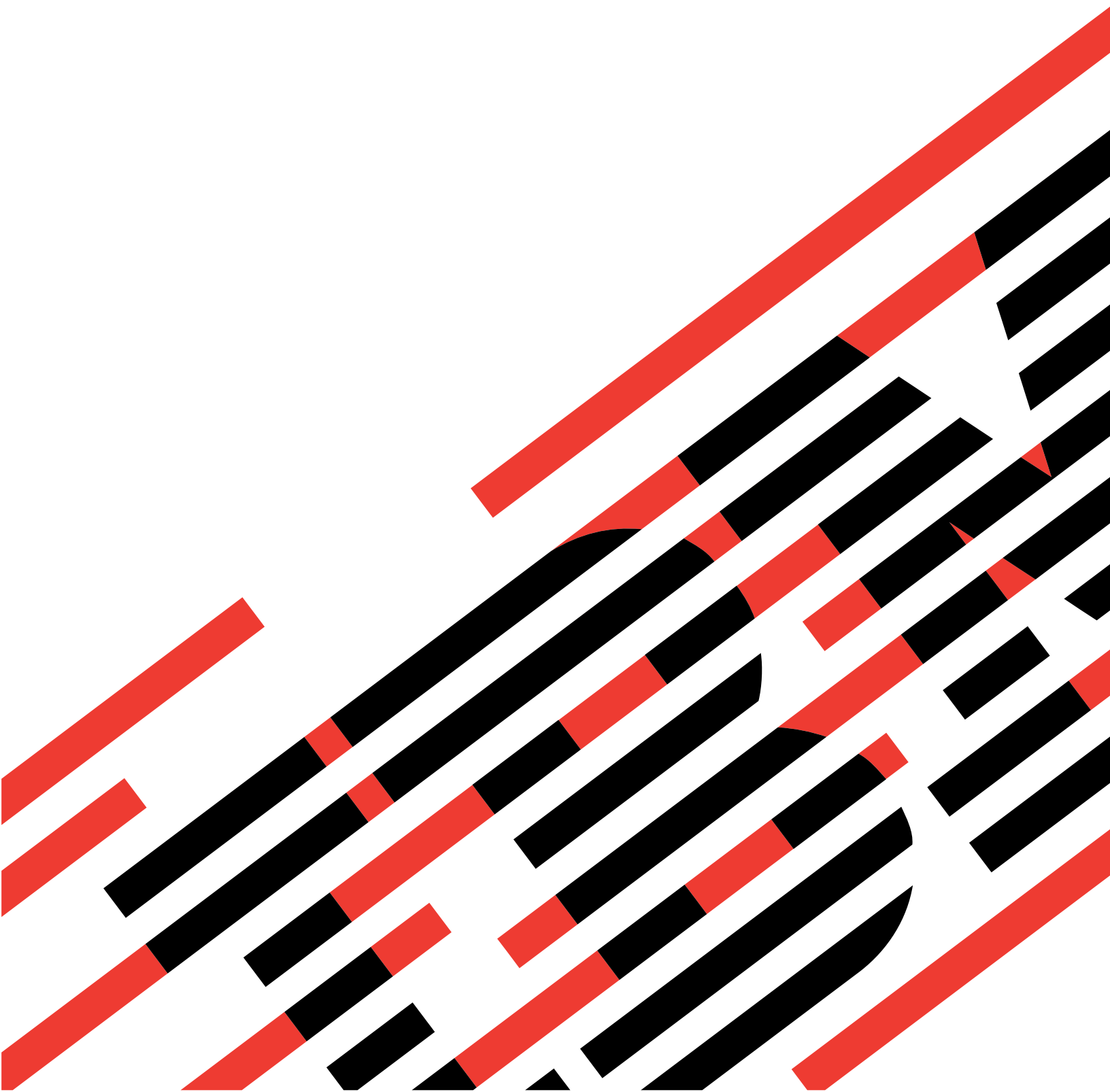


IBM

@server

iSeries

iSeries ナビゲーター・プラグインの開発





@server

iSeries

iSeries ナビゲーター・プラグインの開発

© Copyright International Business Machines Corporation 1998, 2002. All rights reserved.

© Copyright IBM Japan 2002

目次

iSeries ナビゲーター・プラグインの開発	1
iSeries ナビゲーターのプラグイン・サポート	2
プラグインを使ってできること	2
プラグインの仕組み	3
プラグインの要件	5
プラグインの配布	6
iSeries ナビゲーターに対するプラグインの識別	12
サンプル・プラグインの導入と実行	13
サンプル C++ プラグインのセットアップ	13
サンプル Visual Basic プラグインのセットアップ	16
サンプル Java プラグインのセットアップ	18
プラグインのプログラミング・リファレンス	22
iSeries ナビゲーターの構造および C++ プラグインの制御のフロー	22
C++ 用 iSeries ナビゲーター COM インターフェース	24
iSeries ナビゲーター API のリスト	28
iSeries ナビゲーター API に固有の戻りコード	32
iSeries ナビゲーターの構造および Visual Basic プラグインの制御のフロー	33
iSeries ナビゲーターの Visual Basic インターフェース	34
iSeries ナビゲーターの構造および Java プラグインの制御のフロー	35
プラグイン・レジストリー・ファイルのカスタマイズ	36

iSeries ナビゲーター・プラグインの開発

iSeries サーバーの管理タスクとクライアント/サーバー・プログラムを、単一のアプリケーション環境に統合できればよいと思いませんか。iSeries ナビゲーターのプラグイン・フィーチャーを使用すると、まさにそれが可能になります。プラグインを使用することにより、C++、Visual Basic (VB)、または Java で作成されたサード・パーティーのアプリケーションと専用の機能を、iSeries ナビゲーターのインターフェースに統合することができます。この解説を利用して、プラグインとは何か、プラグインを作成してカスタマイズするにはどうすればよいか、プラグインをユーザーに配布するにはどうすればよいか、を学んでください。

プラグインについて学ぶ:

iSeries ナビゲーターのプラグイン・サポート

プラグインを計画するにあたっては、プラグインとは何か、プラグインを使って何ができるか、プラグインをユーザーに配布するにはどうすればよいかを学習してください。

サンプル・プラグインの導入と実行

Programmer's Toolkit は、サンプル・プラグインをダウンロードして実行する際に役立ちます。これらのサンプルを使用すると、iSeries ナビゲーターのプラグイン・サポートを学ぶことができます。また多くの開発者たちは、これらのサンプルをベースとして使用し、それに独自の変更を加えています。

プラグインの開発:

プラグインのプログラミング・リファレンス

プラグインのアーキテクチャーの各タイプおよび iSeries ナビゲーター内の制御のフローに関する情報を参照してください。このトピックにも、API のリスト、戻りコード、および C++ プラグインのための ActiveX と COM 情報へのリンクが、Java プラグインに関連するインターフェースとクラスへのリンクと共に含まれています。

プラグインの配布

iSeries Access の選択セットアップ・フィーチャーを使用すると、作成したプラグインをエンド・ユーザーに簡単に配布することができます。このセクションを利用して、iSeries ナビゲーターに新しいプラグインを識別させる方法、および新しいプラグインをどこに導入するかを学んでください。

コードについての特記事項

このトピックにはプログラミングの例が含まれています。

IBM は、お客様に、すべてのプログラムをサンプルとして使用することができる非独占的な使用権を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

すべてのサンプル・コードは、例として示す目的でのみ、IBM により提供されます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

ここに含まれるすべてのプログラムは、現存するままの状態を提供され、いかなる保証条件も適用されません。不侵害、商品性、特定目的適合性に関する黙示の保証の適用も一切ありません。

iSeries ナビゲーターのプラグイン・サポート

iSeries ナビゲーターのプラグイン・サポートでは、独自の機能とアプリケーションを単一のユーザー・インターフェース、つまり iSeries ナビゲーターに統合する、便利な方法を提供しています。これらの新機能やアプリケーションは、単純なものから複雑なものまで多岐にわたります。プラグインが提供する特定の新しい機能の内容にかかわらず、それを iSeries ナビゲーターに統合することにより、多大な恩恵を受けることができます。例えば、共通のシステム・タスクを iSeries ナビゲーターの単一の場所にバンドルすることにより、共通の管理と操作機能を大幅に単純化することができます。また、iSeries ナビゲーターの GUI インターフェースにより、統合化された機能を簡単に実行でき、しかも難しいスキルは必要ありません。

以下のトピックに関して精通することによって、プラグインの計画に役立てることができます。

- プラグインを使ってできること
プラグインで追加できる新しい機能を学習します。
- プラグインの仕組み
Java プラグインの例を検討して、プラグインの仕組みについて学びます。
- プラグインの要件
プラグインは、C++、VB、または Java で開発できます。このトピックでは、言語ごとに固有の要件を解説します。
- プラグインの配布
新しいプラグインは、iSeries 管理サーバー上に配置することにより、エンド・ユーザーに容易に配布することができます。iSeries Access for Windows の選択セットアップは、この新しいプラグインを検出し、クライアント PC に導入します。

プラグインを使ってできること

プラグインは、iSeries ナビゲーターが特定のユーザー・アクションに応答して呼び出す、定義済みのクラスとメソッドのセットです。プラグインを使用すると、ツールとアプリケーションを表す iSeries ナビゲーター階層のオブジェクトやフォルダーを追加したり、修正したりすることができます。以下の項目を追加あるいは修正することにより、独自のフォルダーおよびオブジェクトのサポートを完全にカスタマイズすることができます。

コンテキスト・メニュー

コンテキスト・メニューは、アプリケーションの立ち上げ、新しいダイアログの表示、および振る舞いの追加または変更に使います。

プロパティ・ページ

プロパティ・ページは、カスタマイズされた属性 (例えば、追加のセキュリティー設定など) をサポートするために使います。プロパティ・シートを持つものであれば、どのオブジェクトまたはフォルダーにもプロパティ・ページを追加することができます。

ツールバー

ツールバーとボタンを完全にカスタマイズすることができます。

カスタム・フォルダーおよびオブジェクト

カスタマイズした独自のフォルダーおよびオブジェクトを、 iSeries ナビゲーターのツリー階層に追加することができます。

プラグインの仕組み

Java プラグインが iSeries ナビゲーターのツリーに新規コンテナを追加する仕組みを以下に示します。

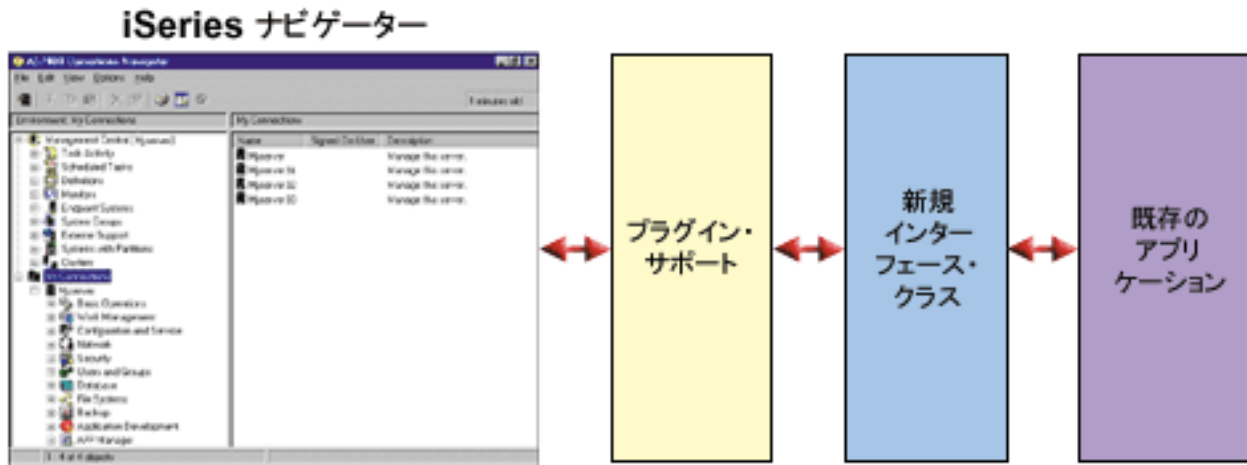
iSeries ナビゲーターは、Windows レジストリーに対する新規プラグインを識別すると、その新規プラグインを検出して、新しい構成に導入します。その後、新規コンテナが iSeries ナビゲーター階層に表示されます。ユーザーがコンテナを選択すると、プラグインの Java コードが呼び出され、コンテナの内容（この例では、ユーザーの省略時メッセージ待ち行列にあるメッセージのリスト）が取得されます。

iSeries ナビゲーター・ダイアログ — メッセージ待ち行列内のメッセージ



iSeries ナビゲーターは、Java インターフェース ListManager に定義されているメソッドを呼び出すことにより、Java プラグインと通信します。このインターフェースにより、Java アプリケーションはナビゲーター・ツリーおよびリスト・ビューにリスト・データを提供します。アプリケーションを iSeries ナビゲーターに組み込むには、このインターフェースをインプリメントする新規の Java クラスを作成します。新規クラスのメソッドは、以下のように既存の Java アプリケーションを呼び出して、リスト・データを取得します。

iSeries ナビゲーターがアプリケーションを呼び出してリスト・データを取得する



ユーザーのいずれかのオブジェクト上で、アクションを実行するものとします。以下の図には、ユーザーがメッセージ・オブジェクト上で、右クリックしてコンテキスト・メニューを表示する様子を示しています。

iSeries ナビゲーター・オブジェクトのコンテキスト・メニュー



iSeries ナビゲーターは、もう 1 つの Java インターフェース `ActionsManager` の定義済みのメソッドを呼び出します。このインターフェースは、メッセージ・オブジェクトでサポートされているメニュー項目のリストを取得します。この場合も、このインターフェースをインプリメントする新規の Java クラスを作成します。このようにして、iSeries ナビゲーターを介して、アプリケーションの特殊機能をユーザーが使用できるようにします。ユーザーがメニュー項目を選択すると、ナビゲーターは、別の `ActionsManager` メソッドを呼び出してアクションを実行します。ユーザーの `ActionsManager` インプリメンテーションは、既存の

Java アプリケーションを呼び出します。これにより、確認ダイアログや、より複雑なユーザー・インターフェース・パネルが表示され、ユーザーが特殊タスクを実行することができます。iSeries ナビゲーターのユーザー・インターフェースは、ユーザーが iSeries サーバー資源のリストを使用して作業し、その資源に対してアクションを実行できるようにすることが目的です。プラグイン機能のアーキテクチャーは、階層内のオブジェクトのリストを処理するためのインターフェースを定義するとともに、これらのオブジェクトへのアクションも定義することにより、このユーザー・インターフェースの設計を反映したものになっています。3 番目のインターフェース DropTargetManager は、ドラッグ・アンド・ドロップ操作を扱います。

プラグインの要件

iSeries ナビゲーターのプラグイン要件は、使用するプログラム言語によって異なります。ただし、すべてのプラグインに、少なくとも iSeries Access for Windows または Client Access Express 95/NT の V3R1M3 と、OS/400 の V4R4 が必要です。Visual Basic および Java プラグインの場合は、Client Access Express V4R4 以上が必要になります。

C++ プラグイン

Microsoft の Visual C++ プログラム言語を使用してプラグインを開発する場合は、バージョン 4.2 以降で作成してください。

C++ プラグインには、以下の iSeries ナビゲーター API も必要です。

ヘッダー・ファイル	インポート・ライブラリー	ダイナミック・リンク・ライブラリー
cwbun.	cwbun.lib	cwbun.dll
cwbunpla.h (アプリケーション管理 API)	cwbapi.lib	cwbunpla.dll

Java プラグイン

Java プラグインは、IBM runtime for Windows(R)、Java Technology Edition で動作します。以下の表に、iSeries Access for Windows と一緒に導入される Java のバージョンを示します。

リリース	JRE	Swing	JavaHelp
V5R2	1.3.1	N/A	1.1.1
V5R1	1.3.0	N/A	1.1.1
V4R5	1.1.8	1.1	N/A
V4R4	1.1.7	1.0.3	N/A

すべての Java プラグインには、そのプラグインに関する特定の情報を含む Windows の資源 DLL が必要です。この DLL を使用すると、iSeries ナビゲーターは、プラグインのインプリメンテーションをロードしなくても、ナビゲーターのオブジェクト階層に機能を表示することができます。サンプルの資源 DLL は、Microsoft Visual C++ バージョン 4.2 を使用して作成されたものですが、Windows の資源のコンパイラとリンクをサポートしている C コンパイラであれば、どれを使用してもかまいません。

iSeries ナビゲーターには、デバッグ支援として、Java コンソールが用意されています。Java コンソールを活動化するには、必要なコンソール標識を Windows のレジストリーに書き込むためのレジストリー・ファイルを選択します。コンソールが活動化されると、JIT コンパイラがオフになり、ソース・コードの行番号がスタック追跡に表示され、ナビゲーターの Java インフラストラクチャーで発生する例外はすべてメッセージ・ボックスに表示されます。iSeries Access for Windows Toolkit に収められているサンプル Java プラグインには、コンソールを活動化、あるいは非活動化するためのレジストリー・ファイルが付属しています。

サンプルのユーザー・インターフェースは、Toolbox for Java コンポーネントに含まれている Graphical Toolbox for Java を使用して開発されたものです。この Toolbox は、オプションとして導入可能な、iSeries Access for Windows のコンポーネントです。iSeries Access for Windows 製品の初期導入時に一緒に導入するか、または後で iSeries Access for Windows 選択セットアップ・プログラムを使用して選択して導入することができます。

Visual Basic プラグイン

Visual Basic プラグインは、Visual Basic のバージョン 5.0 のランタイム環境で動作します。

プラグインの配布

プラグインのコードを iSeries ナビゲーター・ユーザーに配布するには、そのコードを OS/400 アプリケーションに組み込みます。アプリケーションの導入プログラムは、プラグインのコード・バイナリー、レジストリー・ファイル、および変換可能な資源を、iSeries サーバーの統合ファイル・システムのフォルダーに書き込みます。このプロセスが完了すると、エンド・ユーザーは、iSeries Access の選択セットアップ・プログラムを使用して iSeries Access for Window フォルダーからプラグインを取得することができます (iSeries NetServer のマップ済みネットワーク・ドライブを使用します)。選択セットアップでは、プラグインのコードをユーザーのマシンにコピーし、PC の言語設定に応じて、適切な変換可能資源をダウンロードして、レジストリー・ファイルを実行してプラグインのレジストリー情報を Windows レジストリーに書き込みます。初期導入されていない場合は、カスタム・オプションを使用して初期導入時にプラグインをインストールすることもできます。

プラグインのタイプ	導入ディレクトリー	組み込まれるファイル
C++	/QIBM/USERDATA/GUIPLUGIN/ <vendor>.<component> または /QIBM/USERDATA/OpNavPlugin/ <vendor>.<component> (iSeries Access を使わない導入を防ぐため)	<ul style="list-style-type: none"> • プラグインのレジストリー・ファイル • プラグインの iSeries Access for Windows セットアップ・ファイル • プラグインの ActiveX サーバー DLL、および関連コード DLL
Java	/QIBM/USERDATA/OpNavPlugin/ <vendor>.<component> (Java プラグインには iSeries Access が必要)	<ul style="list-style-type: none"> • プラグインのレジストリー・ファイル • プラグインの iSeries Access for Windows セットアップ・ファイル • Java JAR ファイルには、すべての Java クラス、HTML、.gif、PDML、PCML、およびシリアライゼーション・ファイルが入っている。
Visual Basic	/QIBM/USERDATA/OpNavPlugin/ <vendor>.<component> (VB プラグインには iSeries Access が必要)	<ul style="list-style-type: none"> • プラグインのレジストリー・ファイル • プラグインの iSeries Access for Windows セットアップ・ファイル • プラグインの ActiveX サーバー DLL、および関連コード DLL

注:<vendor>.<component> サブディレクトリーは、レジストリー・ファイルに指定されたものと同じでなければなりません。

また、すべてのプラグインは、<vendor>.<component> サブディレクトリ下に最低 1 つのディレクトリー MRI29XX を作成する必要があります (XX はサポートされる言語)。例えば、MRI2924 (英語) ディレクトリーを作成します。このディレクトリーには、以下の項目についての適切な各国語バージョンが含まれなければなりません。

- プラグインの資源 DLL
- プラグインのヘルプ・ファイル
- プラグインの MRI セットアップ・ファイル

プラグインのアップグレードまたは導入解除

ユーザーが新規プラグインを導入した後、後日アップグレードを実行したり、バグ・フィックスを適用したりすることがあるかもしれません。iSeries サーバー上でコードがアップグレードされると、iSeries Access のバージョン・チェック・プログラムがこのプロセスを検出して、アップデートをエンド・ユーザーのマシンに自動的にダウンロードします。iSeries Access には導入解除サポートも用意されており、ユーザーは、いつでもマシンからプラグインを完全に除去することができます。ユーザーは、iSeries サーバーの iSeries ナビゲーター・プロパティで、「プラグイン (Plug-ins)」タブをクリックすることにより、マシンに導入されているプラグインを確認できます。

システム・ポリシーとアプリケーション管理によるプラグインへのアクセスの制限

プラグインに Windows のポリシー・テンプレートを提供した場合、Windows のシステム・ポリシーを使用して、どのネットワーク・ユーザーにそのプラグインの導入を認めるかを制御することもできます。また、iSeries ナビゲーターで iSeries サーバー・ベースのアプリケーション管理サポートを使用することにより、どのユーザーおよびどのユーザー・グループにそのプラグインへのアクセスを認めるかを制御することもできます。

iSeries Access for Windows のセットアップ・ファイル

iSeries Access for Windows のセットアップ・ファイルは、iSeries Access 選択セットアップ・プログラムに、クライアント・ワークステーションに iSeries ナビゲーター・プラグインを導入する場合に必要な情報を提供します。また、このファイルには、プラグインをアップグレードする時期あるいはサービス・リリースを適用する時期を iSeries Access のログイン・サービス検査プログラムが判別する場合に使用する情報も入っています。

ファイル名は SETUP.INI でなければなりません。また、このファイルは、iSeries サーバー上の、プラグインの 1 次ディレクトリーである <vendor>.<component> になければなりません。

ファイルのフォーマットは、Windows の標準構成 (.INI) ファイルのフォーマットに準拠しています。ファイルは、以下の 3 つのパートに分けられます。

- プラグイン情報
- サービス
- クライアント・ワークステーションに導入するファイルの識別を行うためのセクション

例: setup.ini の情報セクション: セットアップ・ファイルの最初のセクション (Plug-in Info) には、以下に示すプラグインに関するグローバル情報が含まれています。

```
[Plugin Info]
Name=Sample plug-in
NameDLL=sampmri.dll
NameResID=128
Description=Sample plug-in description
DescriptionDLL=sampmri.dll
DescriptionResID=129
```

Version=0
 VendorID=IBM.Sample
 SupportExpress=YES
 JavaPlugin=YES

Setup.ini の [Plugin Info] セクションのフィールド	フィールドの説明
Name	プラグインの英語名です。プラグインの導入時に、翻訳名を判別できない場合に、この名前が表示されます。
NameDLL	プラグインの翻訳名を含む資源 DLL の名前です。この DLL は、プラグインの MRI のディレクトリーにあります。
NameResID	MRI DLL 内の翻訳名の資源識別コードです。このフィールドには、プラグインの 1 次レジストリー・キーで定義されている NameID フィールドと同じ値が含まれていなければなりません。
Description	プラグインの英語の説明です。この説明は、プラグインの導入時に、翻訳された説明を判別できない場合に表示されます。
DescriptionDLL	プラグインの翻訳された説明を含む資源 DLL の名前です。この DLL は、プラグインの MRI のディレクトリーにあります。
DescriptionResID	MRI DLL 内の翻訳された説明の資源識別コードです。このフィールドには、プラグインの 1 次レジストリー・キーで定義されている DescriptionID フィールドと同じ値が含まれていなければなりません。
Version	<p>プラグインのリリース・レベルを示す数値です。iSeries Access for Windows のログイン・サービス検査プログラムは、この値を使用して、クライアント・ワークステーション上のプラグインをアップグレードする必要があるかどうかを決定します。プラグインのリリースごとに、この値は、前の値よりも大きい値になります。</p> <p>このバージョン値は、クライアント・ワークステーションに導入済みのプラグインの現行のバージョン値と比較されます。このバージョン値が、すでにクライアント・ワークステーションに存在しているバージョンよりも大きい場合、iSeries Access のログイン・サービス検査プログラムは、このプラグインを新しいバージョンにアップグレードします。</p>
VendorID	プラグインの識別に使用される <VENDOR>.<COMPONENT> という形式のストリングです。このストリングは、iSeries Access のレジストリー・ツリーに、プラグインのレジストリー・キーを作成するために使用されます。VendorID は、プラグインを iSeries サーバーに導入するときのパスの <VENDOR>.<COMPONENT> の部分と同じでなければなりません。
SupportExpress	SupportExpress はオプションです。これは、プラグインが iSeries Access for Windows でサポートされており、正常に動作することを示します。SupportExpress が NO に設定されていたり、存在しない場合、ユーザーがこのプラグインを導入しようとする時、「iSeries ナビゲーターでサポートされないプラグインです (iSeries Navigator Plug-in Not Supported)」というタイトルのダイアログ・ボックスが表示されます。これは、プラグインの導入は可能であっても、iSeries Access でそのプラグインがサポートされていないことを知らせるものです。プラグインを導入するたびにこのメッセージを表示したくない場合は、プラグインが iSeries Access で動作することが分かっている場合、SupportExpress を追加して、それを YES に設定します。
JavaPlugin	JavaPlugin は、これが Java プラグインであるかどうかを示すために使用されます。プラグインが Java プラグインの場合、導入プロセスで特別な処理を行う必要があります。すべての JAR ファイルは、¥PLUGINS¥<VENDOR>.<COMPONENT> ディレクトリーに導入しなければなりません。この値は、特別な処理を行うかどうかを決定するために使用されます。プラグインが Java プラグインであり、この値が NO に設定されているか、存在しない場合は、そのプラグインは導入されても動作しません。

例: setup.ini のサービス・セクション: セットアップ・ファイルの 2 番目のセクション (Service) は、iSeries Access のサービス検査プログラムに、プラグインの新しい修正レベルをクライアント・ワークステーションに適用すべきかどうかを決定するために必要な情報を提供します。

```
[Service]
FixLevel=0
AdditionalSize=0
```

以下のリストに、各フィールドの意味を示します。

Setup.ini の [Service] セクションのフィールド	フィールドの説明
FixLevel	<p>プラグインのサービス・レベルを示す数値です。iSeries Access のサービス検査プログラムは、この値を使用して、プラグインにサービス・リリースを適用する必要があるかどうかを決定します。バージョンのサービス・リリースが改められた場合、この値には、それまでよりも大きな値を使用しなければなりません。</p> <p>この FixLevel 値は、カスタマーのワークステーションに導入済みのプラグインの現行の FixLevel 値と比較されます。この FixLevel 値が、すでにクライアント・ワークステーションに導入されているプラグインの FixLevel 値よりも大きい場合、iSeries Access のサービス検査プログラムは、このプラグインに新しい FixLevel のサービス・リリースを適用します。プラグインが新しいバージョンまたはリリース・レベルにアップグレードされた場合、この値をゼロにリセットしなければなりません。</p>
AdditionalSize	<p>サービス・リリースの適用時にプラグインに追加される、新規または追加の実行可能ファイルを保管するために必要な DASD スペースの量です。導入ではこの値を使用して、ワークステーションにプラグイン用の十分なディスク・スペースがあるかどうかを判別します。</p>

例: setup.ini のファイル識別セクション: セットアップ・ファイルの 3 番目以降のセクションでは、クライアント・ワークステーションに導入するファイルを識別する記述が含まれます。ファイルが示されているセクションは、各ファイルのソースとターゲットの場所を識別します。これらのファイル・セクションは、初期導入時、あるいは新しいバージョンまたはリリース・レベルへのアップグレード時に使用されます。

各ファイル・セクションのファイル項目の書式は `n=file.ext` とします。ここで、`n` はそのセクションのファイルの番号です。この番号は 1 から始まり、すべてのファイルが該当するセクションにリストされるまで 1 ずつ増加します。例えば、以下のようになります。

```
[Base Files]
1=file1.dll
2=file2.dll
3=file3.dll
```

常に、ファイル名とプラグインのみを指定します。ディレクトリー・パスの名前は指定しないでください。ファイル・セクションに項目がない場合、そのセクションは無視されます。

注: Programmer's Toolkit には、C++、Java、および Visual Basic の 3 つの異なるサンプル・プラグイン用のサンプル・セットアップ・ファイルが用意されています。

Setup.ini のセクション	説明
[Base Files]	<p>Client Access の導入ディレクトリー下の ¥PLUGINS¥<VENDOR>.<COMPONENT> にコピーされるファイルです。通常、プラグイン用の ActiveX サーバー DLL (およびそれに関連付けられているコード DLL) はここに指定します。</p> <p>C++ と Visual Basic の場合、ActiveX サーバー DLL (およびそれに関連付けられているコード DLL) をここに指定します。</p> <p>Java の場合、Code JAR ファイル名をここに指定します。</p>
[Shared Files]	Client Access Shared ディレクトリーにコピーされるファイルです。
[System Files]	¥WINDOWS¥SYSTEM または ¥WINNT¥SYSTEM32 ディレクトリーにコピーされるファイルです。
[Core Files]	¥WINDOWS¥SYSTEM または ¥WINNT¥SYSTEM32 ディレクトリーにコピーされるファイルです。これらファイルはレジストリーで使用回数が計数され、除去されることはありません。これらのファイルは、通常は再配布可能ファイルです。
[MRI Files]	iSeries サーバー上のプラグインの MRI ディレクトリーから、ワークステーション上の CLIENT ACCESS¥MRI29XX¥<VENDOR>.<COMPONENT> ディレクトリーにコピーされるファイルです。ここには通常、プラグインのロケール依存資源を指定します。これには、資源の MRI DLL 名を含めます。
[Java MRI29xx] (29xx は、ファイルの NLV フィーチャー・コード)	iSeries サーバー上のプラグインの MRI29xx ディレクトリーから、[Base Files] で指定されたファイルが導入されるのと同じディレクトリーにコピーされる Java ファイルです。ここには通常、JAR MRI29xx の資源を指定します。Java プラグインがサポートする MRI29xx ディレクトリーごとに、それらのファイルをリストした [Java MRI29xx] セクションが必要になります。このセクションを使用するのは、Java プラグインだけです。
[Help files]	iSeries サーバー上のプラグインの MRI ディレクトリーから、ワークステーション上の CLIENT ACCESS¥MRI29XX¥<VENDOR>.<COMPONENT> ディレクトリーにコピーされる .HLP および .CNT ファイルです。これらのファイルへのディレクトリー・パスは、Windows のレジストリーの HKEY_LOCAL_MACHINE¥SOFTWARE¥MICROSOFT¥WINDOWS¥HELP に書き込まれます。
[Registry files]	プラグインに関連付けられている Windows のレジストリー・ファイルです。

[Dependencies]	<p>プラグインを導入する前に導入しておかなければならないサブコンポーネントを定義します。以下で説明する値はオプションです。これらの値が必要になるのは、iSeries ナビゲーターの基本サポートのサブコンポーネントのほかに、別のサブコンポーネントを導入しようプラグインが要求した場合だけです。</p> <p>以下の 2 つの値がサポートされています。 AS400_Operations_Navigator</p> <ul style="list-style-type: none"> この値は、既存システムとの互換性を維持する目的で使用され、iSeries Access V3R2M0 上にプラグインが導入されている場合に導入しなければならないサブコンポーネントを識別します。プラグインが iSeries Access V3R2M0 での実行をサポートしない場合は、この値を指定する必要はありません。 サブコンポーネントは、コンマで区切られたリストで指定されます。1 つのサブコンポーネントは、1 つの番号として指定されます (AS400_Operations_Navigator=3)。CWBUN.H ヘッダー・ファイルには、CWBUN_OPNAV_ という接頭語のついた定数のリストが含まれています。これらの定数は、AS400_Operations_Navigator のコンマ区切りのリストで使用される数値を提供しています。 <p>AS400_Client_Access_Express</p> <ul style="list-style-type: none"> この値を使用して、iSeries Access にプラグインが導入されている場合に導入しなければならないサブコンポーネントを識別します。 サブコンポーネントは、コンマで区切られたリストで指定されます。1 つのサブコンポーネントは、1 つの番号として指定されます (AS400_Client_Access_Express=3)。CWBAD.H ヘッダー・ファイルには、CWBAD_COMP_ という接頭語のついた定数のリストが含まれています。これらの定数は、AS400_Client_Access_Express のコンマ区切りのリストで使用される数値を提供しています。一部の CWBAD_COMP_ コンポーネントは、PC5250 フォントのサブコンポーネントを識別します。以下に示すこれらの定数は、AS400_Client_Access_Express 値では使用しないでください。 <pre>//5250 表示およびプリンター・エミュレーターのサブコンポーネント #define CWBAD_COMP_PC5250_BASE_KOREAN (150) #define CWBAD_COMP_PC5250_PDFPDT_KOREAN (151) #define CWBAD_COMP_PC5250_BASE_SIMPCHIN (152) #define CWBAD_COMP_PC5250_PDFPDT_SIMPCHIN (153) #define CWBAD_COMP_PC5250_BASE_TRADCHIN (154) #define CWBAD_COMP_PC5250_PDFPDT_TRADCHIN (155) #define CWBAD_COMP_PC5250_BASE_STANDARD (156) #define CWBAD_COMP_PC5250_PDFPDT_STANDARD (157) #define CWBAD_COMP_PC5250_FONT_ARABIC (158) #define CWBAD_COMP_PC5250_FONT_BALTIC (159) #define CWBAD_COMP_PC5250_FONT_LATIN2 (160) #define CWBAD_COMP_PC5250_FONT_CYRILLIC (161) #define CWBAD_COMP_PC5250_FONT_GREEK (162) #define CWBAD_COMP_PC5250_FONT_HEBREW (163) #define CWBAD_COMP_PC5250_FONT_LAO (164) #define CWBAD_COMP_PC5250_FONT_THAI (165) #define CWBAD_COMP_PC5250_FONT_TURKISH (166) #define CWBAD_COMP_PC5250_FONT_VIET (167)</pre> <ul style="list-style-type: none"> Client Access V3R2M0 では、この値は無視されます。 <p>注: iSeries Access は、AS400_Client_Access_Express 値がある場合はそれを使用します。その値がなく、AS400_Operations_Navigator 値がある場合は、それを使用します。どちらの値もない場合は、このセクションは無視されます。</p>
----------------	---

[Service Base Files]	iSeries Access の導入ディレクトリーの下 の ¥PLUGINS¥<VENDOR>.<COMPONENT> にコピーされるファイルです。
[Service Shared Files]	iSeries Access の共用ディレクトリーにコピーされるファイルです。
[Service System Files]	¥WINDOWS¥SYSTEM または ¥WINNT¥SYSTEM32 ディレクトリーにコピーされるファイルです。
[Service Core Files]	¥WINDOWS¥SYSTEM または ¥WINNT¥SYSTEM32 ディレクトリーにコピーされるファイルです。これらのファイルはレジストリーで使用回数が計数され、除去されることはありません。また、通常は再配布可能ファイルです。
[Service Registry Files]	プラグインに関連付けられている Windows のレジストリー・ファイルです。

MRI セットアップ・ファイル

MRI セットアップ・ファイルは、iSeries Access の選択セットアップ・プログラムが、iSeries ナビゲーターのプラグインに関連付けられているロケール依存資源をクライアント PC に導入する際に必要となる情報を提供します。

この MRI セットアップ・ファイルは、名前を MRISSETUP.INI にしなければなりません。プラグインがサポートする各国語ごとに、いずれかのバージョンのこのファイルが iSeries サーバー上のサブディレクトリー MRI29XX になければなりません。

ファイルのフォーマットは、Windows の標準構成 (.INI) ファイルのフォーマットに準拠しています。ファイルには、MRI Info セクションだけが含まれています。MRI Info セクションでは、プラグインの MRI のバージョン値が指定されています。プラグインの MRI には、特定の言語のヘルプ・ファイル (.HLP と .CNT) だけでなく、すべての資源 DLL も含まれています。例えば、以下のようになります。

```
[MRI Info]
Version=0
```

iSeries Access の選択セットアップ・プログラムは、プラグインの初期導入時と、バージョン・アップやリリース・アップなどのプラグインのアップグレード時に、MRI のバージョン値を検査します。導入時またはアップグレード時には、このファイル内の MRI のバージョン値は、プラグインの SETUP.INI ファイル内のバージョン値と一致していなければなりません。これらの値が一致していない場合は、MRI ファイルはクライアント PC にコピーされません。Programmer's Toolkit には、サンプル・プラグインとサンプル MRI セットアップ・ファイルが用意されています。

iSeries ナビゲーターに対するプラグインの識別

エンド・ユーザーの Windows デスクトップにプラグイン・ソフトウェアが導入されると、プラグインから Windows レジストリーに情報を提供することで、iSeries がそのプラグインを識別するようにします。レジストリー項目によって、プラグイン・コードの位置を指定し、特殊な iSeries ナビゲーター・インターフェースをインプリメントするクラスを識別します。追加のレジストリー情報を指定することにより、特定の iSeries システムに対しプラグインの機能を活動化するかどうかを iSeries ナビゲーターに判別させることもできます。例えば、プラグインが、OS/400 のある特定のリリースをその最小要件としている場合や、iSeries サーバーに特定の製品が導入されていることを、その動作要件としている場合などが考えられます。

プラグインの導入後、ユーザーが iSeries ナビゲーター階層ツリーの iSeries サーバーをクリックすると、iSeries ナビゲーターは iSeries サーバーを検査して、新規のプラグインをサポートできるかどうかを判別し

ます。プラグインのレジストリー項目に指定されたソフトウェア要件が、iSeries サーバーに導入されたソフトウェアと比較されます。プラグインの要件が満たされている場合、新しい機能が階層ツリーに表示されます。要件が満たされていない場合、レジストリー・ファイルで特に指定されていない限り、その iSeries サーバーにプラグインの機能は表示されません。

サンプル・プラグインの導入と実行

Programmer's Toolkit には、サポートされている各プログラム言語ごとに、サンプル・プラグインが用意されています。これらのサンプルは、プラグインの動作を学習するための優れた教材であり、独自のプラグインを開発する出発点として効果的です。Programmer's Toolkit を導入していない場合は、導入してからでないと、サンプル・プラグインを扱うことはできません。Toolkit は、iSeries Access 選択セットアップによって導入することができます。


- サンプル C++ プラグインのセットアップ
サンプル C++ プラグインをダウンロードし、iSeries ナビゲーターで実行します。
- サンプル Visual Basic プラグインのセットアップ
サンプル Visual Basic プラグインをダウンロードし、iSeries ナビゲーターで実行します。
- サンプル Java プラグインのセットアップ
サンプル Java プラグインをダウンロードし、iSeries ナビゲーターで実行します。


注: サンプル・プラグインを扱う前に、これら 3 つの言語によるプラグインを開発するための、各言語に固有の要件を知っておいてください。

サンプル C++ プラグインのセットアップ

このタスクでは、サンプル ActiveX サーバー DLL をビルドして実行します。このサンプルは、実際に機能する Developer Studio のワークスペースを提供するものです。Developer Studio のワークスペースでは、ブレークポイントを設定して、一般的な iSeries ナビゲーター・プラグインの振る舞いを監視することができます。また、このサンプルを使用すると、プラグイン・コードのコンパイルおよびリンク用に Developer Studio 環境が正しくセットアップされているかどうかを検証することができます。

サンプル C++ プラグインを PC で実行するには、以下のステップを実行してください。

C++ プラグインをダウンロードする	実行可能ファイル <code>cpsmmpq.exe</code>  をダウンロードしてください。このファイルを実行すると、プラグインに関連付けられているファイルがすべて解凍されます。新しいディレクトリー <code>c:\¥MyProject</code> を作成し、解凍したファイルをすべてそこにコピーしてください。別のディレクトリーを作成した場合は、レジストリー・ファイルを修正して、プラグインの正しい場所を指定する必要があります。
--------------------	--

ActiveX サーバー .dll のビルドの準備をする	<ol style="list-style-type: none"> 「MyProject」という名前の新しいディレクトリーをローカルのハード・ディスクに作成します。この例では、ローカル・ドライブとして C: ドライブを想定しています。 注: 新しいディレクトリーが c:\MyProject ではない場合は、レジストリー・ファイルを変更する必要があります。 サンプル・ファイルをすべてこのディレクトリーにコピーします。サンプル・ファイルは、Programmer's Toolkit - iSeries Navigator Plug-ins の Web ページ  からダウンロードできます。 Developer Studio の「ファイル」メニューをオープンし、「ワークスペースを開く (Open Workspace)」を選択します。 「プロジェクトのワークスペースを開く (Open Project Workspace)」ダイアログで、「MyProject」ディレクトリーに移動し、「ファイルの種類:」で「Makefiles (*.mak)」を選択します。 sampext.mak を選択し、「オープン」をクリックします。 「ツール」メニューをオープンし、「オプション...」を選択します。 「ディレクトリー (Directories)」タブで、Client Access の Include ディレクトリーが、組み込みファイルの検索パスの先頭に表示されていることを確認します。 「表示するディレクトリー (Show directories for:)」で、「ライブラリー・ファイル (Library files)」を選択します。Client Access の Lib ディレクトリーが、ライブラリー・ファイルの検索パスの先頭に表示されていることを確認します。 「OK」をクリックして変更を保管し、Developer Studio をクローズした後、再オープンします。Developer Studio で検索パスの変更を強制的にハード・ディスクに保管するには、これ以外の方法はありません。
ActiveX サーバー DLL をビルドする	<ol style="list-style-type: none"> Developer Studio の「ビルド (Build)」メニューをオープンし、「省略時構成の設定... (Set Default Configuration...)」を選択します。 「省略時のプロジェクト構成 (Default Project Configuration)」ダイアログで、sampext の「Win32 デバッグ構成 (Win32 Debug Configuration)」を選択します。 「ビルド (Build)」メニューをオープンし、「すべて再ビルド (Rebuild All)」を選択して DLL をコンパイルし、リンクします。 注: DLL が正常にコンパイルおよびリンクされない場合は、「ビルド (Build)」ウィンドウに表示されているエラー・メッセージをダブルクリックし、エラーの場所を見つけて修正します。次に、「ビルド (Build)」メニューをオープンし、「sampext.dll」を選択して再度ビルドを実行してください。
資源ライブラリーをビルドする	<p>サンプルには、プラグイン用の翻訳可能なテキスト・ストリングとその他のロケール依存資源を含む資源 DLL が含まれています。したがって、この DLL は、独自に作成する必要はありません。プラグインがサポートする言語が 1 つだけであっても、プラグイン・コードは、テキスト・ストリングとロケール固有の資源をこの資源ライブラリーからロードしなければなりません。</p> <p>資源 DLL をビルドするには、以下のステップを実行してください。</p> <ol style="list-style-type: none"> Developer Studio の「ファイル」メニューをオープンして「ワークスペースを開く... (Open Workspace...)」を選択し、「MyProject」ディレクトリーを選択します。 「ファイルの種類:」で「Makefiles (*.mak)」を指定します。 sampmri.mak を選択し、「オープン」をクリックします。 「ビルド (Build)」メニューをオープンし、「すべて再ビルド (Rebuild All)」を選択して DLL をコンパイルし、リンクします。


ActiveX サーバー .dll を登録する	<p>MyProject ディレクトリーの SAMPDBG.REG ファイルに、サンプル・プラグインがあるワークステーション上の場所を iSeries ナビゲーターに認識させるレジストリー・キーが入っています。c:\MyProject と異なるディレクトリーを指定した場合は、以下のステップを実行してください。</p> <ol style="list-style-type: none"> 1. Developer Studio で SAMPDBG.REG ファイルをオープンします (あるいは好みのテキスト・エディターを使用します)。 2. SAMPDBG.REG 内の「c:\MyProject」をすべて「x:\<dir>」で置き換えます。ここで、x は該当するディレクトリーのあるドライブ文字であり、<dir> はそのディレクトリーの名前です。 3. SAMPDBG.REG ファイルを保管します。 4. Windows のエクスプローラーで SAMPDBG.REG ファイルをダブルクリックします。これにより、レジストリー・ファイル内の項目が、使用しているマシンの Windows のレジストリーに書き込まれます。 <p>注: Windows NT の場合、Windows のレジストリーに書き込みを行うには、管理者権限でワークステーションにログインしなければなりません。</p>
デバッガーで iSeries ナビゲーター を実行する	<p>iSeries ナビゲーターを実行してサンプル・プラグインの動作を確認するには、以下のステップを実行してください。</p> <ol style="list-style-type: none"> 1. Developer Studio の「ビルド (Build)」メニューで、「デバッグ (Debug)」->「実行 (Go)」と選択します。 2. プロンプトが表示されるので、ワークステーション上の iSeries Access の導入ディレクトリーにある、iSeries ナビゲーターの実行可能ファイルへの完全修飾パスを入力します。このパスは、C:\PROGRAM FILES\IBM\CLIENT ACCESS\CWBUNNAV.EXE、あるいはこれに類似したものになります。 3. 「OK」をクリックします。iSeries ナビゲーターのメイン・ウィンドウが開きます。 4. 新しいナビゲーターのプラグインを登録したばかりなので、iSeries ナビゲーターのダイアログが表示され、新しいプラグインを見つけるために走査を行うようプロンプトが出されます。 5. 進行状況表示が終了するとダイアログが表示されるので「OK」をクリックします。 6. ナビゲーターのウィンドウが最新表示に更新され、新しいフォルダー (サード・パーティーのサンプル・フォルダー) が、iSeries サーバーの下の、最初に選択された階層に表示されます。これで、iSeries ナビゲーターでプラグインを操作することができ、その動作をデバッガーで確認することができます。

サンプル Visual Basic プラグインのセットアップ

サンプル Visual Basic (VB) プラグインは、OS/400 のライブラリーのリストを提供する iSeries ナビゲーター階層ツリーにフォルダーを追加します。このプラグインでは、それらのライブラリー・オブジェクトにプロパティーとアクションをインプリメントする方法を示します。

導入されるプラグイン・コードの他にも、サンプル・プラグインには、Readme.txt ファイルおよび 2 つのレジストリー・ファイルが含まれています。レジストリー・ファイルの 1 つは開発時に使用するためのもので、もう 1 つは市販バージョンで配布するためのものです。VB プラグインに組み込まれているすべてのファイルの詳細な説明については、『サンプル VB プラグイン・ファイルのディレクトリー』を参照してください。

サンプル VB プラグインを PC で実行するには、以下のステップを実行してください。

VB プラグインをダウンロードする	<p>実行可能ファイル vbopnav.exe  をダウンロードしてください。このファイルを実行すると、プラグインに関連付けられているファイルがすべて解凍されます。新しいディレクトリー c:\¥VBSample を作成し、解凍したファイルをすべてそこにコピーしてください。別のディレクトリーを作成した場合は、レジストリー・ファイルを修正して、プラグインの正しい場所を指定する必要があります。</p>
VB プロジェクトを作成する	<p>Visual Basic で vbsample.vpb をオープンします。参照設定ダイアログで、「IBM Access for Windows ActiveX Object Library」と「iSeries Navigator Visual Basic Plug-in Support」を選択します。</p> <p>注: これらの参照設定項目が「参照設定 (References)」ダイアログに表示されない場合は、「参照」を選択し、iSeries Access for Windows の共用ディレクトリー内の cwbx.dll と cwbunvbi.dll を探してください。IBM iSeries Access の ActiveX Object Library には、サンプル・アプリケーションが iSeries サーバーにリモート・コマンド呼び出しを行うために必要な、OLE オートメーション・オブジェクトが含まれています。iSeries ナビゲーターの Visual Basic プラグイン・サポートには、Visual Basic プラグイン・ディレクトリーを作成するために必要なクラスとインターフェースが含まれています。</p>
ActiveX サーバー DLL をビルドする	<p>Visual Basic の「ファイル」メニューの「dll の作成 (Make)」を選択して DLL をビルドします。コンパイルおよびリンクされない場合は、エラーを見つけて修正し、DLL をもう一度ビルドしてください。</p>
資源ライブラリーをビルドする	<ol style="list-style-type: none"> 1. Microsoft Developer Studio をオープンし、「ファイル」メニューから「ワークスペースを開く (Open Workspace)」を選択し、「VBSample¥win32」ディレクトリーを選択します。 2. 「ファイルの種類:」で、「Makefiles (*.mak)」を選択します。 3. vbsmpmri.mak を選択して「オープン」をクリックします。 4. 「ビルド (Build)」メニューをオープンし、「すべて再ビルド (Rebuild All)」を選択して DLL をコンパイルし、リンクします。 <p>注: この DLL は、独自に作成する必要はありません。サンプルには、プラグイン用の翻訳可能なテキスト・ストリングとその他のロケール依存資源を含む資源 DLL が含まれています。プラグインがサポートする言語が 1 つだけであっても、プラグイン・コードは、テキスト・ストリングとロケール固有の資源をこの資源ライブラリーからロードしなければなりません。</p>
プラグインを登録する	<p>プラグインを登録するには、ファイル vbsmpdbg.reg をダブルクリックします。ディレクトリー c:\¥VBSample を使用しない場合は、レジストリー・ファイルを編集して、ファイル内のすべての「c:\¥¥VBSample¥¥」をプラグイン・コードへの完全修飾パスで置き換えます。パス内の円記号 (¥) は、2 つ続けなければなりません。</p>
iSeries ナビゲーターでプラグインを実行する	<p>iSeries ナビゲーターを開始し、iSeries サーバーの隣にある「+」をクリックしてツリーを展開します。iSeries ナビゲーターは、レジストリーに加えられた変更を検出し、iSeries サーバーが新しいプラグインをサポートできるかどうかを確認するために iSeries サーバーを走査するようプロンプトを出します。走査が完了すると、iSeries ナビゲーターは、新しいプラグインをツリー階層に表示します。</p>

サンプル VB プラグイン・ファイルのディレクトリー

以下に示す表は、V5R2 用のサンプル VB プラグインと共に組み込まれているすべてのファイルについて説明したものです。

Visual Basic のプロジェクト・ファイル	説明
vbsample.vbp	Visual Basic 5.0 のプロジェクト・ファイル

VB のフォーム	説明
authorly.frm	「権限の設定 (Set authority)」フォーム
delete.frm	「削除の確認 (Confirm delete)」フォーム
propsht.frm	「プロパティ・シート (Property Sheet)」フォーム
sysstat.frm	「システム状況」フォーム
wizard.frm	「新しいライブラリーの作成ウィザード (Create new library wizard)」フォーム

VB のモジュール	説明
global.bas	グローバル宣言

VB のクラス・モジュール	説明
actnman.cls	SampleActions Manager クラス
dropman.cls	Sample Drop Target Manager クラス
library.cls	Library クラス
listman.cls	Sample List Manager クラス

VB バイナリー	説明
authorly.frx	「権限の設定 (Set authority)」フォームのバイナリー
delete.frx	「削除の確認 (Confirm delete)」フォームのバイナリー
propsht.frx	「プロパティ・シート (Property Sheet)」フォームのバイナリー
sysstat.frx	「システム状況」フォームのバイナリー
wizard.frx	「新しいライブラリーの作成ウィザード (Create new library wizard)」フォームのバイナリー
vbsample.bin	Vbsample のバイナリー

構成の設定値	説明
mrisetup.ini	プラグインの翻訳可能資源の導入情報

setup.ini	プラグインの実行可能ファイルの導入情報
-----------	---------------------

レジストリー項目	説明
vbsmpdbg.reg	開発時に使用するレジストリー・ファイル
vbsmprls.reg	導入時に iSeries Access が使用するレジストリー・ファイル

資源 DLL の構成のためのファイル	説明
vbsmpmri.mak	Make ファイル
vbsmpmri.rc	RC ファイル
vbsmpres.h	ヘッダー・ファイル


イメージ	説明
compass.bmp	iSeries ナビゲーターのアイコン
lib.ico	
vbsmpflr.ico	オープンおよびクローズ状態の Visual Basic のサンプル・プラグインのフォルダー
vbsmplib.ico	Visual Basic のサンプル・プラグインのライブラリーのアイコン

サンプル Java プラグインのセットアップ

サンプル Java プラグインは、指定された iSeries サーバー上の QUSRSYS のメッセージ待ち行列を操作するものです。最初のプラグインを使用すると、省略時のメッセージ待ち行列 (使用している iSeries のユーザー ID と同じ名前を持つ) にあるメッセージを表示、追加、および削除することができます。2 番目のプラグインは、複数のメッセージ待ち行列をサポートします。最後の 3 番目のプラグインは、メッセージの待ち行列間のドラッグ・アンド・ドロップ機能を追加するものです。

導入されるプラグイン・コードの他にも、サンプル・プラグインには、Java ドキュメント、Readme.txt ファイル、および 2 つのレジストリー・ファイルが含まれています。レジストリー・ファイルの 1 つは開発時に使用するためのもので、もう 1 つは市販バージョンで配布するためのものです。Java プラグインに組み込まれているすべてのファイルの詳細な説明については、『サンプル Java プラグイン・ファイルのディレクトリー』を参照してください。

サンプル Java プラグインをセットアップするには、以下のようにします。

Java プラグインをダウンロードする	実行可能ファイル jvopnav.exe  をダウンロードしてください。このファイルを実行すると、前述のファイルがすべて解凍されます。これらのファイルは、実行可能ファイルにより省略時のディレクトリー <code>jvopnav¥com¥ibm¥as400¥opnav</code> に導入されるようにします。
---------------------	---

iSeries ナビゲーターにプラグインを識別させる	<ol style="list-style-type: none"> 1. jvopnav¥com¥ibm¥as400¥opnav¥MsgQueueSampleX にあるファイル MsgQueueSampleX.reg を編集します。(X=1、2 または 3。これは導入するサンプルによって異なります。) 2. "NLS"="c:¥¥jvopnav¥¥win32¥¥mri¥¥MessageQueuesMRI.dll" および "JavaPath"="c:¥¥jvopnav" の行を見つけます。 3. 「c:¥¥」を、PC の jvopnav ディレクトリーへの完全修飾パスで置き換えます。パス内の円記号 (¥) は、すべて 2 つ続けなければなりません。 4. 変更を保管し、レジストリー・ファイルをダブルクリックします。
サンプル Java プラグインを実行する	<ol style="list-style-type: none"> 1. iSeries ナビゲーターを開始し、iSeries サーバーの隣にある「+」をクリックしてツリーを展開します。 2. iSeries ナビゲーターは、レジストリーに加えられた変更を検出し、iSeries サーバーが新しいプラグインをサポートできるかどうかを確認するために iSeries サーバーを走査するようプロンプトを出します。 3. 「今すぐ走査 (Scan Now)」をクリックします。 4. iSeries ナビゲーターが iSeries サーバーを走査します。走査が完了すると、iSeries サーバーは階層ツリーに新しいフォルダー、Message Queue Sample 1、2 または 3 を表示します。 5. 新しいフォルダーをダブルクリックします。 6. 最初のサンプル・プラグインは、iSeries サーバー上の QUSRSYS の省略時のメッセージ待ち行列の内容を表示します。2 番目と 3 番目のサンプルは、メッセージ待ち行列のリストを表示します。 7. メッセージ待ち行列フォルダーを右クリックして、「新規」->「メッセージ」と選択し、新しいメッセージを追加します。 8. プラグインは PDML ダイアログを表示します。このダイアログからメッセージ・テキストを入力することができます。 9. メッセージを右クリックし、「削除」を選択してメッセージを削除します。この操作は、ツールバーから行うこともできます。 10. 3 番目のサンプル・プラグインを使用している場合は、メッセージを選択し、それを別の待ち行列にドラッグして、ドロップすることができます。 11. この操作により、プラグインは、メッセージを別の待ち行列に移動します。

サンプル Java プラグイン・ファイルのディレクトリー

以下に示す表は、V5R2 用のサンプル Java プラグインと共に組み込まれているすべてのファイルについて説明したものです。詳しくは、プラグインの javadoc 文書をお読みください。javadoc 文書は jvopnav¥com¥ibm¥as400¥opnav¥MsgQueueSample1¥docs ディレクトリーに導入されています。Package-com.ibm.as400.opnav.MsgQueueSample1.html ファイルから読み始めてください。

サンプルのパッケージ名は、com.ibm.as400.opnav.MsgQueueSample1 です。他のパッケージの類似したクラス名と区別するために、すべてのクラス名に「Mq」という接頭部が付けられています。

Java ソース・コード・ファイル (最初のサンプル・プラグイン)	説明
MqMessagesListManager.java	メッセージのリスト用の ListManager です。

Java ソース・コード・ファイル (最初のサンプル・プラグイン)	説明
MqActionsManager.java	プラグイン用のすべてのコンテキスト・メニューを扱う ActionsManager のインプリメンテーションです。
MqMessageQueue.java	メッセージ待ち行列上の、iSeries サーバーのメッセージ・オブジェクトのコレクションです。
MqMessage.java	iSeries サーバーのメッセージを表すオブジェクトです。
MqNewMessageBean.java	「新しいメッセージ (New Message)」ダイアログの UI DataBean のインプリメンテーションです。
MqDeleteMessageBean.java	「削除確認」ダイアログの UI DataBean のインプリメンテーションです。

Java ソース・コード・ファイル (2 番 目のサンプル・プラグイン)	説明
MqListManager.java	プラグインのマスターの ListManager のインプリメンテーションです。
MqMessageQueuesListManager.java	メッセージ待ち行列リスト用のスレーブの ListManager です。
MqMessagesListManager.java	メッセージ・リスト用のスレーブの ListManager です。
MqActionsManager.java	プラグイン用のすべてのコンテキスト・メニューを扱う ActionsManager のインプリメンテーションです。
MqMessageQueueList.java	iSeries サーバーのメッセージ待ち行列のコレクションです。
MqMessageQueue.java	特定の待ち行列上の、iSeries サーバーのメッセージ・オブジェクトのコレクションです。
MqMessage.java	iSeries サーバーのメッセージを表すオブジェクトです。
MqNewMessageBean.java	「新しいメッセージ (New Message)」ダイアログの UI DataBean のインプリメンテーションです。
MqDeleteMessageBean.java	「削除確認」ダイアログの UI DataBean のインプリメンテーションです。

Java ソース・コード・ファイル (3 番 目のサンプル・プラグイン)	説明
MqListManager.java	プラグインのマスターの ListManager のインプリメンテーションです。
MqMessageQueuesListManager.java	メッセージ待ち行列リスト用のスレーブの ListManager です。
MqMessagesListManager.java	メッセージ・リスト用のスレーブの ListManager です。
MqActionsManager.java	プラグイン用のすべてのコンテキスト・メニューを扱う ActionsManager のインプリメンテーションです。
MqDropTargetManager.java	プラグインのドラッグ・アンド・ドロップを扱う DropTargetManager のインプリメンテーションです。
MqMessageQueueList.java	iSeries サーバーのメッセージ待ち行列のコレクションです。
MqMessageQueue.java	特定の待ち行列上の、iSeries サーバーのメッセージ・オブジェクトのコレクションです。
MqMessage.java	iSeries サーバーのメッセージを表すオブジェクトです。

Java ソース・コード・ファイル (3 番目のサンプル・プラグイン)	説明
MqNewMessageBean.java	「新しいメッセージ (New Message)」ダイアログの UI DataBean のインプリメンテーションです。
MqDeleteMessageBean.java	「削除確認」ダイアログの UI DataBean のインプリメンテーションです。

PDML ファイル	説明
MessageQueueGUI.pdml	プラグインのすべての Java UI パネル定義を含んでいます。
MessageQueueGUI.java	関連付けられている Java の資源バンドルです (サブクラス <code>java.util.ListResourceBundle</code>)。

オンライン・ヘルプ・ファイル	説明
IDD_MSGQ_ADD.html	「新しいメッセージ (New Message)」ダイアログのオンライン・ヘルプのスケルトンです。
IDD_MSGQ_CONFIRM_DELETE.html	「削除確認」ダイアログのオンライン・ヘルプのスケルトンです。

シリアライズされたファイル	説明
IDD_MSGQ_ADD.pdml.ser	「新しいメッセージ (New Message)」ダイアログのシリアライズされたパネル定義です。
IDD_MSGQ_CONFIRM_DELETE.pdml.ser	「削除確認」ダイアログのシリアライズされたパネル定義です。 注: MessageQueueGUI.pdml を変更した場合、これらのファイルの名前を変更してください。変更しない場合、変更がパネルに反映されません。

レジストリー項目	説明
MsgQueueSample1.reg MsgQueueSample2.reg MsgQueueSample3.reg	このプラグインが存在していることを iSeries ナビゲーターに対して示す Windows のレジストリー項目であり、プラグインの Java インターフェースのインプリメンテーション・クラスを識別します。

レジストリー項目	説明
MsgQueueSample1install.reg MsgQueueSample2install.reg MsgQueueSample3install.reg	プラグインの市販用バージョンで配布するためのレジストリー・ファイルです。Windows では、このバージョンのレジストリー・ファイルを直接読み取ることはできません。このレジストリー・ファイルには、iSeries Access for Windows の導入ディレクトリーのディレクトリー・パスを表す置換変数が含まれています。ユーザーが iSeries Access の選択セットアップ・プログラムを呼び出して、プラグインを iSeries サーバーから導入するときに、選択セットアップがこのレジストリー・ファイルを読み取って正しいディレクトリー・パスを設定し、ユーザーのマシンのレジストリーに項目を書き込みます。したがって、このファイル内の項目は、開発時に使用されたレジストリー・ファイルと同期している必要があります。

プラグインのプログラミング・リファレンス

iSeries ナビゲーターは、プログラム言語ごとに異なる方法でプラグインを処理します。以下のトピックを利用して、各言語に固有のインターフェースに関する特定の参照情報だけでなく、プラグインのタイプごとに iSeries ナビゲーターの制御のフローについて学ぶことができます。

C++ のリファレンス

- iSeries ナビゲーターの制御のフロー
- COM インターフェース
- API リスト
- 戻りコード

VB のリファレンス

- iSeries ナビゲーターの制御のフロー
- VB インターフェース

Java のリファレンス

- iSeries ナビゲーターの制御のフロー
- Java クラスとインターフェース

各言語に固有の参照情報の他に、各プラグインでは、Windows のレジストリー・ファイルをカスタマイズする必要があります。

プラグインのレジストリー・ファイル

サンプル・プラグインを変更した後は、レジストリー・ファイルにいくつか修正を加える必要があります。このトピックでは、プラグインのタイプごとに、レジストリー・ファイルについて順を追って説明し、変更をいくつかお勧めします。

iSeries ナビゲーターの構造および C++ プラグインの制御のフロー

iSeries ナビゲーター製品の内部アーキテクチャーは、iSeries サーバーの拡張可能な広い基盤に及ぶ操作のインターフェース統合ポイントとして機能することを目的としています。インターフェースの各機能コンポーネントは、ActiveX サーバー DLL としてパッケージされています。iSeries ナビゲーターは、Microsoft の Component Object Model (COM) テクノロジーを使用することにより、現時点のユーザーの要求に応え

る必要のあるコンポーネント・インプリメンテーションのみを活動化します。これにより、始動時にプロダクト全体をロードして Windows 資源の大半を消費することが避けられ、システム全体のパフォーマンスに影響を与えずに済みます。複数のサーバーが、ナビゲーター階層の所定のオブジェクト・タイプに対してメニュー項目やダイアログを追加する要求を登録することができます。

プラグインは、ユーザーのアクションに応答して生成される iSeries ナビゲーターからのメソッド呼び出しに応答することにより機能します。例えば、ユーザーがナビゲーター階層中のオブジェクトを右クリックした場合、ナビゲーターはそのオブジェクトのコンテキスト・メニューを構成して画面にメニューを表示します。ナビゲーターは、選択されたオブジェクト・タイプに対しコンテキスト・メニュー項目を提供するよう登録されている各プラグインを呼び出して、メニュー項目を取得します。

プラグインによって論理的にインプリメントされた関数は、「インターフェース」としてグループ化されません。インターフェースは、iSeries ナビゲーターが特定の機能を実行するために呼び出すことのできるクラスの、論理的に関連するメソッドのセットです。Component Object Model は、一連の純粋な仮想関数を定義する抽象クラスの宣言を行うことにより、C++ でのインターフェースの定義をサポートします。インターフェースを呼び出すクラスは、インプリメンテーション・クラスと呼ばれます。インプリメンテーション・クラスは、抽象クラス定義をサブクラス化して、インターフェースに定義されたそれぞれの関数について C++ コードを提供します。

指定したインプリメンテーション・クラスには、開発者が任意の数のインターフェースをインプリメントすることができます。Developer Studio において ActiveX サーバー DLL 用の新規プロジェクト・ワークスペースを作成する場合、AppWizard がマクロを生成して、インターフェースのインプリメンテーションを容易にします。インターフェースを含むインプリメンテーション・クラスでは、インターフェースをネストされたクラスとして宣言します。ネストされたクラスはメンバー・データを持たず、インターフェースに定義された関数以外の関数は使用しません。通常、そのメソッドは、インプリメンテーション・クラスの関数を呼び出して状態データを取得および設定し、インターフェース仕様により定義された実際の操作を実行します。

C++ 用 iSeries ナビゲーター COM インターフェース

プラグインによって論理的にインプリメントされた関数は、**Component Object Model (COM) インターフェース**としてグループ化されます。インターフェースは、iSeries ナビゲーターが特定の機能を実行するために呼び出すことのできるクラスの、論理的に関連するメソッドのセットです。プラグインは、開発者が提供しようとする機能のタイプに応じ、1 つ以上の COM インターフェースをインプリメントすることができます。例えば、ユーザーがツリー階層中のオブジェクトを右クリックした場合、iSeries ナビゲーターはそのオブジェクトのコンテキスト・メニューを構成して画面にメニューを表示します。ナビゲーターは、選択されたオブジェクト・タイプに対しコンテキスト・メニュー項目を提供するよう登録されている各プラグインを呼び出して、メニュー項目を取得します。プラグインは、ナビゲーターが **IContextMenu インターフェースの QueryContextMenu** メソッドのインプリメンテーションを呼び出すと、ナビゲーターにそのメニュー項目を渡します。

インターフェース	メソッド	説明
IContextMenu	QueryContextMenu	ユーザーがオブジェクトを右クリックした際に、コンテキスト・メニュー項目を提供します。
	GetCommandString	コンテキスト・メニューのヘルプ・テキストを提供し、オブジェクトの状態に応じて、項目が使用可能であるか、ぼかし表示されるかを指示します。
	InvokeCommand	適切なダイアログを表示して、要求されたアクションを実行します。ユーザーが、指定されたメニュー項目をクリックした際に、呼び出されます。
IPropSheetExt	AddPages	標準の Windows API を使用して追加されるプロパティ・ページを作成します。その上で、パラメーターとして渡された関数を呼び出して、ページを追加します。
IDropTarget	DragEnter	ユーザーがドロップ域にオブジェクトをドラッグすると、活動状態になります。
	DragLeave	ユーザーがドロップ域の外にオブジェクトをドラッグすると、活動状態になります。
	DragOver	ユーザーがドロップ域上にある間、活動状態になります。
	Drop	ユーザーがオブジェクトをドロップすると、活動状態になります。
IPersistFile	Load	呼び出されると、選択されたフォルダーの完全修飾オブジェクト名を使って拡張を初期化します。
IA4SortingHierarchyFolder	IsSortingEnabled	フォルダーでソートを使用可能にするかどうかを指定します。
	SortOnColumn	指定されたリスト・ビュー列でリストをソートします。
IA4FilteringHierarchyFolder	GetFilterDescription	現在の組み込み基準のテキスト記述を戻します。
IA4PublicObjectHierarchyFolder	GetPublicListObject	別のプラグインもリスト・オブジェクトを使用できるようにしたい場合にプラグインによってインプリメントされます。
IA4ListObject	GetAttributes	サポートされている属性 ID と、各属性 ID に関連付けられているデータのタイプのリストを戻します。
	GetValue	属性 ID が指定されると、属性の現行値を戻します。

IA4TasksManager	QueryTasks	このオブジェクトでサポートされているタスクのリストを戻します。
	TaskSelected	特定のタスクがユーザーによって選択されたことを IA4TasksManager インプリメンテーションに通知します。

IA4 インターフェース

Microsoft の COM インターフェースの他に、IBM 提供の IA4HierarchyFolder および IA4PropSheetNotify インターフェースがあります。

IA4PropSheetNotify は、メイン・ダイアログが閉じるとサード・パーティーのプロパティ・ページを通知します。また、プラグインに情報を伝えるメソッドも定義します。例えば、表示されているプロパティの iSeries ユーザーがすでに存在するか定義中であるか、または変更を保管するか破棄するかなどといったメソッドを定義します。

IA4HierarchyFolder を使用することにより、プラグインは iSeries ナビゲーター階層に新しいフォルダーを追加することができます。このインターフェースの目的は、ナビゲーター階層に、プラグインにより定義されたフォルダーの内容を表示するときに使用されるリスト・データを提供することです。また、リスト・ビューの列と見出しを指定するためのメソッドや、フォルダーに関連したカスタム・ツールバーを定義するためのメソッドを定義します。

詳しくは、以下のトピックを参照してください。

- IA4HierarchyFolder インターフェース
- IA4HierarchyFolder インターフェース仕様のリスト
- IA4PropSheetNotify インターフェース
- IA4PropSheetNotify インターフェース仕様のリスト

IA4HierarchyFolder インターフェースについて

IA4HierarchyFolder インターフェースは、ISV (independent software vendor) がインプリメントする一連の関数を記述します。IA4HierarchyFolder は、iSeries ナビゲーター階層に、サード・パーティーが新規のフォルダーやオブジェクトを追加できるようにするため IBM が定義した COM (component object model) インターフェースです。Microsoft COM については、Microsoft Web サイトを参照してください。

iSeries ナビゲーター・プログラムは、サード・パーティー・プラグインとの通信が必要になると、IA4HierarchyFolder インターフェースのメソッドを呼び出します。このインターフェースの主な目的は、プラグインにより定義されたフォルダーの内容を表示する際に使用されるリスト・データをナビゲーターに提供することです。インターフェースのメソッドにより、ナビゲーターは特定のサード・パーティー・フォルダーにバインドして、その内容をリストすることができます。詳細ビューの列数とその関連見出しを戻すためのメソッドもあります。また、フォルダーに関連付けるカスタム・ツールバーの仕様を提供するメソッドもあります。

通常、インターフェースのインプリメンテーションはコンパイルされて、ActiveX サーバー DLL (ダイナミック・リンク・ライブラリー) にリンクされます。ナビゲーターは、Windows レジストリー項目によって、新規 DLL の存在を認識します。これらの項目は、ユーザーの PC 上の DLL の位置と、新規フォルダーの挿入先となるオブジェクト階層内の「接合ポイント」を指定します。ナビゲーターは適宜 DLL をロードして、必要に応じて IA4HierarchyFolder インターフェースのメソッドを呼び出します。

ヘッダー・ファイル CWBA4HYF.H には、インターフェース・プロトタイプおよび関連するデータ構造体と戻りコードが宣言されています。

IA4HierarchyFolder インターフェース仕様のリスト

項目識別コード、つまりデータ・エンティティは、Windows ネーム・スペースにあるすべてのフォルダーおよびオブジェクトを識別します。項目識別コードは、階層ファイル・システムにおけるファイル名のようなものです。実際、Windows ネーム・スペースは、デスクトップをルートにする階層ネーム・スペースです。

項目識別コードは、2 バイトのカウント・フィールドと、これに続く可変長のバイナリー・データ構造から構成されます (Microsoft ヘッダー・ファイル SHLOBJ.H の SHITEMID 構造を参照)。この項目識別コードは、オブジェクトを、その親フォルダーとの相対的な関係において一意的に記述するものです。

iSeries ナビゲーターは、以下の所定の構造をとる項目識別コードを使用します。これは IA4HierarchyFolder::ItemAt によって戻さなければなりません。

```
<cb><item name>¥x01<item type>¥x02<item index>
```

ここで、

<cb> は、カウント・フィールド自体を含めた、項目識別コードのバイト単位のサイズです。

<item name> は、ユーザーに表示するために適切な形に変換されたオブジェクト名です。

<item type> は、オブジェクト・タイプを識別する言語依存の固有のストリングです。最小の長さは 4 文字です。

<item index> は、親フォルダー・オブジェクト内でのオブジェクトの位置を識別するゼロ・ベースの索引です。

以下のいずれかの IA4HierarchyFolder 仕様へのリンク

IA4HierarchyFolder::Activate

IA4HierarchyFolder::BindToList

IA4HierarchyFolder::DisplayErrorMessage

IA4HierarchyFolder::GetAttributesOf

IA4HierarchyFolder::GetColumnDataItem

IA4HierarchyFolder::GetColumnInfo

IA4HierarchyFolder::GetIconIndexOf

IA4HierarchyFolder::GetItemCount

IA4HierarchyFolder::GetToolBarInfo

IA4HierarchyFolder::GetListObject

IA4HierarchyFolder::ItemAt

IA4HierarchyFolder::ProcessTerminating

IA4HierarchyFolder::Refresh

IA4PropSheetNotify インターフェースについて

IA4PropSheetNotify インターフェースは IA4HierarchyFolder インターフェースと同じく、ISV (independent software vendor) がインプリメントする一連の関数を記述します。IA4PropSheetNotify は、iSeries サーバー・ユーザーに対して iSeries ナビゲーターが定義するプロパティ・シートに、サード・パーティーが新規プロパティ・ページを追加できるようにするため IBM が定義した COM インターフェースです。

iSeries ナビゲーター・プログラムは、サード・パーティー・プラグインとの通信が必要になると、IA4PropSheetNotify インターフェースのメソッドを呼び出します。このインターフェースの目的は、iSeries ユーザーのメイン・プロパティ・ダイアログが閉じる際に通知を行うことです。通知には、ユーザーが行った変更を保管するか廃棄するかが示されます。IPropSheetExt で使用するものと同じインプリメンテーション・クラスにインターフェースを追加することが意図されています。

インターフェース・インプリメンテーションは、コンパイルされ、プラグインの ActiveX サーバー DLL にリンクされます。ナビゲーターは、Windows レジストリー項目によって、新規 DLL の存在を認識します。これらの項目は、ユーザーの PC 上の DLL の位置を指定します。ナビゲーターは適宜 DLL をロードして、必要に応じて IA4PropSheetNotify インターフェースのメソッドを呼び出します。

CWBA4HYF.H には、インターフェース・プロトタイプおよび関連するデータ構造体と戻りコードの宣言が含まれています。

IA4PropSheetNotify インターフェース仕様のリスト

IA4PropSheetNotify インターフェースは、Users および Groups プロパティ・シートの 1 つに、プロパティ・ページを追加する際に必要な IShellPropSheetExt のインプリメンテーションへの通知を提供します。ユーザーがメインのプロパティ・ダイアログで「OK」をクリックする前に、Users および Groups プロパティ・シートの作成および破棄が頻繁に行われることがあるため、これらの通知が必要になります。IA4PropSheetNotify は、ユーザーが行った変更を保管する必要がある場合に IShellPropSheetExt インプリメンテーションに通知をします。

iSeries ナビゲーターは、iSeries ナビゲーターのプラグインに定義されている通常のレジストリー項目を使用して IA4PropSheetNotify インプリメンテーションに関する情報を取得します。また、Users および

Groups コンポーネントのプロパティ・シート・ハンドラーが登録されると、ページを追加するプロパティ・シートをプラグインが指定できるようにする、特殊なレジストリー値がサポートされます。

以下のいずれかの IA4PropSheetNotify インターフェース仕様へのリンク

- IA4PropSheetNotify::InformUserState
- IA4PropSheetNotify::ApplyChanges
- IA4PropSheetNotify::GetErrorMessage

iSeries ナビゲーター API のリスト

iSeries ナビゲーター API は、プラグインの開発者が特定のタイプのグローバル情報を取得および管理する場合に役立ちます。以下の iSeries ナビゲーター API は、アルファベット順にリストされており、機能ごとに分類されています。

機能	iSeries ナビゲーター API
システム値: この API を使用することにより、プラグイン開発者は iSeries システム値の現行値を取得することができます。	cwbUN_GetSystemValue
システム・ハンドル: これらの API を使用することにより、プラグイン開発者は、指定された iSeries システムに使用する Secure Sockets Layer (SSL) 設定など、接続プロパティを含む iSeries システム・オブジェクト・ハンドルの現行値を取得および解放することができます。	cwbUN_GetSystemHandle cwbUN_ReleaseSystemHandle
ユーザー入力の検証: これらの API を使用することにより、プラグイン開発者は、現行ユーザーが特定の iSeries オブジェクトに対して権限を持っているかどうかを検査することができます。また、この API により、ユーザーが特殊権限を持っているかどうかを判別することができます。	cwbUN_CheckObjectAuthority cwbUN_CheckSpecialAuthority
ユーザー権限の検査: この API を使用することにより、プラグイン開発者は、特定のタイプのユーザー提供ストリングを iSeries サーバーに送信する前に、その妥当性を検査することができます。	cwbUN_CheckAS400Name
ユーザー・プロファイル属性: この API を使用することにより、プラグイン開発者は、現行の iSeries ナビゲーター・ユーザーの任意のユーザー・プロファイル属性の値を取得することができます。	cwbUN_GetUserAttribute

機能	iSeries ナビゲーター API
<p>データ管理: ユーザーが選択したオブジェクトは、サード・パーティー・プラグインに対して 2 つのデータ・エンティティ (項目識別コード・リストおよびオブジェクト名) によって識別されます。データ管理 API を使用することにより、プラグイン開発者はこれらの構造体から情報を取り出すことができます。</p>	<p>cwbUN_ConvertPidlToString</p> <p>cwbUN_GetDisplayNameFromItemId</p> <p>cwbUN_GetDisplayNameFromName</p> <p>cwbUN_GetDisplayPathFromName</p> <p>cwbUN_GetIndexFromItemId</p> <p>cwbUN_GetIndexFromName</p> <p>cwbUN_GetIndexFromPidl</p> <p>cwbUN_GetListObject</p> <p>cwbUN_GetParentFolderNameFromName</p> <p>cwbUN_GetParentFolderPathFromName</p> <p>cwbUN_GetParentFolderPidl</p> <p>cwbUN_GetSystemNameFromName</p> <p>cwbUN_GetSystemNameFromPidl</p> <p>cwbUN_GetTypeFromItemId</p> <p>cwbUN_GetTypeFromName</p> <p>cwbUN_GetTypeFromPidl</p>

機能	iSeries ナビゲーター API
<p>iSeries ナビゲーター・ウィンドウ最新表示: これらの API は、ユーザーに代わって操作を完了した後、プラグインによる要求を実行し、ツリーまたはリスト・ビューを最新表示、またはナビゲーターのステータス・バーにメッセージを表示することができます。</p>	<p>cwbUN_RefreshAll</p> <p>cwbUN_RefreshList</p> <p>cwbUN_RefreshListItems</p> <p>cwbUN_UpdateStatusBar</p>
<p>ODBC 接続: これらの API を使用することにより、プラグイン開発者は、iSeries ナビゲーターのデータベース構成要素により取得された ODBC 接続のハンドルを再利用および終了することができます。</p>	<p>cwbUN_GetODBCConnection</p> <p>cwbUN_EndODBCConnections</p>
<p>iSeries ナビゲーター・アイコンへのアクセス: これらの API を使用することにより、プラグイン開発者は、ナビゲーターのオブジェクト階層に表示されるオブジェクトのアイコン・イメージ・リストにアクセスすることができます。</p>	<p>cwbUN_GetIconIndex</p> <p>cwbUN_GetSharedImageList</p>
<p>アプリケーション管理: これらの API を使用することにより、プラグイン開発者は、ユーザーが管理機能の使用を許可されているかどうかを方針に基づき判断することができます。管理機能とは、iSeries ナビゲーターのアプリケーション管理サブコンポーネントによって使用が制御される機能のことです。</p>	<p>cwbUN_GetAdminValue</p> <p>cwbUN_GetAdminValueEx</p> <p>cwbUN_GetAdminCacheState</p> <p>cwbUN_GetAdminCacheStateEx</p>
<p>導入: この API を使用することにより、プラグイン開発者は、iSeries ナビゲーターのサブコンポーネントが導入されているかどうかを判断することができます。</p>	<p>cwbUN_IsSubcomponentInstalled</p>

機能	iSeries ナビゲーター API
<p>ディレクトリー・サービス: これらの API は、iSeries コンピューター上のディレクトリー・サービス (LDAP) サーバーに関する情報、およびサーバーへの接続機能を提供します。接続機能では、iSeries Access for Windows によりキャッシュに入れられた情報 (識別名やパスワードなど) を使用してサーバーに接続することができます。接続機能は、iSeries Access に付属の LDAP クライアント (LDAP.LIB および LDAP.DLL) を使用するため、ご使用のアプリケーションがこのクライアントを使用している必要があります。</p> <p>文字列を使用する機能は、ANSI および Unicode バージョンで使用可能です。</p> <p>LDAP クライアント API で使用するための識別名およびその他の文字列を戻す機能も、LDAP バージョン 3 サーバー用に UTF-8 バージョンで提供されています。</p>	<p>cwbUN_OpenLocalLdapServer</p> <p>cwbUN_FreeLocalLdapServer</p> <p>cwbUN_GetLdapSvrPort</p> <p>cwbUN_GetLdapSvrSuffixCount</p> <p>cwbUN_GetLdapSvrSuffixName</p> <p>cwbUN_OpenLdapPublishing</p> <p>cwbUN_FreeLdapPublishing</p> <p>cwbUN_GetLdapPublishCount</p> <p>cwbUN_GetLdapPublishType</p> <p>cwbUN_GetLdapPublishServer</p> <p>cwbUN_GetLdapPublishPort</p> <p>cwbUN_GetLdapPublishParentDn</p> <p>cwbUN_OpenLdapBindInfo</p> <p>cwbUN_FreeLdapBindInfo</p> <p>cwbUN_GetLdapServerBindDn</p> <p>cwbUN_BindToLdapServerOnAs400</p> <p>cwbUN_BindToLdapServer</p> <p>cwbUN_NullBindToLdapServerOnAs400</p> <p>cwbUN_NullBindToLdapServer</p>

iSeries ナビゲーター API に固有の戻りコード

6000	CWBUN_BAD_PARAMETER	入力パラメーターが無効です。
6001	CWBUN_FORMAT_NOT_VALID	入力オブジェクト名が無効です。
6002	CWBUN_WINDOW_NOTAVAIL	ビュー・ウィンドウが見つかりません。
6003	CWBUN_INTERNAL_ERROR	処理エラーが発生しました。
6004	CWBUN_USER_NOT_AUTHORIZED	ユーザーには指定された権限がありません。
6005	CWBUN_OBJECT_NOT_FOUND	オブジェクトが iSeries にありません。
6006	CWBUN_INVALID_ITEM_ID	項目識別コード・パラメーターが無効です。
6007	CWBUN_NULL_PARM	NULL パラメーターが渡されました。
6008	CWBUN RTN_STR TOO LONG	ストリングが長すぎるため、戻りバッファーに挿入できません。
6009	CWBUN_INVALID_OBJ_NAME	オブジェクト名パラメーターが無効です。
6010	CWBUN_INVALID_PIDL	PIDL パラメーターが無効です。
6011	CWBUN_NULL_PIDL_RETURNED	親フォルダー PIDL が NULL です。
6012	CWBUN_REFRESH_FAILED	リストの最新表示に失敗しました。
6012	CWBUN_UPDATE_FAILED	ツールバーの更新に失敗しました。
6013	CWBUN_INVALID_NAME_TYPE	iSeries の名前タイプが無効です。
6014	CWBUN_INVALID_AUTH_TYPE	権限タイプが無効です。
6016	CWBUN_HOST_COMM_ERROR	iSeries 通信エラーです。
6017	CWBUN_INVALID_NAME_PARM	名前パラメーターが無効です。
6018	CWBUN_NULL_DISPLAY_STRING	NULL の表示ストリングが戻されました。
6019	CWBUN_GENERAL_FAILURE	一般の iSeries 操作障害です。
6020	CWBUN_INVALID_SYSVAL_ID	無効なシステム値識別コードです。
6021	CWBUN_INVALID_LIST_OBJECT	名前からリスト・オブジェクトを取得できません。
6022	CWBUN_INVALID_IFS_PATH	無効な IFS パスが指定されました。
6023	CWBUN_LANG_NOT_FOUND	拡張機能は、導入されている言語をどれもサポートしていません。
6024	CWBUN_INVALID_USER_ATTR_ID	無効なユーザー属性識別コードです。
6025	CWBUN_GET_USER_ATTR_FAILED	ユーザー属性を取り出すことができません。
6026	CWBUN_INVALID_FLAG_VALUE	無効なフラグ・パラメーター値が設定されました。
6027	CWBUN_CANT_GET_IMAGELIST	アイコン・イメージ・リストを取得できません。

以下に示す戻りコードは、名前検査の API 用です。

6050	CWBUN_NAME_TOO_LONG	名前が長すぎます。
6051	CWBUN_NAME_NULLSTRING	空のストリングです - 文字が含まれていません。
6054	CWBUN_NAME_INVALIDCHAR	無効な文字です。

6055 CWBUN_NAME_STRINGTOOLONG
 スtringが長すぎます。

6056 CWBUN_NAME_MISSINGENDQUOTE
 終了の引用符がありません。

6057 CWBUN_NAME_INVALIDQUOTECHAR
 文字が引用符付きStringに対して無効です。

6058 CWBUN_NAME_ONLYBLANKS
 ブランクのみのStringが検出されました。

6059 CWBUN_NAME_STRINGTOOSHORT
 Stringが短すぎます。

6060 CWBUN_NAME_TOOLONGFORIBM
 Stringには問題がありませんが、IBM コマンドとして長すぎます。

6011 CWBUN_NAME_INVALIDFIRSTCHAR
 最初の文字が無効です。

6020 CWBUN_NAME_CHECK_LAST
 予約済みの範囲です。

以下に示す戻りコードは、LDAP 関連の API 用です。

6101 CWBUN_LDAP_NOT_AVAIL
 LDAP が導入されていないか、構成されていません。

6102 CWBUN_LDAP_BIND_FAILED
 LDAP のバインドに失敗しました。

以下に示す戻りコードは、iSeries 名の検査の API 用です。

1001 CWBUN_NULLSTRING
 Stringが空です。

1004 CWBUN_INVALIDCHAR
 無効な文字です。

1005 CWBUN_STRINGTOOLONG
 Stringが長すぎます。

1006 CWBUN_MISSINGENDQUOTE
 引用符付きStringの終了の引用符がありません。

1007 CWBUN_INVALIDQUOTECHAR
 文字が引用符付きStringに対して無効です。

1008 CWBUN_ONLYBLANKS
 ブランクのみのStringです。

1009 CWBUN_STRINGTOOSHORT
 定義されている最小値よりも短いStringです。

1011 CWBUN_TOOLONGFORIBM
 Stringには問題がありませんが、IBM コマンドとして長すぎます。

1012 CWBUN_INVALIDFIRSTCHAR
 最初の文字が無効です。

1999 CWBUN_GENERALFAILURE
 指定されていないエラーです。

iSeries ナビゲーターの構造および Visual Basic プラグインの制御のフロ

—

iSeries ナビゲーターには、Visual Basic プラグイン用に、ナビゲーターとプラグインのインプリメンテーション間の通信を管理する組み込みの ActiveX サーバーが用意されています。したがって、iSeries ナビゲーターのプラグインを開発する Visual Basic プログラマーは、Microsoft の Visual Basic 5.0 が提供する機能を使用して、自分のプラグイン・クラスを作成し、それらを ActiveX サーバー DLL にパッケージすることができます。

プラグインは、ユーザーのアクションに回答して生成される iSeries ナビゲーターからのメソッド呼び出しに回答することにより機能します。たとえば、ユーザーがナビゲーター階層中のオブジェクトを右クリックした場合、ナビゲーターはそのオブジェクトのコンテキスト・メニューを構成して画面にメニューを表示します。ナビゲーターは、選択されたオブジェクト・タイプに対しコンテキスト・メニュー項目を提供するよう登録されている各プラグインを呼び出して、メニュー項目を取得します。

プラグインによってインプリメントされた関数は、**インターフェース**として論理的にグループ化されます。インターフェースは、iSeries ナビゲーターが特定の機能を実行するために呼び出すことのできるクラスの、論理的に関連するメソッドのセットです。Visual Basic プラグインの場合、以下の 3 つのインターフェースが定義されています。

- ListManager
- ActionsManager
- DropTargetManager

Visual Basic プラグインにおける iSeries ナビゲーター・データ

ナビゲーターがプラグインによりインプリメントされた関数を呼び出す場合、通常、その要求には、ユーザーがナビゲーターのメイン・ウィンドウで選択したオブジェクト (複数の場合もある) が含まれます。プラグインは、どのオブジェクトが選択されているのかを判別できなければなりません。プラグインは、この情報を完全修飾オブジェクト名のリストとして受け取ります。Visual Basic プラグインの場合、選択されたオブジェクトに関する情報を提供する ObjectName クラスが定義されています。オブジェクト階層にフォルダーを追加するプラグインは、フォルダー内の項目を「項目識別コード」の形で iSeries ナビゲーターに戻さなければなりません。Visual Basic プラグインの場合、要求された情報を戻すためにプラグインが使用する ItemIdentifier クラスが定義されています。

Visual Basic プラグインにおける iSeries ナビゲーター・サービス

iSeries ナビゲーターのプラグインは、ナビゲーターのメイン・ウィンドウの動作に影響を与えることがあります。たとえば、ユーザー操作の完了後に、ナビゲーターのリスト・ビューを最新表示にしたり、ナビゲーターの状況域にテキストを挿入したりする必要がある場合です。Visual Basic 環境では、必要なサービスを提供する UIServices というユーティリティ・クラスが提供されています。Visual Basic プラグインは、同様の結果を得るために、cwun.h ヘッダー・ファイル内の C++ API を使用することもできます。このクラスとそのメソッドの詳細については、iSeries ナビゲーターの Visual Basic プラグイン・サポート DLL で提供されるオンライン・ヘルプ (cwunvbi.dll および cwunvbi.hlp) を参照してください。

iSeries ナビゲーターの Visual Basic インターフェース

Visual Basic プラグインは、開発者が iSeries ナビゲーターに提供する予定の機能のタイプに応じて、iSeries ナビゲーターのインターフェース・クラスを 1 つ以上インプリメントしなければなりません。

Programmer's Toolkit に、Visual Basic インターフェース定義のヘルプ・ファイルへのリンクが含まれています。

iSeries ナビゲーターには、以下に示す 3 つのインターフェース・クラスがあります。

- iSeries ナビゲーターの ListManager インターフェース・クラス
- iSeries ナビゲーターの ActionsManager インターフェース・クラス
- iSeries ナビゲーターの DropTargetManager インターフェース・クラス

アプリケーションで、これら 3 つのインターフェース・クラスすべてをインプリメントする必要はありません。

iSeries ナビゲーターの ListManager インターフェース・クラス

ListManager インターフェース・クラスは、iSeries ナビゲーターにおけるデータの提供に使用されます。例えば、リスト・ビューを作成して、それにオブジェクトを設定する必要がある場合、iSeries ナビゲーターが ListManager クラスのメソッドを呼び出して、それを実行します。Visual Basic のサンプル・プラグ

インの listman.cls ファイルには、このクラスの例が提供されています。プラグインが iSeries ナビゲーターのコンポーネント・リストを生成する必要がある場合は、ListManager クラスが必要です。

このクラスとそのメソッドの詳細については、iSeries ナビゲーターの Visual Basic プラグイン・サポート DLL で提供されるオンライン・ヘルプ (cwbunvbi.dll および cwbunvbi.hlp) を参照してください。

iSeries ナビゲーターの ActionsManager インターフェース・クラス

ActionsManager インターフェース・クラスは、コンテキスト・メニューを構成し、コンテキスト・メニューのアクション・コマンドをインプリメントするために使用します。例えば、ユーザーが iSeries ナビゲーターで Visual Basic のリスト・オブジェクトを右クリックすると、ActionsManager インターフェース・クラスの queryActions メソッドが呼び出され、コンテキスト・メニュー項目のストリングが戻されます。Visual Basic のサンプル・プラグインの actnman.cls ファイルには、このクラスの例が提供されています。ActionsManager インターフェース・クラスは、プラグインがサポートする固有のオブジェクト・タイプごとに定義しなければなりません。異なるオブジェクト・タイプに同じ ActionsManager インターフェース・クラスを指定することはできませんが、コード・ロジックで、複数のタイプのオブジェクトでの呼び出しを処理しなければなりません。

このクラスとそのメソッドの詳細については、iSeries ナビゲーターの Visual Basic プラグイン・サポート DLL で提供されるオンライン・ヘルプ (cwbunvbi.dll ファイルおよび cwbunvbi.hlp ファイル) を参照してください。

iSeries ナビゲーターの DropTargetManager インターフェース・クラス

DropTargetManager インターフェース・クラスは、iSeries ナビゲーターでのドラッグ・アンド・ドロップ操作の処理に使用します。ユーザーが Visual Basic のリスト・オブジェクトを選択し、そのオブジェクトに対してマウスでドラッグ・アンド・ドロップを行うと、このクラスのメソッドが呼び出され、ドラッグ・アンド・ドロップ操作が実行されます。

このクラスとそのメソッドの詳細については、iSeries ナビゲーターの Visual Basic プラグイン・サポート DLL で提供されるオンライン・ヘルプ (cwbunvbi.dll および cwbunvbi.hlp) を参照してください。

iSeries ナビゲーターの構造および Java プラグインの制御のフロー

iSeries ナビゲーターは、Java プラグインに対して、ナビゲーターとプラグインの Java クラス間の通信を管理する組み込みの ActiveX サーバーを提供します。このサーバー・コンポーネントは、Java Native Interface (JNI) API を使用してプラグインのオブジェクトを作成し、そのメソッドを呼び出します。したがって、iSeries ナビゲーターのプラグインを開発する Java プログラマーは、ActiveX サーバーのインプリメンテーションの詳細に煩わされることがありません。

ユーザーが iSeries ナビゲーターの Java プラグインを操作する場合には、特定の要求をインプリメントするためのさまざまな登録済み Java インターフェースが呼び出されます。

プラグインは、ユーザーのアクションにตอบสนองして生成される iSeries ナビゲーターからのメソッド呼び出しにตอบสนองすることにより機能します。例えば、ユーザーがナビゲーター階層中のオブジェクトを右クリックした場合、ナビゲーターはそのオブジェクトのコンテキスト・メニューを構成して画面にメニューを表示します。ナビゲーターは、選択されたオブジェクト・タイプに対しコンテキスト・メニュー項目を提供するように登録されている各プラグインを呼び出して、メニュー項目を取得します。

プラグインによって論理的にインプリメントされた関数は、インターフェースとしてグループ化されています。インターフェースは、iSeries ナビゲーターが特定の機能を実行するために呼び出すことができる、クラスの論理的に関連するメソッドのセットです。Java プラグインの場合、以下の 3 つの **Java インターフェース**が定義されています。

- ListManager
- ActionsManager
- DropTargetManager

iSeries ナビゲーターにおけるプラグインの製品アーキテクチャー

iSeries ナビゲーター製品の内部アーキテクチャーは、iSeries サーバーの拡張可能な広い基盤に及ぶ操作のインターフェース統合ポイントとして機能することを目的としています。インターフェースの各機能コンポーネントは、ActiveX サーバーとしてパッケージされています。ナビゲーターは、Windows レジストリー項目によって、特定のサーバー・コンポーネントの存在を認識します。複数のサーバーが要求を登録し、ナビゲーター階層の指定のオブジェクト・タイプに対してメニュー項目やダイアログを追加することができます。

注: iSeries ナビゲーターのユーザーがサード・パーティーの Java プラグインを使用するためには、iSeries Access ユーザーが iSeries Access for Windows バージョン 4 リリース 4 モディフィケーション・レベル 0 を PC に導入していなければなりません。

Java プラグインにおける iSeries ナビゲーター・データ

ナビゲーターがプラグインによりインプリメントされた関数を呼び出す場合、通常、その要求には、ユーザーがナビゲーターのメイン・ウィンドウで選択したオブジェクト (複数の場合もある) が含まれます。プラグインは、どのオブジェクトが選択されているのかを判別できなければなりません。プラグインは、この情報を完全修飾オブジェクト名のリストとして受け取ります。Java プラグインの場合、選択されたオブジェクトに関する情報を提供する `ObjectName` クラスが定義されています。オブジェクト階層にフォルダーを追加するプラグインは、フォルダー内の項目を「項目識別コード」の形で iSeries ナビゲーターに戻さなければなりません。Java プラグインの場合、要求された情報を戻すためにプラグインが使用する `ItemIdentifier` クラスが定義されています。

iSeries ナビゲーターのプラグインは、ナビゲーターのメイン・ウィンドウの動作に影響を与えることがあります。例えば、ユーザー操作の完了後に、ナビゲーターのリスト・ビューを最新表示にしたり、ナビゲーターの状況域にテキストを挿入したりする必要がある場合です。ユーティリティ・クラスは、必須サービスを提供するパッケージ `com.ibm.as400.opnav` に入っています。

プラグイン・レジストリー・ファイルのカスタマイズ

レジストリー・ファイルは、iSeries ナビゲーターのプラグインを識別し、その機能を記述して、プラグインを使用するための前提条件を指定します。サンプル・プラグインには 2 つのレジストリー・ファイル (開発中に使用する Windows 読み取り可能なコピー、および iSeries サーバー上の配布用コピー) が含まれています。独自のプラグインを開発した後は、これらのレジストリー・ファイルにいくつかの修正を加える必要があります。変更の手助けとして、このトピックでは、レジストリー・ファイルの概要および各レジストリー・ファイルの必須セクションの詳細について説明します。

iSeries ナビゲーターは、レジストリー・ファイルを使用してプラグインの有無、要件、および機能を識別します。この情報を提供するため、各プラグインでは少なくとも以下の情報を指定する必要があります。

- プラグインに関するグローバル情報を提供する「1 次」レジストリー・キー。このセクションには、プラグインのベンダーおよびコンポーネント名を指定するプログラム識別コード (ProgID) が含まれ、このセクションで、プラグインが置かれる iSeries サーバー上のフォルダーの名前を付けます。ProgID は、`<vendor>.<component>` (例、`IBM.Sample`) の形式をとります。
- iSeries ナビゲーター階層にあるオブジェクト・タイプのうち、プラグインが追加機能を提供するオブジェクト・タイプを識別するレジストリー・キー。

- プラグインがオブジェクト階層に追加するオブジェクトの各サブツリーのルート用の個別のレジストリー・キー。このキーには、サブツリーのルート・フォルダーに関する情報が含まれます。

レジストリー・ファイルの必須セクションおよび推奨される変更の説明:

- C++ レジストリー・ファイル
- VB レジストリー・ファイル
- Java レジストリー・ファイル

レジストリー・ファイルに関する特別な考慮事項

- C++ におけるプロパティ・シートの処理
- VB におけるプロパティ・シートの処理
- プラグインでの SSL サポート

C++ レジストリー値のカスタマイズ

サンプル・プラグインには、2 つのレジストリー・ファイルが含まれています。1 つは、開発時に使用する、Windows で読み取ることのできるファイル SAMDBG.REG で、もう 1 つは、iSeries サーバー上で配布するためのファイル SAMPRLS.REG です。以下に示す表では、これらのレジストリー・ファイルのセクションについて説明し、独自のプラグインの開発時に奨励される変更内容を示しています。

1 次レジストリー・キー

```

; -----
; プラグインの 1 次レジストリー・キーの定義
; 注: NLS および ServerEntryPoint DLL 名には、
; 修飾ディレクトリー・パスを含めないでください。

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY
plug-in\IBM.Sample]
"Type"="PLUGIN"
"NLS"="sampmri.dll"
"NameID"=dword:00000080
"DescriptionID"=dword:00000081
"MinimumIMPIRelease"="NONE"
"MinimumRISCRRelease"="030701"
"ProductID"="NONE"
"ServerEntryPoint"="sampext.dll"

```

各フィールドの説明と推奨値については、トピック『例: 1 次レジストリー・キー』を参照してください。

データ・サーバーのインプリメンテーション

```

; -----
; このセクションでは、iSeries ナビゲーター階層に追加された新規のフォルダーごとに
; IA4HierarchyFolder のインプリメンテーションを登録します。

[HKEY_CLASSES_ROOT\CLSID\{D09970E1-9073-11d0-82BD-08005AA74F5C}]
@="AS/400 Data Server - Sample Data"

[HKEY_CLASSES_ROOT\CLSID\{D09970E1-9073-11d0-82BD-08005AA74F5C}\InprocServer32]
@="%%CLIENTACCESS%\Plugins\IBM.Sample\sampext.dll"
"ThreadingModel"="Apartment"

```

プラグインがこの階層に複数の新しいフォルダーを追加する場合は、追加フォルダーごとにレジストリー・ファイルのこのセクションを複製し、フォルダーごとに個別の GUID が作成されるようにしなければなりません。プラグインがフォルダーを追加しない場合は、このセクションを除去することができます。

1. 新しいプロジェクト・ワークスペースによって生成される DLL の名前に一致するように、この DLL の名前を変更します。
2. 新しい GUID を生成して、それをコピーします (本節の最後の一括変更のセクションを参照)。
3. レジストリーのこのセクションの CLSID を両方とも、前のステップで生成した GUID のストリングで置き換えます。
4. 独自のファイル SAMPDATA.CPP でストリング "IMPLEMENT_OLECREATE" を探します。
5. コメント行にすでにある CLSID に新しい GUID を貼り付け、次に新しい GUID の 16 進値に一致するよう、IMPLEMENT_OLECREATE マクロ呼び出しの CLSID を変更します。"Sample" という語を新しいフォルダーの名前で置き換えます。
6. 名前変更された SAMPDATA.H と SAMPDATA.CPP のコピーを基にして、新しい GUID ごとに 2 つの新しいソース・ファイルを作成します。

注: ヘッダー・ファイル (.H) には、新しいインプリメンテーション・クラスのクラス宣言が含まれています。インプリメンテーション・ファイル (.CPP) には、新しいフォルダーのデータを取得するコードが含まれていません。

7. 2 つのソース・ファイル内の "CSampleData" というクラス名をすべて、プラグインのコンテキスト内で意味のあるクラス名に置き換えます。
8. 新しいインプリメンテーション・ファイルをプロジェクト・ワークスペースに追加するために、「挿入」メニューをオープンして、「ファイルをプロジェクトに挿入... (Files Into Project...)」を選択します。
9. このように SAMPDATA.CPP を複製すると、すべての新しいフォルダーには最初からライブラリー・オブジェクトが含まれるようになります。

シェル・プラグインのインプリメンテーション

```
-----  
; このセクションでは、シェル・プラグインのインプリメンテーション・クラスを登録します。  
; シェル・プラグインは、階層内の新しいオブジェクトまたは既存のオブジェクトの  
; コンテキスト・メニュー項目またはプロパティ・ページ (あるいはその両方) を追加します。  
  
[HKEY_CLASSES_ROOT\CLSID\{3D7907A1-9080-11d0-82BD-08005AA74F5C}]  
    @="AS/400 Shell plug-ins - Sample"  
  
[HKEY_CLASSES_ROOT\CLSID\{3D7907A1-9080-11d0-82BD-08005AA74F5C}\InprocServer32]  
    @="%CLIENTACCESS%\Plugins\IBM.Sample\sampext.dll"  
    "ThreadingModel"="Apartment"  
  
-----  
; シェル・プラグインの承認 (Windows NT では必須)  
  
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell plug-ins\Approved]  
    "{3D7907A1-9080-11d0-82BD-08005AA74F5C}"="AS/400 Shell plug-ins - Sample"
```

このセクションでは、シェル・プラグインのインプリメンテーション・クラスを登録します。すべての C++ プラグインで、このセクションを使用しなければなりません。

1. 新しいプロジェクト・ワークスペースによって生成された DLL の名前に一致するように、この DLL の名前を変更します。

2. 新しい GUID を生成して、それをコピーします (本節の最後の一括変更のセクションを参照)。
3. 上の例で示した項目内のすべての CLSID を、前のステップで生成した GUID で置き換えます。
4. 独自のファイル EXTINTFC.CPP でストリング "IMPLEMENT_OLECREATE" を探します。
5. コメント行にすでにある CLSID に新しい GUID を貼り付け、次に新しい GUID の 16 進値に一致するように、IMPLEMENT_OLECREATE マクロ呼び出しの CLSID を変更します。

オブジェクトのシェル・プラグインのインプリメンテーション

```

;-----
; 新しいフォルダーとそのオブジェクトのコンテキスト・メニュー・ハンドラーを登録します。

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\shell\Sample\*
\ContextMenuHandlers\{3D7907A1-9080-11d0-82BD-08005AA74F5C}]

;-----
; 新しいフォルダーとそのオブジェクトのプロパティ・シート・ハンドラーを登録します。

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.Sample\shell\Sample\*
\PropertySheetHandlers\{3D7907A1-9080-11d0-82BD-08005AA74F5C}]

;-----
; 新しいフォルダーとそのオブジェクトの「自動最新表示」
; プロパティ・シート・ハンドラーを登録します。
; (これにより、フォルダーで iSeries ナビゲーターの
; 自動最新表示機能を利用することができます。)

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\shell\Sample\*
\PropertySheetHandlers\{5E44E520-2F69-11d1-9318-0004AC946C18}]

;-----
; ドラッグ・アンド・ドロップのコンテキスト・メニュー・ハンドラーを登録します。

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\shell\Sample\*
\DragDropHandlers\{3D7907A1-9080-11d0-82BD-08005AA74F5C}]

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\shell\Sample\*
\DragDropHandlers\{3D7907A1-9080-11d0-82BD-08005AA74F5C}]

;-----
; オブジェクトのドロップを受け入れるためのドロップ・ハンドラーを登録します。

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\shell\Sample\*\DropHandler]
@="{3D7907A1-9080-11d0-82BD-08005AA74F5C}"

;-----
; このプラグインが Secure Socket Layer (SSL) 接続をサポートしていることを登録します。
; 注: "Support Level"=dword:00000001 は、プラグインが SSL をサポートしていることを示しています。
; 注: "Support Level"=dword:00000000 は、プラグインが SSL をサポートしていないことを示しています。

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.Sample\SSL]
"Support Level"=dword:00000001

```

レジストリーの最後のセクションでは、プラグインのインプリメンテーションによって影響を受ける、ナビゲーター階層のオブジェクトを指定します。

1. このセクションの CLSID を、新しい GUID で置き換えます。
2. プラグインが追加のプロパティ・ページを、フォルダーまたはオブジェクトのプロパティ・シートに追加しない場合は、プロパティ・シート・ハンドラーのレジストリー項目を除去します。
3. プラグインをオブジェクトのドロップ・ハンドラーにしない場合は、ドラッグ・アンド・ドロップのコンテキスト・メニュー・ハンドラーとドロップ・ハンドラーのレジストリー項目を除去します。

4. サブキー ¥Sample¥¥ を編集します。詳しくは、『シエルのプラグイン』を参照してください。
5. 独自の EXTINTFC.CPP 内のコードを編集または除去します。このコードでは、サンプルで定義されたオブジェクト・タイプを検査します。
サンプルでは、フォルダー、コンテキスト・メニュー項目、プロパティ・ページ、およびドロップ・アクションを扱っていますが、サンプルの機能をどれだけ残したかによって異なります。

注: サンプル・ファイル EXTINTFC.CPP に基づくコード・ファイルには、コンテキスト・メニュー、プロパティ・ページ、およびドロップ・アクションに対して呼び出されるコードが含まれています。サンプルのコードには、サンプルで定義しているオブジェクト・タイプの検査が含まれています。このファイルを編集し、これらのテストを除去するか、新しい機能を与えようとしているオブジェクト・タイプをチェックするようテストを変更しなければなりません。

一括変更

プラグインのレジストリー・ファイル全体で使用するには、固有の **ProgID** と **GUID** を指定しなければなりません。

プラグインの固有のプログラム識別コード (**ProgID**) を定義する:

ProgID は、<vendor>.<component> テキスト・ストリングに一致していなければなりません。ここで、vendor はプラグインを開発したベンダーの名前を表し、component は提供される機能の説明になります。サンプル・プラグインのストリング "IBM.Sample" の場合は、IBM がベンダーを表し、"Sample" がこのプラグインが提供する機能の説明になります。これはレジストリー・ファイル全体で使用され、プラグインが置かれる、iSeries サーバーとワークステーションの両方のディレクトリーに名前を付けます。レジストリー・ファイル内のすべての "IBM.Sample" を自分の ProgID で置き換えてください。

新しい **GUID** を生成し、レジストリー・ファイル内の **CLSID** 値を置き換える:

iSeries ナビゲーターの C++ プラグインが正常に動作するには、新しいレジストリー・ファイル内の固有の CLSID を、生成する GUID で置き換えなければなりません。

Microsoft の Component Object Model は、16 バイトの 16 進整数を使用して、ActiveX のインプリメンテーション・クラスとインターフェースを一意的に識別します。これらの整数は GUID (大域固有識別コード) として知られます。インプリメンテーション・クラスを識別する GUID を CLSID と言います (「クラス ID」と発音します)。iSeries ナビゲーターは、Windows の ActiveX のランタイム・サポートを使用して、プラグインのコンポーネントをロードし、特定のインターフェースのプラグイン・インプリメンテーションのインスタンスへのポインターを取得します。レジストリー内の CLSID は、特定の ActiveX サーバー DLL にある特定のインプリメンテーション・クラスを一意的に識別します。このマッピングの最初の段階、つまり CLSID からサーバー DLL の名前と場所までは、レジストリー項目によって行われます。したがって、iSeries ナビゲーターのプラグインは、提供するインプリメンテーション・クラスごとに CLSID を登録しなければなりません。

GUID を生成するには、以下のステップに従ってください。

1. Windows のタスクバーから「開始」、「ファイル名を指定して実行 (Run)」と順に選択します。
2. GUIDGEN と入力し、「OK」をクリックします。
3. 「Registry Format」が選択されていることを確認します。
4. 新しい GUID 値を生成する場合は、「New GUID」を選択します。
5. 新しい GUID 値をクリップボードにコピーする場合は、「コピー」を選択します。

例: 1 次レジストリー・キー: 1 次レジストリー・キーは、プラグインのグローバル情報を指定する一連のフィールドを定義します。この情報は必須です。

```

;-----
; プラグインの 1 次レジストリー・キーの定義
; 注: NLS および ServerEntryPoint DLL 名には、修飾ディレクトリー・パスを含めないでください。

```

```

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample]
"Type"="PLUGIN"
"NLS"="sampmri.dll"
"NameID"=dword:00000080
"DescriptionID"=dword:00000081
"MinimumIMPIRelease"="NONE"
"MinimumRISCRlease"="030701"
"ProductID"="NONE"
"ServerEntryPoint"="sampext.dll"

```

1 次レジストリー・キー・フィールド	フィールドの説明
Type	プラグインが iSeries ナビゲーター階層に新しいフォルダーを追加する場合、このフィールドの値は PLUGIN に設定します。それ以外の場合は、 EXT にします。
NLS	プラグインのロケール依存資源を含む資源 DLL の名前を識別します。開発バージョンのレジストリー・ファイルでは、この名前は完全修飾パス名のことがあります。
NameID	iSeries ナビゲーターのユーザー・インターフェースでプラグインを識別するために使用される、資源 DLL 内のテキスト・ストリングの資源識別コードを含むダブルワードです。
DescriptionID	資源 DLL 内のテキスト・ストリングの資源識別コードを含むダブルワードです。この資源 DLL は、iSeries ナビゲーターのユーザー・インターフェースで、プラグインの機能を説明するために使用されます。
MinimumIMPIRelease	<p>プラグインが要求する IMPI ハードウェア上で実行される OS/400 の最小リリースを識別する 6 文字の文字ストリングです。このストリングは、vvrmm という形式でなければなりません。ここで、vv は OS/400 のバージョン、rr はリリース、そして mm はモディフィケーション・レベルです。例えば、プラグインに、バージョン 3 リリース 2 モディフィケーション・レベル 0 が必要な場合は、このフィールドの値は "030200" となります。</p> <p>IMPI ハードウェアで実行される OS/400 のリリース (バージョン 3 リリース 6 よりも前のリリース) をプラグインがどれもサポートしていない場合は、このフィールドの値を "NONE" に設定します。IMPI ハードウェアで実行されるリリースをプラグインがすべてサポートしている場合は、このフィールドの値を "ANY" に設定します。</p>

MinimumRISCRelease	<p>プラグインが要求する RISC ハードウェア上で実行される OS/400 の最小リリースを識別する 6 文字の文字ストリングです。このストリングは、vvrmm という形式でなければなりません。ここで、vv は OS/400 のバージョン、rr はリリース、そして mm はモディフィケーション・レベルです。例えば、プラグインに、バージョン 3 リリース 7 モディフィケーション・レベル 1 が必要な場合は、このフィールドの値は "030701" となります。</p> <p>RISC ハードウェアで実行される OS/400 のリリース (バージョン 3 リリース 6 以降) をプラグインがどれもサポートしていない場合は、このフィールドの値を "NONE" に設定します。RISC ハードウェアで実行されるリリースをプラグインがすべてサポートしている場合は、このフィールドの値を "ANY" に設定します。</p>
ProductID	<p>プラグインが要求する、前提条件の iSeries サーバー・ライセンス・プログラムのプロダクト識別コードを指定する 7 文字の文字ストリングです。iSeries サーバー上に特定のライセンス・プログラムが導入されていることをプラグインが要求しない場合は、このフィールドの値を "NONE" に設定します。</p> <p>同一プロダクトに複数のプロダクト識別コードがある場合は、それらの識別コードをコンマで区切って複数指定することができます。</p>
ServerEntryPoint	<p>サーバーのエントリー・ポイントをインプリメントするコード DLL の名前です。プラグインが特定の iSeries サーバーでサポートされているかどうかを iSeries ナビゲーターが判別する必要があると、このエントリー・ポイントが iSeries ナビゲーターから呼び出されます。プラグインがエントリー・ポイントをインプリメントしていない場合は、このフィールドの値を "NONE" に設定します。開発バージョンのレジストリー・ファイルでは、この名前は完全修飾パス名のことがあります。</p>
JavaPath	<p>プラグインの Java クラスの場所を識別するクラスパスのストリングです。プラグインの開発中は、このフィールドには、クラス・ファイルが存在するディレクトリーのディレクトリー・パスが含まれます。実動バージョンのレジストリー・ファイルでは、iSeries Access for Windows の導入パスを基準とした JAR ファイル名を識別する必要があります。それぞれの前には、導入パスを表す iSeries Access for Windows の置換変数が置かれます。</p>
JavaMRI	<p>プラグインのロケール依存資源を含む JAR ファイルのベース名です。iSeries ナビゲーターは、まず、その名前の接尾部として、該当する Java 言語識別コードと国別識別コードを付けてから、各 JAR ファイルを検索します。指定されたロケールの MRI JAR ファイルが存在しない場合、iSeries ナビゲーターは、基本ロケール (通常は米国英語) の MRI がコード JAR ファイル内にあると見なします。</p>

シェルのプラグイン: これらのレジストリー・キーは、階層内の特定のノードまたはノードのセットを、プラグインが提供する機能のタイプと、その機能をインプリメントするインプリメンテーション・クラスの CLSID にマップします。

ナビゲーター階層の 1 つのオブジェクト・タイプに、任意の数のシェル・プラグインが機能を追加できるよう登録することができます。あるオブジェクト・タイプに機能を提供しているサーバー構成要素が、そのプラグインだけであると想定しないでください。このことは、既存のオブジェクト・タイプだけではなく、プラグインが定義する新しいオブジェクトに対しても適用されます。プラグインが広く使用されている場合は、そのプラグインで定義されているオブジェクト・タイプを、別のベンダーに拡張させないようにすることはできません。

オブジェクト・タイプの識別コード

オブジェクト・タイプの識別コードのペア、つまりサブキー ¥Sample¥¥ は常に、サブキー階層のこのレベルに存在しているものと想定されます。

ペアの最初の識別コードは、ナビゲーター構成要素のルート・フォルダーを指定します。新しいフォルダーを追加するプラグインの場合、この識別コードは、常に、前のセクションで指定されたルート・フォルダーのレジストリー・キー名に一致しなければなりません。既存のオブジェクト・タイプに動作を追加するプラグインの場合は、このサブキーは、通常、iSeries サーバーのコンテナ・オブジェクトの下の、最初のレベルのフォルダーのオブジェクト・タイプでなければなりません。これらのタイプのストリングは、レジストリーの HKEY_CLASSES_ROOT¥IBM.AS400.Network¥TYPES の下に定義されています。

ペアの 2 番目の識別コードは、プラグインが操作対象にする特定のオブジェクト・タイプを識別します。
* が指定されている場合は、親のサブキーで識別されるフォルダー・タイプ、およびそのフォルダーの下の階層に表示されるすべてのフォルダーとオブジェクトに対して、プラグインが呼び出されます。それ以外の場合は、特定のタイプの識別コードを指定しなければなりません。それにより、そのオブジェクト・タイプに対してのみプラグインが呼び出されます。

オブジェクト・タイプの検査

既存のオブジェクト・タイプを検査する場合は、レジストリーの HKEY_CLASSES_ROOT¥IBM.AS400.Network¥TYPES キーの下に定義されている 3 文字の識別コードを使用する必要があります。プラグインで定義された新しいオブジェクト・タイプを検査する場合は、レジストリー・キーを使用してください。プラグインによって定義されているフォルダーに対してデータを提供する場合は、接合点として指定したフォルダーを識別する、またはナビゲーターに戻されるタイプを識別するレジストリー・キーを使用してください。

VB プラグインのレジストリー値のカスタマイズ

サンプル・プラグインには、2 つのレジストリー・ファイルが含まれています。1 つは、開発時に使用する、Windows で読み取ることのできるファイル VBSMPDBG.REG で、もう 1 つは、iSeries サーバー上で配布するためのファイル VBSMPRLS.REG です。以下に示す表では、このレジストリー・ファイルのセクションについて説明し、独自のプラグインの開発時に推奨される変更内容を示しています。

1 次レジストリー・キー

1 次レジストリー・キーは、プラグインのグローバル情報を指定する一連のフィールドを定義します。この情報は必須です。

注: サブキー名は、プラグインの ProgID に一致していなければなりません。

各フィールドの説明については、『例: 1 次レジストリー・キー』を参照してください。

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\
3RD PARTY EXTENSIONS\IBM.VBSample]
"Type"="Plugin"
"NLS"="vbsmpmri.dll"
"NameID"=dword:00000080
"DescriptionID"=dword:00000081
"MinimumMPIRelease"="NONE"
"MinimumRISCRRelease"="040200"
"ProductID"="NONE"
"ServerEntryPoint"="vbsample.dll"
```

推奨する変更:

1. ServerEntryPoint キーの "vbsample.dll" という名前を、プラグインの ActiveX サーバー DLL の名前に一致するように変更します。
2. NLS キーの "vbsmpmri.dll" という名前を、プラグインの C++ MRI の資源 DLL の名前に一致するように変更します。各 Visual Basic プラグインには、固有の C++ MRI DLL 名がなければなりません。

注: これらの変更には、どちらの場合もパスを含めないでください。

新しいフォルダーの登録

このセクションでは、iSeries ナビゲーター階層に追加される新しいフォルダーごとに、Visual Basic プラグインの ListManager クラスのインプリメンテーションを登録します。プラグインが iSeries ナビゲーター階層に新しいフォルダーを追加しない場合は、このセクションを削除して次のタスクに進んでください。

Visual Basic の ListManager クラスは、プラグイン・フォルダーにデータを提供するためのメイン・インターフェイスです。

サンプルでは、Sample Visual Basic Folder を、iSeries ナビゲーター階層の iSeries サーバー・システム名のルート・レベルに配置します。フォルダーをこの階層の別のところに表示する場合は、"Parent" キー値を変更しなければなりません。取り得る値のリストについては、『Parent フィールドの値』を参照してください。

各フィールドの説明とそれが取り得る値については、『例: 新しいフォルダーのレジストリー・キー』を参照してください。

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\
3RD PARTY EXTENSIONS\IBM.VBSample\
folders\SampleVBFolder]
"Parent"="AS4"
"Attributes"=hex:00,01,00,20
"CLSID"="{040606B1-1C19-11d2-AA12-08005AD17735}"
"VBClass"="vbsample.SampleListManager"
"VBInterface"="{0FC5EC72-8E00-11D2-AA9A-08005AD17735}"
"NameID"=dword:00000082
"DescriptionID"=dword:00000083
"DefaultIconIndex"=dword:00000001
"OpenIconIndex"=dword:00000001
```

推奨する変更:

1. レジストリー・ファイル内のすべての "SampleVBFolder" という名前を、フォルダー・オブジェクトを識別する固有の名前に変更します。レジストリー・ファイルで指定された名前は、Visual Basic の、ListManager および ActionsManager クラスで指定されたオブジェクト名に一致していなければなりません。サンプル・プラグインの場合、これらの Visual Basic のソース・ファイルは、**listman.cls** と **actnman.cls** です。
2. VBClass キーの "vbsample.SampleListManager" という名前を、ListManager クラスのプログラム識別コード名に一致するように変更します。例えば、ActiveX サーバー DLL が foo.dll という名前であり、ListManager のインプリメンテーション・クラスが MyListManager の場合、プログラム識別コードは "foo.MyListManager" になります。この名前は大文字小文字を区別します。
3. "VBInterface" キーの値を、ListManager のインプリメンテーション・クラスのインターフェース識別コードに変更します。

VB プラグイン・オブジェクトの登録

レジストリーの最後のセクションでは、Visual Basic プラグインのインプリメンテーションによって影響を受ける、ナビゲーター階層のオブジェクトを指定します。

ActionsManager、ListManager および DropTargetManager クラスのメソッドを実行すると、多くの場合、項目とオブジェクトが渡されます。どのフォルダー・オブジェクトが参照されているのかを判別するには、Windows レジストリーで定義されているオブジェクト・タイプのストリングを使用します。

コンテキスト・メニュー項目を使用して、プロパティ・シートをプラグインに追加することもできます。C++ プラグインに使用されるメカニズムであるプロパティ・シートのレジストリー・キーは使用することができません。自動最新表示プロパティ・シート・ハンドラーを含め、プロパティ・シート・ハンドラーは、Visual Basic プラグインではサポートされていません。

```
;-----  
; 新しいフォルダーとそのオブジェクトのコンテキスト・メニュー・ハンドラーを登録します。
```

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\  
IBM.VBSample\shell\SampleVBFolder**\  
ContextMenuHandlers\{040606B2-1C19-11d2-AA12-08005AD17735}]  
"VBClass"="vbsample.SampleActionsManager"  
"VBInterface"="{0FC5EC7A-8E00-11D2-AA9A-08005AD17735}"
```

```
;-----  
; ドラッグ・アンド・ドロップのコンテキスト・メニュー・ハンドラーを登録します。
```

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\  
IBM.VBSample\shell\SampleVBFolder**\  
DragDropHandlers\{040606B2-1C19-11d2-AA12-08005AD17735}]  
"VBClass"="vbsample.SampleActionsManager"  
"VBInterface"="{0FC5EC7A-8E00-11D2-AA9A-08005AD17735}"
```

```
;-----  
; オブジェクトのドロップを受け入れるためのドロップ・ハンドラーを登録します。
```

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\  
IBM.VBSample\  
shell\SampleVBFolder**\  
DropHandler]  
@="{040606B2-1C19-11d2-AA12-08005AD17735}"  
"VBClass"="vbsample.SampleDropTargetManager"  
"VBInterface"="{0FC5EC6E-8E00-11D2-AA9A-08005AD17735}"
```

推奨する変更:

1. 上に示した項目における CLSID は、常に "{040606B2-1C19-11d2-AA12-08005AD17735}" でなければなりません。
2. "VbClass" キーには、Visual Basic のインプリメンテーション・クラスのプログラム識別コード (ProgID) が含まれます。
3. "VbInterface" キーには、Visual Basic のインプリメンテーション・クラスのインターフェース識別コードが含まれます。
4. プラグインをオブジェクトのドロップ・ハンドラーにしない場合は、ドラッグ・アンド・ドロップのコンテキスト・メニュー・ハンドラーとドロップ・ハンドラーのレジストリー項目を除去します。
5. サブキー ¥SampleVbFolder¥¥ の名前を変更し、フォルダー・オブジェクトを識別する一意のストリングを使用します。この名前は、iSeries ナビゲーターのこのフォルダーに対してアクションが取られたとき、Visual Basic のソースで識別に使用されるオブジェクト・タイプです。
6. ActionsManager インターフェースに基づいて作成されたファイル内で、サンプルによって定義されたオブジェクト・タイプを検査するコードを、新しいフォルダー・オブジェクトの名前を反映するように編集します。サンプル ActionsManager インターフェースは、actnman.cls にあります。

一括変更:

プラグインには、固有のプログラム識別コード、つまり ProgID を定義してください。ProgID は、<vendor>.<component> テキスト・ストリングに一致していなければなりません。ここで、vendor はプラグインを開発したベンダーの名前を表し、component は提供される機能の説明になります。サンプル・プラグインのストリング "IBM.Sample" の場合は、IBM がベンダーを表し、"Sample" がこのプラグインが提供する機能の説明になります。これはレジストリー・ファイル全体で使用され、プラグインが置かれる、iSeries サーバーとワークステーションの両方のディレクトリーに名前を付けます。

"IBM.VBSample" のすべてのインスタンスを、新しい [vender].ProgID で置き換えます。

注: iSeries ナビゲーターには、Java および Visual Basic で作成されたプラグインを管理する、組み込みの ActiveX サーバー DLL が用意されています。したがって、すべての Java および Visual Basic のプラグインは、それぞれ専用の CLSID を登録しています。プログラミング・サンプルで提供されているレジストリー・ファイルには、すでにこれらの定義済みの CLSID が含まれています。

例: 1 次レジストリー・キー: 1 次レジストリー・キーは、プラグインのグローバル情報を指定する一連のフィールドを定義します。この情報は必須です。

```
-----  
; プラグインの 1 次レジストリー・キーの定義  
; 注: NLS および ServerEntryPoint DLL 名には、修飾ディレクトリー・パスを含めないでください。
```

```
[HKEY_CLASSES_ROOT¥IBM.AS400.Network¥3RD PARTY plug-inS¥IBM.Sample]  
"Type"="PLUGIN"  
"NLS"="samppri.dll"  
"NameID"=dword:00000080  
"DescriptionID"=dword:00000081  
"MinimumIMPIRelease"="NONE"  
"MinimumRISRelease"="030701"  
"ProductID"="NONE"  
"ServerEntryPoint"="sampext.dll"
```

1 次レジストリー・キー・フィールド	フィールドの説明
Type	プラグインが iSeries ナビゲーター階層に新しいフォルダーを追加する場合、このフィールドの値は PLUGIN に設定します。それ以外の場合は、EXT にします。
NLS	プラグインのロケール依存資源を含む資源 DLL の名前を識別します。開発バージョンのレジストリー・ファイルでは、この名前は完全修飾パス名のことがあります。
NameID	iSeries ナビゲーターのユーザー・インターフェースでプラグインを識別するために使用される、資源 DLL 内のテキスト・ストリングの資源識別コードを含むダブルワードです。
DescriptionID	資源 DLL 内のテキスト・ストリングの資源識別コードを含むダブルワードです。この資源 DLL は、iSeries ナビゲーターのユーザー・インターフェースで、プラグインの機能を説明するために使用されます。
MinimumIMPIRelease	<p>プラグインが要求する IMPI ハードウェア上で実行される OS/400 の最小リリースを識別する 6 文字の文字ストリングです。このストリングは、vrrmm という形式でなければなりません。ここで、vv は OS/400 のバージョン、rr はリリース、そして mm はモディフィケーション・レベルです。例えば、プラグインに、バージョン 3 リリース 2 モディフィケーション・レベル 0 が必要な場合は、このフィールドの値は "030200" となります。</p> <p>IMPI ハードウェアで実行される OS/400 のリリース (バージョン 3 リリース 6 よりも前のリリース) をプラグインがどれもサポートしていない場合は、このフィールドの値を "NONE" に設定します。IMPI ハードウェアで実行されるリリースをプラグインがすべてサポートしている場合は、このフィールドの値を "ANY" に設定します。</p>
MinimumRISCRelease	<p>プラグインが要求する RISC ハードウェア上で実行される OS/400 の最小リリースを識別する 6 文字の文字ストリングです。このストリングは、vrrmm という形式でなければなりません。ここで、vv は OS/400 のバージョン、rr はリリース、そして mm はモディフィケーション・レベルです。例えば、プラグインに、バージョン 3 リリース 7 モディフィケーション・レベル 1 が必要な場合は、このフィールドの値は "030701" となります。</p> <p>RISC ハードウェアで実行される OS/400 のリリース (バージョン 3 リリース 6 以降) をプラグインがどれもサポートしていない場合は、このフィールドの値を "NONE" に設定します。RISC ハードウェアで実行されるリリースをプラグインがすべてサポートしている場合は、このフィールドの値を "ANY" に設定します。</p>

ProductID	<p>プラグインが要求する、前提条件の iSeries サーバー・ライセンス・プログラムのプロダクト識別コードを指定する 7 文字の文字ストリングです。iSeries サーバー上に特定のライセンス・プログラムが導入されていることをプラグインが要求しない場合は、このフィールドの値を "NONE" に設定します。</p> <p>同一プロダクトに複数のプロダクト識別コードがある場合は、それらの識別コードをコンマで区切って複数指定することができます。</p>
ServerEntryPoint	<p>サーバーのエントリー・ポイントをインプリメントするコード DLL の名前です。プラグインが特定の iSeries サーバーでサポートされているかどうかを iSeries ナビゲーターが判断する必要があると、このエントリー・ポイントが iSeries ナビゲーターから呼び出されます。プラグインがエントリー・ポイントをインプリメントしていない場合は、このフィールドの値を "NONE" に設定します。開発バージョンのレジストリー・ファイルでは、この名前は完全修飾パス名のことがあります。</p>
JavaPath	<p>プラグインの Java クラスの場所を識別するクラスパスのストリングです。プラグインの開発中は、このフィールドには、クラス・ファイルが存在するディレクトリーのディレクトリー・パスが含まれます。実動バージョンのレジストリー・ファイルでは、iSeries Access for Windows の導入パスを基準とした JAR ファイル名を識別する必要があります。それぞれの前には、導入パスを表す iSeries Access for Windows の置換変数が置かれます。</p>
JavaMRI	<p>プラグインのロケール依存資源を含む JAR ファイルのベース名です。iSeries ナビゲーターは、まず、その名前の接尾部として、該当する Java 言語識別コードと国別識別コードを付けてから、各 JAR ファイルを検索します。指定されたロケールの MRI JAR ファイルが存在しない場合、iSeries ナビゲーターは、基本ロケール (通常は米国英語) の MRI がコード JAR ファイル内にあると見なします。</p>

Parent フィールドの値: 追加されるフォルダーの親を識別する、3 文字の識別コードです。以下の識別コードの 1 つを指定できます。

ADF	アプリケーション開発フォルダー
AS4	iSeries サーバー・フォルダー
BKF	バックアップ・フォルダー
BOF	基本操作フォルダー
CFG	構成およびサービス・フォルダー
DBF	データベース・フォルダー
FSF	ファイル・システム・フォルダー
JMF	ジョブ管理フォルダー
MCN	マネージメント・セントラル・フォルダー
MCS	マネージメント・セントラルの構成およびサービス・フォルダー
MDF	マネージメント・セントラル定義フォルダー

MMF マルチメディア・フォルダー
 NSR ネットワーク・サーバー・フォルダー
 NWF ネットワーク・フォルダー
 SCF セキュリティ・フォルダー
 UGF ユーザーおよびグループ・フォルダー

例: 新しいフォルダーのレジストリー・キー: プラグインによってオブジェクト階層に追加されるオブジェクトの各サブツリーのルートに対しては、個別のレジストリー・キーを定義しなければなりません。このキーには、サブツリーのルート・フォルダーに固有の情報が含まれています。

このレジストリー・キーには、少なくとも 4 文字から成る、意味のあるフォルダー名を割り当ててください。

```

;-----
; 新しいフォルダーの登録

[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\folders\Sample]
"Parent"="AS4"
"Attributes"=hex:00,01,00,20
"CLSID"="{D09970E1-9073-11d0-82BD-08005AA74F5C}"
"NameID"=dword:00000082
"DescriptionID"=dword:00000083
"DefaultIconIndex"=dword:00000000
"OpenIconIndex"=dword:00000001
"AdminItem"="QIBM_SAMPLE_SMPFLR"

```

Parent	追加されるフォルダーの親を識別する、3 文字の識別コードです。 有効な値のリストについては、『Parent フィールドの値』を参照してください。
Attributes	標識バイトを逆順にした、フォルダーの属性を含む 4 バイトの 2 進数フィールドです。Microsoft の組み込みファイル SHLOBJ.H 内で IShellFolder::GetAttributesOf メソッドに対して定義されているフォルダーの属性フラグを参照してください。
CLSID	フォルダーの内容を取得するために iSeries ナビゲーターによって呼び出される IA4HierarchyFolder インプリメンテーションの CLSID です。 Java プラグイン の場合は、CLSID は常に 1827A856-9C20-11d1-96C3-00062912C9B2 でなければなりません。 Visual Basic プラグイン の場合は、CLSID は常に 040606B1-1C19-11d2-AA12-08005AD17735} でなければなりません。
JavaClass	フォルダーの内容を取得するために iSeries ナビゲーターによって呼び出される ListManager インプリメンテーションの完全修飾 Java クラス名です。プラグインが Java プラグインではない場合、このフィールドは省略します。
VBClass	フォルダーの内容を取得するために iSeries ナビゲーターによって呼び出される ListManager インプリメンテーション・クラスのプログラム識別コード (ProgID) です。
VBInterface	ListManager インプリメンテーション・クラスのインターフェースの GUID です。
NameID	iSeries ナビゲーター階層に、フォルダーの名前として表示されるストリングの資源 ID を含むダブルワードです。
DescriptionID	iSeries ナビゲーター階層に、フォルダーの説明として表示されるストリングの資源 ID を含むダブルワードです。

DefaultIconIndex	プラグインの NLS 資源 DLL に組み込まれた、iSeries ナビゲーター階層のフォルダー用として表示されるアイコンの索引を含むダブルワードです。これは、資源 DLL に組み込まれたゼロ・ベースの索引であり、アイコンの資源識別コードではありません。索引付けが正常に行われるためには、アイコンの資源識別コードが順番に割り当てられている必要があります。
OpenIconIndex	プラグインの NLS 資源 DLL に組み込まれた、ユーザーがフォルダーを選択したときに iSeries ナビゲーター階層のフォルダー用として必ず表示されるアイコンの索引を含むダブルワードです。
AdminItem	フォルダーへのアクセスを制御するアプリケーション管理機能の機能 ID を含むストリングです。このフィールドを省略した場合、アプリケーション管理機能はフォルダーへのアクセスを管理しません。このフィールドを指定した場合は、これはグループ機能あるいは管理可能機能の機能識別コードでなければなりません。これをプロダクト機能の機能識別コードにはできません。

サンプル Java レジストリー・ファイル

Java で作成されたサンプル・プラグインには、それぞれ専用のレジストリー・ファイルがあります。以下のセクションでは、レジストリー・ファイルの重要な部分を説明し、プラグイン用に適切な項目を作成する方法を示します。例は、説明する機能に適したサンプルに基づいています。

プログラム識別コード (ProgID)

作成したプラグインは、<vendor>.<component> というフォーマットのテキスト・ストリングによって、iSeries ナビゲーターに一意的に識別されます。ここで、vendor はプラグインを開発したベンダーを表し、component は提供される機能の説明になります。以下に示す例のストリング IBM.MsgQueueSample3 の場合は、IBM がベンダーを表し、"MsgQueueSample3" がプラグインが提供する機能の説明になります。このストリングは、プログラム識別コード、つまり ProgID として知られます。プラグインが提供する機能を指定する場合は、これがレジストリー・ファイル全体で使用され、プラグインが置かれる、iSeries サーバーとクライアント・ワークステーションの両方のディレクトリーに名前を付けます。

大域固有識別コード (GUID)

Microsoft の Component Object Model は、16 バイトの 16 進整数を使用して、ActiveX のインプリメンテーション・クラスとインターフェースを一意的に識別します。これらの整数は、大域固有識別コード、つまり GUID として知られます。インプリメンテーション・クラスを識別する GUID は、CLSID (「クラス ID」と発音します) と呼ばれます。

Java で作成された iSeries ナビゲーターの構成要素の場合は、新しい GUID を定義する必要はありません。すべての Java プラグインは、Java プラグインを管理する組み込みの ActiveX サーバー構成要素を指定する標準の GUID のセットを使用します。使用する標準の CLSID は、以下の例のとおりです。

プラグインの 1 次属性の定義:

```
;-----  
; Message Queue Sample 3 の 1 次レジストリー・キーの定義  
  
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY  
EXTENSIONS\IBM.MsgQueueSample3] "Type"="PLUGIN"  
"NLS"="MessageQueuesMRI.dll"  
"NameID"=dword:00000001  
"DescriptionID"=dword:00000002  
"MinimumIMPIRelease"="NONE"  
"MinimumRISCRRelease"="ANY"  
"ProductID"="NONE"  
"ServerEntryPoint"="NONE"  
"JavaPath"="MsgQueueSample3.jar"  
"JavaMRI"="MsgQueueSample3MRI.jar"
```

Type

プラグインが iSeries ナビゲーター階層に新しいフォルダーを追加する場合は、このフィールドの値を PLUGIN に設定します。それ以外の場合は、EXT に設定します。

NLS

プラグインのロケール依存資源を含む資源 DLL の名前を識別します。開発バージョンのレジストリー・ファイルでは、この名前は完全修飾パス名のことがあります。

NameID

iSeries ナビゲーターのユーザー・インターフェースでプラグインを識別するために使用される、資源 DLL 内のテキスト・ストリングの資源 ID を含むダブルワードです。

DescriptionID

資源 DLL 内のテキスト・ストリングの資源 ID を含むダブルワードです。この資源 DLL は、iSeries ナビゲーターのユーザー・インターフェースでプラグインの機能を説明するために使用されます。

MinimumIMPIRelease

プラグインが要求する、IMPI ハードウェア上で実行される OS/400 の最小リリースを識別する 6 文字の文字ストリングです。このストリングは、vvrmm という形式でなければなりません。ここで、vv は OS/400 のバージョン、rr はリリース、そして mm はモディフィケーション・レベルです。例えば、プラグインに、バージョン 3 リリース 2 モディフィケーション・レベル 0 が必要な場合は、このフィールドの値は "030200" となります。

プラグインが、IMPI ハードウェア上で実行される OS/400 のリリース (バージョン 3 リリース 6 よりも前のリリース) をサポートしていない場合は、このフィールドの値を "NONE" に設定します。IMPI ハードウェア上で実行されるリリースをプラグインがすべてサポートしている場合は、このフィールドの値を "ANY" に設定します。

MinimumRISCRRelease

プラグインが要求する、RISC ハードウェア上で実行される OS/400 の最小リリースを識別する 6 文字の文字ストリングです。このストリングは、vvrmm という形式でなければなりません。ここで、vv は OS/400 のバージョン、rr はリリース、そして mm はモディフィケーション・レベルです。例えば、プラグインに、バージョン 3 リリース 7 モディフィケーション・レベル 1 が必要な場合は、このフィールドの値は "030701" となります。

プラグインが、RISC ハードウェアで実行される OS/400 のリリース (バージョン 3 リリース 6 以降) をサポートしていない場合は、このフィールドの値を "NONE" に設定します。RISC ハードウェアで実行されるリリースをプラグインがすべてサポートしている場合は、このフィールドの値を "ANY" に設定します。

ProductID

プラグインが要求する、前提条件の iSeries サーバーのライセンス・プログラムのプロダクト識別コードを指定する 7 文字の文字ストリングです。iSeries サーバー上に特定のライセンス・プログラムが導入されていることをプラグインが要求しない場合は、このフィールドの値を "NONE" に設定します。

同一プロダクトに複数のプロダクト識別コードがある場合は、それらの識別コードをコンマで区切って複数指定することができます。

ServerEntryPoint

サーバーのエントリー・ポイントをインプリメントするコード DLL の名前です。プラグインが特定の iSeries サーバーでサポートされているかどうかを iSeries ナビゲーターが判別する必要が生じると、このエントリー・ポイントが iSeries ナビゲーターから呼び出されます。プラグインがエントリー・ポイントをインプリメントしていない場合は、このフィールドの値を "NONE" に設定します。開発バージョンのレジストリー・ファイルでは、この名前は完全修飾パス名のことがあります。

JavaPath

プラグインの Java クラスの場所を識別するクラスパスのストリングです。プラグインの開発中は、このフィールドには、クラス・ファイルが存在するディレクトリーのディレクトリー・パスが含まれます。実動バージョンのレジストリー・ファイルでは、JAR ファイルを識別する必要があります。JAR ファイル名は、ディレクトリー名で修飾しないでください。iSeries ナビゲーターは、Java VM に渡すクラスパスのストリングを構成する際に、それらのファイル名を自動的に修飾します。

JavaMRI

プラグインのロケール依存資源を含む JAR ファイルのベース名です。iSeries ナビゲーターは、まず、その名前の接尾部として、該当する Java 言語識別コードと国別識別コードを付けてから、各 JAR ファイルを検索します。コード JAR にはベース・ロケール (通常は米国英語) の資源があるため、開発バージョンのレジストリー・ファイルでは、このフィールドに空のストリングを含めることができます。

新しいフォルダーの定義:

```
;-----  
; 新しいフォルダーの登録  
  
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.MsgQueueSample3\Folders\Sample3]  
"Parent"="AS4"  
"Attributes"=hex:00,01,00,a0  
"CLSID"="{1827A856-9C20-11d1-96C3-00062912C9B2}"  
"JavaClass"="com.ibm.as400.opnav.MsgQueueSample3.MqListManager"  
"NameID"=dword:0000000b  
"DescriptionID"=dword:0000000c  
"DefaultIconIndex"=dword:00000001  
"OpenIconIndex"=dword:00000000  
"AdminItem"="QIBM_SAMPLE_SMPFLR"  
"TaskpadNameID"=dword:00000003  
"TaskpadDescriptionID"=dword:00000004
```

Type

プラグインが iSeries ナビゲーター階層に追加する新しいフォルダーごとに、一意の論理タイプが割り当てられます。上の例では、ストリング `Sample3` が、実行時に制御がプラグインに渡されるときに、現在選択されているフォルダーを識別するために使用されるタイプです。

Parent

追加されるフォルダーの親を識別する、3 文字の識別コードです。以下の識別コードの 1 つを指定できます。

ADF	アプリケーション開発フォルダー
AS4	iSeries サーバー・フォルダー
BKF	バックアップ・フォルダー
BOF	基本操作フォルダー
CFG	構成およびサービス・フォルダー
DBF	データベース・フォルダー
FSF	ファイル・システム・フォルダー
JMF	ジョブ管理フォルダー
MCN	マネージメント・セントラル・フォルダー
MCS	マネージメント・セントラル構成およびサービス・フォルダー
MDF	マネージメント・セントラル定義フォルダー
MMN	マネージメント・セントラル・モニター
MST	マネージメント・セントラル・スケジュール済みタスク
MTA	マネージメント・セントラル・タスク・アクティビティ
MXS	マネージメント・セントラル・エクストリーム・サポート
NSR	ネットワーク・サーバー・フォルダー
NWF	ネットワーク・フォルダー
SCF	セキュリティ・フォルダー
UGF	ユーザーおよびグループ・フォルダー

属性

標識バイトを逆順にした、フォルダーの属性を含む 4 バイトの 2 進数フィールドです。Microsoft の組み込みファイル `SHLOBJ.H` 内で `IShellFolder::GetAttributesOf` メソッドに対して定義されているフォルダーの属性フラグを参照してください。フォルダーにタスクパッドがあることを示すには、`0x00000008` を使用してください。

CLSID

フォルダーの内容を取得するために、iSeries ナビゲーターによって呼び出される `IA4HierarchyFolder` インプリメンテーションの CLSID です。Java プラグインの場合、この CLSID は、常に、`{1827A856-9C20-11d1-96C3-00062912C9B2}` でなければなりません。

JavaClass

フォルダーの内容を取得するために、iSeries ナビゲーターによって呼び出される `ListManager` インプリメンテーションの完全修飾 Java クラス名です。

NameID

iSeries ナビゲーター階層に、フォルダーの名前として表示されるストリングの資源 ID を含むダブルワードです。

DescriptionID

iSeries ナビゲーター階層に、フォルダーの説明として表示されるストリングの資源 ID を含むダブルワードです。

DefaultIconIndex

プラグインの NLS 資源 DLL に組み込まれた、iSeries ナビゲーター階層のフォルダー用に表示されるアイコンの索引を含むダブルワードです。これは、資源 DLL に組み込まれたゼロ・ベースの索引であり、アイコンの資源識別コードではありません。索引付けが正常に行われるためには、アイコンの資源識別コードが順番に割り当てられている必要があります。

OpenIconIndex

ユーザーがフォルダーを選択したときに、iSeries ナビゲーター階層のフォルダー用に必ず表示されるアイコンの索引を含むダブルワードです。これは、省略時のアイコン索引と同じであってもかまいません。

AdminItem

フォルダーへのアクセスを制御するアプリケーション管理機能の機能識別コードを含むストリングです。このフィールドを省略した場合、アプリケーション管理機能はフォルダーへのアクセスを管理しません。このフィールドを指定した場合は、これはグループ機能あるいは管理可能機能の機能識別コードでなければなりません。これをプロダクト機能の機能識別コードにはできません。

TaskpadNameID

iSeries ナビゲーター階層に、フォルダーの名前として表示されるストリングの資源 ID を含むダブルワードです。

TaskpadDescriptionID

資源 DLL 内のテキスト・ストリングの資源 ID を含むダブルワードです。この資源 DLL は、iSeries ナビゲーターのユーザー・インターフェースで、タスクパッドの機能を説明するために使用されます。

コンテキスト・メニュー項目の追加:

```
-----  
; 新しいフォルダーとそのオブジェクトのコンテキスト・メニュー・ハンドラーを登録します。
```

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.MsgQueueSample3\shell\Sample3\*\ContextMenuHandlers\{1827A857-9C20-11d1-96C3-00062912C9B2}]  
"JavaClass"="com.ibm.as400.opnav.MsgQueueSample3.MqActionsManager"
```

```
-----  
; 新しいフォルダーとそのオブジェクトのドラッグ・アンド・ドロップの  
; コンテキスト・メニュー・ハンドラーを登録します。
```

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.MsgQueueSample3\shell\Sample3\*\DragDropHandlers\{1827A857-9C20-11d1-96C3-00062912C9B2}]  
"JavaClass"="com.ibm.as400.opnav.MsgQueueSample3.MqActionsManager"
```

タスクパッドのタスクの追加:

```
-----  
; 新しいフォルダーとそのオブジェクトのタスク・ハンドラーを登録します。
```

```
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.MsgQueueSample5\shell\Sample5\*\TaskHandlers\{1827A857-9C20-11d1-96C3-00062912C9B2}]  
"JavaClass"="com.ibm.as400.opnav.MsgQueueSample5.MqTasksManager"  
"JavaClassType"="TasksManager"
```

ドラッグ・アンド・ドロップのサポート:

```
;-----  
; 新しいフォルダーとそのオブジェクトのドロップ・ハンドラーを登録します。  
  
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.MsgQueueSample3\  
shelllex\Sample3*\DropHandler]  
@="{1827A857-9C20-11d1-96C3-00062912C9B2}"  
"JavaClass"="com.ibm.as400.opnav.MsgQueueSample3.MqDropTargetManager"
```

管理されるオブジェクトの指定

オブジェクト・タイプ識別コードのペアが `shelllex` キーの下に必要です。ペアの最初の識別コードは、`iSeries` ナビゲーターの構成要素のルート・フォルダーを指定します。プラグインが追加する新しいフォルダーの場合、この識別コードは、接合点として指定したフォルダーの論理タイプに一致する必要があります。既存のフォルダーの場合、このサブキーは、通常、`iSeries` サーバーのコンテナ・オブジェクトの下の、最初のレベルのフォルダーのオブジェクト・タイプでなければなりません。これらのタイプのストリングは、レジストリーの `HKEY_CLASSES_ROOT\IBM.AS400.Network\TYPES` の下に定義されています。

ペアの 2 番目の識別コードは、プラグインが操作対象にする特定のオブジェクト・タイプを識別します。「*」が指定されている場合は、最初の識別コードで識別されるフォルダー・タイプ、およびそのフォルダーの下の階層に表示されるすべてのフォルダーとオブジェクトに対して、プラグインが呼び出されます。それ以外の場合は、特定のタイプの識別コードを指定しなければなりません。それにより、そのタイプのオブジェクトに対してユーザーがアクションを行った場合のみ、そのプラグインが呼び出されます。

ナビゲーター階層の 1 のオブジェクト・タイプに、任意の数のプラグインが機能を追加できるように登録することができます。あるオブジェクト・タイプに機能を提供しているサーバー構成要素が、そのプラグインだけであると想定しないでください。このことは、既存のオブジェクト・タイプだけでなく、プラグインが定義する新しいオブジェクトに対しても適用されます。プラグインが広く使用されている場合は、そのプラグインで定義されているオブジェクト・タイプを、別のベンダーに拡張させないようにすることはできません。

CLSID

上の例で示されている CLSID は、Java プラグインを管理する組み込みの ActiveX サーバー構成要素を指定します。フォルダーに関係しないすべての機能に対しては、この CLSID は `{1827A857-9C20-11d1-96C3-00062912C9B2}` でなければなりません。

JavaClass

指定された機能をサポートするために、`iSeries` ナビゲーターによって呼び出されるインターフェースのインプリメンテーションの完全修飾 Java クラス名です。

SSL のサポート: プラグインと `iSeries` サーバーとの通信を、ソケット API またはその他の低レベルの通信サービスを使用して行う場合、SSL が要求されていれば、プラグインはそれをサポートしなければなりません。プラグインがこのサポートを提供していない場合は、以下に示すように、プラグインは SSL をサポートしていないことを示さなければなりません。これにより、ユーザーがセキュア接続を要求しても、プラグインの機能は使用不可になっています。

```
;-----  
; このプラグインが SSL をサポートしていることを示します。  
  
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.MsgQueueSample3\SSL]  
"Support Level"=dword:00000001
```

サポート・レベル

プラグインが SSL をサポートしている場合は、サポート・レベルの値を 1 に設定します。それ以外の場合は 0 に設定します。

プロパティ・シート・ハンドラーのプロパティ・ページ

プロパティ・シート・ハンドラーのプロパティ・シートは、Microsoft Foundation Class Library のクラスを使って構成することはできません。ただし、IBM が提供している **CExtPropertyPage** を MFC のクラス **CPropertyPage** の代わりに使用することができます。iSeries ナビゲーターのプラグインによってインプリメントされるプロパティ・ページは、**CExtPropertyPage** をサブクラスにする必要があります。クラス宣言はヘッダー・ファイル **PROPEXT.H** に、インプリメンテーションはファイル **PROPEXT.CPP** にあります。どちらのファイルも、サンプル・プラグインのパーツとして提供されています。

注 プラグインのプロジェクト・ワークスペースに **PROPEXT.CPP** を組み込む必要があります。

1 つのプロパティ・シートをプラグインのオブジェクト・タイプの 1 つに関連付けることを、プラグインが要求している場合は、**SFGAO_HASPROPSHEET** フラグをそのオブジェクトの属性の一部として戻さなければなりません。このフラグがオンの場合は、ナビゲーターは、そのオブジェクトのコンテキスト・メニューにプロパティを自動的に追加します。また、このフラグがオンの場合、コンテキスト・メニュー項目が選択されると、ナビゲーターは登録済みのプロパティ・シート・ハンドラーを呼び出して、プロパティ・シートにページを追加します。

プラグインは、プロパティ・シートではなく、独自のオブジェクト・タイプの 1 つに定義された「プロパティ」コンテキスト・メニュー項目を標準の Windows ダイアログとしてインプリメントすることができます。このような場合、フラグが定義され、**IContextMenu::QueryContextMenu** への呼び出しでナビゲーターに戻すことができます。このフラグが戻されると、プロパティに対する自動処理は実行されないため、コンテキスト・メニュー項目の追加と、関連付けられているダイアログのインプリメントは、プラグインが行います。このフラグについては、『**QueryContextMenu フラグについて**』で説明されています。

プラグインで iSeries ユーザーのプロパティ・シートの 1 つにプロパティ・ページを追加する場合は、そのプロパティ・シート・ハンドラーの **CLSID** を指定するキーに **PropSheet** フィールドを指定して、指定されたハンドラーがページを追加するプロパティ・シートを識別しなければなりません。以下に例を示します。

```
----- ;
iSeries ユーザー用に Network プロパティ・シートのプロパティ・シート・ハンドラーを登録する
[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY plug-in\IBM.Sample\shell\Users
and Groups\User\PropertySheetHandlers\{3D7907A1-9080-11d0-82BD-08005AA74F5C}]
"PropSheet"="Networks"
```

PropSheet フィールドの有効な値は以下のとおりです。

PropSheet フィールドの有効な値				
グループ	個人	セキュリティーまたは機能	ジョブ	ネットワーク
Groups-Before-All	Personal-Before-All	Capabilities-Before-All	Jobs-Before-All	Networks-Before-All
Groups-After-Info	Personal-After-Name	Capabilities-After-Privileges	Jobs-After-General	Networks-After-Servers
	Personal-After-Location	Capabilities-After-Auditing	Jobs-After-Startup	Networks-After-General
	Personal-After-Mail	Capabilities-Before-Other	Jobs-After-Display	
		Capabilities-After-Other	Jobs-After-Output	
			Jobs-After-International	

iSeries ユーザーのプロパティ・シートにページを追加するには、プラグインが IA4PropSheetNotify インターフェースをインプリメントしていなければなりません (『IA4PropSheetNotify インターフェース仕様のリスト』を参照)。

制約事項:

現在、iSeries ユーザー・オブジェクトのプロパティ・シートには、以下の制約があります。

- 1 人の iSeries ユーザーに関連付けられている各種のプロパティ・シートに対して複数のプロパティ・シート・ハンドラーを同じインプリメンテーション・クラスでインプリメントすることはできません。各プロパティ・シートには、個別の CLSID が必要です。

QueryContextMenu フラグについて: iSeries ナビゲーターでは、IContextMenu インターフェースに対する以下の拡張がサポートされています。

コンテキスト・メニュー項目の順序

iSeries ナビゲーターは、IContextMenu インターフェースを拡張して、特定のフォルダーまたはオブジェクトのメニューにメニュー項目を追加する順序を、より詳細に制御しています。ナビゲーターでは、そのコンテキスト・メニューは 3 つのセクションに構造化されています。この構造により、オブジェクトのコンテキスト・メニューに複数のコンポーネントが項目を追加する場合でも、項目が Windows ユーザー・インターフェースに定義されている適切な順序で表示されるようになります。

最初のセクションには、データベース・テーブルの再編成など、オブジェクト・タイプに固有のアクションが含まれます。2 番目のセクションには、「オブジェクトの作成」項目が含まれます。これらの項目は、「新規作成 (New)」メニュー項目のカスケード・メニューとなるオブジェクト・タイプです。最後に、削除やプロパティなど「標準の」Windows メニュー項目があります。コンテキスト・メニューのいずれのセクションにもメニュー項目を追加することができます。

iSeries ナビゲーターは、1 つのコンポーネントに対してメニューのセクションごとに 1 回ずつ、計 3 回連続して QueryContextMenu メソッドを呼び出します。uFlags パラメーターには、以下の追加フラグが定義されているため、コンテキスト・メニューのどのセクションが現在使用されているかを判別することができます。

UNITY_CMF_CUSTOM

このフラグは、メニューにオブジェクト固有のアクションを追加する必要があることを示します。

UNITY_CMF_NEW

このフラグは、メニューにオブジェクト作成項目を追加する必要があることを示します。

UNITY_CMF_STANDARD

このフラグは、メニューに標準のアクションを追加する必要があることを示します。

UNITY_CMF_FILEMENU

このフラグは、UNITY_CMF_STANDARD を変更します。これは、ユーザーがマウス・ボタンを右クリックした際に表示されるメニューとは対照的な、オブジェクトの「ファイル」プルダウン・メニュー構造を指し示します。

「ファイル」プルダウンの項目は、やや異なる方法で配置されます。メニューに「プロパティ」を追加する場合、通常この項目の前に挿入するセパレーターは挿入しないでください。また、「ファイル」メニューに「コピー」や「貼り付け」などの編集アクションを追加しないようにしてください。これらは「編集 (Edit)」プルダウンに表示されます。(iSeries ナビゲーターは、ユーザーのシェル・プラグインを適宜呼び出し、「編集 (Edit)」メニューの項目を取得します。UNITY_CMF_FILEMENU は設定しません。)

固有のプロパティ・ダイアログ

プラグインは、プロパティ・シートではなく、独自のオブジェクト・タイプの 1 つに定義された「プロパティ (Properties)」コンテキスト・メニュー項目を標準の Windows ダイアログとしてインプリメントすることがあります。UNITY_CMF_STANDARD フラグが設定されている場合、このような状況で定義されたフラグは、IContextMenu::QueryContextMenu への呼び出し時にナビゲーターに戻されます。この A4HYF_INFO_PROPERTIESADDED フラグは、QueryContextMenu により戻される HRESULT 値と論理和演算されます。

このフラグが戻るということは、「プロパティ (Properties)」の自動処理が行われないということです。この場合、プラグインはコンテキスト・メニュー項目を追加し、関連ダイアログを構成しなければなりません。

例: プロパティ・シート・ハンドラーの Visual Basic プロパティ・ページの作成

iSeries ナビゲーター Visual Basic プラグインによりインプリメントされるプロパティ・ページでは、プロパティ・ページを指定するためのレジストリー・キーを使用できません。特定のプロパティ・ページ・コンテキスト・メニュー項目を ListManager クラスに追加して、プロパティ・ページをインプリメントする必要があります。既存のプロパティ・シート・オブジェクトには、プロパティ・ページを追加することはできません。

Visual Basic のサンプル・プラグインでは、iSeries ナビゲーター・リストの Library に対するプロパティ・ページがサポートされています。これは、以下のステップで行います。

1. listman.cls 内で、Library オブジェクト・タイプが getAttributes メソッドのプロパティ・ページを指定します。


```

' Returns the attributes of an object in the list.
Public Function ListManager_getAttributes(ByVal item As Object) As Long
    Dim uItem As ItemIdentifier
    Dim nAttributes As ObjectTypeConstants

    If Not IsEmpty(item) Then
        Set uItem = item
    End If

    If uItem.getType = "SampleVBFolder" Then
        nAttributes = OBJECT_ISCONTAINER
    ElseIf item.getType = "SampleLibrary" Then
        nAttributes = OBJECT_IMPLEMENTSPROPERTIES
    Else
        nAttributes = 0
    End If

    ListManager_getAttributes = nAttributes
End Function

```

2. actnman.cls 内で、queryActions メソッドが Library オブジェクトのコンテキスト・メニューにプロパティを表示するよう指定します。

```

Public Function ActionsManager_queryActions(ByVal flags As Long) As Variant
    :
    :
    ' Add menu items to a Sample Library
    If selectedFolderType = "SampleLibrary" Then
        ' Standard Actions
        If (flags And STANDARD_ACTIONS) = STANDARD_ACTIONS Then
            ReDim actions(0)

            ' Properties
            Set actions(0) = New ActionDescriptor
            With actions(0)
                .Create
                .setID IDPROPERTIES
                .SetText m_uLoader.getString(IDS_ACTIONTEXT_PROPERTIES)
                .setHelpText m_uLoader.getString(IDS_ACTIONHELP_PROPERTIES)
                .setVerb "PROPERTIES"
                .setEnabled True
                .setDefault True
            End With

            ' Properties is only selectable if there is ONLY 1 object selected
            If Not IsEmpty(m_ObjectNames) Then
                If UBound(m_ObjectNames) > 0 Then
                    actions(2).setEnabled False
                End If
            End If
        End If
    End If
    :
    :
End Function

```

3. actnman.cls 内で、actionsSelected メソッドが、プロパティのコンテキスト・メニューが選択されるとプロパティ・フォームを表示します。

```

Public Sub ActionsManager_actionSelected(ByVal action As Integer, ByVal owner As Long)
    :
    :
    Select Case action
        :
        :
        Case IDPROPERTIES
            If (Not IsEmpty(m_ObjectNames)) Then

```

```

        ' Pass the System Name into a hidden field on the form for later use
        frmProperties.lblSystemName = m_ObjectNames(0).getSystemName

        ' Pass the Display Name of the selected object into a hidden field on the form
        frmProperties.lblLibName = m_ObjectNames(0).getDisplayname

        ' Show the properties
        frmProperties.Show vbModal
    End If

    .
    .
    Case Else
        'Do Nothing
    End Select

    .
End Sub

```

注: プロパティ・シートを作成して表示するためのコードは、**propsht.frm** にあります。

Java におけるプロパティ・シートの処理

V5R1 では、Java プラグインのプロパティ・シートにプロパティ・ページを追加することができます。これにより、オブジェクト名の作成、プロパティの表示、サード・パーティーとのオブジェクトの共用、および同一プラグインでの C++ と Java コードの混合が可能になります。

プロパティ・ページを使用するには、以下のメソッドを提供するプロパティ・マネージャー・インターフェースを構築しなければなりません。

- **Initialize**
プロパティのコンテナ・オブジェクトを識別します。
- **getPages**
PanelManager オブジェクトのベクトルを構成し、提供します。
- **CommitHandlers**
コミット時に呼び出されるハンドラーのベクトルを戻します。
- **CancelHandlers**
取り消し時に呼び出されるハンドラーのベクトルを戻します。

次に、**ListManager** の **getAttributes** メソッドで **ListManager.OBJECT_HASPROPERTIES** を戻して、プロパティ・メニューを使用可能にします。

最後に、**PopertiesManagerInterface** を識別するレジストリー項目を作成します。例えば、以下のようになります。

```

[HKEY_CLASSES_ROOT\IBM.AS400.Network\AS/400 Network\*
\shell\PropertySheetHandlers\{1827A857-9C20-11d1-96C3-00062912C9B2}]
"JavaClass"="com.ibm.as400.opnav.TestPages.TestPropertiesManager"
"JavaClassType"="PropertiesManager"

```

注: 複数の **PropertiesManager** のインプリメンテーションが、ある 1 つのオブジェクト・タイプ用のプロパティ・ページを提供するように登録していることがあります。したがって、自分の項目だけがページを提供していると想定したり、ページが追加される順序を想定したりしないでください。

詳しくは、『例: Java プロパティ・マネージャー』を参照してください。

Secure Sockets Layer (SSL) のレジストリー項目

iSeries ナビゲーターのユーザーは、iSeries オブジェクトのプロパティ・シートにある「接続」タブの「**Secure Sockets Layer を使用する (Use Secure Sockets Layer)**」チェック・ボックスを選択することにより、iSeries サーバーにセキュア接続を要求することができます。このチェック・ボックスが選択されている場合は、SSL 通信をサポートする iSeries ナビゲーター構成要素だけを活動化することができます。

プラグインと iSeries サーバーとの通信がすべて、iSeries Access for Windows のシステム・ハンドルを使用することで管理されている場合 (cwbCO_SysHandle を入力する)、あるいは Java プラグインの場合であれば、クラス **com.ibm.as400.access.AS400** を使用することで管理されている場合は、プラグインが iSeries サーバーとのセキュア接続をサポートしていることを示します。C++ プラグインの場合は、cwbCO_SysHandle は cwbUN_GetSystemHandle API を呼び出すことにより取得されます。ユーザーがセキュア接続を要求している場合、ナビゲーターは自動的に SSL を使用可能にします。Java プラグインの場合、クラス **com.ibm.as400.opnav.ObjectName** の **getSystemObject** メソッドを呼び出すことにより取得された iSeries サーバー・オブジェクトは、実際には **com.ibm.as400.access.SecureAS400** のインスタンスになります。

注: SSL で Java を実行し、独自の CA 証明書を作成する場合は、iSeries Access for Windows GA サービス・パックが必要です。

プラグインと iSeries サーバーとの通信を、ソケット API またはその他の低レベルの通信サービスを使用して行う場合、SSL が要求されていれば、プラグインはそれをサポートしなければなりません。プラグインがこのサポートを提供していない場合は、以下に示すように、プラグインは SSL をサポートしていないことを示さなければなりません。これにより、ユーザーがセキュア接続を要求しても、プラグインの機能は使用不可になっています。

例: SSL を使用可能にするためのレジストリー・キーを追加する

キーは、[HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.Sample\SSL]
"Support Level"=dword:00000001 の下の SSL です。ここで、IBM.Sample はプラグインが提供するプロダクト・コンポーネントです。

注: "Support Level"=dword:00000001 は SSL をサポートしています。"Support Level"=dword:00000000 は SSL をサポートしていません。

```
-----  
; このプラグインが SSL をサポートしていることを示しているレジストリー・キーの例  
{HKEY_CLASSES_ROOT\IBM.AS400.Network\3RD PARTY EXTENSIONS\IBM.Sample\SSL}  
"Support Level"=dword:00000001
```




Printed in Japan