



IBM Systems - iSeries

iSeries 接続

iSeries ナビゲーター プラグインの開発

バージョン 5 リリース 4





IBM Systems - iSeries

iSeries 接続

iSeries ナビゲーター プラグインの開発

バージョン 5 リリース 4

ご注意

本書および本書で紹介する製品をご使用になる前に、107 ページの『特記事項』に記載されている情報をお読みください。

本書は、i5/OS™ (5722-SS1) バージョン 5、リリース 4、モディフィケーション 0 に適用されます。また、改訂版で特に断りがない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。このバージョンは、すべての RISC モデルで稼働するとは限りません。また CISC モデルでは稼働しません。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： IBM Systems - iSeries
Connecting to iSeries
Developing iSeries Navigator Plug-ins
Version 5 Release 4

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.2

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2004. All rights reserved.

© Copyright IBM Japan 2006

目次

iSeries ナビゲーター プラグインの開発	1	サンプル Java プラグインのセットアップ	20
V5R4 の新機能	1	プラグインのプログラミング・リファレンス	24
トピックの印刷	1	C++ のリファレンス	24
iSeries ナビゲーターのプラグイン・サポート	2	Visual Basic のリファレンス	75
プラグインを使ってできること	2	Java のリファレンス	77
プラグインの仕組み	2	プラグイン・レジストリー・ファイルのカスタマイズ	79
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	15	付録. 特記事項	107
サンプル・プラグインのインストールと実行	15	プログラミング・インターフェース情報	108
サンプル C++ プラグインのセットアップ	15	商標	109
サンプル Visual Basic プラグインのセットアップ	18	資料に関するご使用条件	109

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。

注: コードのサンプルを使用すると、105 ページの『コードに関するライセンス情報および特記事項』の条件に同意したものとみなされます。

V5R4 の新機能

このリリースの新機能を検索します。

V5R4 には、Java Runtime Environment (JRE) 1.4.2 のサポートが含まれています。

iSeries ナビゲーター・プラグインの開発には、setup.ini ファイルの拡張セクションがあり、出口プログラム・セクションの新規の例があります。

新機能または変更点を確認する方法

技術上の変更が行われた箇所を見分ける上で役立つように、この情報では以下の記号を使用しています。

-  イメージは、新規情報または変更情報の開始位置を示します。
-  イメージは、新規情報または変更情報の終了位置を示します。

本リリースの新機能または変更点についてのその他の情報を見るには、「プログラム資料説明書」を参照してください。

トピックの印刷

この情報の PDF を表示および印刷する場合に使用します。

この文書の PDF 版をダウンロードし、表示するには、「iSeries ナビゲーター プラグインの開発」を選択します。

PDF ファイルの保存

表示用または印刷用の PDF ファイルをワークステーションに保存するには、次のようにします。

1. ブラウザーで PDF を開く (上記のリンクをクリックする)。
2. PDF をローカルに保管するオプションをクリックする。
3. PDF を保存したいディレクトリーに進む。
4. 「保存」をクリックする。

Adobe Reader のダウンロード

PDF ファイルを表示したり印刷したりするには、ご使用のシステムに Adobe Reader がインストールされている必要があります。これは、Adobe Web サイト (www.adobe.com/products/acrobat/readstep.html) から、ダウンロードできます。

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。

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用します。

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: 追加のセキュリティ設定) をサポートするために使用します。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

iSeries ナビゲーターが新規プラグインを検索し、このプラグインを新規構成でインストールする方法の詳細な例

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、新しい構成にインストールします。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、ユーザーのデフォルト・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。

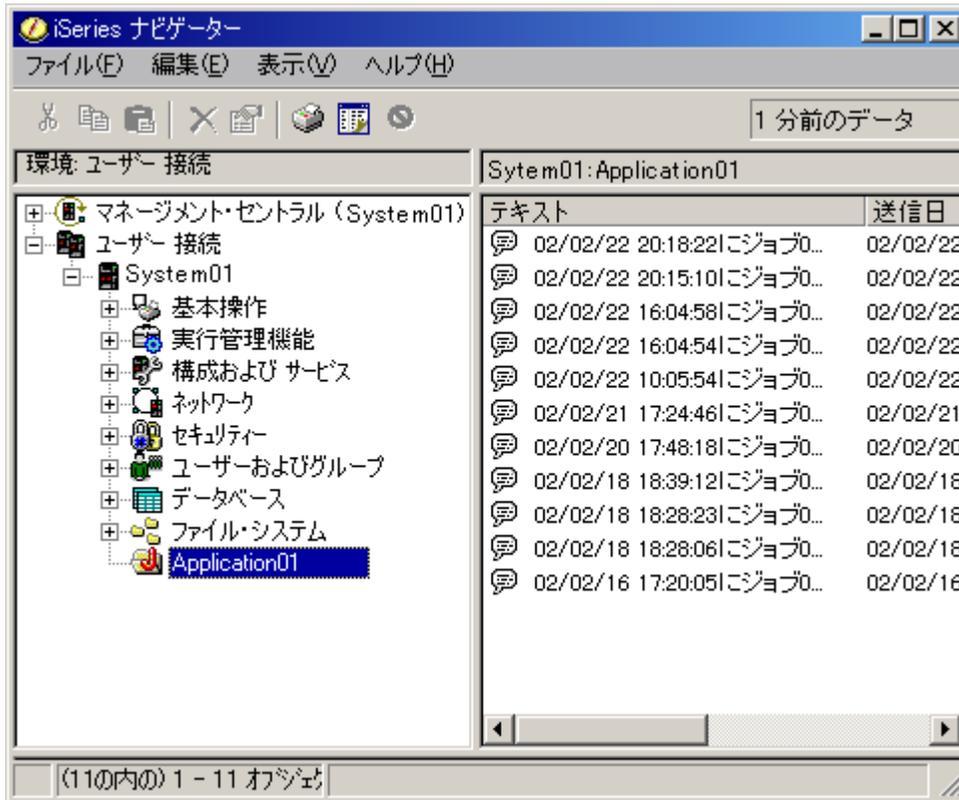
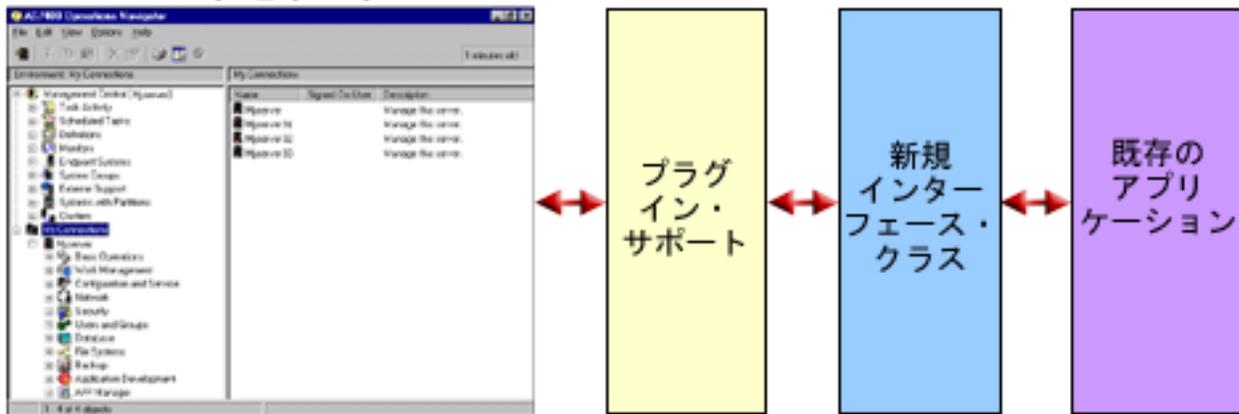


図1. iSeries ナビゲーター・ダイアログ -- メッセージ待ち行列内のメッセージ

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。

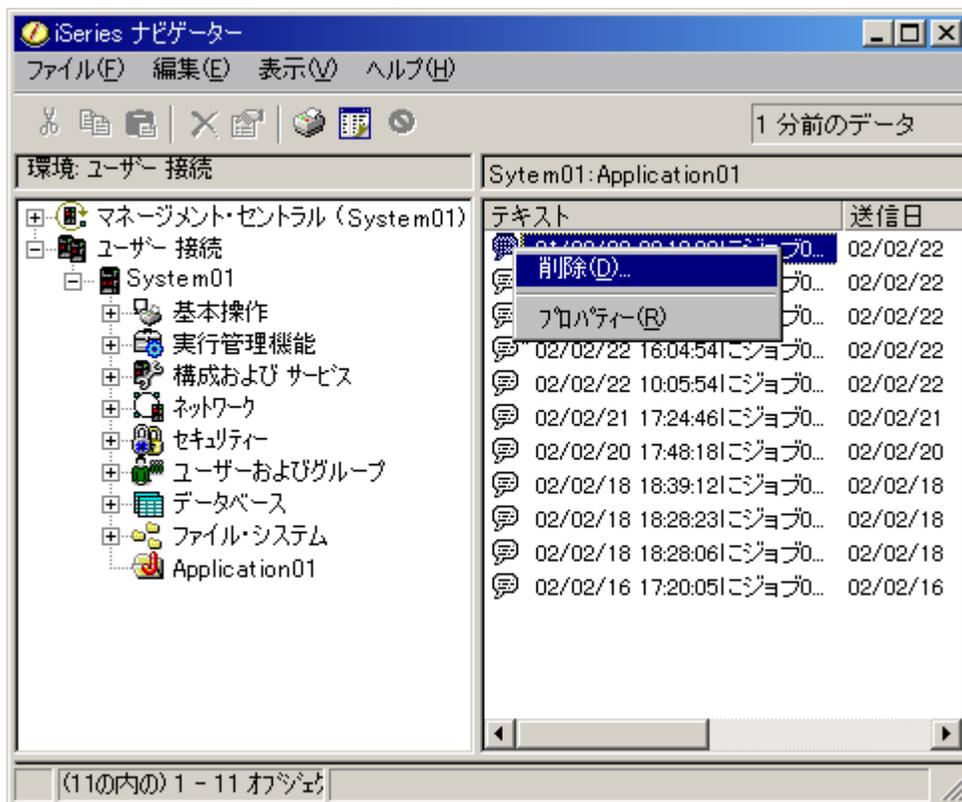
図2. iSeries ナビゲーターがアプリケーションを呼び出してリスト・データを取得する方法

iSeries ナビゲーター



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。

図3. 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。

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.h	cwbunapi.lib	cwbunapi.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、Java Technology Edition で動作します。以下の表に、iSeries Access for Windows と一緒にインストールされる Java のバージョンを示します。

リリース	JRE	Swing	JavaHelp
V5R4	1.4.2	N/A	1.1.1
V5R3	1.4.1	N/A	1.1.1
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 コンパイラーであれば、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 コンパイラーがオフになり、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

ページ・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、IBM 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、オプションとしてインストール可能な、iSeries Access for Windows のコンポーネントです。 iSeries Access for Windows 製品の初期インストール時に一緒にインストールするか、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インストール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。

関連概念

15 ページの『サンプル・プラグインのインストールと実行』

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。

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 i5/OS[®] アプリケーションに組み込みます。

アプリケーションのインストール・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 for Windows の選択セットアップ・プログラムを使用して iSeries Access for Windows 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。初期インストール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インストール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	インストール・ディレクトリー	組み込まれるファイル
C++	/QIBM/USERDATA/OpNavPlugin/ <vendor>.<component> (iSeries Access for Windows なしでのインストールを避けるため)	<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、およびシリアルライゼーション・ファイルが入っている。

プラグインのタイプ	インストール・ディレクトリー	組み込まれるファイル
Visual Basic	/QIBM/USERDATA/OpNavPlugin/ <vendor>.<component> (VB プラグインには iSeries Access for Windows 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。
- GUIPlugin の場所は将来のリリースでサポートされなくなるため、プラグインを OpNavPlugin の場所に移行する必要があります。

また、すべてのプラグインは、<vendor>.<component> サブディレクトリー下に最低 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アンインストール

ユーザーが新規プラグインをインストールした後、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 for Windows にはアンインストール・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・プロパティーで、「プラグイン (Plug-ins)」タブをクリックすることにより、マシンにインストール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インストール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。

Setup.ini ファイル

プラグインの setup.ini ファイルは、インストール・ウィザードに、クライアント・ワークステーションで iSeries ナビゲーター・プラグインをインストール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、チェック・サービス・レベル・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 4 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インストールするファイルの識別を行うためのセクション
- クライアント・ワークステーションで実行する出口プログラムの識別を行うためのセクション

関連概念

19 ページの『サンプル VB プラグイン・ファイルのディレクトリー』

例: setup.ini の情報セクション:

setup.ini ファイルで必要なグローバル情報フィールドの説明

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
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。プラグインのインストール時に、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。
説明	プラグインの英語の説明です。この説明は、プラグインのインストール時に、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。

Setup.ini の [Plugin Info] セクションのフィールド	フィールドの説明
Version	<p>プラグインのリリース・レベルを示す数値です。チェック・サービス・レベル・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インストール済みのプラグインの現在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、チェック・サービス・レベル・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	<p>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 for Windows 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 サーバーにインストールするときのパスの <VENDOR>.<COMPONENT> の部分と同じでなければなりません。</p>
JavaPlugin	<p>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、インストール・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> ディレクトリーにインストール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、そのプラグインはインストールされても動作しません。</p>

例: setup.ini のサービス・セクション:

setup.ini ファイルのこのセクションのフィールドには、Check Service Level プログラムで使用される情報が含まれています。

セットアップ・ファイルの 2 番目のセクション (Service) は、チェック・サービス・レベル・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。チェック・サービス・レベル・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インストール済みのプラグインの現在の FixLevel 値と比較されます。この FixLevel 値が、クライアント・ワークステーションにインストールされているプラグインの FixLevel 値よりも大きい場合、チェック・サービス・レベル・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>

Setup.ini の [Service] セクションのフィールド	フィールドの説明
AdditionalSize	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。インストール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。

例: setup.ini のファイル識別セクション:

setup.ini ファイルのこの部分には、クライアント・ワークステーションにインストールされるファイルを識別する情報が含まれています。

セットアップ・ファイルの 3 番目のセクションでは、クライアント・ワークステーションにインストール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、初期インストール時、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]	Client Access のインストール・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。 C++ と Visual Basic の場合 、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) をここに指定します。 Java の場合 、Code JAR ファイル名をここに指定します。
[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。

Setup.ini のセクション	説明
[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インストール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、プラグイン用の 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>プラグインをインストールする前にインストール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。AS400_Client_Access_Express が必要となるのは、プラグインが、iSeries ナビゲーターの基本のサポート・サブコンポーネント以外に、他のサブコンポーネントのインストールを必要としている場合のみです。</p> <p>AS400_Client_Access_Express</p> <ul style="list-style-type: none"> サブコンポーネントは、コマンド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 のコマンド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_ 定数は、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> <p>注: AS400_Client_Access_Express の値は存在する場合にのみ使用されます。存在しない場合は、このセクションは無視されます。</p>
[Service Base Files]	iSeries Access for Windows のインストール・ディレクトリーの下での \PLUGINS\<VENDOR>.\<COMPONENT> にコピーされるファイルです。
[Service Shared Files]	iSeries Access for Windows 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。

Setup.ini のセクション	説明
[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。

例: setup.ini の出口プログラム・セクション:

セットアップ・ファイルの最終部分には、インストール、アップグレード、またはアンインストールの前後にクライアント・ワークステーションで実行するプログラムを識別する記述が含まれます。

以下は、インストール、アップグレード、またはアンインストールの前後に、クライアント・ワークステーションでプログラムを識別および実行するこれらの出口プログラムのセクションで使用される構文の例です。

例 1 - ファイルのインストール前に呼び出されるオプションのプログラム。これは、初期インストール中のみに実行されます。

```
[PreInstallProgram]
Program=whatever.exe
CmdLine=
CheckReturnCode=
Wait=
```

[PreInstallProgram] のフィールド	説明
Program	必須。パスが指定されている場合、ファイル名のみが使用されます。プログラムは、インストール・ソースのプラグインの <vendor>.<component> パスに存在している必要があります。
CmdLine	オプション。特定のプログラムによりさまざまなコマンドが必要とされます。
CheckReturnCode	オプション。デフォルトは、N です。Y に設定され、ゼロ以外が戻される場合、このプラグインのインストールは継続されません。プログラムがゼロ以外の戻りコードを戻す場合、メッセージは表示されませんが、メッセージは、INSTLOG.TXT に記録されます。
Wait	オプション。実行の継続前にプログラムの終了を待機します。デフォルトは、Y です。CheckReturnCode=Y の場合、ここで指定されているものに関係なく Wait=Y が使用されます。

例 2 - ファイルのインストール後に呼び出されるオプションのプログラム。これは、初期インストール中のみに実行されます。

```
[PostInstallProgram]
Program=whatever.exe
CmdLine=
CheckReturnCode=
Wait=
```

[PostInstallProgram] のフィールド	説明
Program	必須。パスが指定されている場合、ファイル名のみが使用されます。プログラムは、PC のプラグインの <vendor>.<component> ディレクトリーに存在している必要があります。

[PostInstallProgram] のフィールド	説明
CmdLine	オプション。特定のプログラムによりさまざまなコマンドが必要とされます。
CheckReturnCode	オプション。PreInstallProgram でサポートされているものと同じ値が PostInstallProgram に対してサポートされていますが、この時点ではプラグインは既にインストールされているため、CheckReturnCode 値はそれほど重要ではありません。
Wait	オプション。実行の継続前にプログラムの終了を待機します。また、PreInstallProgram でサポートされているものと同じ値が PostInstallProgram に対してサポートされていますが、この時点ではプラグインは既にインストールされているため、CheckReturnCode 値はそれほど重要ではありません。

例 3 - ファイルのアンインストール前に呼び出されるオプションのプログラム。

```
[UninstallProgram]
Program=whatever.exe
CmdLine=
CheckReturnCode=
Wait=
```

[UninstallProgram] のフィールド	説明
Program	必須。パスが指定されている場合、ファイル名のみが使用されます。プログラムは、PC のプラグインの <vendor>.<component> ディレクトリーに存在する必要があります。
CmdLine	オプション。特定のプログラムによりさまざまなコマンドが必要とされます。
CheckReturnCode	オプション。デフォルトは、N です。CheckReturnCode 値が Y に設定されている場合、戻り値がゼロ以外ならプラグインのアンインストールは継続しません。プログラムがゼロ以外の戻りコードを戻す場合、メッセージは表示されませんが、メッセージは、INSTLOG.TXT に記録されます。
Wait	オプション。実行の継続前にプログラムの終了を待機します。デフォルトは、Y です。CheckReturnCode=Y の場合、ここで指定されているものに関係なく Wait=Y が使用されます。

例 4 - ファイルのアップグレード前に呼び出されるオプションのプログラム。

```
[PreUpgradeProgram]
Program=whatever.exe
CmdLine=
CheckReturnCode=
Wait=
```

[PreUpgradeProgram] のフィールド	説明
Program	必須。パスが指定されている場合、ファイル名のみが使用されます。プログラムは、インストール・ソースのプラグインの <vendor>.<component> パスに存在する必要があります。
CmdLine	オプション。特定のプログラムによりさまざまなコマンドが必要とされます。
CheckReturnCode	オプション。デフォルトは、N です。Y に設定され、ゼロ以外が戻される場合、このプラグインのインストールは継続されません。プログラムがゼロ以外の戻りコードを戻す場合、メッセージは表示されませんが、メッセージは、INSTLOG.TXT に記録されます。

[PreUpgradeProgram] のフィールド	説明
Wait	オプション。実行の継続前にプログラムの終了を待機します。デフォルトは、Y です。CheckReturnCode=Y の場合、ここで指定されているものに関係なく Wait=Y が使用されます。

例 5 - ファイルのアップグレード後に呼び出されるオプションのプログラム。

```
[PostUpgradeProgram]
Program=whatever.exe
CmdLine=
CheckReturnCode=
Wait=
```

[PostUpgradeProgram] のフィールド	説明
Program	必須。パスが指定されている場合、ファイル名のみが使用されます。プログラムは、インストール・ソースのプラグインの <vendor>.<component> パスに存在している必要があります。
CmdLine	オプション。特定のプログラムによりさまざまなコマンドが必要とされます。
CheckReturnCode	オプション。PreUpgradeProgram でサポートされているものと同じ値が PostUpgradeProgram に対してサポートされていますが、この時点ではプラグインは既にインストールされているため、CheckReturnCode 値はそれほど重要ではありません。
Wait	オプション。実行の継続前にプログラムの終了を待機します。デフォルトは、Y です。CheckReturnCode=Y の場合、ここで指定されているものに関係なく Wait=Y が使用されます。

MRI セットアップ・ファイル

iSeries Access for Windows 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 にインストール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 - 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 for Windows の選択セットアップ・プログラムは、プラグインの初期インストール時と、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。インストール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。

関連概念

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・ソフトウェアがインストールされると、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 ナビゲーター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、i5/OS 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インストールされていることを、その動作要件としている場合などが考えられます。

プラグインのインストール後、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前提条件が、iSeries サーバーにインストール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・プラグインのインストールと実行

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 をインストールしていない場合は、インストールしてからでないと、サンプル・プラグインを扱うことはできません。Toolkit は、iSeries Access for Windows 選択セットアップによってインストールすることができます。

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。

関連概念

5 ページの『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。

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>cppsmppq.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"> 1. 「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 <code>c:\MyProject</code> 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。 2. 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。Programmer's Toolkit - iSeries Navigator Plug-ins Web ページからサンプルをダウンロードできます。 3. 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。 4. 「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。 5. <code>sampext.mak</code> を選択し、「オープン」をクリックします。 6. 「ツール」メニューをオープンし、「オプション...」を選択します。 7. 「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。 8. 「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。 9. 「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"> 1. 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。 2. 「省略時のプロジェクト構成 (Default Project Configuration)」ダイアログで、<code>sampext</code> の「Win32 デバッグ構成 (Win32 Debug Configuration)」を選択します。 3. 「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、「<code>sampext.dll</code>」を選択して再度ビルドを実行してください。

<p>資源ライブラリーをビルドする</p>	<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"> 1. 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。 2. 「ファイルの種類:」で「Makefiles (*.mak)」を指定します。 3. sampmri.mak を選択し、「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>ActiveX サーバー .dll を登録する</p>	<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 (あるいはご使用のテキスト・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>デバッガーで iSeries ナビゲーターを実行する</p>	<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 for Windows のインストール・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。

関連概念

Programmer's Toolkit - iSeriesNavigator Plug-ins Web page

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、i5/OS 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。

インストールされるプラグイン・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	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。
VB プロジェクトを作成する	Visual Basic で vbsample.vpb をオープンします。参照設定ダイアログで、「 IBM iSeries Access for Windows ActiveX Object Library 」と「 iSeries Navigator Visual Basic Plug-in Support 」を選択します。 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 cwibunvi.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。
ActiveX サーバー DLL をビルドする	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。
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	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。
iSeries ナビゲーターでプラグインを実行する	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。

関連概念

『サンプル VB プラグイン・ファイルのディレクトリー』

関連タスク

実行可能ファイル vbopnav.exe のダウンロード

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、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
authorfy.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
authorfy.frx	「権限の設定 (Set authority)」フォーム・バイナリー
delete.frx	「削除の確認 (Confirm delete)」フォーム・バイナリー
propsht.frx	「プロパティ・シート (Property Sheet)」フォーム・バイナリー
sysstat.frx	「システム状況」フォーム・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・バイナリー
vbsample.bin	Vbsample バイナリー

構成の設定値	説明
mrsetup.ini	プラグインの翻訳可能資源のインストール情報
setup.ini	プラグインの実行可能ファイルのインストール情報

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	インストール時に使用されるレジストリー・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

関連概念

18 ページの『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、i5/OS 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。

14 ページの『MRI セットアップ・ファイル』

iSeries Access for Windows 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 にインストールする際に必要となる情報を提供します。

7 ページの『Setup.ini ファイル』

プラグインの setup.ini ファイルは、インストール・ウィザードに、クライアント・ワークステーションで iSeries ナビゲーター・プラグインをインストール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、チェック・サービス・レベル・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。

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、デフォルト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。

インストールされるプラグイン・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、以下のようになります。

<p>サンプル Java プラグインをダウンロードする</p>	<p>実行可能ファイル <code>javopnav.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、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デフォルトのディレクトリー <code>javopnav¥com¥ibm¥as400¥opnav</code> にインストールされるようになります。</p>
<p>iSeries ナビゲーターにプラグインを識別させる</p>	<ol style="list-style-type: none"> 1. <code>javopnav¥com¥ibm¥as400¥opnav¥MsgQueueSampleX</code> にあるファイル <code>MsgQueueSampleX.reg</code> を編集します。(X=1、2 または 3。これはインストールするサンプルによって異なります。) 2. "NLS"="c:¥¥javopnav¥¥win32¥¥mri¥¥MessageQueuesMRI.dll" および "JavaPath"="c:¥¥javopnav" の行を見つけます。 3. 「c:¥¥」を、PC の <code>javopnav</code> 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。
<p>サンプル Java プラグインを実行する</p>	<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 ナビゲーターは階層ツリーに新しいフォルダー、Java 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 のデフォルト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。

関連概念

『サンプル Java プラグイン・ファイルのディレクトリー』

関連情報

実行可能ファイル `javopnav.exe` のダウンロード

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、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 `javopnav¥com¥ibm¥as400¥opnav¥MsgQueueSample1¥docs` ディレクトリーにインストールされています。

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。
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 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。
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にはできません。このレジストリー・ファイルには、iSeries Access for Windows のインストール・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 for Windows 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 サーバーからインストールするときに、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が、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。

関連概念

20 ページの『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。

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。

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。

C++ のリファレンス

以下のトピックを利用して、C++ 用 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。

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あります。

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機能は、「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	指定されたりスト・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。

インターフェース	メソッド	説明
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返します。
	GetValue	属性 ID が指定されると、属性の現行値を返します。
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返します。
	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。このメソッドで、例えば、プロパティが表示されている iSeries ユーザーが既存または定義中であることや、変更を保管するか破棄するかなどを伝えます。

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、プラグインがナビゲーター階層に追加した新規のフォルダーの内容を取り込む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 レジストリーの項目によって、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。

関連情報

Microsoft Web サイト

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・エンティティーは、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: 目的

IA4HierarchyFolder インスタンスを活動状態にします。また、クライアント上のフォルダー・オブジェクトのキャッシュを準備するための iSeries サーバーの呼び出しなど、列挙用のフォルダーを準備するために必要な処理もすべて実行します。この関数は、長時間の操作によりユーザー・インターフェースのパフォーマンスが劣化することがないように、データ・スレッドから呼び出されます。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE Activate();
```

パラメーター

なし。

戻りコード

正常な場合は NOERROR を、フォルダーの内容を取得できない場合は E_FAIL を戻します。

注記

ユーザーが最初にフォルダーを選択または拡張したときに、iSeries ナビゲーターによってこの関数が呼び出されます。ユーザーがフォルダー内容の最新表示を要求すると、Close を呼び出した後に、再度この関数は呼び出されます。

これ以外にも、フォルダー・インターフェースへのポインターを再度確立する必要がある場合に呼び出すことがあります。例えば、ユーザーがフォルダーを 2 度目に選択する場合などです。別のフォルダーを選択した後の場合は、関連処理が実行済みであれば TRUE が戻されます。

極端に大きなリストの場合は、最初に作業員スレッドを作成してリストの作成を継続してから、リストを完全に構成する前に Activate から戻るよう選択することができます。この場合、GetListSize のインプリメンテーションが、リストが完全に構成されたかどうかの適切な指示を必ず戻すようにしてください。

IA4HierarchyFolder::BindToList: 目的

iSeries ナビゲーター階層の特定のフォルダーに対応する IA4HierarchyFolder のインスタンスを戻します。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE BindToList(  
    HWND hwnd,  
    LPCITEMIDLIST pidl,  
    REFIID riid,  
    LPVOID* ppvOut  
);
```

パラメーター

hwnd ツリーまたはリスト・コントロールのいずれかとしてリストを表示するビュー・ウィンドウのハンドル。クライアント上でこのビューについてオブジェクトのリストがキャッシュされているかどうかを判別するには、コンポーネントがこのハンドルを使用する必要があります。

pidl 列挙するフォルダーを一意的に識別する ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

riid 戻されるインターフェースの識別コード。このパラメーターは、IID_IA4HierarchyFolder インターフェースの識別コードを指します。

ppvOut

インターフェース・ポインターを受け取るアドレス。エラーが発生した場合は、このアドレスに NULL ポインターが戻されなければなりません。

戻りコード

正常な場合は NOERROR を、一般エラーが発生した場合は E_FAIL を戻します。

注記

指定されたフォルダーについて *IA4HierarchyFolder* のインスタンスが既に存在する場合、このメンバー関数は、別個のインスタンスをインスタンス化して初期化するのではなく、キャッシュされたインスタンスを戻さなければなりません。ただし、キャッシュされたオブジェクトに関連したウィンドウ・ハンドルが、*hwnd* パラメーターに指定された値と同じでない場合は、新しいインスタンスを作成する必要があります。

この関数は、提供されたパラメーターからインプリメンテーション・クラスのメンバー変数を初期化する必要があります。

***IA4HierarchyFolder::DisplayErrorMessage:* 目的**

Activate がエラーを戻した際に、エンド・ユーザーにエラー・メッセージを表示するために呼び出されます。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE DisplayErrorMessage();
```

パラメーター

なし。

戻りコード

正常な場合は *NOERROR* を、表示するメッセージがない場合は *E_FAIL* を戻します。

注記

なし。

***IA4HierarchyFolder::GetAttributesOf:* 目的**

iSeries ナビゲーター階層の特定のフォルダーの属性を戻します。属性標識は、Microsoft のインターフェース・メソッド *IShellFolder::GetAttributesOf* に定義されているものと同じです。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetAttributesOf(  
    LPCITEMIDLIST pidl,  
    ULONG* ulfInOut  
);
```

パラメーター

pidl 検索する属性を持つオブジェクトを一意的に識別する *ITEMIDLIST* (項目識別コード・リスト) 構造体へのポインター。

ulfInOut

戻されるオブジェクト属性。入力時、このパラメーターは検索するオブジェクト属性を示すよう設定されます。

戻りコード

正常な場合は *NOERROR* を、オブジェクト属性を見付けられない場合は *E_FAIL* を戻します。

注記

ビット・フラグを定義する定数については、Windows 組み込みファイル shlobj.h を参照してください。

この関数は、ツリーまたはリスト・ビューを配置する際に、iSeries ナビゲーターにより繰り返し呼び出されます。したがって、長時間の実行操作は避けるようにしてください。

IA4HierarchyFolder::GetColumnDataItem: 目的

iSeries ナビゲーターのリスト・ビューの列に表示するフォルダーまたはオブジェクトのデータ・フィールドを戻します。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetColumnDataItem(  
    LPCITEMIDLIST pidl,  
    LPARAM lParam,  
    char * lpszColumnData,  
    UINT cchMax  
);
```

パラメーター

pidl 取得する列データを持つオブジェクトを一意的に識別する ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

lParam

コンポーネントによってデータが要求される列に、以前関連のあった値 (GetColumnInfo を参照)。

lpszColumnData

NULL 終了のデータ・ストリングを受け取るバッファのアドレス。

cchMax

NULL 終了のデータ・ストリングを受け取るバッファのサイズ。

戻りコード

正常な場合は NOERROR を、列データを検索できなかった場合は E_FAIL を戻します。

注記

この関数は、リスト・ビューを配置する際に、iSeries ナビゲーターにより繰り返し呼び出されます。したがって、長時間の実行操作は避けるようにしてください。

IA4HierarchyFolder::GetColumnInfo: 目的

詳細ビューにある特定のフォルダーの内容表示に必要な列を記述するデータ構造体を戻します。これは、オプションのメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetColumnInfo(  
    LPVOID* ppvInfo  
);
```

パラメーター

ppvInfo

戻されるデータ構造体。戻される構造体は、A4hyfColumnInfo 構造体のインスタンスから構成されている必要があります。この構造体は、リスト・ビューの列ごとに 1 つずつ A4hyfColumnItem 構造体の配列を含みます。

それぞれの列項目構造体は、列見出し用に変換されたストリング、デフォルトの列幅、および列にデータを与えるデータ・フィールドを一意的に識別する整数値を提供します。CWBA4HYF.H を参照してください。

戻りコード

正常な場合は NOERROR を、関数をインプリメントできない場合は E_NOTIMPL を戻します。

注記

iSeries ナビゲーターは、Open 呼び出しが戻った後にこの関数を呼び出し、詳細ビューの列見出しを作成します。

この関数がインプリメントされていない場合、ナビゲーターは名前と説明の 2 つの列を挿入します。GetColumnDataItem により、これらの 2 つのフィールドにデータを戻さなければなりません。これらのフィールドは、それぞれ整数値 0 および 1 により識別されます。

戻された構造体に記憶域を割り振るには、Windows IMalloc インターフェースを使用します。この記憶域の削除は、ナビゲーターにより行います。

IA4HierarchyFolder::GetIconIndexOf: 目的

階層フォルダーのアイコンのロードに使用するコンポーネント資源 DLL に、索引を戻します。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetIconIndexOf(  
    LPCITEMIDLIST pidl,  
    UINT uFlags,  
    int* piIndex  
);
```

パラメーター

pidl 検索されるアイコン索引を持つオブジェクトを一意的に識別する ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

uFlags このパラメーターはゼロか、GIL_OPENICON (提供されるアイコンが開いたフォルダーであることを指す) にすることができます。GIL_OPENICON は、Windows 組み込みファイル SHLOBJ.H に定義されています。

piIndex

アイコン索引を受け取る整数へのポインター。

戻りコード

正常な場合は NOERROR を、索引を判別できない場合は E_FAIL を戻します。

注記

この関数は、ツリーまたはリスト・ビューを配置する際に、iSeries ナビゲーターにより繰り返し呼び出されます。したがって、長時間の実行操作は避けるようにしてください。

IA4HierarchyFolder::GetItemCount: 目的

iSeries Navigator 階層の特定のフォルダーに含まれるオブジェクトの合計数を戻します。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetItemCount(  
    ULONG* pCount  
);
```

パラメーター

pCount

リスト内の項目の数を受け取る長整数へのポインター。

戻りコード

リストが完全に作成されて項目の合計数が判別された場合、A4HYF_OK_LISTCOMPLETE が戻されます。リストが構成中の場合は A4HYF_OK_LISTNOTCOMPLETE が戻され、この場合、項目数は部分的に構成されたリストの項目数を示します。リストの構成中にエラーが発生した場合は、A4HYF_E_LISTDATAERROR が戻されます。この場合、項目数はクライアントにキャッシュされている項目のみを表します。

注記

Activate からの戻り値が正常な場合、ナビゲーターはこの関数を呼び出して、配置するフォルダーのオブジェクトの数を取得します。この関数の呼び出し後、ナビゲーターはフォルダー内にあるオブジェクトの項目識別コードを取得するため、ItemAt を繰り返し呼び出します。

極端に大きなリストの場合は、リスト全体がクライアントのキャッシュに入れられる前に、Activate から戻るよう選択することができます。この場合、GetItemCount から A4HYF_OK_LISTNOTCOMPLETE を戻す必要があります。この時点より、A4HYF_OK_LISTCOMPLETE または A4HYF_E_LISTDATAERROR が戻されるまで、GetItemCount は iSeries ナビゲーターにより 10 秒間隔で呼び出されます。

IA4HierarchyFolder::GetToolBarInfo: 目的

iSeries ナビゲーター階層にある指定のフォルダーに関連するカスタム・ツールバーを記述する構造体を戻します。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetToolBarInfo(  
    LPCITEMIDLIST pidl,  
    LPVOID* ppvInfo  
);
```

パラメーター

pidl ツールバー情報を検索するオブジェクトを一意的に識別する ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

ppvInfo

戻されるデータ構造体。このポインターにおいては A4hyfToolBarInfo のインスタンスが戻される必要があります。この構造体は、オブジェクトのツールバー・ボタンの数、各ボタンの属性を含む TBBUTTON 構造体の配列のアドレス、およびプラグインのインスタンス・ハンドルを提供します。CWBA4HYF.H を参照してください。

戻りコード

正常な場合は `NOERROR` を、関数をインプリメントしないよう選択する場合は `E_NOTIMPL` を戻します。

注記

この関数は、ユーザーが `iSeries` ナビゲーターのプラグインに属するフォルダーまたはオブジェクトを選択するたびに呼び出されます。

戻された構造体に記憶域を割り振るには、`Windows IMalloc` インターフェースを使用します。この記憶域の削除は、ナビゲーターにより行います。

このメンバー関数がインプリメントされていない場合は、デフォルトの `iSeries` ナビゲーター・ツールバーが使用されます。このツールバーには、「コピー」、「貼り付け」、「削除」、「プロパティ」、4つのリスト・ビューのボタン、および「最新表示」が含まれます。`iSeries` ナビゲーターは、製品内の `IContextMenu::GetCommandString` (`GCS_VALIDATE` フラグを設定) のインプリメンテーションを呼び出し、オブジェクトに対して使用可能にするツールバー・ボタンを判別します。

IA4HierarchyFolder::GetListObject: 目的

完全修飾オブジェクト名を指定することにより、この関数は、プラグインが作成したキャッシュ内のプロキシ・オブジェクトへのポインターを戻します。これは、オプションのメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE GetListObject(  
    const char * lpszObjectName,  
    LPVOID* ppvObj  
);
```

パラメーター

lpszObjectName

リスト・オブジェクトが戻される完全修飾オブジェクト名。

ppvObj

インプリメンテーション定義のオブジェクトを指す戻りポインターです。呼び出しルーチンは、適切なオブジェクト・タイプにこのポインターをキャストする必要があります。

戻りコード

正常な場合は `NOERROR` を、関数をインプリメントしないよう選択する場合は `E_NOTIMPL` を戻します。

注記

`Activate` メソッドによりインスタンス化された「プロキシ」オブジェクトを取得するために、プラグイン・コードが `cwbUN_GetListObjectFromName` または `cwbUN_GetListObjectFromPidl` を呼び出すと、この関数への呼び出しが行われます。プラグインは、このプロキシ・オブジェクトを使用して `iSeries` サーバーのデータにアクセスするか、`iSeries` サーバー上でアクションを実行します。`IA4HierarchyFolder` のインプリメンテーションにより、プロキシ・オブジェクトのキャッシュが保持されるため、呼び出し側プログラムがオブジェクトを削除しないようにしてください。

IA4HierarchyFolder::ItemAt: 目的

フォルダー内容のリストの指定位置に、フォルダー・オブジェクトの SHITEMID (項目識別コード) 構造体として戻ります。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE ItemAt(  
    ULONG ulIndex,  
    LPITEMIDLIST* ppidl  
);
```

パラメーター

ulIndex

項目識別コードが要求される項目のゼロ・ベースの索引。

ppidl 要求された項目識別コードを受け取るポインターのアドレス。

戻りコード

正常な場合は NOERROR を、項目が使用できない場合は E_FAIL を戻します。項目識別コードに使用する記憶域が不足している場合は、E_OUTOFMEMORY を戻します。

注記

この関数は、リアルタイムでフォルダーを配置する際に、iSeries ナビゲーターにより繰り返し呼び出されます。したがって、長時間の実行操作は避けるようにしてください。iSeries ナビゲーターの項目識別コードの形式については、CWBA4HYF.H を参照してください。項目識別コードに記憶域を割り振るには、Windows IMalloc インターフェースを使用します。

IA4HierarchyFolder::ProcessTerminating: 目的

プラグインが永続データを保管できるようにするため、ユーザーが iSeries ナビゲーター・ウィンドウをクローズする際に、この関数を呼び出します。これは、オプションのメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE ProcessTerminating();
```

戻りコード

正常な場合は NOERROR を、関数をインプリメントしないよう選択する場合は E_NOTIMPL を戻します。戻りエラーは無視されます。

注記

なし

IA4HierarchyFolder::Refresh: 目的

キャッシュに入れられたフォルダー・オブジェクトを破棄し、iSeries サーバーから取得された新規データを使用して、キャッシュを再ビルドします。これは、必須のメンバー関数です。

構文

```
HRESULT STDMETHODCALLTYPE Refresh();
```

戻りコード

正常な場合は NOERROR を、フォルダーのオブジェクトへのアクセス時にエラーが発生した場合は A4HYF_E_LISTDATAERROR を戻します。

注記

メインの iSeries ナビゲーター・ウィンドウのグローバル最新表示が行われると、この関数が呼び出されません。

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

関連概念

97 ページの『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。

IA4PropSheetNotify::InformUserState: 目的

このユーザーが iSeries サーバー上に既に存在するか、あるいははじめて作成されるのかをインプリメンテーションに通知するために、IShellPropSheetExt インスタンスの作成直後に呼び出されます。

構文

```
HRESULT STDMETHODCALLTYPE InformUserState(  
    UINT wUserState  
);
```

パラメーター

wUserState

iSeries ユーザーの現在の状態。3 つの相互排他値のうちのいずれかを提供します。

IUS_NEWUSER

iSeries ナビゲーターのユーザーにより提供される属性に基づきユーザーを作成します。

IUS_NEWUSERBASEDON

既存の iSeries ユーザーの属性に基づきユーザーを作成します。

IUS_USEREXISTS

ユーザーが iSeries 上に存在します。

戻りコード

正常な場合は NOERROR を、一般エラーが発生した場合は E_FAIL を戻します。

注記

なし

IA4PropSheetNotify::ApplyChanges: 目的

ユーザーに属するデータを保管する必要のあることをインプリメンテーションに通知するために呼び出されます。

構文

```
HRESULT STDMETHODCALLTYPE ApplyChanges(  
    const char * pszNewUserName  
);
```

パラメーター

pszNewUserName

新規 iSeries ユーザーをはじめて作成する場合 (例えば InformUserState が IUS_USEREXISTS 以外の値を指定する場合)、そのユーザー名を指定します。

戻りコード

正常な場合は NOERROR を、一般エラーが発生した場合は E_FAIL を戻します。

注記

なし

IA4PropSheetNotify::GetErrorMessage: 目的

ApplyChanges でエラーが戻された際に、インプリメンテーションのエラー・メッセージ・テキストを検索するために呼び出されます。

構文

```
HRESULT STDMETHODCALLTYPE GetErrorMessage(
    char * pszErrMsg,
    UINT cchMax
);
```

パラメーター

pszErrMsg

NULL 終了のエラー・メッセージを受け取るバッファのアドレス。

cchMax

NULL 終了のエラー・メッセージを受け取るバッファのサイズ。

戻りコード

正常な場合は NOERROR を、メッセージ・テキストを検索できない、またはメッセージ・テキストが大きすぎてバッファに収まらない場合は E_FAIL を戻します。

注記

なし

iSeries ナビゲーター API

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。

iSeries ナビゲーター API のリスト:

機能別にリストおよびグループ化された iSeries ナビゲーター API。

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。	39 ページの『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。	40 ページの『cwbUN_GetSystemHandle』 41 ページの『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、開発者はユーザーが 1 つ以上の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。	42 ページの『cwbUN_CheckObjectAuthority』 42 ページの『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。	43 ページの『cwbUN_CheckAS400Name』

機能	iSeries ナビゲーター API
<p>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。</p>	<p>44 ページの 『cwbUN_GetUserAttribute』</p>
<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>45 ページの 『cwbUN_ConvertPidlToString』</p> <p>46 ページの 『cwbUN_GetDisplayNameFromItemId』</p> <p>46 ページの 『cwbUN_GetDisplayNameFromName』</p> <p>47 ページの 『cwbUN_GetDisplayPathFromName』</p> <p>48 ページの 『cwbUN_GetIndexFromItemId』</p> <p>48 ページの 『cwbUN_GetIndexFromName』</p> <p>49 ページの 『cwbUN_GetIndexFromPidl』</p> <p>49 ページの 『cwbUN_GetListObject』</p> <p>50 ページの 『cwbUN_GetParentFolderNameFromName』</p> <p>50 ページの 『cwbUN_GetParentFolderPathFromName』</p> <p>51 ページの 『cwbUN_GetParentFolderPidl』</p> <p>52 ページの 『cwbUN_GetSystemNameFromName』</p> <p>52 ページの 『cwbUN_GetSystemNameFromPidl』</p> <p>52 ページの 『cwbUN_GetSystemNameFromPidl』</p> <p>53 ページの 『cwbUN_GetTypeFromName』</p> <p>54 ページの 『cwbUN_GetTypeFromPidl』</p>
<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>54 ページの 『cwbUN_RefreshAll』</p> <p>55 ページの 『cwbUN_RefreshList』</p> <p>55 ページの 『cwbUN_RefreshListItems』</p> <p>56 ページの 『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>57 ページの 『cwbUN_GetODBCConnection』</p> <p>57 ページの 『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>58 ページの 『cwbUN_GetIconIndex』</p> <p>58 ページの 『cwbUN_GetSharedImageList』</p>

機能	iSeries ナビゲーター API
<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>59 ページの『cwbUN_GetAdminValue』</p> <p>60 ページの『cwbUN_GetAdminValueEx』</p> <p>61 ページの『cwbUN_GetAdminCacheState』</p> <p>62 ページの『cwbUN_GetAdminCacheStateEx』</p>
<p>インストール: 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インストールされているかどうかを判断することができます。</p>	<p>63 ページの『cwbUN_IsSubcomponentInstalled』</p>
<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。この接続機能により、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できます。接続機能は、iSeries Access for Windows 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>ストリングを使用する機能は、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ストリングを戻す機能も、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>63 ページの『cwbUN_OpenLocalLdapServer』</p> <p>64 ページの『cwbUN_FreeLocalLdapServer』</p> <p>65 ページの『cwbUN_GetLdapSvrPort』</p> <p>65 ページの『cwbUN_GetLdapSvrSuffixCount』</p> <p>66 ページの『cwbUN_GetLdapSvrSuffixName』</p> <p>67 ページの『cwbUN_OpenLdapPublishing』</p> <p>68 ページの『cwbUN_FreeLdapPublishing』</p> <p>68 ページの『cwbUN_GetLdapPublishCount』</p> <p>69 ページの『cwbUN_GetLdapPublishType』</p> <p>70 ページの『cwbUN_GetLdapPublishServer』</p> <p>71 ページの『cwbUN_GetLdapPublishPort』</p> <p>72 ページの『cwbUN_GetLdapPublishParentDn』</p>

cwbUN_GetSystemValue: 目的

iSeries システム値の値を含むストリングを戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetSystemValue(
    USHORT usSystemValueId,
    const char * szSystemName,
    char * szSystemValue,
    UINT cchMax
);
```

パラメーター

const char * szSystemValueId - input

検索する iSeries システム値を識別する数値。システム値定数の定義については、ヘッダー・ファイル「CWBA4SVL.H」に記されています。

char * szSystemValue - output

NULL 終了のシステム値ストリングを受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の値ストリングを受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_INTERNAL_ERROR

システム値を検索できませんでした。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

この API により戻される値は、NLS (各国語サポート) ストリングではなく、変換もされません。例えば、'*NONE' の代わりに 'None' が戻されます。

cwbUN_GetSystemHandle: 目的

この iSeries システムで使用するセキュリティー (SSL)、ユーザー ID、およびパスワード設定を含むシステム・ハンドルを戻します。このシステム・ハンドルには、入力 iSeries システム名用に iSeries ナビゲーターで構成された設定が割り振られます。

アプリケーション名が NULL に設定されている場合、戻されるシステム・ハンドルは固有のものになります。アプリケーション名が設定されている場合、このアプリケーション名に一致するシステム・ハンドルが戻されます。

アプリケーションがシステムの固有な OS/400 ジョブを要求する場合、アプリケーション名として NULL または固有名を渡す必要があります。

アプリケーションが OS/400 ジョブの共用を要求する場合は、この関数のすべての呼び出し元で同じアプリケーション名を渡す必要があります。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetSystemHandle(  
    char * szSystemName,  
    char * szAppName,  
    cwbCO_SysHandle * systemHandle  
);
```

パラメーター

char * szSystemName - input

システム・ハンドルを作成するシステムの名前を含む ASCIIZ ストリングへのポインター。

char * szAppName - input

12 文字以内の ASCIIZ ストリングへのポインター。これにより、単一のシステム・ハンドルを共用するアプリケーションを一意的に識別します。

cwbCO_SysHandle * systemHandle - output

このシステム名のシステム・ハンドルへのポインター。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_NULL_PARM

システム名が NULL でした。

CWBUN_INVALID_NAME_PARM

システム名が無効です。

CWB_NON_REPRESENTABLE_UNICODE_CHAR

1 つ以上の入力 UNICODE 文字が、使用中のコード・ページにおける表記を持ちません。

CWB_API_ERROR

システム・ハンドルを戻すことができませんでした。

使用法

iSeries Access for Windows API を使用して SSL をサポートするすべてのサード・パーティー・アプリケーションは、この関数を使用しなければなりません。例えば、すべての iSeries Access for Windows の通信 API には、SSL をサポートするシステム・ハンドルが必要です。

この関数の呼び出し元が、通信においてシステム・ハンドルを必要としなくなった場合、関数 **cwbUN_ReleaseSystemHandle** を呼び出してハンドルを解放することができます。

すべてのハンドルは、iSeries ナビゲーター・アプリケーション (cwbunnav.exe) が終了すると解放されます。

cwbUN_ReleaseSystemHandle: 目的

この iSeries[™] システムで使用するセキュリティー (SSL) 設定を含むシステム・ハンドルを解放します。システム・ハンドルは、**cwbUN_GetSystemHandle** 関数を使用して取得されます。

この関数の呼び出し元がハンドルへの最後の参照を持つ場合、実際にハンドル資源が破棄されます。

構文

```
CWBAPI unsigned int WINAPI cwbUN_ReleaseSystemHandle(  
    cwbCO_SysHandle * systemHandle  
);
```

パラメーター

cwbCO_SysHandle * systemHandle - input

cwbUN_GetSystemHandle 呼び出しで取得されたシステム・ハンドルへのポインター。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_API_ERROR

システム・ハンドルを解放できませんでした。

使用法

この関数の呼び出し元が、通信においてシステム・ハンドルを必要としなくなった場合、ハンドルを解放することができます。

cwbUN_CheckObjectAuthority: 目的

iSeries ナビゲーター・ユーザーが iSeries サーバー上の特定のオブジェクトに対して権限を持つかどうかを戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_CheckObjectAuthority(  
    const char * szObjectPath,  
    const char * szObjectType,  
    const char * szAuthorityType,  
    const char * szSystemName  
);
```

パラメーター

const char * szObjectPath - input

権限を検査する対象の iSeries オブジェクト・パス。

const char * szObjectType - input

権限を検査する対象のオブジェクトの iSeries オブジェクト・タイプ。例えば、*DTAQ。

const char * szAuthorityType - input

検査する iSeries オブジェクト権限。以下の値が使用可能です。

複数の権限を検査する場合、権限を連結する必要があります (例、*OBJMGT*OBJEXIST)。単一の呼び出しで、最大 11 の権限タイプを指定することができます。オブジェクトに対し、指定された権限をユーザーがすべて所有している場合にのみ、この関数は CWB_OK を戻します。

const char * szSystemName - input

検査を実行する iSeries システムの名前。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

オブジェクトに対し、ユーザーは指定された権限を所有しています。

CWBUN_USER_NOT_AUTHORIZED

ユーザーは指定された権限を所有していません。

CWBUN_OBJECT_NOT_FOUND

指定されたオブジェクトを検査できませんでした。

CWBUN_INTERNAL_ERROR

オブジェクト権限を検査できませんでした。

使用法

*EXCLUDE が権限として指定されている場合は、その他の権限タイプを指定することはできません。

*AUTLMGT は、「szObjectType」が *AUTL の場合にのみ有効です。

cwbUN_CheckSpecialAuthority: 目的

iSeries ナビゲーター・ユーザーが iSeries サーバー上で特定の特殊権限を持つかどうかを戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_CheckSpecialAuthority(  
    const char * szSpecialAuthority,  
    const char * szSystemName  
);
```

パラメーター

const char * szSpecialAuthority - input

検査する iSeries サーバー特殊権限。以下の値が使用可能です。

const char * szSystemName - input

検査を実行する iSeries システムの名前。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

ユーザーは指定された特殊権限を所有しています。

CWBUN_USER_NOT_AUTHORIZED

ユーザーは指定された権限を所有していません。

CWBUN_INTERNAL_ERROR

特殊権限を検査できませんでした。

使用法

なし

cwbUN_CheckAS400Name: 目的

指定されたストリングが iSeries サーバー上の有効な名前パラメーターであるかどうかを戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_CheckAS400Name(  
    const char * szAS400Name,  
    const char * szSystemName,  
    USHORT usTypeId  
);
```

パラメーター

const char * szAS400Name - input

妥当性を検査する iSeries 名。

const char * szSystemName - input

検査を実行する iSeries システムの名前。

USHORT usTypeId - input

入力ストリングの解釈方法を指示する数値。解釈の方法には、長いオブジェクト名、短いオブジェクト名、通信名、またはストリングがあります (通常、型や定数定義のインクルード・ファイルへのリンクが記述される)。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_NAME_TOO_LONG

名前が長すぎます。

CWBUN_NAME_NULLSTRING

ストリングが空です。文字がありません。

CWBUN_NAME_INVALIDCHAR

無効な文字です。

CWBUN_NAME_STRINGTOOLONG

ストリングが長すぎます。

CWBUN_NAME_MISSINGENDQUOTE

終了の引用符がありません。

CWBUN_NAME_INVALIDQUOTECHAR

引用符ストリングには無効な文字です。

CWBUN_NAME_ONLYBLANKS

ブランクのみのストリングが検出されました。

CWBUN_NAME_STRINGTOOSHORT

ストリングが短すぎます。

CWBUN_NAME_TOOLONGFORIBM

ストリングには問題がありませんが、IBM コマンドとして長すぎます。

CWBUN_NAME_INVALIDFIRSTCHAR

最初の文字が無効です。

使用法

なし

cwbUN_GetUserAttribute: 目的

現在の iSeries ナビゲーター・ユーザーのユーザー・プロファイル属性の値を含むストリングを戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetUserAttribute(  
    USHORT usAttributeId,  
    const char * szSystemName,  
    char * szValue,  
    UINT cchMax  
);
```

パラメーター

USHORT usAttributeId - input

検索するユーザー属性値を識別する数値。ユーザー属性定数の定義については、ヘッダー・ファイル「CWBA4USR.H」に記されています。

const char * szSystemName - input

ユーザー属性を検索する iSeries システムの名前。

char * szValue - output

NULL 終了の属性値ストリングを受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の値ストリングを受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_INTERNAL_ERROR

属性値を検索できませんでした。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

この API により戻される値は、NLS ストリングではなく、変換もされません。例えば、'*NONE' の代わりに 'None' が戻されます。

cwbUN_ConvertPidlToString: 目的

iSeries ナビゲーターの項目識別コード・リストを、完全修飾オブジェクト名に変換します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_ConvertPidlToString(  
    LPCITEMIDLIST pidl,  
    char * szObjectName,  
    UINT cchMax  
);
```

パラメーター

LPCITEMIDLIST pidl - input

変換する ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

char * szObjectName - output

NULL 終了のオブジェクト名を受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了のオブジェクト名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コード・リストが無効です。

WB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetDisplayNameFromItemId: 目的

Unity 項目識別コードから項目名フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetDisplayNameFromItemId(  
    const char * szItemId,  
    char * szItemName,  
    UINT cchMax  
);
```

パラメーター

const char * szItemId - input

項目名を取り出す Unity 項目識別コード。

char * szItemName - output

NULL 終了の項目名を受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の項目名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コードが無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetDisplayNameFromName: 目的

完全修飾 Unity オブジェクト名から項目名フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetDisplayNameFromName(  
    const char * szObjectName,  
    char * szItemName,  
    UINT cchMax  
);
```

パラメーター

const char * szObjectName - input

項目名を取り出す Unity オブジェクト名。

char * szItemName - output
NULL 終了の項目名を受け取るバッファのアドレス。

UINT cchMax - input
NULL 終了の項目名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK
正常終了。

CWBUN_FORMAT_NOT_VALID
指定されたオブジェクト名が無効です。

CWB_BUFFER_OVERFLOW
バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetDisplayPathFromName: 目的

完全修飾 Unity オブジェクト名を、ユーザーへの表示に適している完全修飾パス名に変換します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetDisplayPathFromName(  
    const char * szObjectName,  
    char * szPathName,  
    UINT cchMax  
);
```

パラメーター

const char * szObjectName - input
パス名の派生元の Unity オブジェクト名。

char * szPathName - output
NULL 終了のパス名を受け取るバッファのアドレス。

UINT cchMax - input
NULL 終了のパス名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK
正常終了。

CWBUN_FORMAT_NOT_VALID
指定されたオブジェクト名が無効です。

CWB_BUFFER_OVERFLOW
バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetIndexFromItemId: 目的

Unity 項目識別コードから項目索引フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetIndexFromItemId(  
    const char * szItemId,  
    ULONG* piIndex  
);
```

パラメーター

const char * szItemId - input

項目索引を取り出す Unity 項目識別コード。

ULONG* piIndex - output

項目索引を受け取る符号なし長整数のアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コードが無効です。

使用法

なし

cwbUN_GetIndexFromName: 目的

完全修飾 Unity オブジェクト名から項目索引フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetIndexFromName(  
    const char * szObjectName,  
    ULONG* piIndex  
);
```

パラメーター

const char * szObjectName - input

項目索引を取り出す Unity オブジェクト名。

ULONG* piIndex - output

項目索引を受け取る符号なし長整数のアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定されたオブジェクト名が無効です。

使用法

なし

cwbUN_GetIndexFromPidl: 目的

完全修飾 Unity 項目識別コード・リストから項目索引フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetIndexFromPidl(  
    LPCITEMIDLIST pidl,  
    ULONG* piIndex  
);
```

パラメーター

LPCITEMIDLIST pidl - input

項目索引が取り出される ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

ULONG* piIndex - output

項目索引を受け取る符号なし長整数のアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コード・リストが無効です。

使用法

なし

cwbUN_GetListObject: 目的

指定されたリスト・オブジェクト名に関連したオブジェクトへのポインターを取得します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetListObject(  
    const char * szFileName,  
    LPVOID *pListObject  
);
```

パラメーター

const char * szFileName - input

オブジェクト・ポインターを検出して戻す対象の Unity オブジェクト名。

LPVOID pListObject - output

要求 Unity オブジェクトへのポインターのアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

使用法

なし

cwbUN_GetParentFolderNameFromName: 目的

完全修飾 Unity オブジェクト名からオブジェクトの親フォルダーの名前を取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetParentFolderNameFromName(  
    const char * szObjectName,  
    char * szParentFolderName,  
    UINT cchMax  
);
```

パラメーター

const char * szObjectName - input

親フォルダー名を取り出す Unity オブジェクト名。

char * szParentFolderPath - output

NULL 終了の親フォルダー名を受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の親フォルダー名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定されたオブジェクト名が無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetParentFolderPathFromName: 目的

完全修飾 Unity オブジェクト名を指定することにより、オブジェクトの親フォルダーの完全修飾オブジェクト名を戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetParentFolderPathFromName(  
    const char * szObjectName,  
    char * szParentFolderPath,  
    UINT cchMax  
);
```

パラメーター

const char * szObjectName - input

親フォルダー・オブジェクト名を取り出す Unity オブジェクト名。

char * szParentFolderPath - output

NULL 終了の親フォルダー・オブジェクト名を受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の親フォルダー・オブジェクト名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定されたオブジェクト名が無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻り文字列を含むことができません。

使用法

なし

cwbUN_GetParentFolderPidl: 目的

完全修飾 Unity 項目識別コード・リストを指定することにより、オブジェクトの親フォルダーの完全修飾項目識別コード・リストを戻します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetParentFolderPidl(  
    LPCITEMIDLIST pidl,  
    LPITEMIDLIST *ppidl  
);
```

パラメーター

LPCITEMIDLIST pidl - input

親フォルダー項目識別コード・リストを取り出す ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

LPITEMIDLIST* ppidl - output

親フォルダー項目識別コード・リストを受け取る項目識別コード・リスト・ポインターのアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コード・リストが無効です。

使用法

なし

cwbUN_GetSystemNameFromName: 目的

完全修飾 Unity オブジェクト名から iSeries システム名を取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetSystemNameFromName(  
    const char * szObjectName,  
    char * szSystemName,  
    UINT cchMax  
);
```

パラメーター

const char * szObjectName - input

システム名を取り出す Unity オブジェクト名。

char * szSystemName - output

NULL 終了のシステム名を受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了のシステム名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定されたオブジェクト名が無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetSystemNameFromPidl: 目的

完全修飾 Unity 項目識別コード・リストから iSeries システム名を取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetSystemNameFromPidl(  
    LPCITEMIDLIST pidl,  
    char * szSystemName,  
    UINT cchMax  
);
```

パラメーター

LPCITEMIDLIST pidl - input

システム名を取り出す ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

char * szSystemName - output

NULL 終了のシステム名を受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了のシステム名を受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コード・リストが無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetTypeFromName: 目的

完全修飾 Unity オブジェクト名から項目タイプ・フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetTypeFromName(  
    const char * szObjectName,  
    char * szType,  
    UINT cchMax  
);
```

パラメーター

const char * szObjectName - input

項目索引を取り出す Unity オブジェクト名。

char * szType - output

NULL 終了の項目タイプを受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の項目タイプを受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定されたオブジェクト名が無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_GetTypeFromPidl: 目的

完全修飾 Unity 項目識別コード・リストから項目索引フィールドを取り出します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetTypeFromPidl(  
    LPCITEMIDLIST pidl,  
    char * szType,  
    UINT cchMax  
);
```

パラメーター

LPCITEMIDLIST pidl - input

項目索引が取り出される ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

char * szType - output

NULL 終了の項目タイプを受け取るバッファのアドレス。

UINT cchMax - input

NULL 終了の項目タイプを受け取るバッファのサイズ。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_FORMAT_NOT_VALID

指定された項目識別コード・リストが無効です。

CWB_BUFFER_OVERFLOW

バッファが小さすぎるため、戻りストリングを含むことができません。

使用法

なし

cwbUN_RefreshAll: 目的

iSeries ナビゲーターのツリー・ウィンドウおよびリスト・ウィンドウの内容を最新表示します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_RefreshAll(  
    const char * pszStatusText  
);
```

パラメーター

const char * pszStatusText - input

完了時に、ステータス・バー・ウィンドウに配置される NULL 終了ストリング。このパラメーターは、NULL にすることができます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_WINDOW_NOTAVAIL

ビュー・ウィンドウを検出できませんでした。

使用法

この関数は、ユーザーにより要求されたアクションの実行後に、iSeries ナビゲーターの内容全体を最新表示するために使用します。

cwbUN_RefreshList: 目的

iSeries ナビゲーターのリスト・ビュー・ウィンドウの内容を最新表示します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_RefreshList(  
    const char * pszStatusText  
);
```

パラメーター

const char * pszStatusText - input

完了時に、ステータス・バー・ウィンドウに配置される NULL 終了ストリング。このパラメーターは、NULL にすることができます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_WINDOW_NOTAVAIL

リスト・ビュー・ウィンドウを検出できませんでした。

使用法

この関数は、ユーザーにより要求されたアクションの実行後に、リスト・ウィンドウの内容を最新表示するために使用します。

cwbUN_RefreshListItems: 目的

iSeries ナビゲーターのリスト・ビュー・ウィンドウで現在選択されている項目を最新表示します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_RefreshListItems(  
    const char * pszStatusText  
);
```

パラメーター

const char * pszStatusText - input

完了時に、ステータス・バー・ウィンドウに配置される NULL 終了ストリング。このパラメーターは、NULL にすることができます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_WINDOW_NOTAVAIL

リスト・ビュー・ウィンドウを検出できませんでした。

使用法

この関数は、ユーザーにより要求されたアクションの実行後に、リスト・ウィンドウ内の選択項目を最新表示するために使用します。

cwbUN_UpdateStatusBar: 目的

iSeries ナビゲーター・ウィンドウのステータス・バーにテキスト・ストリングを挿入します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_UpdateStatusBar(  
    const char * pszStatusText  
);
```

パラメーター

const char * pszStatusText - input

完了時に、ステータス・バー・ウィンドウに配置される NULL 終了ストリング。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_WINDOW_NOTAVAIL

ステータス・バー・ウィンドウを検出できませんでした。

使用法

この関数は、ダイアログの「OK」ボタンをクリックして要求されたアクションが正常に完了したことをユーザーに通知するために使用します。

cwbUN_GetODBCConnection: 目的

指定された iSeries サーバーのサーバー・システムにおける ODBC 接続へのハンドルを戻します。指定された iSeries サーバーに対する接続が存在しない場合、API は新しいハンドルを取得します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetODBCConnection(  
    const char * szSystemName,  
    HDBC *phDBC  
);
```

パラメーター

const char * szSystemName - input

ODBC 接続を検索する iSeries システムの名前。

HDBC *phDBC - output

ODBC 接続ハンドルを戻すアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

使用法

なし

cwbUN_EndODBCConnections: 目的

cwbUN_GetODBCConnection によって以前に開かれたすべての ODBC 接続を終了します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_EndODBCConnections(  
);
```

パラメーター

なし

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

ハンドルが作成されませんでした。

使用法

EndODBCConnections 関数は、**GetODBCConnection** 関数を使用して開いた接続のみを閉じることに注意してください。**EndODBCConnections** 関数は、直接またはその他のインターフェースを使用して開いた ODBC 接続は認識しません。

また、拡張機能中のコードで **GetODBCConnection** を使用する場合、アプリケーション拡張機能のフォルダのデストラクターが必ず **EndODBCConnections** を呼び出すようにしてください。

cwbUN_GetODBCConnection も参照してください。

cwbUN_GetIconIndex: 目的

指定されたアイコンのイメージ・リストの索引を取得します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetIconIndex(  
    LPCITEMIDLIST pidl,  
    UINT uFlags,  
    int* piIndex  
);
```

パラメーター

LPCITEMIDLIST pidl - input

参照するアイコンの識別に使用される ITEMIDLIST (項目識別コード・リスト) 構造体へのポインター。

UINT uFlags - input

検索するアイコン索引タイプの指定 (上記に定義)。以下のフラグ・タイプが使用できます。

int * piIndex - output

アイコン索引を受け取る整数のアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_INVALID_FLAG_VALUE

サポートされる有効なフラグ値ではありません。

使用法

なし

cwbUN_GetSharedImageList: 目的

iSeries ナビゲーターに関連したアイコン・イメージ・リストを検索します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetSharedImageList(  
    UINT uFlags,  
    HIMAGELIST *phImageList  
);
```

パラメーター

UINT uFlags - input

検索するイメージ・リスト・タイプの指定 (上記に定義)。以下のフラグ・タイプが使用可能です。

HIMAGELIST* phImageList -

イメージ・リスト・ハンドルを受け取る変数のアドレス。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWBUN_INVALID_FLAG_VALUE

サポートされる有効なフラグ値ではありません。

CWBUN_CANT_GET_IMAGELIST

アイコン・イメージ・リストの取得を試みた際に障害が発生しました。

使用法

なし

cwbUN_GetAdminValue: 目的

この API は、指定された iSeries サーバー上の現行の iSeries ナビゲーター・ユーザーによる特定の管理機能の使用を認めるか、拒否するかを戻します。**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。

例えば、アプリケーション管理サブコンポーネントを使用することにより、管理者は iSeries ナビゲーターのいくつかの機能に対してユーザーのアクセスを制御することができます。これらの機能の 1 つにジョブ管理があります。cwbUN_GetAdminValue API を使用することにより、ジョブ管理に対応する管理機能の名前を指定して、現行の iSeries ナビゲーター・ユーザーにジョブ管理機能の使用を認めるかどうかを方針に基づいて決めることができます。iSeries ナビゲーターでサポートされている管理機能の名前のリストについては、cwbunpla.h ヘッダー・ファイルを参照してください。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetAdminValue(  
    const char * szSystemName,  
    char* adminFunction,  
    cwbUN_Usage& usageValue);
```

パラメーター

const char * szSystemName

検査を実行する iSeries システムの名前。

char* adminFunction

管理機能の名前を含む ASCII ストリングへのポインター。ストリングは NULL 終了である必要があり、最大長は 30 バイト + 1 バイト (NULL 終了文字) です。サポートされる入力値については、cwbunpla.h を参照してください。

cwbUN_Usage & usageValue

この値は、CWB_OK の戻りコードが戻される場合のみ有効です。以下の 2 つの値のいずれかが戻されます。

- cwbUN_granted -- ユーザーは機能の使用を許可されます。
- cwbUN_denied -- ユーザーは機能の使用を拒否されます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

API は正常です。

CWBSY_USER_CANCELLED

ユーザーが、API により示されたユーザー ID とパスワードを取り消しました。

使用法

この API は、特定の iSeries サーバーの現行の iSeries ナビゲーター・ユーザーに指定機能の使用を認めるかどうかを判別します。指定された iSeries サーバーに、現在、ユーザーがサインオンしていない場合、API によりユーザーがサインオンされ、ユーザー ID とパスワード・プロンプトが表示されます。

この API は、iSeries ナビゲーターまたはクライアント・アプリケーションの機能カテゴリーにある管理機能の検査にのみ使用することができます。

cwbUN_GetAdminValueEx: 目的

この API は、指定された iSeries サーバー上の現行のユーザーが特定の管理機能の使用を許可されているかどうかを戻します。**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。

注: iSeries ナビゲーターのプラグインは、cwbUN_GetAdminValueEx ではなく cwbUN_GetAdminValue API を使用する必要があります。

例えば、アプリケーション管理サブコンポーネントを使用することにより、管理者は iSeries ナビゲーターのいくつかの機能に対してユーザーのアクセスを制御することができます。これらの機能の 1 つに「ジョブ管理」があります。cwbUN_GetAdminValueEx API を使用することにより、ジョブ管理に対応する管理機能の名前を指定して、現行のユーザーがジョブ管理機能を使用できるかどうかを方針に基づいて決定することができます。iSeries ナビゲーターでサポートされている管理機能の名前のリストについては、CWBUNPLA.H ヘッダー・ファイルを参照してください。

この API は cwbUN_GetAdminValue と同じ機能を提供しますが、システム名の代わりにシステム・オブジェクト・ハンドルを受け入れる点が異なります。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetAdminValueEx(  
    cwbCO_SysHandle* pSysHandle,  
    char* adminFunction,  
    cwbUN_Usage& usageValue);
```

パラメーター

cwbCO_SysHandle* pSysHandle

システム・オブジェクト・ハンドルへのポインター。この API への呼び出しの前に、システム・オブジェクト中にシステム名を指定する必要があります。cwbUN_GetAdminValueEx API の振る舞いは、システム・オブジェクトが iSeries サーバーへのサインオンを取得したかどうかにより異なります。

サインオンしていない場合 ->

cwbUN_GetAdminValueEx は iSeries サーバーにサインオンをします。ユーザーの前のアプリケーション管理設定がクライアント PC にキャッシュされていない場合は、iSeries サーバーからダウンロードされます。

サインオンしている場合 ->

システム・オブジェクトが iSeries サーバーにサインオンしており、iSeries ユーザー ID とパスワードの妥当性検査を行うよう指定している場合は (妥当性検査モード)、cwbUN_GetAdminValueEx API は、サインオンが完了した時点で正確であったアプリケーション管理設定のスナップショットを使用します。ユーザー ID とパスワードの妥当性検査を行わずにサインオンした場合は、cwbUN_GetAdminValueEx は、アプリケーション管理設定のコピー (24 時間以内のもの) を使用します。

char* adminFunction

管理機能の名前を含む ASCII スtringへのポインター。Stringは NULL 終了である必要があり、最大長は 30 バイト + 1 バイト (NULL 終了文字) です。サポートされる入力値については、CWBUNPLA.H を参照してください。

cwbUN_Usage& usageValue

この値は、CWB_OK の戻りコードが戻される場合にのみ有効です。以下の 2 つの値のいずれかが戻されます。

cwbUN_granted

ユーザーは機能の使用を許可されます。

cwbUN_denied

ユーザーは機能の使用を拒否されます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

API は正常です。

CWBSY_USER_CANCELLED

ユーザーが、API により示されたユーザー ID とパスワードを取り消しました。

使用法

この API は、入力システム・オブジェクトによって定義された現行の iSeries ユーザーが指定された機能の使用を許可されているかどうかを判別します。指定された iSeries サーバーに、現在、ユーザーがサインオンしていない場合、API によりユーザーがサインオンされ、ユーザー ID とパスワード・プロンプトが表示されます。

この API は、iSeries ナビゲーターまたはクライアント・アプリケーションの機能カテゴリーにある管理機能の検査にのみ使用することができます。

cwbUN_GetAdminCacheState: 目的

この API は、長時間かかるであろう cwbUN_GetAdminValue API の次回の呼び出しを行うかどうかを指示します。cwbUN_GetAdminValue API は、PC 上のデータをキャッシュに入れます。キャッシュが現行のものでない場合、cwbUN_GetAdminValue はキャッシュを更新するためサインオン・プロンプトを表示するか、その他の処理を実行します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetAdminCacheState(  
    const char * szSystemName,  
    cwbUN_State& adminState);
```

パラメーター

const char * szSystemName

検査を実行する iSeries システムの名前。

cwbUN_State& adminState

長時間かかるであろう cwbUN_GetAdminValue API の次回の呼び出しを行うか、内部キャッシュを使用してホスト iSeries サーバーにアクセスせずに戻るかを指示します。

以下の 3 つの値のいずれかが戻されます。

cwbUN_logon

指定された iSeries サーバーには現行ユーザーが存在しません。cwbUN_GetAdminValue API はサインオン・プロンプトを示すことができます。

cwbUN_refresh

cwbUN_GetAdminValue は iSeries サーバーにアクセスし、内部キャッシュを更新します。

cwbUN_cache

cwbUN_GetAdminValue は現行のキャッシュを持っており、長時間実行することはできません。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

API は正常です。

使用法

この API は、cwbUN_GetAdminValue のユーザーが、長時間かかる cwbUN_GetAdminValue の次回の呼び出しを実行するかどうかを判別する場合に使用できます。

cwbUN_GetAdminCacheStateEx: 目的

この API は、長時間かかるであろう cwbUN_GetAdminValueEx API の次回の呼び出しを行うかどうかを指示します。cwbUN_GetAdminValueEx API は、PC 上のデータをキャッシュに入れます。キャッシュが現行のものでない場合、cwbUN_GetAdminValueEx API はキャッシュを更新するためサインオン・プロンプトを表示するか、その他の処理を実行します。

構文

```
CWBAPI unsigned int WINAPI cwbUN_GetAdminCacheStateEx(  
    cwbCO_SysHandle* pSysHandle,  
    cwbUN_State& adminState);
```

パラメーター

cwbCO_SysHandle* pSysHandle - input

システム・オブジェクト・ハンドルへのポインター。この API への呼び出しの前に、システム・オブジェクト中にシステム名を指定する必要があります。

cwbUN_State& adminState

長時間かかるであろう `cwbUN_GetAdminValue` API の次回の呼び出しを行うか、内部キャッシュを使用してホスト iSeries サーバーにアクセスせずに戻るかを指示します。

以下の 3 つの値のいずれかが戻されます。

cwbUN_logon

指定された iSeries サーバーには現行ユーザーが存在しません。`cwbUN_GetAdminValue` API はサインオン・プロンプトを示すことができます。

cwbUN_refresh

`cwbUN_GetAdminValue` は iSeries サーバーにアクセスし、内部キャッシュを更新します。

cwbUN_cache

`cwbUN_GetAdminValue` は現行のキャッシュを持っており、長時間実行することはできません。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

API は正常です。

使用法

この API は、`cwbUN_GetAdminValueEx` のユーザーが、長時間かかる `cwbUN_GetAdminValueEx` の次回の呼び出しを実行するかどうかを判別する場合に使用できます。

cwbUN_IsSubcomponentInstalled: 目的

iSeries ナビゲーターのサブコンポーネントが PC にインストールされているかどうかを判別します。

構文

```
CWBAPI BOOL WINAPI cwbUN_IsSubcomponentInstalled(  
    UNIT uOption);
```

パラメーター

UNIT uOption

検査する iSeries ナビゲーター・サブコンポーネントを指定します。サポートされる値のリストについては、`cwbun.h` にある API のまえがきを参照してください。

戻りコード

ブール値を戻します。

TRUE サブコンポーネントがインストールされています。

FALSE

サブコンポーネントがインストールされていません。

使用法

なし。

cwbUN_OpenLocalLdapServer: 目的

iSeries サーバー上のディレクトリー・サービス (LDAP) サーバーに関する構成情報へのアクセスに使用可能なハンドルを作成します。

構文

```
int cwbUN_OpenLocalLdapServerW
( LPCWSTR      system,
  cwbUN_ldapSvrHandle *pHandle
);

int cwbUN_OpenLocalLdapServerA
( LPCSTR      system,
  cwbUN_ldapSvrHandle *pHandle
);
```

パラメーター

LPCSTR system - input

iSeries システム名へのポインター。

cwbUN_ldapSvrHandle *pHandle - output

戻される際に、以下の API で使用可能なハンドルが含まれます。

- cwbUN_FreeLocalLdapServer
- cwbUN_GetLdapSvrPort
- cwbUN_GetLdapSvrSuffixCount
- cwbUN_GetLdapSuffixName

注: このハンドルは、cwbUN_FreeLocalLdapServer への呼び出しにより開放する必要があります。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_PARAMETER

無効なパラメーターが指定されています。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

CWBUN_LDAP_NOT_AVAIL

ディレクトリー・サービスがインストールされていないか、サーバーが構成されていません。

使用法

なし

cwbUN_FreeLocalLdapServer: 目的

入力ハンドルに関連した資源を解放します。

構文

```
int cwbUN_FreeLocalLdapServer
( cwbUN_ldapSvrHandle handle
);
```

パラメーター

cwbUN_LdapSvrHandle handle - input
資源を解放する対象となるハンドル。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK
正常終了。

CWB_INVALID_API_HANDLE
cwbUN_OpenLocalLdapServer() によりハンドルが作成されませんでした。

使用法

ハンドルは、cwbUN_OpenLocalLdapServer 呼び出しを実行して取得します。

cwbUN_GetLdapSvrPort: 目的

LDAP サーバーにより使用されるポート番号を戻します。

構文

```
int cwbUN_GetLdapSvrPort
( cwbUN_LdapSvrHandle handle,
  int *port,
  int *sslPort
);
```

パラメーター

cwbUN_LdapSvrHandle handle - input
cwbUN_OpenLocalLdapServer() への呼び出しにより以前に取得されたハンドル。

int * port - output
LDAP 接続に使用するポート番号。

int * sslPort - output
SSL 接続に使用するポート番号。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK
正常終了。

CWB_INVALID_API_HANDLE
無効なハンドル。

CWB_INVALID_POINTER
NULL ポインターが指定されました。

使用法

なし

cwbUN_GetLdapSvrSuffixCount: 目的

このサーバーに対して構成されたサフィックスの数を戻します。サフィックスは、ディレクトリー・ツリー内の開始点の識別名 (DN) です。

構文

```
int cwbUN_GetLdapSvrSuffixCount
( cwbUN_ldapSvrHandle handle,
  int *count
);
```

パラメーター

cwbUN_ldapSvrHandle handle - input

cwbUN_OpenLocalLdapServer() への呼び出しにより以前に取得されたハンドル。

int * count - output

サーバー上に存在するサフィックスの数。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

使用法

なし

cwbUN_GetLdapSvrSuffixName: 目的

サフィックスの識別名を戻します。

構文

```
int cwbUN_GetLdapSuffixNameA
( cwbUN_ldapSvrHandle handle,
  int index,
  LPSTR suffix,
  int *length
);
```

```
int cwbUN_GetLdapSuffixNameW
( cwbUN_ldapSvrHandle handle,
  int index,
  LPWSTR suffix,
  int *length
);
```

```
int cwbUN_GetLdapSuffixName8 /* returns suffix in UTF-8 */
( cwbUN_ldapSvrHandle handle,
  int index,
  LPSTR suffix,
  int *length
);
```

パラメーター

cwbUN_LdapSuffixHandle handle - input

cwbUN_OpenLocalLdapServer() への呼び出しにより以前に取得されたハンドル。

int index - input

サフィックスのゼロ・ベースの索引。この値は、cwbUN_GetLdapSvrSuffixCount() が戻すカウントより小さくなければなりません。

LPSTR suffix - output

サフィックスの識別名を含むバッファーへのポインター。

int * length - input/output

サフィックス・バッファーの長さへのポインター。バッファーが小さいためにストリング (NULL を含む) を保持できない場合、必要なバッファーのサイズがこのパラメーターに設定されます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_API_PARAMETER

無効な索引。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

CWB_BUFFER_OVERFLOW

サフィックス・バッファーが小さいため、結果全体を保持できません。

使用法

なし

cwbUN_OpenLdapPublishing: 目的

この iSeries サーバーにより LDAP ディレクトリーに公開された情報に関する構成情報へのアクセスに使用可能なハンドルを作成します。

構文

```
int cwbUN_OpenLdapPublishingW
( LPCWSTR      system,
  cwbUN_LdapPubHandle *pHandle
);
```

```
int cwbUN_OpenLdapPublishingA
( LPCSTR      system,
  cwbUN_LdapPubHandle *pHandle
);
```

パラメーター

LPCSTR system - input

iSeries システム名へのポインター。

cwbUN_ldapSvrHandle *pHandle - output

戻される際に、以下の API で使用可能なハンドルが含まれます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_PARAMETER

無効なパラメーターが指定されています。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

CWBUN_LDAP_NOT_AVAIL

ディレクトリー・サービスがインストールされていないか、サーバーが構成されていません。

使用法

なし

cwbUN_FreeLdapPublishing: 目的

入力ハンドルに関連した資源を解放します。

構文

```
int cwbUN_FreeLdapPublishing
( cwbUN_ldapPubHandle handle
);
```

パラメーター

cwbUN_ldapPubHandle handle - input

資源を解放する対象となるハンドル。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

cwbUN_OpenLdapPublishing() によりハンドルが作成されませんでした。

使用法

ハンドルは、cwbUN_OpenLdapPublishing() 呼び出しを実行して取得します。

cwbUN_GetLdapPublishCount: 目的

この iSeries サーバー用に構成された公開レコードの数を戻します。公開レコードは、公開する情報のカテゴリー、およびこれを公開する方法と場所を識別します。

構文

```
int cwbUN_GetLdapPublishCount
( cwbUN_ldapPubHandle handle,
  int *count
);
```

パラメーター

cwbUN_ldapPubHandle handle - input

cwbUN_OpenLdapPublishing() への呼び出しにより以前に取得されたハンドル。

int * count - output

iSeries サーバーに構成されている公開レコードの数。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

使用法

なし

cwbUN_GetLdapPublishType: 目的

この公開レコードが対象とする情報のタイプを戻します。

構文

```
int cwbUN_GetLdapPublishType
( cwbUN_ldapPubHandle handle,
  int index,
  cwbUN_LdapPubCategories *information
);
```

パラメーター

cwbUN_ldapPubHandle handle - input

cwbUN_OpenLdapPublishing() への呼び出しにより以前に取得されたハンドル。

int index - input

公開レコードのゼロ・ベースの索引。この値は、cwbUN_GetLdapPublishCount() が戻すカウントより小さくなければなりません。

cwbUN_LdapPubCategories * information - output

この公開レコードが対象とする情報のタイプ。以下の値が使用可能です。

CWBUN_LDAP_PUBLISH_USERS

ユーザー情報

CWBUN_LDAP_PUBLISH_COMPUTERS

iSeries システム

CWBUN_LDAP_PUBLISH_NETWORK_INVENTORY

NetFinity

CWBUN_LDAP_PUBLISH_PRINTERS

iSeries プリンター

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_API_PARAMETER

無効な索引。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

使用法

なし

cwbUN_GetLdapPublishServer: 目的

この情報の公開先のサーバーの名前を戻します。

構文

```
int cwbUN_GetLdapPublishServerW  
( cwbUN_ldapPubHandle handle,  
  int index,  
  LPWSTR server,  
  int *length  
);
```

```
int cwbUN_GetLdapPublishServerA  
( cwbUN_ldapPubHandle handle,  
  int index,  
  LPSTR server,  
  int *length  
);
```

パラメーター

cwbUN_ldapPubHandle handle - input

cwbUN_OpenLdapPublishing() への呼び出しにより以前に取得されたハンドル。

int index - input

公開レコードのゼロ・ベースの索引。この値は、cwbUN_GetLdapPublishCount() が戻すカウントより小さくなければなりません。

LPSTR server - output

サーバーの名前を含むバッファーへのポインター。

int * length - input/output

サーバー・バッファの長さへのポインター。バッファが小さいためにストリング (NULL を含む) を保持できない場合、必要なバッファのサイズがこのパラメーターに設定されます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_API_PARAMETER

無効な索引。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

CWB_BUFFER_OVERFLOW

サフィックス・バッファが小さいため、結果全体を保持できません。

使用法

なし

cwbUN_GetLdapPublishPort: 目的

この情報を公開するために使用されるサーバーのポート番号を戻します。

構文

```
int cwbUN_GetLdapPublishPort
( cwbUN_ldapPubHandle   handle,
  int                   index,
  int                   *port,
  cwbUN_LdapCnnSecurity *connectionSecurity
);
```

パラメーター**cwbUN_ldapPubHandle handle - input**

cwbUN_OpenLdapPublishing() への呼び出しにより以前に取得されたハンドル。

int index - input

公開レコードのゼロ・ベースの索引。この値は、cwbUN_GetLdapPublishCount() が戻すカウントより小さくなければなりません。

int * port - output

サーバーへの接続に使用されるポート番号。

cwbUN_LdapCnnSecurity * connectionSecurity - output

サーバーへの接続に使用される接続タイプ。これは、関連ポートを介して確立することのできる接続タイプを示します。以下の値が使用可能です。

CWBUN_LDAPCNN_NORMAL

通常接続が使用されます。

CWBUN_LDAPCNN_SSL

SSL 接続が使用されます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_API_PARAMETER

無効な索引。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

使用法

なし

cwbUN_GetLdapPublishParentDn: 目的

公開されたオブジェクトの親識別名を戻します。例えば、公開ユーザーの parentDN が cn=users,o=ace industry,c=us で、John Smith のユーザー情報が公開された場合、公開されたオブジェクトの DN は、cn=john smith,cn=users,ou=ace industry,c=us になります。

構文

```
int cwbUN_GetLdapPublishParentDnW
( cwbUN_ldapPubHandle handle,
  int index,
  LPWSTR parentDn,
  int *length
);

int cwbUN_GetLdapPublishParentDnA
( cwbUN_ldapPubHandle handle,
  int index,
  LPSTR parentDn,
  int *length
);

int cwbUN_GetLdapPublishParentDn8 /* return parentDn in UTF-8 */
( cwbUN_ldapPubHandle handle,
  int index,
  LPSTR parentDn,
  int *length
);
```

パラメーター

cwbUN_ldapPubHandle handle - input

cwbUN_OpenLdapPublishing() への呼び出しにより以前に取得されたハンドル。

int index - input

公開レコードのゼロ・ベースの索引。この値は、cwbUN_GetLdapPublishCount() が戻すカウントより小さくなければなりません。

LPSTR parentDn - output

parentDn の名前を含むバッファーへのポインター。

int * length - input/output

parentDn バッファーの長さへのポインター。バッファーが小さいためにストリング (NULL を含む) を保持できない場合、必要なバッファーのサイズがこのパラメーターに設定されます。

戻りコード

以下のリストは共通の戻り値を示します。

CWB_OK

正常終了。

CWB_INVALID_API_HANDLE

無効なハンドル。

CWB_INVALID_API_PARAMETER

無効な索引。

CWB_INVALID_POINTER

NULL ポインターが指定されました。

CWB_BUFFER_OVERFLOW

サフィックス・バッファーが小さいため、結果全体を保持できません。

使用法

なし

iSeries ナビゲーター API に固有の戻りコード

iSeries ナビゲーター API 戻りコードおよび関連する意味のリスト

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	

- 6014 CWBUN_INVALID_AUTH_TYPE
iSeries の名前タイプが無効です。
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、インストール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
ストリングが長すぎます。
- 6056 CWBUN_NAME_MISSINGENDQUOTE
終了の引用符がありません。
- 6057 CWBUN_NAME_INVALIDQUOTECHAR
文字が引用符付きストリングに対して無効です。
- 6058 CWBUN_NAME_ONLYBLANKS
ブランクのみのストリングが検出されました。
- 6059 CWBUN_NAME_STRINGTOOSHORT
ストリングが短すぎます。
- 6060 CWBUN_NAME_TOOLONGFORIBM
ストリングには問題がありませんが、IBM^(R) コマンドとしては長すぎます。
- 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 がインストールされていないか、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^(TM) 名の検査の API 用です。

- 1001 CWBUN_NULLSTRING
ストリングが空です。
- 1004 CWBUN_INVALIDCHAR
無効な文字です。
- 1005 CWBUN_STRINGTOOLONG
ストリングが長すぎます。

1006	CWBUN_MISSINGENDQUOTE	引用符付きストリングの終了の引用符がありません。
1007	CWBUN_INVALIDQUOTECHAR	文字が引用符付きストリングに対して無効です。
1008	CWBUN_ONLYBLANKS	空白のみのストリングです。
1009	CWBUN_STRINGTOOSHORT	定義されている最小値よりも短いストリングです。
1011	CWBUN_TOOLONGFORIBM	ストリング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。

Visual Basic のリファレンス

以下のトピックを利用して、Visual Basic 用 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。

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

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。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。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。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、`cwbun.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 (`cwbunvbi.dll` および `cwbunvbi.hlp`) を参照してください。

関連概念

『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。

77 ページの『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。

77 ページの『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。

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。

注: 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` ファイルおよび `cwbunvbi.hlp`) を参照してください。

関連概念

75 ページの『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』
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。

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すが、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 (cwbnvbi.dll ファイルおよび cwbnvbi.hlp ファイル) を参照してください。

関連概念

75 ページの『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』
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。

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 (cwbnvbi.dll ファイルおよび cwbnvbi.hlp) を参照してください。

関連概念

75 ページの『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』
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。

Java のリファレンス

以下のトピックを利用して、Java 用 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。

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・クラ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機能は、「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。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、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あります。

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 for Windows 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 にインストール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

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。

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  
"MinimumIMPRelease"="NONE"  
"MinimumRISRelease"="030701"  
"ProductID"="NONE"  
"ServerEntryPoint"="sampext.dll"
```

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

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す。

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 を複写すると、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(R) では必須)

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell Extensions\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、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¥File Systems¥*  
¥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、シェ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" をユーザーの 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「スタート」、「ファイル名を指定して実行」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であることがあります。
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 i5/OS 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 i5/OS 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 i5/OS 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>

1 次レジストリー・キー・フィールド	フィールドの説明
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 i5/OS 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 i5/OS 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 i5/OS 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・プログラムがインストール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であることがあります。</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 のインストール・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、インストール・パスを表す iSeries Access for Windows の置換変数が置かれます。</p>

1 次レジストリー・キー・フィールド	フィールドの説明
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。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。

関連タスク

87 ページの『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。

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、レジストリー内の 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

"MinimumIMPIRelease"="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、このセクションを飛ば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・インターフェースです。

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が区別されます。
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・キーを参照してください。

関連概念

83 ページの『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。

Parent フィールドの値:

親フォルダーの ID コードのリスト

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。

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
MST	マネージメント・セントラル・スケジュール・タスク
MSM	マネージメント・セントラル・モニター
MTA	マネージメント・セントラル・タスク・アクティビティー
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	セキュリティー・フォルダー
UGF	ユーザーおよびグループ・フォルダー

関連概念

『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。

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。 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。
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	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。

関連概念

90 ページの『Parent フィールドの値』
親フォルダーの ID コードのリスト

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識別し、component は提供される機能を説明しています。以下に示す例では、ストリング IBM.MsgQueueSample3 は、IBM をベンダーとして、「MsgQueueSample3」をプラグインが提供する機能の説明として識別します。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.d11"  
"NameID"=dword:00000001  
"DescriptionID"=dword:00000002  
"MinimumIMPIRelease"="NONE"  
"MinimumRISCRelease"="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、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・ファイルでは、この名前は完全修飾パス名であることがあります。

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

プラグインが要求する **IMPI** ハードウェア上で実行される **i5/OS** 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、**vvrrmm** という形式でなければなりません。ここで、**vv** は **i5/OS** 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。

IMPI ハードウェア上で実行される **i5/OS** 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。

MinimumRISCRelease

プラグインが要求する **RISC** ハードウェア上で実行される **i5/OS** 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、**vvrrmm** という形式でなければなりません。ここで、**vv** は **i5/OS** 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。

RISC ハードウェア上で実行される **i5/OS** 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。

ProductID

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・プログラムがインストール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であることがあります。

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文字列です。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文字列を構成する際に、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文字列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 ナビゲーターの階層に追加する個々の新しいフォルダーには、固有の論理タイプがあります。上の例では、文字列 **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
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
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・タスク
MTA	マネージメント・セントラル・タスク・アクティビティー
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	セキュリティー・フォルダー
UGF	ユーザーおよびグループ・フォルダー
WMF	ワーク・マネージメント・フォルダー

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。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。これは、デフォルト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 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。

TaskpadNameID

iSeries ナビゲーター階層に、タスクパッド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、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\shelllex\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\shelllex\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\shelllex\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、レジストリー内の 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。

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 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
```

Support Level

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 (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。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-Output	
		Capabilities-After-Other	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。

関連概念

- 99 ページの『QueryContextMenu フラグについて』
- SHORT DESCRIPTION の追加

35 ページの『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。

QueryContextMenu フラグについて:

SHORT DESCRIPTION の追加

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。

関連概念

97 ページの『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。

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
    End If
End Function
```

```

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 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

プロパティ・ページを Java プラグインのプロパティ・シートに追加する。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、ユーザーのエンティティ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、Properties Manager の例を参照してください。

関連概念

『例: Java Properties Manager』

以下は、Java Properties Manager のコードの外観です。

例: Java Properties Manager:

以下は、Java Properties Manager のコードの外観です。

```
package com.ibm.as400.opnav.Sample;
import com.ibm.as400.opnav.*;

import java.awt.Frame;

import com.ibm.as400.ui.framework.java.*;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class SamplePropertiesManager implements
PropertiesManager
{

// The list of selected objects.
ObjectName[] m_objectNames;

// Save the array of selected object names
```

```

//
public void initialize(ObjectName[] objectNames)
{
    m_objectNames = objectNames;
}

// Return an array of Panel Managers
//
public PanelManager[] getPages()
{
    // Instantiate the data beans
    MyDataBean dataBean = new MyDataBean();
    dataBean.load();
    AnotherDataBean dataBean2 = new AnotherDataBean();
    dataBean2.load();

    DataBean[] dataBeans = { dataBean };
    DataBean[] dataBeans2 = { dataBean2 };

    // Create the panel
    PanelManager pm = null;
    PanelManager pm2 = null; try
    {

        pm = new PanelManager("com.ibm.as400.opnav.Sample.Sample",
            "PAGE1",
            dataBeans);

        pm2 = new PanelManager("com.ibm.as400.opnav.Sample.Sample",
            "PAGE2",
            dataBeans2);

    }

    catch (com.ibm.as400.ui.framework.java.DisplayManagerException
    e)
    {

        Monitor.logError("SamplePropertiesManager: Exception when
        creating pages "+e);

    }

    pm.setTitle("First Java Page");pm2.setTitle("Second Java Page");

    PanelManager[] PMArray = {pm, pm2};

    return PMArray;

}

// Return a list of ActionListener objects to be notified when
commit is processed

public ActionListener[] getCommitListeners()
{

    ActionListener[] a1 = new ActionListener[1];
    a1[0] = new ActionListener()
    {

        public void actionPerformed(ActionEvent evt)
        {

```

```

Monitor.logError("SamplePropertiesManager: Processing Commit
Listener");
}
};
return al;
}
// Return a list of ActionListener objects to be notified when
cancel is selected
public ActionListener[] getCancelListeners()
{
ActionListener[] al = new ActionListener[1];
al[0] = new ActionListener()
{
public void actionPerformed(ActionEvent evt)
{
Monitor.logError("SamplePropertiesManager: Processing Cancel
Listener");
}
};
return al;
}
}
}

```

関連概念

101 ページの『Java におけるプロパティー・シートの処理』

プロパティー・ページを Java プラグインのプロパティー・シートに追加する。これにより、オブジェクト名の作成、プロパティーの表示、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。

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ー・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 cwvUN_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って取得された 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
```

コードに関するライセンス情報および特記事項

IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作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。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

IBM、そのプログラム開発者、または供給者は、いかなる場合においてもその予見の有無を問わず、以下に対する責任を負いません。

1. データの喪失、または損傷。
2. 直接損害、特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。

付録. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとしします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

- | 本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム
- | 契約の契約条項、IBM プログラムのご使用条件、IBM 機械コードのご使用条件、またはそれと同等の条項
- | に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。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。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

この「iSeries ナビゲーター プラグインの開発」資料には、プログラムを作成するユーザーが IBM i5/OS のサービスを使用するためのプログラミング・インターフェースが記述されています。

商標

以下は、IBM Corporation の商標です。

i5/OS
IBM
iSeries
NetServer
OS/400

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

他の会社名、製品名およびサービス名などはそれぞれ各社の商標または登録商標です。

資料に関するご使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。



Printed in Japan