第1章

Windowsサービスの概要

本章では，Windowsサービスの概要，Microsoft.NETについて説明します．

1.1 Windowsサービス

1.1.1 概要

近年，「サービス」という単語を耳にすると「XML Web サービス」をイメージされる方が多いかと思われますが，XML Webサービスが登場する前からWindowsサービスは存在していました．

WindowsサービスはWindows NTと同時に登場した，歴史のあるものです．従来は，動作可能OSがWindows NTのみであったため「NTサービス」あるいは「システムサービス」と呼ばれていましたが，Windows NTのアーキテクチャを踏襲しているWindows 2000やWindows XP，Windows Server 2003でも動作するため「Windowsサービス」と呼ばれるようになりました．厳密な定義では，システムサービスにはハードウェア制御を行う「デバイスドライバ」と，システムサービスを提供する「Win32サービス」がありますが，「Win32サービス＝Windowsサービス」です．（**図**1-1）

Windowsサービスは，UNIXのデーモンのようにバックグランドで長時間動作し，ユーザーインターフェースを持たないアプリケーションプログラムです．通常のデスクトップアプリケーションと異なりバッググランドで動作するため，コンピュータが起動後，Windowsにユーザーがログオンする前に自動的に起動することができます．Windowsサービスの特徴を，**表**1-1に示します．

図1-1　Windowsサービスの概要

●標準で提供されるWindowsサービス

Windows NT/2000/XP/Server 2003では，OSとともに標準で提供されるWindowsサービスがあり，OSの基本的な機能やネットワークプロトコルごとの機能などが実現されています．また，SQL Serverなどのサーバ製品やVisual Studio .NET(以下，VS.NET)などの開発製品の多くでは，Windowsサービスを使用しています．

Windows 2000 Professionalで提供される主なWindowsサービスを，**表1-2**に示します．

表1-1　Windowsサービスの特徴

項　目	説　明
バックグランド動作	バックグランドで，ログオンユーザーの有無にかかわらず動作する．また，コンピュータ起動時に自動的に開始することもでき，一時停止・再開なども可能
長時間動作	通常Windowsサービスは，ユーザーやデスクトップアプリケーションの要求に応じて動作を開始・停止し，長時間システムに常駐する
ユーザーインターフェースなし	Windowsサービスは，バックグランドで基本的なサービスを提供するため，ユーザーインターフェースを持たない．サーバーで使用するときや，コンピュータを利用する他のログオンユーザーの邪魔をせずに，バックグランドで長時間稼動させる必要がある場合に最適
特定のログオンアカウント	Windowsサービスは，ログオンユーザーや既定のコンピュータアカウントとは異なる，特定のユーザーアカウントのセキュリティコンテキストで実行される
管理ツール	Windowsサービスがパラメータなどを必要とする場合は，Windowsサービス専用の管理ツール(MMCアプレット，コントロールパネルアプレット)などを経由して設定する

表1-2　主なWindowsサービス

システム基本機能		ネットワーク機能
Alerter	Smart Card	Computer Browser
COM+ Event System	Task Scheduler	DHCP Client
Event Log	Windows Time	DNS Client
Logical Disk Manager	Workstation	FAX Service
Messenger		FTP Publishing Service
Net Logon		IIS Admin Service
NT LM Security Support Provider		IPSEC Policy Agent
Performance Logs and Alerts		Network DDE
Plag and Play		QoS RSVP
Print Spooler		Routing and Remote Access
Protected Storage		SMTP
Removable Storage		Telephony
RunAs Service		Telnet
Server		World Wide Web Publishing Service

1.1.2 ソフトウェア構成

Windowsサービスは「サービス制御マネージャ」により管理され，Windowsサービスで何らかの問題が発生した場合などには，「イベントログ」にイベントを書き出すことで，システム管理者に情報を提供します．（**図**1-2）

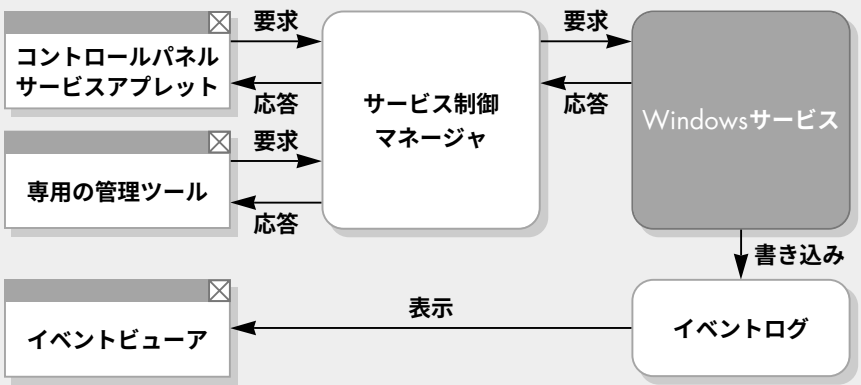
●サービス制御マネージャ

サービス制御マネージャ（サービスコントロールマネージャ，SCM）は，Windowsサービス管理の中核となるWindows標準のユーティリティです．サービス制御マネージャには，コントロールパネルのサービスアプレットからアクセスできます．Windowsサービスにカスタマイズ動作を組み込む場合は，そのWindowsサービス専用の管理ツールを開発し，その管理ツールから.NET Frameworkのクラスライブラリを使用することで，サービス制御マネージャ経由でWindowsサービスへカスタマイズ動作の開始が指示できます．

サービス制御マネージャは，Windowsサービスに関する各種の処理を行います．サービス制御マネージャの実体は，「SERVICES.EXE」という名前で，Windowsシステム起動時に自動的に起動され，各種Windowsサービスを自動開始します．また，Windowsサービスをインストールする際も，サービス制御マネージャが介在し，Windowsサービスに関する情報を，レジストリ（HKEY_LOCALMACHINE¥SYSTEM¥CurrentControlSet¥Services以下）に保存します．サービス制御マネージャの主な役割は，以下のとおりです．

1. Windows**サービスのインストール／アンインストール**
2. **サービスアプレットなどから要求を受け付け，該当の**Windows**サービスに送信**
3. **スタートアップの種類が自動開始の**Windows**サービスを，OS起動時に開始**
4. **インストール済みの**Windows**サービスをレジストリで管理**
5. **動作中の**Windows**サービスの状態をレジストリで管理**

図1-2　ソフトウェア構成



1.1.3 動作概要

Windowsサービスは，サービス制御マネージャからの開始・停止・一時停止・再開などの要求により動作します．Windowsサービスの状態としては，停止状態・実行状態・一時停止状態があります(**図**1-3)．また，通常のデスクトップアプリケーションと異なり，Windowsサービスの実行可能ファイルは，開始前にインストールしておく必要があります．

●開始-停止，一時停止-再開のサポート

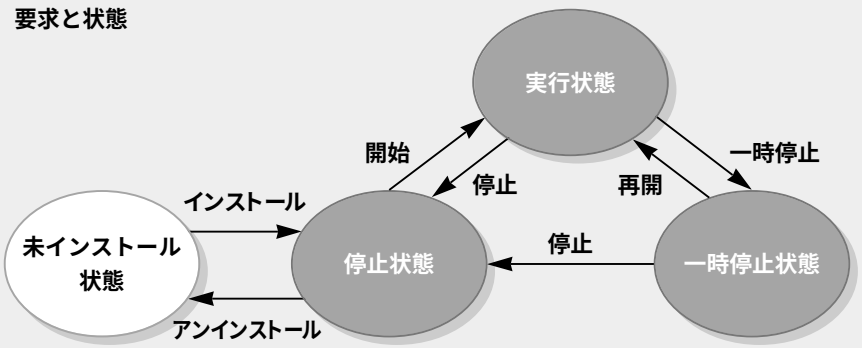
基本的な処理だけを行うWindowsサービスであれば，開始と停止だけをサポートしていれば問題はありません．一時停止・再開については，処理の中断と再開が可能でなければ，サポートすべきではありません．例えば，通信を行うWindowsサービスが実行状態のときにサービス制御マネージャからの一時停止の要求に対して，Windowsサービスがどのように動作するかを考えてみてください．この場合，通信中のパケットやコネクションを破棄するか，一時停止の要求を遅らせるか，一時停止を受け付けた後はクライアントからの要求を受け付けないなど，さまざまな方法が考えられますが，それらに応じた例外処理などを組み込んでみて，テストする必要があります．

●シャットダウン時の通知と低電力モードの通知

Windowsサービスでは，サービス制御マネージャからシャットダウン時の通知や，低電力モードの通知を受け取ることもできます．Windowsシステムがシャットダウンを開始すると，すべてのWindowsサービスは20秒経過する前に停止処理を行う必要があります．その後，Windowsシステムはまだ停止していないWindowsサービスを強制的に終了する処理を開始します．強制的な終了が実行される前(つまり，20秒以内)に，Windowsサービスでは現在行っている処理に関連するデータをすみやかに退避する必要があります．シャットダウン要求は，停止要求と異なりサービスを提供しているクライアントへの配慮は不要です．

低電力モードの通知は，ラップトップコンピュータなどで，バッテリー電力の低下，サスペンド状態への変更など電力モードが変更された場合に通知され，それぞれの状態や要求に対して適切に応答する必要があります．

図1-3　要求と状態



●スタートアップの種類

Windowsサービスは，Windowsシステム起動時に自動的に開始するか，手動で開始するか，あるいは無効とするかの3種類が選択できます．スタートアップの種類は，コントロールパネルのサービスアプレットで確認・変更ができます．

●依存関係

Windowsシステム起動時に，Windowsサービスは自動的に開始することができますが，Windowsサービスの起動順序は通常，Windowsシステム(OS)によって決定されます．Windowsシステムでは，他のWindowsサービスの起動完了を待たずに，次々とWindowsサービスが起動されます．

複数のWindowsサービスを，ある一定の順序で起動したい場合は，Windowsサービスの依存関係を設定します．このWindowsサービスの依存関係は，レジストリで設定します．例として，Service1 -> Service2の順にWindowsサービスを起動したい場合，Service2のレジストリに，依存関係を設定します(**表**1-3)．

先に起動させたいWindowsサービスが複数ある場合，サービス名を各行に1つずつ入力します．

> **Windows NT/2000では，REG_MULTI_SZのデータを編集するために，「regedit.exe」ではなく「regedt32.exe」を使用してください．**

依存関係は，コントロールパネルのサービスアプレットで，各Windowsサービスのプロパティを表示して，確認できます．

1.1.4 開発方法

●従来の開発方法

VS.NET発売以前では，WindowsサービスはC++(またはC言語)でしか開発できず，Windowsサービス固有のWIN32APIを呼び出すなど，開発はとても手間のかかるものでした．

また，Visual Studio 6.0のATL(Active Template Library)で，ある程度のスケルトンは作成できましたが，インストーラは別に開発する必要があり，イベントログへの書き込みなども貧弱なもので，開発効率はよくありませんでした．

表1-3　**依存関係のレジストリ**

ハイブ	HKEY_LOCAL_MACHINE¥ SYSTEM¥ CurrentControlSet¥ Services¥ Service2
キー	DependOnServices
タイプ	REG_MULTI_SZ
値	Service1

●Windowsサービスの開発

VS .NETでは，Windowsサービスは，VS .NETのプロジェクトとして作成でき，Visual Basic .NET(以下，VB.NET)をはじめさまざまなプログラミング言語で開発できます．Windowsサービスには，サービス制御マネージャから受け付ける要求(コマンド)をプロパティに設定し，Windowsサービスが受け付けた要求に対する処理をプログラミングします．Windowsサービスの開発方法については，「1.3　VS.NETでのWindowsサービス開発」「第3章　Windowsサービスの開発・導入」を参照してください．

●Windowsサービスのインストール

Windowsサービスのインストールには，コマンドラインユーティリティを使用する方法と，VS.NETの配置機能を使用する方法があります．コマンドラインユーティリティを使用する方法では，Windowsサービスの実行可能ファイル(*.EXE)のパスを指定して「InstallUtil.exe」を実行します．InstallUtil.exeについては「付録1.2　InstallUtil.exe」を参照してください．VS.NETの配置機能を使用する方法では，VS.NETのGUIを使用して簡単にインストール用のプロジェクトおよびインストーラを作成することができます．

Windowsサービスをインストールすると，サービス制御マネージャを使用して，Windowsサービスを起動，停止，一時停止，再開，および設定できるようになります．Windowsサービスのインストールについては，「第3章　Windowsサービスの開発・導入」を参照してください．

●Windowsサービスのデバッグ

Windowsサービスは，VS.NETから直接デバッグまたは実行することはできません．Windowsサービスのインストールと起動を行ってから，デバッガをWindowsサービスのプロセスにアタッチする必要があります．Windowsサービスのデバッグについては，「第8章　デバッグ」を参照してください．

1.2 Microsoft.NET

1.2.1 .NET構想

●概要

2000年6月にマイクロソフト社は.NET構想を発表しました．.NET構想は，インターネットをプラットフォームとし，インターネット接続されたさまざまな環境に，さまざまなサービスを提供することを主眼としています．発表から2年経過した現在では，.NETのインフラとなる.NET Enterprise Servers製品，.NETの実行環境・開発環境である.NET Framework, Visual Studio .NETがリリースされ，企業システムを構築・開発する準備が整いました(**図**1-4)．

●.NETで実現可能なシステム

.NETは近年の技術動向を取り入れ，インターネットをプラットフォームとしたアプリケーション環境の実現を目指していますが，.NET構想に登場する.NET Framework SDK, Visual Studio .NETは，XML Webサービスを構築・開発するためだけに利用するものではなく，デスクトップアプリケーションや，Windowsサービスの開発にも，利用できます(**図**1-5)．

図1-4　.NETの概要

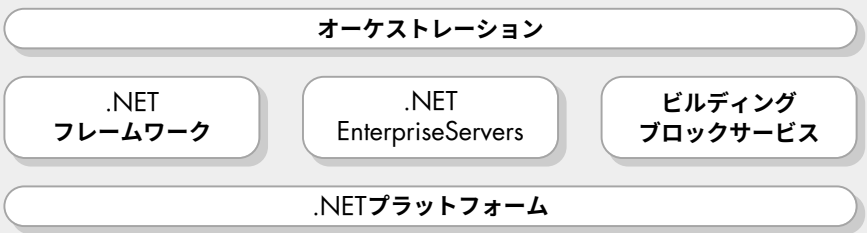
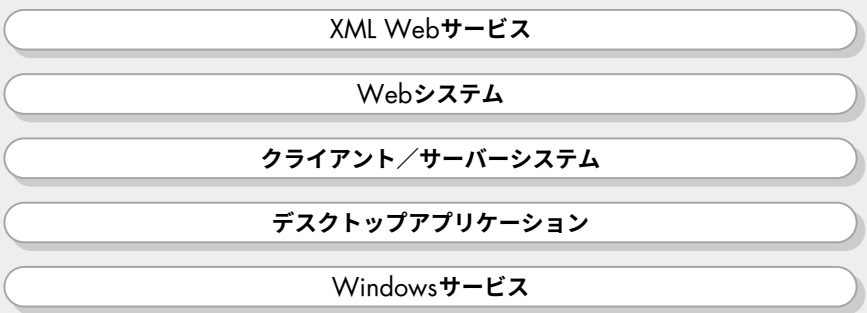


図1-5　.NETで実現可能なシステム



1.2.2 .NET Framework

.NET Frameworkは，主にXML Webサービスを構築・配布するための環境です．.NET Frameworkには，開発ツールセットである「Visual Studio .NET」，開発言語に依存しないランタイムエンジンである「CLR（Common Language Runtime）」，クラスライブラリなどが含まれます．.NET Frameworkの概要を，**図**1-6に示します．

VS.NETで開発されたプログラムは，中間コード（IL：Intermediate Language）に変換され，CLRがその中間コードを，JIT（Just In Time）コンパイルして実行します．クラスライブラリは，VB, VC#, VC++などプログラミング言語が異なっても共通のクラスライブラリを利用でき，ネームスペース（名前空間）ごとにグループ分割されています．クラスライブラリは，基本的には1つのネームスペースごとに1つのDLLというように，実装されています．

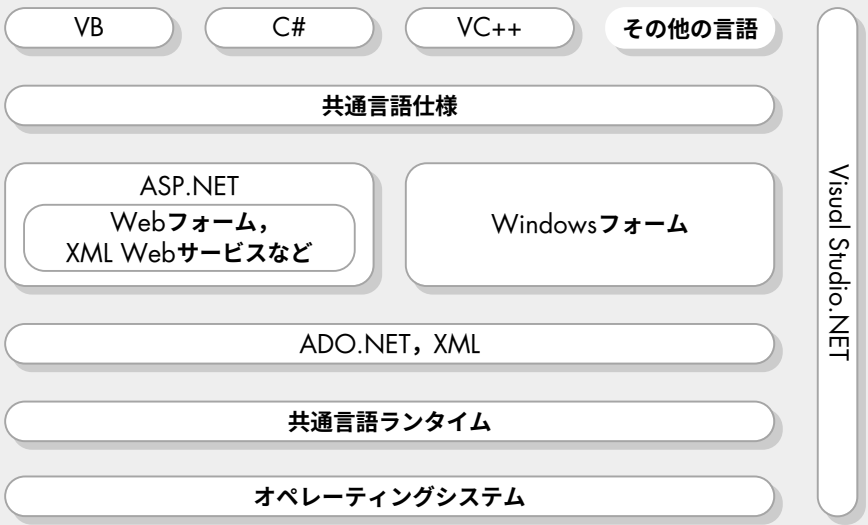
1.2.3 Visual Studio .NET

●概要

Visual Studio .NETは，XML Webサービスおよびアプリケーションを迅速に構築し統合するための開発ツール・開発環境です．旧バージョンと比較して，さまざまな点で操作性が向上し，生産性の向上が期待できます．

> **「Windowsサービス」プロジェクトのテンプレートおよび関連する機能は，VB.NETとC# .NETのStandard Editionでは使用できません．**

図1-6 .NET Frameworkの概要



●エディション

Visual Studio .NETにはEnterprise Architect，Enterprise Developer，Professional，Academicの4つのエディションがあります．エディションごとの機能の違いを，**表**1-4に示します．

表1-4　Visual Studio .NETエディションごとの機能

機能＼エディション	Enterprise Architect	Enterprise Developer	Professional	Academic
Visual Basic .NET	○	○	○	○
Visual C++ .NET	○	○	○	○
Visual C# .NET	○	○	○	○
XML Webサービスの作成と使用	○	○	○	○
Webアプリケーションの構築	○	○	○	○
Windowsアプリケーションの構築	○	○	○	○
MSDEでのテーブルとビューのデザイン	○	○	○	○
MSDE，SQL Server，Oracleでのテーブル，ビュー，プロシージャ，トリガ，関数，その他のデザイン	○	○	−	−
Windows 2000 Server	○	○	−	−
SQL Server 2000 Developer Edition	○	○	−	−
Commerce Server 2000 Developer Edition	○	○	−	−
Host Integration Server 2000 Developer Edition	○	○	−	−
Exchange Server 2000 Developer Edition	○	○	−	−
Visual SourceSafe	○	○	−	−
エンタープライズフレームワークとテンプレートの実行	○	○	−	−
エンタープライズフレームワークとテンプレートの編集	○	−	−	−
BizTalk Server 2000 Developer Edition	○	−	−	−
Visioベースのソフトウェアモデル	○	−	−	−
Visioベースのデータベースモデル	○	−	−	−
アプリケーションウィザードや課題の管理など，学生用ツール	−	−	−	○

1.3 VS .NETでのWindowsサービス開発

1.3.1 概要

VS .NETでは，Windowsサービスは，VS .NETのプロジェクト(Windowsサービステンプレート)として作成でき，VB.NETをはじめさまざまなプログラミング言語で開発できます．Windowsサービスには，サービス制御マネージャから受け付ける要求(コマンド)をプロパティに設定し，Windowsサービスが受け付けた要求に対する処理をプログラミングします．

●System.ServiceProcessクラス

VS.NETでWindowsサービス，インストーラ，管理ツールを開発する場合，.NET Framework のServiceProcessクラスとその列挙体を使用します．ServiceProcessクラスは，Windowsサービスを実行，インストールするためのクラスで，Windowsサービスは，このクラスの中のServiceBaseクラスから派生して実装されます．ServiceProcessクラスに含まれるクラスとその説明を，**表**1-5に示します．

表1-5　ServiceProcessクラス

クラス名	説　明
ServiceBase	Windowsサービスの一部として存在するWindowsサービスの基本クラスを提供する．ServiceBaseは，新しいWindowsサービスクラスの作成時に派生される必要がある
ServiceController	Windowsサービスを表し，実行中のWindowsサービスまたは停止したWindowsサービスへの接続，Windowsサービスの操作，またはWindowsサービスに関する情報の取得を実現する
ServiceController Permission	サービス制御に対するコードアクセスセキュリティ，アクセス許可を制御できるようにする
ServiceController PermissionAttribute	サービス制御のアクセス許可をチェックできるようにする
ServiceController PermissionEntry	ServiceControllerに対して設定するコードアクセスセキュリティ，アクセス許可の最小単位を定義する
ServiceController PermissionEntryCollection	ServiceControllerPermissionEntryオブジェクトの厳密に型指定されたコレクションを格納する
ServiceInstaller	ServiceBaseを拡張するクラスをインストールしてWindowsサービスを実装する
ServiceProcess DescriptionAttribute	プロパティまたはイベントの説明を指定する
ServiceProcessInstaller	ServiceBaseを拡張するクラスを含む実行可能ファイルをインストールする
STimeoutException	指定したタイムアウト時間が経過するとスローされる例外