法については、第4章「Windows Visual C++/ActiveS」の「4-1-3 Microsoft Platform SDKの導入」を参照してほしい。使用しているのはMFCのヘッダのみなので、ディレクトリの参照設定は、インクルードパスに以下を設定するだけでよい。

```
C:\Program Files\Microsoft Platform SDK\Include\mfc
```

- Main.cppを本章末のソースコード全文と書き換える。
oyajin氏のプログラムは先頭文字「##」で日→英、英→日の両方を自動判別しているが、筆者のプログラムでは多言語対応を考慮して「##」で日→英、「#!」で英→日とした。また、参照するSkypeのイベントやインターネット接続方法等に若干の変更を行っている。
- ビルドに成功したら、Skypeが起動している環境で実行してみよう。チャットウィンドウを開き、「## こんにちは、世界！」と送信すると、「こんにちは、世界！ /-----/Hello World!」と返事が返ってくるはずだ。



図 19-2 Translator 実行結果 (Skype チャット画面)

著者紹介

岩田真一（個人 blog - <http://www.siwata.net/blog/>）

Skype Technologies S.A. 日本マーケット開発マネージャ兼 Developer Relations

Skype 公式ブログ：<http://share.skype.com/sites/ja/>

ロータス、マイクロソフトを経て、2001 年アリエル・ネットワークの設立に参加。P2P アプリケーションの設計、開発、開発マネージメント、また P2P 技術の啓蒙活動を行う。2005 年 8 月より現職。Skype の日本マーケット開発および、Skype API の啓蒙を通じて日本の Skype 普及に努める。

著書に『なるほどナットク！ P2P がわかる本』（オーム社）がある。

※「まえがき」「第 1 章 Skype API 概要」を執筆。

r ㄥ /xai (ryu@nyanyan.to、<http://nyanyan.to/>)

Skype Japanese Developer Forum Moderator

Skype 公式日本語開発者フォーラムのモデレータやっています。

Skype 公式開発者 Forum(J): <<http://forum.skype.com/viewforum.php?f=29>>

質問などは上記フォーラムで。

※「第 17 章 A2A」を執筆。

池嶋 俊 (e@ikejisoft.com)

筑波大学第三学群情報学類卒業。現在、筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻所属

大学生時代から、プログラミング、サークル、など精力的に活動し、ソフトイサー社の立ち上げにも参加。現在も筑波大学生によるベンチャー企業「てつくてつく」にも所属。『Skype の仕組み入門』という書籍を執筆するなど、Skype、P2P に関する造詣が深い。卒業研究の題目は「P2P 技術を利用した Web システムに関する研究」であり、AsagumoWeb をアウトプットしている。

※「第 10 章 Windows Ruby/ActiveS」「第 11 章 Windows Java/ActiveS」「第 13 章 Mac OS X AppleScript」を執筆。

著者紹介

大谷弘喜 (liris.pp@gmail.com、<http://www.liris.org/>)

アリエル・ネットワーク株式会社開発マネージャ (<http://www.ariel-networks.com/>)

ロータスで開発を経て、2001年アリエルネットワークの設立に参加。P2Pアプリケーションの設計、開発に携わり、現在はWebアプリケーションの開発を行っている。

※「第8章 Windows Python」「第9章 Windows Python/ActiveS」「第15章 Linux Python」を執筆。

須崎雅道 (msuzaki@gmail.com)

アリエル・ネットワーク株式会社開発部主任研究員

ロータス、IBMを経て2002年にアリエル・ネットワークに入社。P2Pアプリケーションの開発に携わり、現在はWebアプリケーションの開発に従事している。

※「第7章 Windows Visual C#/ActiveS」を執筆。

寺田亮

(有) インサイフォス代表取締役 (<http://www.insighfos.co.jp/>)

Skype関連では、某社コードレスハンドセットの連携アプリケーションを開発。

中学生で親に買ってもらったPC-8801(初代)でのN88-Basicから始めて、今ではデータセンター向けネットワークアプリケーションの開発を専門とすることに。

週末は家族でボランティア活動を楽しみながら、仕事に追われる毎日です。

※「第2章 さまざまな開発環境」「第3章 Windows Visual C++」を執筆。

谷萩毅之 (yahagi@gatagata.jp、<http://www.gatagata.jp>)

1983年「RC構造計算ソフト(N88-Basic)」からソフトウェアを作成、Skype API公開当初よりSKSweetsを公開、その後もwS、SKMapなどのフリーウェア/サービスを公開している。

職歴:Digital Equipment → Compaq → EC Architect (Softbank)を経て、現在のDigital Genes Co. 役員。IT関連のソフトウェア開発およびコンサルタントを行っている。

※「第5章 Windows Visual Basic」「第6章 Windows Visual Basic/ActiveS」「第12章 Mac OS X ObjectiveC」「第14章 Linux C」「第18章 SKSweets For Mac」を執筆。

山本達也 (support@tapur.com、<http://www.tapur.com/>)

(有) パークソフト代表取締役 (<http://www.paak.co.jp/>)

本業の傍ら、気の向くままに以下の製品・サービスを開発・公開しています。

・Skype用留守番・録音・録画フリーウェア『Tapur』(※「たぴゅーる」と読みます)

・Skype用ビデオテストサービス『videoecho123』(現在、英語版のみ)

※「第4章 Windows Visual C++/ActiveS」「第16章 Skypeの音声処理」「第19章 Skype Web Gateway」を執筆。

Skype API Book Vol.1

2006年8月23日 初版発行 (v.1.0)

監修 Japanese Skype Developers Forum API Book Project

著者 岩田真一、rゆ/xai、池嶋 俊、大谷弘喜、須崎雅道、寺田 亮、谷萩毅之、山本達也

発行者 鈴木三佳

発行所 有限会社 Talpa-Tech

〒182-0023

東京都調布市染地 3-1-68-W129

mail info@talpa-tech.com

URL <http://www.talpa-tech.com/>

印刷・製本 コンテンツワークス株式会社

© 2006 Shinichi Iwata, rYu/xai, Syun Ikejima, Hiroki Ohtani, Masayuki Suzaki, Ryo Terada, Takeshi Yahagi, Tatsuya Yamamoto

本書の一部あるいはすべてについて、法律の定めのある場合または権利者の承諾がある場合を除き、いかなる方法においても複製・複写することはできません。

ISBN 4-903408-01-9

Printed in Japan