

CICS® Transaction Gateway



COM オートメーション・プログラミング

バージョン 4.0

CICS® Transaction Gateway



COM オートメーション・プログラミング

バージョン 4.0

ご注意

本書の情報およびそれによってサポートされる製品を使用する前に、113ページの『特記事項』に記載する一般情報をお読みください。

本書は、CICS Transaction Gateway、プログラム番号 5724-A75 に適用されます。また、新版で特に明示されない限り、これ以降のすべてのバージョン、リリース、および修正レベルにも適用されます。

本書は、SC88-7780 の改訂版です。ページの左側にある縦線は、この版で新規の内容であることを示しています。

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)

原典： SC34-5946-00
CICS® Transaction Gateway
COM Automation Programming
Version 4.0

発行： 日本アイ・ピー・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 2001.6

この文書では、平成明朝体™W3、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™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 1996,1999. All rights reserved.

Translation: © Copyright IBM Japan 2001

目次

本書について	vii
本書の対象読者	vii
本書で使用される規則と用語	vii
前提条件と関連情報	viii
変更の要約	ix

第1部 クライアント・クラス - 指針 1

第1章 CICS オブジェクト指向プログラミングの紹介	3
CICS クライアントでのオブジェクト指向サポート	3
プログラミング言語サポート	4

第2章 作業環境の確立	5
必要なソフトウェアおよびオペレーティング・システム	5
COM ライブラリー	5
サーバー	5
登録	6
COM ライブラリーを使用可能にする	7
COM ライブラリー: オブジェクトおよびアパートメント	7

第3章 COM クラスの使用	9
オブジェクト作成とインターフェース	9
タイプ・ライブラリーと Visual Basic	11
Intellisense	11
クラスの要約	11
例外処理	12
移行に関して報告されている問題点	14
Visual Basic を使用した CICS への ECI リンク呼び出し	14
VBScript を使用した CICS への ECI リンク呼び出し	16
ECI 呼び出し同期タイプ	16
CICS サーバー情報および接続状況	17
作業単位内での ECI リンク呼び出し	18
VB で COMAREAS を処理する	19
ECI ユーザー ID およびパスワード管理	20

Visual Basic を使用した CICS 3270 アプリケーションへの接続	21
CICS 3270 セッションの実行	22
EPI 呼び出し同期タイプ	23
CICS サーバー情報	24
BMS マップ・データの EPI COM クラスでの使用	25
VBScript を使用した CICS 3270 アプリケーションへの接続	27
自動トランザクション始動 (ATI) のサポート	27
セキュリティ管理	28

第2部 COM クラス - 参照 31

第4章 Buffer COM クラス	35
インターフェースの選択	35
オブジェクト作成	35
メソッド	36
AppendString	36
Data	36
ExtractString	36
InsertString	37
Length	37
Overlay	37
SetData	37
SetLength	38
SetString	38
String	38

第5章 Connect COM クラス	39
インターフェースの選択	39
オブジェクト作成	39
メソッド	40
AlterSecurity	40
Cancel	40
Changed	40
ChangePassword	41
Details	41
Link	42
MakeSecurityDefault	43
Password	43

ServerName	43	BackgroundColor	58
ServerStatus	43	BaseAttribute	58
ServerStatusText	44	Column	58
Status	44	DataTag	58
TranDetails	44	ForegroundColor	59
UnpaddedPassword	45	Highlight	59
UnpaddedServerName	45	InputProt	59
UnpaddedUserId	45	InputType	60
UserId	45	Intensity	60
VerifyPassword	45	Length	60
第6章 ECI COM クラス	47	Position	61
インターフェースの選択	47	ResetDataTag	61
オブジェクト作成	47	Row	61
メソッド	47	SetBaseAttribute	61
ErrorFormat	47	SetExtAttribute	61
ErrorOffset	48	SetText	62
ErrorWindow	48	Text	62
ExCode	48	TextLength	62
ExCodeText	49	Transparency	62
ServerCount	49	第9章 Flow COM クラス	63
ServerDesc	49	インターフェースの選択	63
ServerName	50	オブジェクト作成	63
SetErrorFormat	50	メソッド	63
第7章 EPI COM クラス	51	AbendCode	63
インターフェースの選択	51	CallType	64
オブジェクト作成	51	CallTypeText	64
メソッド	51	Diagnose	64
Diagnose	51	Flowid	64
ErrorFormat	52	ForceReset	64
ErrorOffset	52	Poll	64
ErrorWindow	52	SetSyncType	65
ExCode	52	SetTimeout	65
ExCodeText	53	SyncType	66
ServerCount	53	Timeout	66
ServerDesc	54	Wait	66
ServerName	54	第10章 Map COM クラス	67
SetErrorFormat	54	インターフェースの選択	67
State	54	オブジェクト作成	67
Terminate	55	メソッド	67
第8章 Field COM クラス	57	ExCode	67
インターフェースの選択	57	FieldByName	68
メソッド	57	Validate	68
AppendText	57	第11章 Screen COM クラス	71

インターフェースの選択	71	AlterSecurity	83
メソッド	71	CCSID	84
CursorCol	71	ChangePassword	84
CursorRow	71	Connect	84
Depth	72	Devtype	85
FieldByIndex	72	Diagnose	85
FieldByPosition	72	Disconnect	85
FieldCount	72	DisconnectWithPurge	85
MapName	72	DiscReason	86
MapSetName	73	ExCode	86
SetAID	73	ExCodeText	86
SetCursor	73	Install	87
Width	74	MakeSecurityDefault	87
第12章 SecAttr COM クラス	75	NetName	87
インターフェースの選択	75	Password	87
パブリック・メソッド	75	Poll	87
ExpiryTime	75	PollForReply	88
InvalidCount	75	QueryATI	88
LastAccessTime	76	ReadTimeout	89
LastVerifiedTime	76	ReceiveATI	89
第13章 SecTime COM クラス	77	Screen	89
インターフェースの選択	77	Send	89
パブリック・メソッド	77	ServerName	90
Day	77	SetATI	90
GetDate	77	SetTermDefns	90
Hours	77	SignonCapability	91
Hundredths	78	Start	92
Minutes	78	State	92
Month	78	TermId	92
Seconds	78	TransId	93
Year	78	Userid	93
第14章 Session COM クラス	79	VerifyPassword	93
インターフェースの選択	79	第16章 UOW COM クラス	95
オブジェクト作成	79	インターフェースの選択	95
メソッド	79	オブジェクト作成	95
Diagnose	79	メソッド	95
SetSyncType	79	BackOut	95
State	80	Commit	96
TransId	81	ForceReset	96
第15章 Terminal COM クラス	83	UowId	96
インターフェースの選択	83	第3部 付録	97
オブジェクト作成	83	付録A. COM グローバル定数	99
メソッド	83		

付録B. CICS クライアント ECI 定数 . . .	101
同期タイプ	101
フロー状況タイプ	101
接続状況コード	101

付録C. CICS クライアント EPI に特有の定 数	103
同期タイプ	103
CclEPI 状態	103
CclSession 状態	103
CclTerminal 状態	104
CclTerminal ATI 状態	104
CclTerminal EndTermReason	104
CclTerminal サインオン・タイプ	105
CclScreen AID キー・コード	105
CclField 保護状態属性	106
CclField 数値属性	106
CclField 輝度属性	106
CclField 変更属性	107
CclField 強調表示属性	107
CclField 透過性属性	107

CclField カラー属性	108
--------------------------	-----

付録D. エラー・コード参照	109
--------------------------	-----

特記事項	113
商標	115

参考文献	117
CICS Transaction Gateway および CICS ユニ バーサル・クライアントのライブラリー . . .	117
CICS Transaction Gateway 資料	117
CICS ユニバーサル・クライアント資料	118
CICS ファミリー資料	118
資料のファイル名	118
サンプル構成資料	119
その他の資料	119
オンライン資料の表示	120

用語集	121
---------------	-----

索引	123
--------------	-----

本書について

本書は CICS 外部呼び出しインターフェース (ECI) および CICS 外部表示インターフェース (EPI) のためのオブジェクト指向プログラミングについて説明します。本書では、システムが提供しているクラスとメソッドを使用して、例を使って、プログラム作成の指針を説明します。

本書の対象読者

本書は、IBM CICS ユニバーサル・クライアント バージョン 3.1 で提供されたオブジェクト指向クラスを使用して、オブジェクト指向 CICS クライアント・プログラムを開発する方法を会得したい CICS アプリケーション・プログラマーを対象としています。

クライアントは、外部呼び出しインターフェース (ECI)、外部表示インターフェース (EPI) および 外部セキュリティー・インターフェース (ESI) によって CICS サービスを使用することができます。CICS クライアントの COM クラスを使用すると、プログラマーはオブジェクト指向の手法で ECI および EPI インターフェースをアクセスすることができます。

このバージョンでは、Microsoft® Visual Basic、Visual Basic for Applications、および VBScript をサポートしています。

本書の コンポーネント・オブジェクト・モデル (COM) クラスの詳細は、Visual Basic および VBScript のポイントから表示できます。これらのクラスは COM オートメーションをサポートする、他の開発環境および言語から作業することがありますが、IBM ではこれらをテストしていません。

本書の直前の旧版、CICS ファミリー: CICS クライアント用 BASIC オブジェクト指向プログラミング、SC88-7780-00 は、CICS クライアントの旧バージョンにご使用ください。

本書で使用される規則と用語

本書では、用語 *CICS* ユニバーサル・クライアント は、CICS Transaction Gateway のクライアント・コンポーネントを示しています。

- CICS® for MVS/ESA™
- CICS® Transaction Server (OS/390® 版)

- CICS® Transaction Server (VSE/ESA™ 版)
- CICS/VSE®

CICS ユニバーサル・クライアントは、Windows NT® および Windows 2000 上で実行します。本書で Windows® は、特定の Windows のバージョンが指定されていない限り、Windows NT および Windows 2000 の両方を指しています。

本書で OS/390 は、OS/390 オペレーティング・システムおよび z/OS オペレーティング・システムの両方を指しています。

前提条件と関連情報

本書は、読者がオブジェクト指向の概念と Visual Basic 言語に精通し、かつ CICS が提供する既存のサービスをひとつと理解していることを前提としています。

CICS サービスを使用する際の詳しい情報は、「CICS® ファミリー: クライアント / サーバー プログラミング」を参照してください。

変更の要約

以下は、CICS ユニバーサル・クライアント バージョン 4.0 で行われた機能変更です。

- TCP62 のサポート。
- Windows 2000 のサポート。
- Windows 2000 Terminal Service のサポート。これによって、単一の Windows 2000 サーバー上で複数のデスクトップ・セッションが実行可能です。
- 以下のサポートはなくなりました。
 - Cobol
 - PL1
 - REXX
 - CICSTELD
 - OS/2[®]、Windows 95、Windows 98
 - 16 ビット・アプリケーション
 - NetBIOS および DCE プロトコル
 - IBM VisualAge[®] C++ for Windows[®]

本書は、SC88-7780 の改訂版です。ページの左側にある縦線は、この版で新規の内容であることを示しています。

第1部 クライアント・クラス ー 指針

第1章 CICS オブジェクト指向プログラミングの紹介	3
CICS クライアントでのオブジェクト指向サポート	3
プログラミング言語サポート	4

第2章 作業環境の確立	5
必要なソフトウェアおよびオペレーティング・システム	5
COM ライブラリー	5
サーバー	5
登録	6
処理中サーバーの登録	6
処理外サーバーの登録	6
COM ライブラリーを使用可能にする	7
COM ライブラリー: オブジェクトおよびアパートメント	7

第3章 COM クラスの使用	9
オブジェクト作成とインターフェース	9
タイプ・ライブラリーと Visual Basic	
Intellisense	11
クラスの要約	11
例外処理	12
移行に関して報告されている問題点	14
Visual Basic を使用した CICS への ECI リンク呼び出し	14
VBScript を使用した CICS への ECI リンク呼び出し	16
ECI 呼び出し同期タイプ	16
CICS サーバー情報および接続状況	17
作業単位内での ECI リンク呼び出し	18
VB で COMAREAS を処理する	19
ECI ユーザー ID およびパスワード管理	20
Visual Basic を使用した CICS 3270 アプリケーションへの接続	21
CICS 3270 セッションの実行	22
EPI 呼び出し同期タイプ	23
CICS サーバー情報	24
BMS マップ・データの EPI COM クラスでの使用	25

VBScript を使用した CICS 3270 アプリケーションへの接続	27
自動トランザクション始動 (ATI) のサポート	27
セキュリティ管理	28

第1章 CICS オブジェクト指向プログラミングの紹介

CICS[®] ファミリーは、IBM[®] が提供する主なハードウェア・プラットフォーム、および UNIX[®] などの非 IBM の主要なプラットフォームにわたる、強力なトランザクション処理機能を備えています。また、クライアント / サーバーをサポートする幅広い機能を提供して、エンド・ユーザーに情報を提供する最新のグラフィカル・インターフェースの使用を可能にします。CICS ファミリーは、オブジェクト指向プログラミング用に出現するテクノロジーをサポートするようになっており、CICS ユーザーには、CICS 技術、データ、ならびにアプリケーションへの投資を活用しながら、オブジェクト・テクノロジーの利点の多くを利用する方が提供されています。

オブジェクト指向 プログラミングでは、現実性の高いモデルを、より柔軟性に富んだプログラミング言語で作成することができます。オブジェクトの新しい型もしくはクラスを定義できると同時に、これらのオブジェクトをさまざまな構造を採り入れて表すことができます。

オブジェクト指向 プログラミングを用いると、特定の型のオブジェクトに関連する動作を定義するメソッド (メンバー関数) を作成し、これにより、基本的なデータに関連するセマンティックを多く取り込むことで、データにより多くのセマンティックを関連づけることもできます。

ソフトウェアの複雑な部分を、より単純な外部シェルに包みこむか、カプセル化することで、コード再使用の手掛かりが得られます。このようにして定義されたオブジェクトは、広範囲のさまざまなアプリケーションが使用することができます。オブジェクトを適切に定義し個別に用意することが、その後のアプリケーションのより迅速かつ経済的な作成を可能にする、再使用可能なパーツのライブラリーの基礎となります。既存のパーツは、すでにテスト済みで、他のアプリケーションで使用されているため、その再使用によってより高いレベルのソフトウェア品質が得られることとなります。

CICS クライアントでのオブジェクト指向サポート

CICS クライアントで提供される基本的な通信メカニズムは、外部呼び出しインターフェース (ECI) です。ECI の完全機能をオブジェクト指向の方法でモデル化する、ECI クラス・ライブラリーを用意することで、拡張サポート構築の基礎が与えられます。

CICS オブジェクト指向プログラミングの紹介

クラスが提供されることで、サーバーへのリンク、サーバー上の CICS プログラムの呼び出し、状況の検出、そして作業単位 (UOW) の活用を行う機能が得られます。

CICS クライアントのアプリケーション・プログラマーが使用できるもう一つのインターフェースは、外部表示インターフェース (EPI) です。CICS アプリケーションにもとづく 3270 端末との通信は、端末、画面、フィールドおよび BMS マップをカプセル化するクラスによって提供されます。EPI クラスを用いると、3270 データ・ストリームを直接処理する必要がなくなり、出力画面の内容の検査、更新がさらに容易になります。

プログラミング言語サポート

オブジェクト指向ライブラリーは、C++ および COM プログラマー用の CICS クライアント バージョン 3.1 と一緒に提供されます。本書は、COM プログラミングに関するトピックを記載しています。C++ を使用してプログラムを作成したい場合は、サンプルやクラスの説明などのユーザーの手引きとなる情報が記載された「*CICS Transaction Gateway: C++ プログラミング*」を参照してください。

第2章 作業環境の確立

Windows NT 環境では、CICS クライアント用のコンポーネント・オブジェクト・モデル (COM) オブジェクト指向 (OO) サポートが提供されます。これには、COM ランタイム、タイプ・ライブラリー、BMS マップ・ユーティリティーおよびサンプル・コードが含まれています。

必要なソフトウェアおよびオペレーティング・システム

- Windows NT ワークステーション バージョン 4.00 SP3 以降
- Microsoft Visual Basic 5.0 以降
- Microsoft VBScript バージョン 5.0 以降

「CICS クライアント 管理の手引き (SC88-7631)」で、CICS クライアントがサポートする CICS サーバー・プラットフォームに関する詳細を参照してください。

COM ライブラリー

COM ライブラリーはオートメーション・コンパチブルです。オートメーション・コンパチブルとは、COM ライブラリー Visual Basic、VBScript および Delphi などのプログラム言語から簡単に使用できるということです。しかし、これらのライブラリーは Visual Basic および VBScript でのみテストされ、その他の環境ではフル・サポートは提供されません。

サーバー

ライブラリーは処理中サーバー (cclicci.dll および ccliapi.dll) および処理外またはローカル・サーバー (ccloeci.exe および ccloapi.exe) として提供されます。ローカル・サーバーは提供されていますが、それらは DCOM 環境ではサポートされておらず、テストもされていません。

処理中サーバーは処理外サーバーよりパフォーマンスが良いので、処理中サーバーを使用することをお勧めします。

登録

COM ライブラリーはインストール時に登録しなければなりません。登録はライブラリーのそれぞれの、COM クラス、関連する ProgID (後で参照) およびタイプ・ライブラリーを登録します。

タイプ・ライブラリーをそれぞれの Visual Basic Project に登録する場合に、Visual Basic はタイプ・ライブラリーのみを使用します。これらのタイプ・ライブラリーの機能とパフォーマンスを完全に使用するためには、このような使い方をお勧めします。プロジェクト使用可能化についての詳細は 7ページの『COM ライブラリーを使用可能にする』を参照してください。

VBScript はタイプ・ライブラリーを使用しません。

COM ライブラリーのすべては自動登録および登録抹消をサポートし、処理中または処理外ライブラリーを登録できます。

処理中ライブラリーと処理外ライブラリーの両方が登録されると、デフォルトでは Visual Basic と VBScript が処理中バージョンを使用します。

処理中サーバーの登録

処理中サーバーを登録するには、Microsoft 提供のプログラム REGSVR32 を使用してください。

たとえば、ECI COM ライブラリーを登録するには、次のコマンドを発行します。

```
REGSVR32 CCLIECI.DLL
```

登録抹消するには次のコマンドを発行します。

```
REGSVR32 /U CCLIECI.DLL
```

処理外サーバーの登録

処理外サーバーを登録するには、実行可能プログラムを実行し、登録が完了したことを示すウィンドウが表示されたら、それをクローズしてください。たとえば、EPI ライブラリーを登録するにはプログラム CCLOEPI を実行してください。

処理外サーバーを登録抹消するには、/unregister パラメーターを使用して実行可能プログラムを実行します。たとえば、EPI ライブラリーを登録抹消するにはプログラム CCLOEPI /unregister を実行してください。

COM ライブラリーを使用可能にする

タイプ・ライブラリーを使用するために Visual Basic をセットアップするには、Visual Basic Project/References... ライブラリーに進み、ユーザー・アプリケーションの必要に応じて以下のいずれかを選択します。

IBM CICS Client EPI

または

IBM CICS Client ECI

OK を押してください。

タイプ・ライブラリーがリストされていない場合は、COM ライブラリーは登録されていないと考えられます。COM ライブラリーの登録についての情報は直前のセクションを参照してください。

COM ライブラリー: オブジェクトおよびアパートメント

CICS クライアント COM ライブラリーの設計では、COM オブジェクトを他の COM オブジェクトへ渡す必要があります。この機能を実現するには関連する COM オブジェクトを、同一アパートメント内に作成する必要があります。たとえば、ECI で、Connect COM オブジェクト上でリンク・メソッド呼び出しを作成するには、Flow、Buffer、および UOW オブジェクトが渡される必要があります。正しく機能させるには、これらはすべて同一のアパートメントに作成されている必要があります。

また、EPI では、端末セッションおよび Map COM オブジェクトが、必ず同一のアパートメントで生成されていることが重要です。端末は、Screen オブジェクトを作成する必要があります。このオブジェクトを、その端末と同じアパートメントに作成します。次にこの Screen オブジェクトは、フィールド・オブジェクトを作成する責任があり、この場合も、その Screen オブジェクトと同じアパートメントに作成します。out-of-process サーバーは、すべてのオブジェクトが同一アパートメントに作成されることを保証します。ただし、COM オブジェクトが in-process サーバーを使用する場合にそれらのオブジェクトが作成されるアパートメントについては、プログラマーが制御します。

Visual Basic ではほとんどの場合、ユーザーは単一スレッド・アプリケーションを作成するため、アパートメントについて考慮する必要はありません。

第3章 COM クラスの使用

CICS クライアントは、Microsoft Windows NT 上の CICS ECI および EPI 用の COM クラスを提供します。COM オートメーションをサポートするすべてのアプリケーション開発環境から、インターフェースを使用することができます。しかし、本書では Visual Basic および VBScript についてのみを説明します。これらの環境のみがサポートされます。

これらの COM クラスは、Microsoft Visual Basic、Visual Basic for Applications (Microsoft Excel バージョン 5.0 などのアプリケーションに組み込まれています)、および VBScript からアクセスできます。

COM クラスを使用するプログラムの例については、個別のサンプル文書を参照してください。

オブジェクト作成とインターフェース

COM オブジェクトと対話するためには、インターフェースを使用する必要があります。ECI および EPI COM ライブラリーは、COM クラスごとに 2 つのインターフェースを提供します。

最初のインターフェースは IDispatch と呼ばれ、以前の Visual Basic アプリケーションと VBScript をサポートするために提供されています。2 番目のインターフェースは、Custom インターフェースで、これも Visual Basic で使用するために提供されています。このインターフェースは IDispatch インターフェースよりも処理速度が早く、Visual Basic と共に使用することをお勧めします。それぞれの COM クラスが 1 つの IDispatch インターフェースと 1 つの Custom インターフェースを提供します。

Visual Basic は COM オブジェクトを作成し、そのオブジェクトと対話するインターフェースを選択するための複数の方法を提供します。オブジェクトを作成するために、CreateObject ファンクションと新規ファンクションがあります。Visual Basic でオブジェクトを作成するために、新規のファンクションを使用することをお勧めします。

VBScript はより単純です。これは、オブジェクトを作成するただ 1 つの方法、すなわち CreateObject ファンクションを提供するもので、この場合は IDispatch インターフェースを使用しなければなりません。

COM クラス

次に COM オブジェクトを作成する幾つかの例を示します。

```
set eci = CreateObject("Ccl.ECI")
set eci = new Cc1oECI
set connection = CreateObject("Ccl.Connect")
set connection = New Cc1oConn
```

オブジェクト・クラスを要求できる 2 つの方法があることに注意してください。CreateObject を使用する場合は、Programmatic ID または省略して ProgID と呼ばれるストリングを指定します。新規のファンクションを使用する場合は、タイプ・ライブラリーに登録されているクラス名を指定します。

Visual Basic を使用する場合は、どのインターフェースを使用したいかを選択することができます。オブジェクトとしてユーザーの変数を DIM 宣言する場合は、IDispatch インターフェースを選択します。クラス名としてユーザーの変数を DIM 宣言する場合は、custom インターフェースを選択します。Visual Basic で terminal オブジェクトを作成するには、以下のコードを使用します。

```
Dim Terminal as Cc1oTerminal
Set Terminal = New Cc1oTerminal
```

図1. Visual Basic で terminal オブジェクトを作成する

または、必要であれば、上記を単一ステートメントに結合することもできます。

```
Dim Terminal as New Cc1oTerminal
```

VBScript を使用する場合は、VBScript がユーザーのために IDispatch インターフェースを自動的に選択します。たとえば、VBScript に terminal オブジェクトを作成するには、以下のコードを使用します。

```
Dim Terminal
Set Terminal = CreateObject("Ccl.Terminal")
```

図2. VBScript で terminal オブジェクトを作成する

オブジェクトを作成するには、下記をお勧めします。

- 1 つのインターフェース・タイプを選択する。
- ユーザーのプログラムには複数のオブジェクト・インターフェース・タイプを混在させない。タイプが混在するような環境はサポートされません。
- パフォーマンスが改善されるので、custom インターフェースを選択する。

どのインターフェースが選択された場合でも、オブジェクトがどの方法により作成された場合でも、ユーザーのプログラム内ではそのオブジェクトは全く同様に使用されます。

タイプ・ライブラリーと Visual Basic Intellisense

タイプ・ライブラリーは COM ライブラリーに多くの役立つ機能を追加します。その 1 つが Visual Basic Intellisense です。タイプ・ライブラリーは Visual Basic に情報を提供しこれにより、Visual Basic はコードを完了させることができます。また、メソッドの形式についてプロンプトが出され、また該当する場合はメソッド・パラメーターと関連する定数またはテストできる戻り値の形式についてプロンプトが出されます。たとえば、10ページの図1 の表示にしたがって Visual Basic 用の terminal オブジェクトを作成する場合に、terminal オブジェクトでメソッドを選択したい場合、' ' キーを押すと、選択可能なメソッドのリストが表示されます。メソッドを選択し、スペースを押すかブラケットをオープンすると必須パラメーターが表示されます。ECI および EPI クラスについての参照情報を、Visual Basic Object Browser を使用してタイプ・ライブラリーでブラウズすることができます。ECI クラスの CcIECILib または、EPI クラスの CcIEPILib 参照情報を選択してください。タイプ・ライブラリーは処理中ライブラリー・ファイル cclieci.dll および ccliepi.dll 内に組み込まれます。

クラスの要約

CICS COM サーバーの提供する COM クラスを、次の表にリストしていません。

表1. ECI COM クラス

COM クラス	説明
バッファー	CICS サーバーとのデータの受け渡しに使用するバッファー。
Connection	CICS サーバーへの接続を制御する。
ECI	クライアントで構成される CICS サーバーのリストへのアクセスを提供する。
フロー	CICS サーバー・プログラムとの単一の会話を制御する。
SecAttr	セキュリティー属性 (パスワード) について情報を提供する。
SecTime	日付と時間情報を提供する。
UOW	CICS サーバーへの呼び出しのリカバリー可能セットを調整する。

COM クラス

表2. EPI COM クラス

COM クラス	説明
EPI	CICS EPI の初期化および終了を行い、クライアントで構成される CICS サーバーのリストへのアクセスを提供する。
フィールド	画面上の 1 つの 3270 フィールドへのアクセスを提供する。
マップ	CICS サーバーの BMS マップによって定義されている 3270 フィールドへのアクセスを提供する。
画面	3270 端末画面へのアクセスを提供する。
セッション	3270 端末の CICS サーバーとの会話の順序列を制御する。
端末	3270 端末接続を制御する。

Visual Basic から ECI および EPI COM クラスを使用する方法について説明するサンプルが、CICS クライアントのディレクトリ `xxxxxxx\SAMPLES\VB` に提供されています。本書の以降のセクションでは、このサンプルからの引用を示しています。

例外処理

ECI および EPI クラスを使用する場合、メソッドを呼び出すときの問題をチェックする 2 つの方法が表示されます。

1 つの方法は `ErrorWindow` メソッドを使用し、それを偽に設定し、それから呼び出しの後でリターン・コードを調べるために `ExCode` および `ExCodeText` メソッドを調べるやり方です。これはメソッド呼出時の問題をチェックする際のお勧めする方法ではなく、旧アプリケーションとの後向きの互換性をサポートする目的でのみ存在しています。

お勧めする方法は、Visual Basic および VBScript が提供する `Err` オブジェクトを使用することです。`Err` オブジェクトには、エラー情報が含まれています。Visual Basic は、`On Error Goto` および `On Error Resume` フィーチャーをサポートし、エラーの発生を検出します。VBScript は `On Error Resume Next` 機能のみをサポートします。Visual Basic または VBScript で `On Error Resume Next` を使用する場合は、エラーを戻す可能性があるすべての COM オブジェクトの前にこの行を必ず入力しておく必要があります。ユーザーが行わない限り Visual Basic/VBScript では `Err` 変数をリセットしません。

選択したインターフェースのタイプ (ユーザーがオブジェクトまたはクラス名として `DIM` 宣言した変数) は、`Err.number` 特性に含まれる値に影響します。`Err.Number` のすべての値を処理する汎用ルーチンを作成し、使用可能なドキュ

メント化された ExCode エラー・コードに変換することが可能です。以下の例のコードは、これをアーカイブする方法を示しています。

この技法を用いて十分な効果を上げるには、Err オブジェクトに情報をすべて入手していることを確認してください。ECI または EPI オブジェクトを作成した後で次の呼び出しを発行してください。

```
ECI.SetErrorFormat 1
```

また、EPI の場合は次の呼び出しを発行してください。

```
EPI.SetErrorFormat 1
```

図3 では Visual Basic でエラーをハンドルする方法を示します。

```
Private Sub Command1_Click()
'
' The following code assumes you have created the
' required objects first, ECI, Connect, Flow, UOW,
' Buffer
'
On Error GoTo ErrorHandler
conn.Link flow, "EC01", buf, uow
Exit Sub
ErrorHandler:
'
' Ok, the Connect call failed
' Parse the Error Number, this will work regardless of
' how the ECI objects were Dimmed
'
Dim RealError As Cc1ECIExceptionCodes
RealError = (Err.Number And 65535) - eci.ErrorOffset

If RealError = cclTransaction Then
'
' Transaction abend, so query the Abend code
'
    AbendCode = flow.AbandCode
    If AbendCode = "AEY7" Then
        MsgBox "Invalid Userid/Password to execute CICS Program", , "CICS ECI Error"
    Else
        MsgBox "Unable to execute EC01, transaction abend:" + AbendCode, , "CICS ECI Error"
    End If
Else
    MsgBox Err.Description, , "CICS ECI Error"
End If
End Sub
```

図3. Visual Basic 例外処理サンプル

COM クラス

図4 では VBScript のエラー処理コードを示します。

```
On Error Resume Next
con.Link flow, "EC01", buf, uow
if Err.Number <> 0 then
'
' Ok, the Connect call failed
' Parse the Error Number, this will work regardless of
' how the ECI objects were Dimmed
'
RealError = Err.Number And 65535 - eci.ErrorOffset
'
' 13 = CclTransaction, a transaction abend.
'
If RealError = 13 Then
'
' Transaction abend, so query the Abend code
'
AbendCode = flow.AbandCode
If AbendCode = "AEY7" Then
Wscript.Echo "Invalid Userid/Password to execute CICS Program"
Else
Wscript.Echo "Unable to execute EC01, transaction abend:", AbendCode
End If
Else
Wscript.Echo Err.Description
End If
End If
```

図4. VBScript 例外処理サンプル

移行に関して報告されている問題点

例外処理の項で説明したように、Err.Number の値はユーザーがどのインターフェースとエラー・フォーマットを選択したかによって変わります。旧アプリケーションで、クラス名として変数を定義するために DIM 宣言を使用し、Err.Number で 8600 の値を検査した場合は (旧リリースでは他の値は利用不能)、この検査は失敗します。

Visual Basic を使用した CICS への ECI リンク呼び出し

最初に、使用する ECI インターフェースのオブジェクト変数を宣言します。通常は、Visual Basic プログラムの General Declarations セクションで以下のように行います。

```
Dim ECI As Cc10ECI
Dim Connect As Cc10Conn
Dim Flow As Cc10Flow
Dim Buffer As Cc10Buf
Dim UOW As Cc10UOW
```

必要な ECI オブジェクトが、Visual Basic **New** 関数によりインスタンス化されます。これは、**Form_Load** サブルーチンによって、または後の工程でなんらかのユーザーのアクションに応じて行われます。最初に Cc10ECI オブジェクトを作成する必要がある点に注意してください。

```
Sub ECILink_Click()
    Set ECI = New Cc10ECI
    Set Connect = New Cc10Conn
    Set Flow = New Cc10Flow
    Set Buffer = New Cc10Buf
```

使用する CICS サーバーについての詳細、すなわち (クライアント初期設定ファイルで構成された) サーバー名、ユーザー ID、およびパスワードは、Connect オブジェクトの **Details** メソッドを通じて提供されます。Buffer オブジェクトは、CICS に送られる任意のデータによって初期化されます。

```
Connect.Details "CICSNAME", "sysad", "sysad"
Buffer.SetString "Hello"
```

CICS へ呼び出しを行うことができるようになりました。**Link** メソッドは、パラメーターとして、Flow オブジェクト、呼び出される CICS サーバー・プログラムの名前、Buffer オブジェクト、および UOW オブジェクトを取ります。この例では、NULL 変数が UOW パラメーターに提供されており、したがってこの呼び出しはリカバリー可能な作業単位の一部にはなりません。CICS から戻される Buffer の内容は、以下の Visual Basic テキスト・ボックス「Text1」に出力されます。

```
Connect.Link Flow, "ECIWT0", Buffer, UOW
Text1.Text = Buffer.String
```

最後に、CICS COM オブジェクトが削除されます。

```
Set Connect = Nothing
Set Flow = Nothing
Set Buffer = Nothing
End Sub
```

この例では、単純なテキスト・ストリングを送受信しています。実際には、Buffer オブジェクトには多くの複雑なデータ (たとえば C データ構造) が含ま

COM CICS ECI へのガイド

れています。バイナリー・データでは、**Buffer.SetData** および **Buffer.Data** メソッドが提供され、これによってその内容は、Byte 配列としてアクセスできます。

一般的なクライアント・アプリケーションは、1 つまたは複数の

Connect.Link 呼び出しを通じて CICS にアクセスすることができ、エンド・ユーザーの Basic プログラムで使用する 'ビジネス・オブジェクト' を構成することができます。その 1 つのアプローチとしては、'ビジネス・オブジェクト' を CclIOBuffer オブジェクトの内容を処理する論理を含んだ、独立した COM オートメーション・サーバーとして実施することが挙げられます。

VBScript を使用した CICS への ECI リンク呼び出し

前述のセクションの visual basic と似ていますが、オブジェクトの作成方法は異なります。

VBScript で変数をすべて DIM 宣言する必要はありませんが、DIM 宣言することは良いプログラミングの練習になります。

```
Dim ECI, Connect, Flow, Buffer, UOW
```

オブジェクトを作成するために、以下のコードを使用してください。

```
Set ECI = CreateObject("Ccl.ECI")  
Set Connect = CreateObject("Ccl.Connect")  
Set Flow = CreateObject("Ccl.Flow")  
Set Buffer = CreateObject("Ccl.Buffer")  
Set UOW = Nothing
```

UOW を使用する予定ではない場合は、それを VBScript で明示的に 'Nothing' に設定します。

ECI 呼び出し同期タイプ

CICS クライアント ECI COM クラスは、同期 (『ブロッキング』) および遅延同期 (『ポーリング』) プロトコルをサポートします。これらのクラスは、C++ クラスで使用可能な非同期呼び出しをサポートしません。

前出の例では、デフォルトの同期タイプである cclSync を指定して Flow オブジェクトが使用されています。この Flow オブジェクトが、**Connect.Link** の最初のパラメーターとして使用されると、同期リンク呼び出しが CICS に対して行われます。Visual Basic プログラムは、CICS からの応答を受け取るまでの間、ブロックされました。CICS からの応答を戻されたリンク呼び出しは、Buffer オブジェクト内で即時に使用可能になります。

遅延同期呼び出しを行う場合、Flow オブジェクトで **SetSyncType** メソッドを使用して、Flow を `ccIDSync` に設定します。この Flow オブジェクトが **Connect.Link** 呼び出しで使用されると、ECI 呼び出しが CICS に行われますが、制御は即時に Visual Basic プログラムに戻され、CICS からの応答を Flow オブジェクトで **Poll** メソッドを使用して検索することが必要になります。

```
Sub ECIDsync_Click()
    Set Connect = New Cc10Conn
    Set Flow = New Cc10Flow
    Set Buffer = New Cc10Buf
    Connect.Details "CICSNAME", "sysad", "sysad"
    Flow.SetSyncType ccIDSync
    Buffer.SetString "Hello"
    Connect.Link Flow, "ECIWT0", Buffer, UOW
End Sub
```

CICS への呼び出しが現在実行中です。後の工程で (ユーザーのアクションに応じて、あるいは Visual Basic プログラムがその他のタスクを完了するとき)、**Poll** メソッドを Flow オブジェクトで使用して CICS からの応答が収集されます。CICS からの応答データを期待している場合には、**Poll** メソッドはパラメーターとして Buffer オブジェクトを必要とする点に注意してください。

```
Sub ECIDreply_Click()
    If Flow.Poll(Buffer) Then
        Text1.Text = Buffer.String
    Else
        Text1.Text = "No reply from CICS yet"
    End If
End Sub
```

CICS サーバー情報および接続状況

ECI COM クラスは、クライアント初期設定ファイルで構成されている CICS の名前および記述を提供します。**Connect COM** クラスは、特定の CICS サーバーの可用性を照会するためのメソッドを提供します。

オブジェクト変数は先に宣言されたとおりですが、今回は、**ECI**、**Connect** および **Flow COM** クラスを使用します。

```
'Declare object variables
Dim ECI As Cc10ECI
Dim Connect As Cc10Conn
Dim Flow As Cc10Flow
```

COM CICS ECI へのガイド

ユーザーの要求に応じて、オブジェクトが作成され、CICS サーバー名のリストとそれぞれについての記述が構成されます。

```
Sub ECIServers_Click()  
    Dim I as Integer  
  
    'Instantiate CICS ECI objects  
    Set ECI = New Cc10ECI  
    Set Connect = New Cc10Conn  
    Set Flow = New Cc10Flow  
  
    'List CICS server information  
    For I = 1 To ECI.ServerCount  
        List1.AddItem ECI.ServerName(I)  
        List1.AddItem ECI.ServerDesc(I)  
    Next  
End Sub
```

最初のサーバーへの同期状況呼び出しが行われ、その呼び出しの結果がテキスト・フィールドに表示されます。

```
Connect.Details ECI.ServerName(1)  
Connect.Status Flow  
Text1.Text = Connect.ServerStatusText
```

作業単位内での ECI リンク呼び出し

UOW COM のクラスを使用して、1 つの作業単位内で CICS サーバーへ複数回のリンク呼び出しを行うことができます。そうすると、クライアント・プログラムは必要に応じて CICS サーバー内のリカバリー可能リソースへの更新をコミットあるいはバックアウトすることができるようになります。

この例では、UOW オブジェクトが作成され、**Connect.Link** 呼び出しへのパラメーターとして使用されています。

```

Sub ECIStartUOW_Click()
    'Instantiate CICS ECI objects
    Set Connect = New Cc10Conn
    Set Flow = New Cc10Flow
    Set UOW = New Cc10UOW
    Set Buffer = New Cc10Buf
    Connect.Details "CICSNAME", "sysad", "sysad"
End Sub

Sub ECILink_Click()
    'Set up the commarea buffer
    Buffer.SetString Text1.Text
    Buffer.SetLength 80
    'Make the link call as part of a Unit of Work
    Connect.link Flow, "ECITSQ", Buffer, UOW
End Sub

```

何回かのリンク呼び出しを行った後では、**Cc1 UOW** インターフェースの **Commit** または **Backout** メソッドを使用することができます。

```

Sub Commit_Click()
    'Commit the CICS updates
    UOW.Commit Flow
End Sub
Sub Backout_Click()
    'Backout the CICS updates
    UOW.Backout Flow
End Sub

```

UOW オブジェクトが使用されない (**Connect.Link** 呼び出しに NULL 値が指定される) と、各リンク呼び出しが完全な作業単位 (CICS サーバーの LINK SYNCONRETURN に相当) になります。

作業論理単位を使用するときは、アクティブな作業単位のバックアウトまたはコミットを必ず行うようにしてください。特にプログラムの終了時には必ず行ってください。作業論理単位がまだアクティブかどうかは、`uowId` メソッドでゼロ以外の値が検知されるかどうかで知ることができます。

Visual Basic では、UOW 変数を DIM 宣言し、オブジェクトを作成しないと、それは値 **Nothing** になるとみなされ、リンク呼び出しは作業単位を呼び出しと関連させません。VBScript では、しかし、変数が何にもセットされていないことを明示的に確認する必要があります。これを行なうには、下記を

```
Set UOW=Nothing
```

ユーザーのリンク呼び出しの前に行なってください。

VB で COMAREAS を処理する

CommArea は、ユーザーがサーバーと送受信するすべての情報を含むストレージのブロックです。このため、この情報を収容できる十分な大きさで

COM CICS ECI へのガイド

CommArea を作成しなければなりません。たとえば、12 バイトのシリアル番号をサーバーに送信する必要がある場合でも、サーバーからは最大で 20 KB 戻される可能性もあります。すなわち、サイズ 20 KB の Commarea を作成しなければなりません。これを行なうには、以下のようにコード化します。

```
Set Buf = new CcIOBuf ' create extensible buffer object
Buf.SetString(serialNo)
Buf.setLength(20480) ' stores Nulls in the unused area
```

上記の例では、Commarea にシリアル番号が与えられ、必要な量までバッファが増やされ、余分なエリアはヌルで埋められます。ヌルで埋めることは重要な意味を持ち、これにより、確実にサーバーに伝送される情報が最少に抑えられます。クライアントは余分なヌルをストリップし、サーバーに 12 バイトのみを伝送します。

ECI ユーザー ID およびパスワード管理

パスワード満了管理をサポートするサーバーでセキュリティー管理を行えるようになりました。サポートされているサーバーとプロトコルについて詳しくは「CICS® ファミリー: クライアント / サーバー プログラミング」を参照してください。

これらの機能を使用するには、最初に Connection オブジェクトを作成し、さらに **Details** メソッドを呼び出し、ユーザー ID とパスワードをオブジェクトに関連させます。選択可能な 2 つのメソッドは、サーバー・セキュリティー・システムを持つ connection オブジェクト内でユーザー ID とパスワードを検査する **VerifyPassword** と、サーバーでパスワードを変更することができる **ChangePassword** です。成功すると、続いて connection オブジェクトのパスワードが更新されます。

いずれかの呼び出しが成功すると、CcIOSecAttr オブジェクトが戻されます。このオブジェクトにより最新の検査済み日付、満了時間および最新のアクセスされた日付などの情報へのアクセス可能になります。たとえば、最新の検査済み時刻を照会すると、CcIOSecTime オブジェクトがユーザーに戻され、SecTime COM クラス・メソッドを使用してさまざまな形式で情報を入手することができます。以下に示すコードは、これらのさまざまなオブジェクトの使用方法を示します。


```

' Connection object already created called conn

on error goto pemhandler

dim SecAttr as Cc10SecAttr
dim LastVerified as Cc10SecTime
dim lvdate as Date

set SecAttr = conn.VerifySecurity
set LastVerified = SecAttr.LastVerifiedTime

lvdate = LastVerified.GetDate
strout = Format(lvdate, "hh:mm:ss, dddd, mmm d yyyy")
Text1.Text = strout

exit sub

pemhandler:

' handle a expired password here maybe
end sub

```

Visual Basic を使用した CICS 3270 アプリケーションへの接続

注: COM クラスには、DBCS フィールドを含む 3270 データ・ストリームの特定のサポートは含まれていません。また、これらのクラスは、SBCS と DBCS との混合フィールドを含む 3270 データ・ストリームをサポートしていません。

最初に、使用する EPI インターフェースのオブジェクト変数を宣言します。通常は、Visual Basic プログラムの General Declarations セクションで行います。

```

Dim EPI As Cc10EPI
Dim Terminal As Cc10Terminal
Dim Session As Cc10Session
Dim Screen As Cc10Screen
Dim Field As Cc10Field

```

必要な EPI オブジェクトが、Visual Basic **New** ファンクションを使用してインスタンス化されます。これは、**Form_Load** サブルーチンによって、または後のどれかの工程でなんらかのユーザーのアクションに応じて行われます。

CICS クライアント EPI を初期化するために、Cc10EPI オブジェクトを最初に作成する必要があります。次に、Cc10Terminal オブジェクトを作成して、**Terminal.Connect** メソッドを使用して、特定の CICS サーバーに対する接続を確立します。このメソッドの最初のパラメータは、CICS サーバー名 (クライアント初期設定ファイルで構成されている) であり、それ以外のパラメータ

COM CICS EPI へのガイド

一はその他の接続の詳細を指定します (84ページの『Connect』を参照してください)。

```
Sub EPIConnect_Click()  
    'Create Cc1.EPI first to initialize EPI  
    Set EPI = New Cc1OEPI  
    'Create a terminal object and connect to CICS  
    Set Terminal = New Cc1OTerminal  
    Terminal.Connect "CICSNAME", "", ""  
    'Create a session object (defaults to synchronous)  
    Set Session = New Cc1OSession  
End Sub
```

CICS 3270 セッションの実行

Cc1OTerminal オブジェクトを必要な CICS サーバーに接続すると、**Terminal**、**Session**、**Screen** および **Field** COM クラスを使用して CICS でトランザクションが開始され、アプリケーションで必要な 3270 フィールドにアクセスしながら、3270 パネルをナビゲートします。

必要な CICS トランザクションが、4 文字のトランザクション・コードを使用して開始されます。最初のトランザクション・データは、**Terminal.Start** メソッドでも提供できますが、この例ではデータの請求は行われていません。CICS から戻された 3270 データにアクセスするには、**terminal** オブジェクトから **screen** オブジェクトを入手してから、さまざまなメソッドを使用してその画面のフィールドを入手し、そのフィールドのテキストや属性を読んだり更新します。

```
Sub EPIStart_Click()  
    'Start CESN transaction  
    Terminal.Start Session, "CESN", ""  
    'Get the screen object  
    Set Screen = Terminal.Screen  
    'Output the text from some 3270 fields  
    Set Field = Screen.FieldByIndex(5)  
    List1.AddItem Field.Text  
    Set Field = Screen.FieldByIndex(6)  
    List1.AddItem Field.Text
```

CESN トランザクションがユーザーからの入力を待機しています。いくつかのフィールドにプログラムからテキストを入力してトランザクションを続行することができます。この例では、CICS へ PF3 を送って単にトランザクションを終了しています。

```
    'Send PF3 back to CICS to end CESN  
    Screen.SetAID cclPF3  
    Terminal.Send Session
```

```

'Output the text from a 3270 field
Set Field = Screen.FieldByIndex(1)
List1.AddItem Field.Text
End Sub

```

最後に、端末を切断して EPI を終了します。これが行なわれる順序は非常に重要です。

```

Sub EPIDone_Click()
Term.Disconnect
'Delete the EPI COM objects
Set Field = Nothing
Set Screen = Nothing
Set Session = Nothing
Set Terminal = Nothing
Set EPI = Nothing
End Sub

```

EPI 呼び出し同期タイプ

CICS クライアント EPI COM クラスは、同期（「ブロッキング」）および遅延同期（「ポーリング」）プロトコルをサポートします。Visual Basic 環境は、C++ クラスで使用可能な非同期呼び出しをサポートしません。

前出の例では、デフォルトの同期タイプである `cclSync` を指定して `Session` オブジェクトが使用されています。`Session` オブジェクトが **Terminal.Start** または **Terminal.Send** の最初のパラメーターとして使用されると、CICS に同期リンク呼び出しが行われました。Visual Basic プログラムは、CICS からの応答を受け取るまでの間、ブロックされました。呼び出しが戻されると、CICS から更新された画面データが戻され、即時に `Screen` オブジェクトで使用可能になります。

遅延同期呼び出しを行う場合、**Session.SetSyncType** メソッドを使用して、`Session` を `cclDSync` に設定します。この `Session` オブジェクトを **Terminal.Start** または **Terminal.Send** 呼び出しで使用すると、画面の内容が CICS に 3270 データ・ストリームとして伝送されますが、メソッドは即時に戻されます。これにより、Visual Basic プログラムは、CICS サーバーのトランザクションを実行しながら、ユーザーの会話を含むその他のタスクを続行することができます。それ以降に CICS から更新された 3270 画面は、後で `Terminal` オブジェクトの **Poll** メソッドを使用して検索する必要があります。

COM CICS EPI へのガイド

```
Sub EPIDSync_Click()  
    'Create a session object (deferred synchronous)  
    Set Session = New CcIOSession  
    Session.SetSyncType cclDSync  
    Terminal.Start Session, "CESN", ""  
End Sub
```

トランザクションは、CICS サーバーで現在実行中です。後の工程で (ユーザーのアクションに応じて、あるいは Visual Basic プログラムがその他のタスクを完了するとき)、**Terminal.PollForReply** メソッドを使用して CICS からの応答が収集されます。

```
Sub EPIReply_Click()  
    If terminal.State <> cclDiscon And terminal.State <> cclError Then  
        If terminal.PollForReply Then  
            'Screen has been updated, output some fields  
            Set Screen = Terminal.Screen  
            Set Field = Screen.FieldByIndex(1)  
            List1.AddItem Field.Text  
        Else  
            List1.AddItem "No Reply from CICS yet"  
        End If  
    End If  
End Sub
```

CICS サーバー・トランザクションは、1 回の **Terminal.Start** 呼び出しまたは **Terminal.Send** 呼び出しに応じて、複数の応答を送ることができます。したがって、すべての応答を収集するには、複数の **Terminal.PollForReply** 呼び出しが必要になります。**Terminal.State** メソッドを使用して、それ以上の応答が予期されているかどうかを調べます。予期される応答がある場合には、戻される値は `cclServer` になります。

CICS サーバー情報

EPI COM クラスは、クライアント初期設定ファイルで構成されている CICS サーバーの名前と記述を提供します。

EPI オブジェクトが前出の例のように作成され、CICS サーバー名のリストと各サーバーについての記述がリスト・ボックス「List1」に出力されます。

```
Sub EPIServers_Click()  
    Dim I  
    'Instantiate CICS EPI object  
    Set EPI = New CcIOEPI  
    'List CICS server information  
    For I = 1 To EPI.ServerCount  
        List1.AddItem EPI.ServerName(I)  
        List1.AddItem EPI.ServerDesc(I)  
    Next
```

BMS マップ・データの EPI COM クラスでの使用

多くの CICS サーバー・プログラムが、基本マッピング・サポート (Basic Mapping Support (BMS)) を使用して、3270 画面設計を実施します。サーバー・プログラムは、個別の画面マップと、そのマップの 3270 フィールドに記号名を使用します。BMS ソース・ファイルが使用可能であれば、それらを CICS クライアント開発環境にコピーして、Visual Basic EPI プログラムの実施に使用します。

CICS クライアントから提供される CICS BMS Conversion Utility (CICSBMSC.EXE) は、Visual Basic 定義ファイル (.BAS ファイル) を、ソース BMS ファイル (.BMS ファイル) から作成します。そうすると、この定義ファイルを Visual Basic プログラムに組み込むことができ、サーバー・プログラム内でマップとそのフィールドを識別するために使用されたものと同じ記号名を、EPI **Map** COM クラスを持つクライアント・プログラムでも使用することができます。

Visual Basic 定義を作成するために変換ユーティリティを実行する際には、/B オプションを指定する必要があります。

```
CICSBMSC /B <filename>.BMS
```

提供されているサンプル・サーバー・プログラム EPIINQ の BMS ファイルで変換ユーティリティを実行すると、以下の Visual Basic 定義が作成されます。

```
Public Const MAPINQ = xx.xx.xx.xx.xx.xx.xx
Public Const MAPINQ1_PRODNAME = xx
Public Const MAPINQ1_APPLID = xx
Public Const MAPINQ1_TIME = xx
```

最初の定数 MAPINQ1 は、マップを識別し、そのマップの配置、サイズおよびレイアウトについて記述した情報を提供します。それ以外の定数は、そのマップ内の指定されたフィールドについて定義します。

次の例は、**Map** COM クラスを使用して、フィールドに BMS 記号名でアクセスする方法について示しています。

```
Dim EPI As Cc10EPI
Dim Terminal As Cc10Terminal
Dim Session As Cc10Session
Dim Screen As Cc10Screen
Dim Map as Cc10Map
Dim Field As Cc10Field
```

COM CICS EPI へのガイド

最初に、EPI が初期化され、3270 端末の CICS との接続が、前出の例にあるように開始されます。

```
Sub EPIConnect_Click()  
  'Create Cc1.EPI first to initialize EPI  
  Set EPI = New Cc10EPI  
  'Create a terminal object and connect to CICS  
  Set Terminal = New Cc10Terminal  
  Terminal.Connect "CICSNAME", "", ""  
  'Create a session object (defaults to synchronous)  
  Set Session = New Cc10Session  
End Sub
```

次に、BMS アプリケーションが開始されます。この例では、提供されるサーバー・プログラム EPIINQ を実行するトランザクション・コード「EPIC」を使用します。

```
Sub EPIRunBMS_Click()  
  Terminal.Start Session, "EPIC", ""  
  Set Screen = Terminal.Screen
```

ここで、CICS サーバー・プログラムはクライアントに最初の画面を戻します。すでに認識済みのマップ『MAPINQ1』が戻されることが预期されています。したがって、Map オブジェクトを作成し、**Map.Validate** メソッドを使用して初期化を行い、预期したとおりの 3270 画面が戻されたかどうかを調べます。そうすると、**Map.FieldByName** メソッドを使用してフィールドにアクセスすることが可能になります。

```
Set Map = New Cc10Map  
If (Map.Validate(Screen,MAPINQ1)) Then  
  Set Field = Map.FieldByName(MAPINQ1_PRODNAM)  
  List1.AddItem Field.Text  
  Set Field = Map.FieldByName(MAPINQ1_TIME)  
  List1.AddItem Field.Text  
Else  
  List1.Text= "Unexpected screen data"  
End If
```

さらに複雑なアプリケーションの場合には、選択したフィールドにデータを入力して、必要な AID キー (Enter、Clear、PF、または PA キー) を設定して、必要に応じて画面を通じて操作を行ってください。クライアント・アプリケーションは、**Screen COM** クラス (およびその **FieldByIndex** および **FieldByPosition** メソッド) と **Map COM** クラスを使い分けることができます。

VBScript を使用した CICS 3270 アプリケーションへの接続

これはまた、Visual Basic に酷似していますがオブジェクト作成方法が異なります。ユーザーの変数を DIM 宣言する必要はありませんが、そうするなら良いコーディング練習になります。ビジュアル・ベーシックと同様に、COM オブジェクトは 3270 データ・ストリーム内の DBCS フィールドをサポートしません。オブジェクトを作成する場合は、**CreateObject** 関数を次のように使用する必要があります。

```
Sub EPICheck_Click()  
    ' Create Ccl.EPI first to initialise EPI  
    Set EPI = CreateObject("Ccl.EPI")  
    ' Create a terminal object and connect to CICS  
    Set Terminal = CreateObject("Ccl.Terminal")  
    Terminal.Connect "CICSNAME", "", ""  
    ' Create a session object (defaults to synchronous)  
    Set Session = CreateObject("Ccl.Session")  
End Sub
```

同様な方法で、Map オブジェクトを作成するには、次のように発行します。

```
Set Map = CreateObject("Ccl.MAP")
```

Screen オブジェクトと Fields オブジェクトがユーザーのために作成されま
す。

自動トランザクション始動 (ATI) のサポート

CICS サーバーの API 呼び出し EXEC CICS START を使用すると、サーバー・プログラムは特定の端末でトランザクションを開始することができます。自動トランザクション開始 (ATI) と呼ばれるこの機構では、これらのトランザクションと通常のクライアント始動トランザクションとの間の対話をハンドルするために、クライアント・サイドで追加のプログラミングが必要です。

クライアント・アプリケーションは、**setATI** および **queryATI** メソッドを端末 COM クラスで使用して、ATI トランザクションを使用可能にするかどうかを制御することができます。デフォルトでは、ATI は設定は使用不可に設定されています。次に示すコード・フラグメントは、特定の端末で ATI を使用可能にする方法です。

COM CICS EPI へのガイド

```
// Create terminal connection to CICS server
Dim terminal as CclTerminal
Set terminal = new CclTerminal
terminal.details "MYSERVER", "", ""
terminal.setATI CclATIEnabled
```

CICS クライアントは、トランザクションの処理中は端末の ATI をキューします。Ccl Terminal クラスはトランザクションが終了するとすぐに未解決の ATI を実行し、ATI 応答をハンドルするのに必要な追加のプログラミングを呼び出し、使用されている呼び出し同期タイプにしたがって、クライアント始動トランザクションの前または間で ATI を実行します。

同期 端末 **送信** メソッドを呼び出す場合は、未解決の ATI はクライアント始動トランザクションが完了した後で実行されます。端末クラスは ATI 応答を待ち、それから CclOScreen オブジェクトの内容を synchronous **send** 呼び出しの一部として更新します。クライアント始動トランザクションの前または間で ATI を生じさせたい場合は、Ccl Terminal **receiveATI** メソッドを呼び出して ATI について同期がとられるのを待ってください。

遅延同期

CclTerminal **Start** または **Send** メソッドが、据え置き同期セッションに対して呼び出された後、**Poll** メソッドまたは **PollForReply** メソッドが、応答を受信するために使用されます。未解決の ATI は最後の応答が受信されると (すなわち最後の **Poll** または **PollForReply** メソッドで) 開始されます。**Poll** または **PollForReply** メソッドを呼び出して、クライアント始動トランザクション間で、ATI の開始および応答の受信をすることもできます。

Poll または **PollForReply** メソッドはクライアント始動トランザクションの前または間で呼び出すことができるので、**receiveATI** メソッドは遅延同期セッションに不要 (また無効) です。

セキュリティ管理

パスワード満了管理をサポートするサーバーで、セキュリティ管理を行えるようになりました。サポートされているサーバーとプロトコルについては「CICS® ファミリー: クライアント / サーバー プログラミング」を参照してください。

これらの機能を使用するには、最初に Terminal オブジェクトを作成し、**SetTerminalDefinition** メソッドを呼び出し、ユーザー ID とパスワードをオブジェクトに関連させます。選択可能な 2 つのメソッドは、サーバー・セキュ

リティー・システムを持つ terminal オブジェクト内でユーザー ID とパスワードを検査する **VerifyPassword** と、そのサーバーでパスワードを変更するために使用できる **ChangePassword** です。成功すると、続いて terminal オブジェクトのパスワードが更新されます。

いずれかの呼び出しが成功すると、CclOSecAttr オブジェクトが戻されます。このオブジェクトにより最新の検査済み日付と時間、満了日および最新のアクセスされた日付と時間などの情報へのアクセスが可能となります。たとえば、最新の検査済み日付を照会すると、さまざまな形式で情報を入手できるようにする CclOSecTime オブジェクトが戻されます。次のコードでは、これらのさまざまなオブジェクトの使用方法を示します。

```
' Terminal object already created called term
on error goto pemhandler

dim SecAttr as CclOSecAttr
dim LastVerified as CclOSecTime
dim lvddate as Date

set SecAttr = term.VerifyPassword
set LastVerified = SecAttr.LastVerifiedTime

lvddate = LastVerified.GetDate
strout = Format(lvddate, "hh:mm:ss, dddd, mmm d yyyy")
Text1.Text = strout

exit sub

pemhandler:
' handle a expired password here maybe

end sub
```


第2部 COM クラス – 参照

第4章 Buffer COM クラス	35	メソッド	47
インターフェースの選択	35	ErrorFormat	47
オブジェクト作成	35	ErrorOffset	48
メソッド	36	ErrorWindow	48
AppendString	36	ExCode	48
Data	36	ExCodeText	49
ExtractString	36	ServerCount	49
InsertString	37	ServerDesc	49
Length	37	ServerName	50
Overlay	37	SetErrorFormat	50
SetData	37		
SetLength	38	第7章 EPI COM クラス	51
SetString	38	インターフェースの選択	51
String	38	オブジェクト作成	51
		メソッド	51
第5章 Connect COM クラス	39	Diagnose	51
インターフェースの選択	39	ErrorFormat	52
オブジェクト作成	39	ErrorOffset	52
メソッド	40	ErrorWindow	52
AlterSecurity	40	ExCode	52
Cancel	40	ExCodeText	53
Changed	40	ServerCount	53
ChangePassword	41	ServerDesc	54
Details	41	ServerName	54
Link	42	SetErrorFormat	54
MakeSecurityDefault	43	State	54
Password	43	Terminate	55
ServerName	43		
ServerStatus	43	第8章 Field COM クラス	57
ServerStatusText	44	インターフェースの選択	57
Status	44	メソッド	57
TranDetails	44	AppendText	57
UnpaddedPassword	45	BackgroundColor	58
UnpaddedServerName	45	BaseAttribute	58
UnpaddedUserid	45	Column	58
UserId	45	DataTag	58
VerifyPassword	45	ForegroundColor	59
		Highlight	59
第6章 ECI COM クラス	47	InputProt	59
インターフェースの選択	47	InputType	60
オブジェクト作成	47	Intensity	60

Length	60	SetAID	73
Position	61	SetCursor	73
ResetDataTag	61	Width	74
Row	61	第12章 SecAttr COM クラス	75
SetBaseAttribute	61	インターフェースの選択	75
SetExtAttribute	61	パブリック・メソッド	75
SetText	62	ExpiryTime	75
Text	62	InvalidCount	75
TextLength	62	LastAccessTime	76
Transparency	62	LastVerifiedTime	76
第9章 Flow COM クラス	63	第13章 SecTime COM クラス	77
インターフェースの選択	63	インターフェースの選択	77
オブジェクト作成	63	パブリック・メソッド	77
メソッド	63	Day	77
AbendCode	63	GetDate	77
CallType	64	Hours	77
CallTypeText	64	Hundredths	78
Diagnose	64	Minutes	78
Flowid	64	Month	78
ForceReset	64	Seconds	78
Poll	64	Year	78
SetSyncType	65	第14章 Session COM クラス	79
SetTimeout	65	インターフェースの選択	79
SyncType	66	オブジェクト作成	79
Timeout	66	メソッド	79
Wait	66	Diagnose	79
第10章 Map COM クラス	67	SetSyncType	79
インターフェースの選択	67	State	80
オブジェクト作成	67	TransId	81
メソッド	67	第15章 Terminal COM クラス	83
ExCode	67	インターフェースの選択	83
FieldByName	68	オブジェクト作成	83
Validate	68	メソッド	83
第11章 Screen COM クラス	71	AlterSecurity	83
インターフェースの選択	71	CCSID	84
メソッド	71	ChangePassword	84
CursorCol	71	Connect	84
CursorRow	71	Devtype	85
Depth	72	Diagnose	85
FieldByIndex	72	Disconnect	85
FieldByPosition	72	DisconnectWithPurge	85
FieldCount	72	DiscReason	86
MapName	72	ExCode	86
MapSetName	73		

ExCodeText	86	Start	92
Install	87	State.	92
MakeSecurityDefault.	87	TermId	92
NetName	87	TransId	93
Password	87	Userid	93
Poll	87	VerifyPassword	93
PollForReply	88		
QueryATI	88	第16章 UOW COM クラス.	95
ReadTimeout	89	インターフェースの選択	95
ReceiveATI	89	オブジェクト作成	95
Screen	89	メソッド	95
Send.	89	BackOut	95
ServerName	90	Commit.	96
SetATI	90	ForceReset.	96
SetTermDefns	90	UowId	96
SignonCapability	91		

以下の章では、すべてのクライアント COM クラスに関する説明をアルファベット順に記載しています。インターフェース内で、メソッドのアルファベット順リストが示されています。これらのインターフェースの使用方法に関する詳細については、第 1 部を参照してください。

Visual Basic オブジェクト・ブラウザーを使用して、これらのメソッドの要約を検索できます。 ECI クラスおよびメソッドは CcIECILib ライブラリーにリストされ、 EPI クラスは CcIEPILib ライブラリーにリストされています。ユーザーの Visual Basic Project の最初に、適切な解説書を追加してあることを確認してください。

第4章 Buffer COM クラス

CclOBuffer オブジェクトには、情報の保持に使用できるメモリー内のデータ域が入っています。CclOBuffer オブジェクト特有の使用法は、CICS サーバーとの間でデータを受け渡しするときに使用する COMMAREA の保持です。

CclOBuffer オブジェクトは、主にバイト (2 進) データでの使用を対象としています。一般に、COMMAREA には、CICS サーバー C プログラムにより作られることが多い、アプリケーション固有のデータ構造が入っています。Visual Basic で 2 進データを処理する場合にお勧めする方法は、バイト・データ・タイプです。**SetData** および **Data** メソッドを使用すると、CclOBuffer オブジェクトの内容に、バイト配列としてアクセスすることができるようになります。CclOBuffer オブジェクトは、ストリング・データとして使用することができ、ストリングは 1 バイトの ANSI 文字として保管されますが、コード・ページ変換や DBCS はサポートしていません。32 ビット環境では Visual Basic は、2 バイト 1 コードの文字表示を使用し、COM クラスはこの 2 バイト 1 コード文字を 1 バイトの ANSI へ変換したり、あるいは 1 バイトの ANSI から変換したりします。

CclOBuffer オブジェクトを作成すると、メモリーの区域をバッファーとして割り振ることができます。このバッファーの長さは、**SetLength** メソッドを通じて明示的に設定することができます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as Ccl0Buf
```

お勧めしたい方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

COM クラス: Buffer

```
set var = CreateObject("Ccl.Buffer")
set var = New Cc10Buf
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

AppendString

AppendString(*string* as String)

string

ソース・ストリング

ストリングを **Ccl.Buffer** オブジェクトの既存のデータに追加します。

Data

Data() as Variant

バッファの内容を 1 バイト配列として戻します。

ExtractString

ExtractString (*offset* as Integer[,
length as Integer]) as String

offset

データ域へのオフセット。

length

抽出されるストリングのバイト単位の長さ (オプション)。

指定のオフセットで開始されるデータ域からストリングを戻します。

length を指定しないと、**ExtractString** は最初の NULL 文字を見つけるまでデータを戻します。*length* を指定すると、**ExtractString** はストリング内の NULL 文字も含めて、要求されたバイト数ほどデータを戻します。

InsertString

InsertString (*offset* as Integer,
string as String)

offset

ストリングが挿入される、データ域内のオフセット。

string

ソース・ストリング

指定のオフセットのデータ域へ、指定のストリングを挿入します。

Length

Length() as Integer

データ域の長さ (バイト) を戻します。

Overlay

Overlay (*offset* as Integer,
string as String)

offset

ストリングが挿入される、データ域内のオフセット。

string

ソース・ストリング

指定のオフセットから始まるデータ域を、指定のストリングでオーバーレイします。

SetData

SetData(*array* as Variant)

array

ソース・データを含んでいる配列。

提供された配列をバッファへコピーします。バイト、整数、および長い配列がサポートされます。

COM クラス: Buffer

SetLength

SetLength(*length* as Integer)

length

データ域の新しい長さ (バイト単位)。
データ域の現在の長さを変更します。buffer オブジェクトの長さを増やした場合は、余分なスペースはヌルで埋められます。クライアントは CICS サーバーにバッファを送信する前にすべてのヌルを切り捨てます。

SetString

SetString(*string* as String)

string

ソース・ストリング
提供されたストリングをバッファへコピーします。

String

String() as String

Ccl.Buffer オブジェクトの内容を、ストリングとして戻します。

第5章 Connect COM クラス

Connect COM クラスは、クライアントと指定されたサーバーとの間の ECI 接続を保守および表示するために使用します。サーバーへのアクセスは、オプションとしてユーザー ID およびパスワードで制御されます。**Connect** COM クラスは、サーバー内のプログラムを呼び出すことも、接続の状態に関する情報を得ることもできます。

Connect COM クラスは、これを使用して CICS への呼び出しを行う前に、**Details** メソッド (あるいはオプションとして、**TranDetails** メソッド) で初期化を行う必要があります。

クライアントとサーバー間の会話は、すべて、**CclOFlow** オブジェクトおよび **CclOConnect** オブジェクトを必要とします。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CclOConn
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.Connect")  
set var = New CclOConn
```

Visual Basic でお勧めする方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

AlterSecurity

AlterSecurity(*newUserid* as String, *newPassword* as String)

newUserid

新規のユーザー ID。

newPassword

新規のユーザー ID に対応する新規パスワード。

次のリンク呼び出しで使用するユーザー ID とパスワードをセットします。

Cancel

Cancel(*flow* as Object)

または

Cancel(*flow* as CclOFlow)

flow

クライアント / サーバー呼び出しを制御するために使用する CclOFlow オブジェクト。

この接続と関連付けられるサーバーに対して、それ以前に出された **Changed** 呼び出しをすべて取り消します。

Changed

Changed(*flow* as Object)

または

Changed(*flow* as CclOFlow)

flow

クライアント / サーバー呼び出しを制御するために使用する CclOFlow オブジェクト。

現行接続状況が変更になったときに、クライアントに通知するようサーバーに要求します。この接続に、未解決の **Changed** 呼び出しがあると、呼び出しは無視されます。

ChangePassword

ChangePassword (*newPassword* as String) as Object

または

ChangePassword (*newPassword* as String) as CclIOSecAttr

newPassword

新規パスワード。

Connect オブジェクトに保持されているパスワードと、Connect オブジェクトに保持されているユーザー ID 用に外部セキュリティー・マネージャーが記録したパスワードを、クライアント・アプリケーションが変更できるようにします。外部セキュリティー・マネージャーは、Connect オブジェクトによって定義されたサーバーに配置されているとみなされます。エラーが生じない場合は、CclIOSecAttr オブジェクトが戻されます。

Details

Details (*serverName* as String, *userId* as String, *password* as String)

serverName

サーバーの名前。名前が指定されていない場合は、デフォルトのサーバー (クライアント初期設定ファイルで指定された最初のサーバー) が使用されます。この名前は、サーバーに最初の呼び出しが行われた後で、

ServerName メソッドを使用して探すことができます。長さは、ブランクの埋め込みによって 8 文字に調整されます。

userId

(必要に応じて) ユーザー ID。長さは、ブランクの埋め込みによって 16 文字に調整されます。

password

(必要に応じて) *userID* に指定されたユーザー ID に対応するパスワード。長さは、ブランクの埋め込みによって 16 文字に調整されます。

このメソッドは、CICS サーバーの詳細を提供するために使用します。CICS サーバーとの会話は、**Link**、**Status**、または **Changed** メソッドが呼び出される

COM クラス: Connect

までは行われません。接続が状況呼び出しにのみ使用される場合、あるいはサーバーにセキュリティーがない場合、ユーザー ID およびパスワードは必要ありません。

Link

Link (*flow* as Object,
 programName as String,
 commArea as Object,
 unitOfWork as Object)

または

Link (*flow* as CclOFlow,
 programName as String,
 commArea as CclOBuf,
 unitOfWork as CclOUOW)

flow

クライアント / サーバー呼び出しを制御するために使用する CclOFlow。

programName

呼び出しを受けているサーバー・プログラムの名前。必要なら、ブランクの埋め込みか切り捨てによって、長さを 8 文字に調整することができます。

commArea

呼び出し先プログラムに渡すデータを COMMAREA に保持する CclOBuffer オブジェクト。COMMAREA が送られない場合には、NULL 値を指定する必要があります。

unitOfWork

この呼び出しが関連付けられている作業単位 (UOW) を識別する CclOUOW オブジェクト。UOW を使用しない場合には、NULL 値を指定する必要があります。

サーバーで指定のプログラムを呼び出します。サーバー・プログラムは、着信呼び出しを EXEC CICS LINK 呼び出しとみなします。

MakeSecurityDefault

MakeSecurityDefault()

Connect オブジェクトの構造に指定したように、サーバーに渡される ECI および EPI 要求のデフォルトとして、このオブジェクトの現行ユーザー ID とパスワードが使用されることを、クライアントに通知します。

Password

Password() as String

CclOConnect オブジェクトが保持する、スペースで埋められたパスワード。

ServerName

ServerName() as String

CclOConnect オブジェクトが保持しており、クライアント初期設定ファイルがリストしている サーバー・システムの名前か、あるいは、(デフォルト・サーバーが使用され、呼び出しが一回も行われていない場合は) ブランクを戻します。

ServerStatus

ServerStatus() as Integer

または

ServerStatus() as CclConnectStatusCodes

それ以前の **status** または **changed** 要求で設定されたサーバー接続の状況を戻します。以下の値を使用することができます。

cclUnknown

CICS サーバー状況は未確認です。

cclAvailable

CICS サーバーは使用可能です。

cclUnavailable

CICS サーバーは使用不能です。

COM クラス: Connect

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

ServerStatusText

ServerStatusText() as String

前の **status** または **changed** 要求によって設定された、サーバーの可用性を示すストリングを戻します。

Status

Status(*flow* as Object)

または

Status(*flow* as CclOFlow)

flow

クライアント / サーバー呼び出しを制御するために使用する CclOFlow オブジェクト。

サーバー・コネクションの状況を要求します。

TranDetails

TranDetails (*runTran* as String, *attachTran* as String)

runTran

呼び出されたプログラムを実行する CICS トランザクション。デフォルトでは、デフォルトのサーバー・トランザクションを使用します。長さは、ブランクの埋め込みによって 4 文字に調整されます。

attachTran

呼び出されたプログラムを接続する CICS トランザクション。デフォルトは、デフォルト CPMI の使用です。長さは、ブランクの埋め込みによって 4 文字に調整されます。

このメソッドは、CICS に補足的な情報を提供するために使用します。情報はオプションですが、プログラムが CICS サーバーで実行する環境に反映させるために使用することができます。

UnpaddedPassword

UnpaddedPassword() as String

CclIOConnect が保持するパスワードを戻しますが、末端の部分にスペースが埋め込まれていないパスワードを戻します。

UnpaddedServerName

UnpaddedServerName() as String

CclIOConnect オブジェクトが保持するサーバー名を戻しますが、末端部分にスペースが埋め込まれないサーバー名を戻します。

UnpaddedUserid

UnpaddedUserid() as String

CclIOConnect オブジェクトが保持するユーザー ID を戻しますが、末端部分にスペースが埋め込まれないユーザー ID を戻します。

Userid

Userid() as String

CclIOConnect オブジェクトが保持するユーザー ID をスペースで埋めて戻すか、またはユーザー ID がなければブランクを戻します。

VerifyPassword

VerifyPassword() as Object

または

VerifyPassword() as CclIOSecAttr

Connect オブジェクトに保持されているパスワードと、Connect オブジェクトに保持されているユーザー ID 用に外部セキュリティー・マネージャーが記録したパスワードが一致しているかどうかを、クライアント・アプリケーション

COM クラス: Connect

が検査できるようにします。外部セキュリティー・マネージャーは、Connect オブジェクトによって定義されたサーバーに配置されているとみなされます。エラーが生じない場合は、 CclOSecAttr オブジェクトが戻されます。

第6章 ECI COM クラス

ECI COM クラスを使用するすべてのアプリケーションは、最初に Cc10ECI オブジェクトを作成する必要があります。

ECI COM クラスは、指定可能な CICS サーバーについての詳細を提供します。エラー情報を入手するために使用することもできます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが使用可能です。

```
Dim var as Object  
Dim var as Cc10ECI
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.ECI")  
set var = New Cc10ECI
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

ErrorFormat

ErrorFormat() as Integer

エラー・メッセージ・フォーマットのための現行の設定を示す値を戻します。有効な値の現行リストについては、SetErrorFormat を参照してください。

ErrorOffset

ErrorOffset() as Long

ERR.Number メソッドで検索された CICS クライアントのエラー値を、ドキュメント化された ExCode エラー値に変換するために使用できる値を戻します。この方法の詳細については、12ページの『例外処理』を参照してください。

ErrorWindow

— 使用すべきでないメソッド —

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ErrorWindow(*display* as Boolean)

display

- true** エラー・ウィンドウがユーザーに対して表示されるようにします。これは、デフォルトの設定です。
- false** エラー・ウィンドウはユーザーに対して表示されません。アプリケーションは、ExCode メソッドを使用してエラーの有無を検査する必要があります。

ExCode

— 使用すべきでないメソッド —

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ExCode() as Integer

または

ExCode() as CciECIExceptionCodes

最も新しい ECI エラーを示す列挙型を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

ExCodeText メソッドは、エラー値に関して説明するテキスト・ストリングを戻す点に注意してください。

ExCodeText

使用すべきでないメソッド

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ExCodeText() as String

最新の ECI エラーについて説明するテキストを含んでいるストリングを戻します。

ServerCount

ServerCount() as Integer

クライアント初期設定ファイルの構成に基づいて、クライアントを接続することのできる使用可能なサーバーの数を戻します。

ServerDesc

ServerDesc(*index* as Integer) as String

index

リスト内の接続されたサーバーの数 (1 から始まる)。

indexth サーバーの記述を戻します。

ServerName

ServerName(*index* as Integer) as String

index

リスト内の接続されたサーバーの数 (1 から始まる)。

indexth サーバーの名前を戻します。

SetErrorFormat

SetErrorFormat(*format* as Integer)

format

- 0** 旧形式のもので、旧版との互換性のみの目的で提供されています。
- 1** 新規形式のもので、Visual Basic および VBScript **Err** オブジェクトの中により詳しい情報を提供します。この形式を使用することをお勧めします。

このメソッドを使用するとエラー・メッセージ形式を選択できます。

第7章 EPI COM クラス

EPI COM クラスは、CICS クライアントの EPI 関数を初期化します。使用可能な CICS サーバーに関する情報を入手するためのメソッドも持っています。CICS サーバーに接続する CclOTerminal オブジェクトを作成する前には、CclOEPI オブジェクトを作成してください。**Diagnose** メソッド、**ExCode** メソッド、および **State** メソッドは、エラー条件に関する情報を提供します。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CclOEPI
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.EPI")  
set var = New CclOEPI
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

Diagnose

Diagnose() as String

最新のエラーについての説明を含んでいる文字ストリングを戻します。

ErrorFormat

ErrorFormat() as Integer

エラー・メッセージ・フォーマットのための現行の設定を示す値を戻します。
有効な値の現行リストについては、54ページの『SetErrorFormat』を参照してください。

ErrorOffset

ErrorOffset() as Long

ERR.Number メソッドで検索された CICS クライアントのエラー値を、ドキュメント化された ExCode エラー値に変換するために使用できる値を戻します。
この方法の詳細については、12ページの『例外処理』を参照してください。

ErrorWindow

使用すべきでないメソッド

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ErrorWindow(*display* as Boolean)

display

- true** エラー・ウィンドウがユーザーに対して表示されるようにします。これは、デフォルトの設定です。
- false** エラー・ウィンドウはユーザーに対して表示されません。アプリケーションは、**ExCode** メソッドを使用してエラーの有無を検査する必要があります。

ExCode

使用すべきでないメソッド

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ExCode() as Integer

または

ExCode() as CciEPIExceptionCodes

条件コードを戻します。以下の値を使用することができます。

cclSystemError

内部 CICS クライアント・システム・エラーが発生しました。

cclUnknownServer

ServerDesc または **ServerName** メソッドに提供された *index* (索引) に対応する CICS サーバーがありません。

cclNoError

呼び出しは正常に実行されました。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

ExCodeText**使用すべきでないメソッド**

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ExCodeText() as String

最新の例外について説明するテキストを含んでいる文字列を戻します。

ServerCount**ServerCount() as Integer**

クライアント初期設定ファイルの構成に基づいて、クライアントを接続することのできる使用可能なサーバーの数を戻します。

ServerDesc

ServerDesc(*index* as Integer) as String

index

接続されたサーバーの索引番号 (1 から開始される)。
選択した CICS サーバーに関する記述か、あるいは、(指定したサーバーのクライアント初期設定ファイルに使用可能な情報がない場合には) NULL スtring を返します。

ServerName

ServerName(*index* as Integer) as String

index

接続されたサーバーの索引番号 (1 から開始される)。
要求した CICS サーバーの名前か、あるいは、(指定したサーバーのクライアント初期設定ファイルに使用可能な情報がない場合には) NULL スtring を返します。

SetErrorFormat

SetErrorFormat(*format* as Integer)

format

- 0 旧形式のもので、旧版との互換性のみの目的で提供されています。
- 1 新規形式のもので、Visual Basic および VBScript **Err** オブジェクトの中により詳しい情報を提供します。この形式を使用することをお勧めします。

このメソッドを使用するとエラー・メッセージ形式を選択できます。

State

State() as Integer

または

State() as CcIEPIStates

EPI の状態を示す値を返します。以下の値を使用することができます。

cclActive

初期化されました。

cclDiscon

終了しました。

cclError

エラー、12ページの『例外処理』を参照してください。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

Terminate**使用すべきでないメソッド**

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

Terminate()

CICS クライアント EPI を、制御された方法で終了します。

第8章 Field COM クラス

Field COM クラスは、3270 画面の 1 つのフィールドにアクセスするために使用します。

CclOField オブジェクトは、CICS サーバーからの 3270 データが CclOScreen オブジェクトによって処理されるときに、作成され削除されます。

Field オブジェクトは、CclOScreen オブジェクトの **fieldbyIndex** または **fieldbyPosition** メソッドを呼び出すと戻されます。例を以下に示します。

```
set var=Screen.fieldbyIndex(1)
```

このクラスのメソッドによって、フィールドのテキストと属性を読み取り、かつ更新することができます。次の伝送時に、更新済みのフィールドが CICS サーバーに送られます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CclOField
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

メソッド

AppendText

AppendText(*textString* as String)

textString

フィールドに追加されるテキスト・ストリング

COM クラス: Field

textString 内の文字を、フィールド内に既に存在するテキストの最後に追加します。

BackgroundColor

BackgroundColor() as Integer

または

BackgroundColor() as CclColorAttributes

108ページの『CclField カラー属性』にリストされているフィールドの背景色を示す値を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

BaseAttribute

BaseAttribute() as Integer

フィールドの 3270 基本属性を戻します。

Column

Column() as Integer

画面のフィールドの開始位置の列番号 (左端列を 1 とした) を戻します。

DataTag

DataTag() as Integer

または

DataTag() as CclModifiedAttributes

フィールド内のデータが修正されたかどうかを示す値を戻します。以下の値を使用することができます。

cclModified
cclUnmodified

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

ForegroundColor

ForegroundColor() as Integer

または

ForegroundColor() as CclColorAttributes

108ページの『CclField カラー属性』にリストされているフィールドの前景色を示す値を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

Highlight

Highlight() as Integer

または

Highlight() as CclHighlightAttributes

107ページの『CclField 強調表示属性』にリストされている、どのタイプの強調表示が使用されているかを示す値を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

InputProt

InputProt() as Integer

または

InputProt() as CclProtAttributes

フィールドが保護されているかどうかを示す値を戻します。以下の値を使用することができます。

```
cclProtect  
cclUnprotect
```

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

InputType

InputType() as Integer

または

InputType() as CclNumericAttributes

フィールドが英数字であるか数字であるかを示す値を戻します。以下の値を使用することができます。

```
cclAlphanumeric  
cclNumeric
```

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

Intensity

Intensity() as Integer

または

Intensity() as CclIntensityAttributes

フィールドが、標準、高輝度、ぼかし表示のいずれであるかを示す値を戻します。以下の値を使用することができます。

```
cclDark  
cclNormal  
cclIntense
```

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

Length

Length() as Integer

フィールドの全長を戻します。ここには、3270 属性バイト情報を保管するために使用される 1 バイトが含まれています。そのため、実際のデータ用のスペースは、このメソッドによって戻される値より 1 小さいです。TextLength メソッドも参照してください。

Position

Position() as Integer

画面の左上隅からのオフセットとして、フィールドの開始位置を戻します(最上行が 0 から 79、2 番目の行が 80 から 159... というように)。

ResetDataTag

ResetDataTag()

修正済みのデータ・タグ (MDT) を cclUnmodified にリセットします。

Row

Row() as Integer

画面のフィールドの開始位置の行番号を戻します。最上行は 1 です。

SetBaseAttribute

SetBaseAttribute(*Attribute* as Integer)

Attribute

フィールドに入力される基本 3270 属性の値。
3270 基本属性を設定します。

SetExtAttribute

SetExtAttribute(*Attribute* as Integer, *Value* as Integer)

Attribute

設定される拡張属性のタイプ。

Value

拡張 3270 属性を設定します。無効な 3270 属性タイプまたは値が指定されると、パラメーター例外が生じます。

COM クラス: Field

SetText

SetText(*textString* as String)

textString

フィールドに入力される NULL 文字で終了するテキスト。
textString をフィールドにコピーします。

Text

Text() as String

現在フィールドに保留されているテキストを戻します。

TextLength

TextLength() as Integer

現在フィールドに保留されている文字数を戻します。

Transparency

Transparency() as Integer

または

Transparency() as CclTransparencyAttributes

107ページの『CclField 透過性属性』にリストされているフィールドの背景透過性を示す値を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

第9章 Flow COM クラス

CclOFlow オブジェクトは、クライアント / サーバーのペアの ECI 通信を制御するために使用します。

CclOFlow オブジェクトは、各クライアントとサーバーとの会話 (クライアントからの呼び出しおよびサーバーからの応答) 用に作成され、使用後には破棄されます。CclOFlow オブジェクトは、再利用することができますが、すでに使用中の CclOFlow オブジェクトへの再利用要求は拒否されます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CclOFlow
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.Flow")  
set var = New CclOFlow
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

AbendCode

AbendCode() as String

4 文字の CICS トランザクション異常終了コードか、あるいは (異常終了が起っていない場合には)、スペースを戻します。

COM クラス: Flow

CallType

CallType() as Integer

または

CallType() as CclFlowCallTypes

フローが現在実行している呼び出しのタイプを返します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

CallTypeText

CallTypeText() as String

フローが現在実行している呼び出しのタイプをテキスト形式で返します。

Diagnose

Diagnose() as String

Flow オブジェクトの現在の状態を説明するテキストを返します。

Flowid

Flowid() as Integer

この Flow オブジェクトの固有な ID を返します。

ForceReset

ForceReset()

フローを非活動にし、リセットします。代表的には、このメソッドはフローを破棄した後に、Flow オブジェクトの再使用または削除に備えるのに使用します。

Poll

Poll(*commArea* as Object) as Boolean

または

Poll(*commArea* as CclIOBuf) as Boolean

commArea

戻された COMMAREA が配置される CclOBuffer オブジェクト。このパラメーターは、COMMAREA が戻されない場合には **Nothing** に設定することができます。

遅延同期の各 **Backout**、**Cancel**、**Changed**、**Commit**、**Link**、または **Status** のうちのどの呼び出し要求から応答を受け取ったかを示します。この方法は、遅延同期通信においてのみ有効です。以下の値を使用することができます。

True 応答を受信しました。
False 応答を受信していません。

SetSyncType**SetSyncType(syncType as Integer)**

または

SetSyncType(syncType as CclFlowSyncTypes)*syncType*

この CclOFlow オブジェクトに必要な同期タイプ。以下の値を使用することができます。

cclSync
cclDSync

この CclOFlow オブジェクトに必要な同期タイプをセットします。cclSync が使用される場合、このフローを使用する **link** 呼び出しおよび **status** 呼び出しは、CICS からの応答を受信するまでは呼び出しプログラムをブロックします。cclDSync が使用される場合、このフローを使用する **link** 呼び出しおよび **status** 呼び出しは即時に呼び出しプログラムに戻されます。このプログラムは、後に CICS からの応答を受信する際には **Poll** メソッドを使用することができるようになります。

SetTimeout**SetTimeout(Timeout as Integer)**

フローの次の活動化用に Flow オブジェクトのタイムアウト値をセットします。フローがアクティブな間にこの値をセットすることができますが、現行のアクティブなフローには影響を与えません。

COM クラス: Flow

SyncType

SyncType() as Integer

または

SyncType() as CclFlowSyncTypes

使用する同期タイプを戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

Timeout

Timeout() as Integer

Flow オブジェクトにセットされた現行のタイムアウト値を戻します。

Wait

Wait()

サーバーからの応答を待ち、その間クライアント処理をブロックします。このメソッドは、遅延同期呼び出しが実行されたが、現在アプリケーションは応答を同期的に待機している場合に使用されます。

第10章 Map COM クラス

Map COM クラスは、CICS BMS マップから入手した記号情報を使用して、3270 画面データに妥当性検査およびアクセスを提供します。このインターフェースを使用するには、サーバー・プログラムの BMS マップで CICSBMSC ユーティリティを実行する必要があります。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as Cc10Map
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Cc1.Map")  
set var = New Cc10Map
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

ExCode

使用すべきでないメソッド

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

COM クラス: Map

ExCode() as Integer

または

ExCode() as CclEPIExceptionCodes

現在の条件コードを示す値を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

FieldByName

FieldByName(*name* as Integer) as Object

または

FieldByName(*name* as Integer) as CclOField

name

必須フィールドの記号値。この値は、CICSBMSC ユーティリティーによってソース BMS から生成された <mapname>.BAS ファイルに提供されます。指定した CclOField オブジェクトを戻します。

Validate

Validate (*screenRef* as Object, *mapname* as String) as Boolean

または

Validate (*screenRef* as CclOScreen, *mapname* as String) as Boolean

screenRef

CclOScreen オブジェクト

mapname

CICSBMSC ユーティリティーによりソース BMS ユーティリティーから生成された <mapname>.BAS ファイルに提供される文字列値。現行画面に照らしてマップを妥当性検査します。

このメソッドを使用して、特定の BMS マップが CICS サーバーから検索されているかどうかを調べることができます。戻される値は以下のいずれかです。

- TRUE** 指定された BMS マップが現行の画面の内容に一致していません。
- FALSE** 指定された BMS マップの内容が現行の画面の内容と一致していません。

TRUE が戻されると、**FieldByName** メソッドを使用して、フィールドに BMS 名でアクセスすることが可能になります。

第11章 Screen COM クラス

Screen COM クラスは、3270 仮想画面上のすべてのデータを保守し、このデータにアクセスします。これには、現行の 3270 画面上のフィールドを表す一連の CclOField オブジェクトが入っています。

単一の Screen オブジェクトは、端末が Ccl Terminal 接続またはインストール・メソッドのいずれかでインストールされるときに、Terminal オブジェクトによって作成されます。アプリケーションは、Ccl Terminal **Screen** メソッドを通じて、CclOScreen オブジェクトにアクセスすることができます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CclOScreen
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

メソッド

CursorCol

CursorCol() as Integer

現在カーソルのある列 (左端列 = 1) を戻します。

CursorRow

CursorRow() as Integer

現在カーソルのある行 (最高行 = 1) を戻します。

COM クラス: Screen

Depth

Depth() as Integer

画面上の列数を返します。

FieldByIndex

FieldByIndex(*index* as Integer) as Object

または

FieldByIndex(*index* as Integer) as CcIOField

index

必須のフィールドの索引番号。最初のフィールドの番号は 1 です。

FieldByPosition

FieldByPosition (*rowPos* as Integer, *colPos* as Integer) as Object

または

FieldByPosition (*rowPos* as Integer, *colPos* as Integer) as CcIOField

rowPos

フィールドの行番号 (最上行 = 1)。

colPos

フィールドの列番号 (最左列 = 1)。

FieldCount

FieldCount() as Integer

画面上のフィールド数を返します。

MapName

MapName() as String

端末リソース用に処理された SEND MAP コマンドの、MAP オプションで参照された最新のマップ名を指定している文字列を返します。端末リソース

が BMS でサポートされない場合、またはサーバーに送信されるいずれのマップにもレコードがない場合は、戻される値はブランクです。

MapSetName

MapSetName() as String

端末リソース用に処理された SEND MAP コマンドの、MAPSET オプションで参照された最新のマップ・セット名を指定している文字列を返します。最新の要求で MAPSET オプションが指定されていない場合は、BMS はマップ・セット名としてマップ名を使用しました。両方の事例で、使用されるマップ・セット名は端末接尾部によって接尾部が付けられます。端末リソースが BMS でサポートされない場合、またはサーバーに送信されるいずれのマップ・セットにもレコードがない場合は、戻される値はブランクです。

SetAID

SetAID(*key* as Integer)

または

SetAID(*key* as CclADIKeys)

key

105ページの『CclScreen AID キー・コード』にリストされた AID キー値。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

次の伝送でサーバーに渡される AID キー値を設定します。

SetCursor

SetCursor (*rowPos* as Integer, *colPos* as Integer)

rowPos

カーソルの必須の行番号 (最上行 = 1)。

colPos

カーソルの必須の列番号 (最左列 = 1)。

COM クラス: Screen

Width

Width() as Integer

画面上の列数を戻します。

第12章 SecAttr COM クラス

SecAttr COM クラスは、**CclOConnect** または **CclOTerminal** オブジェクトで、**verifySecurity** または **changePassword** メソッドを発行したときに、外部セキュリティ・マネージャーによって報告されるパスワードについての情報を提供します。

このオブジェクトは **CclOConnect** または **CclOTerminal** オブジェクトによって作成および所有され、またこのオブジェクトへのアクセスは **VerifyPassword** または **ChangePassword** メソッドを呼び出すと提供されます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CclOSecAttr
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

パブリック・メソッド

ExpiryTime

ExpiryTime() as Object

または

ExpiryTime() as CclOSecTime

パスワードの有効期限が切れる日付と時間を含む **CclOSecTime** オブジェクトを戻します。

InvalidCount

InvalidCount() as Integer

そのユーザー ID に無効なパスワードを入力した回数を戻します。

LastAccessTime

LastAccessTime() as Object

または

LastAccessTime() as CclIOSecTime

そのユーザー ID が最後にアクセスされた日付と時間を含む CclIOSecTime オブジェクトを戻します。

LastVerifiedTime

LastVerifiedTime() as Object

または

LastVerifiedTime() as CclIOSecTime

最新の検査が行なわれた日付と時間を含む CclIOSecTime オブジェクトを戻します。

第13章 SecTime COM クラス

SecTime COM クラスは、**Connect** または **Terminal** オブジェクトで、**verifySecurity** または **changePassword** メソッドを発行したときに、外部セキュリティ・マネージャーによって報告されるさまざまなエントリー用の **CcIOSecAttr** オブジェクトに日付と時間情報を提供します。

これらのオブジェクトは **CcIOSecAttr** オブジェクトによって作成および所有され、オブジェクトへのアクセスはこのオブジェクトで選択可能なさまざまなメソッドを通じて獲得できます。選択可能なコンストラクターとデストラクターはありません。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CcIOSecTime
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

パブリック・メソッド

Day

unsigned short Day() as Integer

1 から 31 の範囲で日付を戻します。

GetDate

GetDate() as Date

Visual Basic DATE タイプ形式で日付と時間を戻します。

Hours

unsigned short Hours() as Integer

0 から 23 の範囲で時間を戻します。

Hundredths

unsigned short Hundredths() as Integer

0 から 99 の範囲で百分の一秒単位の時間を戻します。

Minutes

unsigned short Minutes() as Integer

0 から 59 の範囲で分を戻します。

Month

unsigned short Month() as Integer

1 から 12 の範囲で月を戻します。

Seconds

unsigned short Seconds() as Integer

0 から 59 の範囲で秒単位の時間を戻します。

Year

unsigned short Year() as Integer

4 桁の年を戻します。

第14章 Session COM クラス

Session COM クラスは、1 つの EPI セッション内の CICS におけるデータの送受信の流れを制御します。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as Cc10Session
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.Session")  
set var = New Cc10Session
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

Diagnose

Diagnose() as String

セッションの現在の状態を説明するテキストを戻します。

SetSyncType

SetSyncType(syncType as Integer)

または

COM クラス: Session

SetSyncType(syncType as CclFlowSyncTypes)

syncType

この CclOSession オブジェクトに必要な同期タイプ。以下の値を使用することができます。

`cclSync`

`cclDSync`

この CclOSession オブジェクトに必要な同期タイプをセットします。cclSync を使用する場合、このフローを使用している **Start** および **Send** 呼び出しは、CICS から応答が戻されるまで呼び出しプログラムをブロックします。cclDSync を使用する場合、このフローを利用している **Start** および **Send** 呼び出しは、即時に呼び出しプログラムに戻されます。このプログラムは、後に CICS からの応答を受信する際には **Poll** メソッドを使用することができるようになります。

State

State() as Integer

または

State() as CclEPIStates

セッションの現在の状態を示す値を戻します。以下の値を使用することができます。

cclActive

接続されています。

cclServer

CICS サーバーでトランザクションが実行中。

cclClient

CICS サーバーはクライアントからの応答を待機中。

cclDiscon

切断されています。

cclError

エラーが発生しました。**ExCode** メソッドおよび **Diagnose** メソッドを呼び出して、詳細情報を参照してください。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

TransId

TransId() as String

現行トランザクションの 4 文字の名前を戻します。

第15章 Terminal COM クラス

Terminal COM クラスは、3270 端末の CICS サーバーとの接続を表しています。CICS 接続は、**Connect** メソッドが呼び出されると確立されます。これで、CICS サーバーの 3270 端末アプリケーション (BMS アプリケーションの場合が多い) との会話に、メソッドを使用することができます。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as Cc10Terminal
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.Terminal")  
set var = New Cc10Terminal
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

AlterSecurity

AlterSecurity(*newUserId* as String,*newPassword* as String)

newPassword

newUserId に指定する新規パスワード。

newUserId

新規のユーザー ID。

COM クラス: Terminal

まだユーザー ID とパスワードを指定して構成されたことがない端末リソース、すなわちサインオン不能で作成済みの端末に、ユーザー ID とパスワードを再定義できるようにします。端末をインストールする前にメソッドを呼び出すことができます。端末定義を変更するので、新規のユーザー ID とパスワードは `install` が呼び出されるとその端末で使用されます。

CCSId

CCSId() *as long*

選択済みコード・ページを表示している長形式を戻します。

ChangePassword

ChangePassword(*newPassword as String*) as Object

または

ChangePassword(*newPassword as String*) as CcIOSecAttr

newPassword

指定する新規パスワード。

Terminal オブジェクトに保持されているパスワードと、Terminal オブジェクトに保持されているユーザー ID 用に外部セキュリティー・マネージャーが記録したパスワードを、クライアント・アプリケーションが変更できるようにします。外部セキュリティー・マネージャーは Terminal オブジェクトによって定義されたサーバーに配置されているとみなされます。エラーが生じない場合は、CcIOSecAttr オブジェクトが戻されます。

Connect

**Connect(*servName as String*,
devType as String,
nworkName as String)**

servName

通信する必要があるサーバーの名前。NULL ストリングが提供されると、クライアント初期設定ファイルで定義されたデフォルトのサーバー・システムが使用されます。名前が 8 文字未満の場合は、必要に応じて空白で埋め込まれます。

devType

サーバーが端末リソース定義の生成に使用するモデル端末定義の名前。NULL スtringが提供されると、デフォルトのモデルが使用されます。名前が 16 文字未満の場合は、必要に応じて空白で埋め込まれます。

nworkName

インストールもしくは予約する端末リソースの名前。名前が 8 文字未満の場合は、必要に応じて空白で埋め込まれます。NULL Stringが提供されると、CICS サーバーから名前が割り振られます。

3270 通信を、指定の CICS サーバーに対して確立します。

Devtype**Devtype() as String**

Stringで端末装置タイプを戻します。

Diagnose**Diagnose() as String**

最新のサーバー呼び出しが戻すエラーの記述を保留する、文字Stringを戻します。

Disconnect**Disconnect()**

CICS から端末を切断します。実行中の未解決のトランザクションはページされません。

DisconnectWithPurge**DisconnectWithPurge()**

CICS から端末を切断し、実行中の未解決トランザクションのページが試行されます。

DiscReason

DiscReason() as CclEndTermReasons

このメソッドは、端末が切断された理由を示す列挙型を戻します。使用可能な値を、104ページの『CclTerminal EndTermReason』に示します。

ExCode

— 使用すべきでないメソッド —

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ExCode() as Integer

または

ExCode() as CclEPIExceptionCodes

サーバーから戻された最新の条件コードを示す値を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

ExCodeText

— 使用すべきでないメソッド —

新規アプリケーションではこのメソッドを使用しないでください。このメソッドは使用すべきでなく、旧バージョンとの互換性の目的でのみ提供されています。

ExCodeText() as String

サーバーから戻された最新の条件コードを説明するテキスト列を戻します。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

Install

Install(session as Object, timeout as Integer)

または

Install(session as CclIOSession, timeout as Integer)

session

この terminal オブジェクトで使用する session オブジェクト。

InstallTimeout

0 から 3600 の範囲の値で、端末リソースのインストールで使うことのできる時間の最大値を秒数で指定します。0 の値は限界をセットしないこと意味します。

このメソッドは非接続端末リソースをインストールします。端末がすでにインストールされている場合は、`cclInvalidState` エラーが生じます。

MakeSecurityDefault

MakeSecurityDefault()

Terminal オブジェクトの構造に指定したように、サーバーに渡される ECI および EPI 要求のデフォルトにこのオブジェクトの現行ユーザー ID とパスワードがなることを、クライアントに通知します。

NetName

NetName() as String

端末のネットワーク名を戻します。

Password

Password() as String

端末と関連するユーザー ID についての現行パスワードを含むテキスト列を戻します。パスワードなしの場合はストリングは空です。

Poll

Poll() as Boolean

COM クラス: Terminal

応答が遅延同期 **Start** または **Send** 要求から受信されたかどうかを調べます。以下の値を使用することができます。

True 未解決の応答はありません。
False 未解決の応答がまだあります。

CICS サーバー・トランザクションは、1 回の **Terminal.Start** 呼び出しまたは **Terminal.Send** 呼び出しに応じて、複数の応答を送ることができます。したがって、すべての応答を収集するには、複数の **Terminal.Poll** 呼び出しが必要になります。リターン・コードは、さらにポーリング要求を実行する必要があるかどうかを示します。

PollForReply

PollForReply() as Boolean

応答が遅延同期の **Start** 要求または **Send** 要求から受信されたかどうかを調べます。以下の値を使用することができます。

true 応答が受信されました。
false 応答は受信されませんでした。

CICS サーバー・トランザクションは、1 回の **Terminal.Start** 呼び出しまたは **Terminal.Send** 呼び出しに応じて複数の応答を送ることができます。したがって、すべての応答を収集するには、複数の **Terminal.PollForReply** 呼び出しが必要になります。**Terminal.State** メソッドを使用して、それ以上の応答が期待されているかどうかを調べます。期待されている場合には、戻される値は **cclServer** になります。

QueryATI

QueryATI() as Integer

または

QueryATI() as CclATIStates

「自動トランザクション開始」(ATI) が使用可能であるか使用不能であるかを示す値を返します。以下の値を使用することができます。

cclATIEnabled
cclATIDisabled

ReadTimeout

ReadTimeout() as Integer

端末の読み取りタイムアウト設定値を戻します。

ReceiveATI

ReceiveATI (*session* as Object)

または

ReceiveATI (*session* as CclIOSession)

session

CICS サーバー対話で使用される CclIOSession オブジェクトへのポインター。

CICS ATI トランザクションで 3270 データ・ストリームを待ち、受信します。提供される CclIOSession オブジェクトは同期だけです。

Screen

Screen() as Object

この端末と関連付けられている 3270 画面を操作する CclOScreen オブジェクトを戻します。

Send

Send(*session* as Object)

または

Send(*session* as CclIOSession)

session

使用されるセッションを制御する CclIOSession オブジェクト。

CclIOSession オブジェクトが使用されていない場合は、NULL に設定されます。

COM クラス: Terminal

CclOScreen オブジェクトの現在の内容から 3270 データ・ストリームを生成して、CICS サーバーに転送します。

ServerName

ServerName() as String

CclOTerminal オブジェクトが保持しており、クライアント初期設定ファイルがリストしている サーバー・システムの名前か、あるいは、(デフォルト・サーバーが使用され、呼び出しが一回も行われていない場合は) ブランクを返しません。

SetATI

SetATI(*stateVal* as Integer)

または

SetATI(*stateVal* as CclATISates)

stateVal

ATI が使用可能であるか使用不能であることを示す値。以下の値を使用することができます。

cclATIEnabled

cclATIDisabled

SetTermDefns

**SetTermDefns (*servName* as String,
devType as String,
nworkName as String
signonCapability as CclSignonTypes
userid as String
password as String
ReadTimeout as Integer
CCSid as Long)**

servName

通信する必要があるサーバーの名前。NULL スtringが指定されると、クライアント初期設定ファイルで定義されたデフォルトのサーバー・システムが使用されます。名前が 8 文字未満の場合は、必要に応じて空白で埋め込まれます。

devType

サーバーが端末リソース定義の生成に使用するモデル端末定義の名前。NULL スtringが提供されると、デフォルトのモデルが使用されます。名前が 16 文字未満の場合は、必要に応じて空白で埋め込まれます。

nworkName

インストールもしくは予約する端末リソースの名前。名前が 8 文字未満の場合は、必要に応じて空白で埋め込まれます。NULL Stringが指定されると、CICS サーバーから名前が割り振られます。

signonCapability

サインオン機能を次の 1 つに設定します。

cclSignonCapable

cclSignonIncapable

ReadTimeout

0 から 3600 の範囲の値で、そのクラスが `clientrepl` 状態にすむ時間と、アプリケーション・プログラムが応答メソッドを呼び出す時間の間の最大時間を秒数で指定します。

userid この端末リソースに関連させるユーザー ID の名前。

password

このユーザー ID と関連するパスワード。

CCSid 端末リソースと CICS トランザクション間で渡されるデータについて、クライアント・アプリケーションが使用するコード化図形文字セットを識別するための、コード化文字セット ID (CCSID) を指定する長形式。ゼロはデフォルトの使用を暗黙指定します。

端末リソースを作成しますが、サーバーへの接続は行いません。

SignonCapability

SignonCapability() as Integer

または

SignonCapability() as CclSignonTypes

インストールされた端末のタイプを戻します。以下の値を使用することができません。

- `cclSignonCapable`
- `cclSignonIncapable`

COM クラス: Terminal

Start

Start (*session* as Object,
tranCode as String,
startData as String)

または

Start (*session* as CclOSession,
tranCode as String,
startData as String)

session

使用されるセッションを制御する CclOSession オブジェクト。
CclOSession オブジェクトが使用されていない場合は、NULL に設定されます。

tranCode

開始されるトランザクションの名前。

startData

トランザクション・データを開始します。NULL 値は、開始されているトランザクションにデータが必要ないということを示します。
提供されたデータに基づいて 3270 データ・ストリームを生成し、指定のトランザクションを開始する CICS サーバーに転送します。

State

State() as Integer

または

State() as CclEPIStates

セッションの現在の状態を示す値を戻します。これらの値は、**Session** COM クラスの **state** メソッドから戻される値と同じです。

タイプ・ライブラリーでは定数が利用可能です。それらを表示するには Visual Basic Object Browser を使用してください。

TermId

TermId() as String

端末 ID を戻します。

TransId

TransId() as String

現行 CICS トランザクションの 4 文字の名前を戻します。現行のトランザクションから RETURN IMMEDIATE を実行すると、TransId は新規トランザクションの名前を提供せず、TransId にはまだ最初のトランザクションの名前が含まれていることに注意してください。

Userid

Userid() as String

端末の現行ユーザー ID を含むテキスト列を戻します。ユーザー ID なしの場合にはストリングは空です。

VerifyPassword

VerifyPassword() as Object

または

VerifyPassword() as CcIOSecAttr

Terminal オブジェクトに保持されているパスワードと、Terminal オブジェクトに保持されているユーザー ID 用に外部セキュリティー・マネージャーが記録したパスワードが一致しているかどうかを、クライアント・アプリケーションが検査できるようにします。外部セキュリティー・マネージャーは Terminal オブジェクトによって定義されたサーバーに配置されているとみなされます。エラーが生じない場合は、CcIOSecAttr オブジェクトが戻されます。

第16章 UOW COM クラス

“作業単位” (UOW) 内のサーバーのリカバリー可能リソースを更新するときは、この COM クラスを使用します。UOW 内の各更新は、CcIOUOW オブジェクトを参照して調べることができます。**Connect** COM クラス (42ページの『Link』) の **Link** メソッドを参照してください。

インターフェースの選択

Visual Basic では、次のタイプのインターフェースが選択可能です。

```
Dim var as Object  
Dim var as CcIOUOW
```

お勧めする方法は 2 番目の例です。

変数を DIM 宣言しないか、またはタイプの指定なしで DIM 宣言するか、または VBScript を使用している場合は、変数は **Object** のタイプであると想定されます。

オブジェクト作成

2 通りの方法でオブジェクトを作成できます。

```
set var = CreateObject("Ccl.UOW")  
set var = New CcIOUOW
```

Visual Basic で推奨する方法は **New** を使用する方法です。VBScript の場合は、**CreateObject** メソッドのみを使用することができます。

メソッド

BackOut

BackOut(*flow* as Object)

または

BackOut(*flow* as CcIOFlow)

COM クラス: UOW

flow

クライアント / サーバー呼び出しを制御するために使用する CclOFlow オブジェクト。

この UOW を終了し、サーバーのリカバリー可能リソースに加えられたすべての変更をバックアウトします。

Commit

Commit(*flow* as Object)

または

Commit(*flow* as CclOFlow)

flow

クライアント / サーバー呼び出しを制御するために使用する CclOFlow オブジェクト。

この UOW を終了し、サーバーのリカバリー可能リソースに加えられたすべての変更をコミットします。

ForceReset

ForceReset()

この UOW を非活動にし、リセットします。UOW はコミットされたか、バックアウトされました。

UowId

UowId() as long

UOW の ID を戻します。ゼロの戻りは、UOW が完了しているか、まだ開始していない、すなわち非活動であるかのどちらかを示します。

第3部 付録

付録A. COM グローバル定数

定数は CICS クライアント COM ライブラリー用のタイプ・ライブラリーに提供されています。ライブラリーは CCLIECI.DLL と CCLIEPI.DLL に提供されています。

Visual Basic を使用している場合は、Visual Basic オブジェクト・ビューアーまたは他のタイプのライブラリー・ビューアーを使用してタイプ・ライブラリーの定義を調べることができます。

VBScript を使用している場合は、タイプ・ライブラリーに定義されている列挙型をアクセスすることはできないので、ここに定義されている数値を使用する必要があります。

例外コード定数は 109ページの『付録D. エラー・コード参照』にリストされています。

付録B. CICS クライアント ECI 定数

同期タイプ

表 3. 同期タイプ

VB 列挙型	値	説明
cclSync	0	同期呼び出しタイプ
cclDsync	1	遅延同期呼び出しタイプ

フロー状況タイプ

表 4. フロー状況タイプ

VB 列挙型	値	説明
cclInactive	0	フローは非アクティブです。
cclLink	1	フローはリンクを現在呼び出中です。
cclBackout	2	フローは UOW を現在バックアウト中です。
cclCommit	3	フローは現在 UOW をコミット中です。
cclStatus	4	フローは状況を要求中です。
cclChanged	5	フローは状況変更を要求中です。
cclCancel	6	フローは状況取り消しを要求中です。

接続状況コード

表 5. 接続状況コード

VB 列挙型	値	説明
cclUnknown	0	CICS サーバー状況は未確認です。
cclAvailable	1	CICS サーバー状況は選択可能です。
cclUnavailable	2	CICS サーバー状況は選択不可能です。

付録C. CICS クライアント EPI に特有の定数

同期タイプ

表 6. 同期タイプ

VB 列挙型	値	説明
cclSync	0	同期呼び出しタイプ
cclDsync	1	遅延同期呼び出しタイプ

CcIEPI 状態

表 7. CcIEPI 状態

VB 列挙型	値	説明
cclEPIActive	0	EPI の初期化が完了。
cclDiscon	1	EPI は終了しました。
cclEPIError	2	EPI は初期化に失敗しました。詳しくは例外をハンドルしてください。

CclSession 状態

表 8. CclSession 状態

VB 列挙型	値	説明
cclSessionIdle	0	アイドル、クライアントはトランザクションを開始する必要があります。
cclSessionServer	1	サーバーを待ちます。
ccISessionClient	2	クライアントが応答するのを待ちます。
ccISessionDiscon	3	切断されています。
ccISessionError	4	セッション・エラー。例外をハンドルすると詳しい情報が得られます。

CclTerminal 状態

表 9. CclTerminal 状態

VB 列挙型	値	説明
cclInit	0	端末は定義されているが未インストールです。
cclActive	1	端末は接続されています (使用されていません)。
cclIdle	2	アイドル状態。クライアントはトランザクションを開始する必要があります。
cclServer	3	サーバーを待ちます。
cclClient	4	クライアントが応答するのを待ちます。
cclDiscon	5	切断されています。
cclError	6	端末エラー。例外をハンドルすると詳しい情報が得られます。

CclTerminal ATI 状態

表 10. CclTerminal ATI 状態

VB 列挙型	値	説明
cclATIEnabled	0	ATI は許可されています。
cclATIDisabled	1	ATI は許可されていません。

CclTerminal EndTermReason

表 11. CclTerminal ATI 状態

VB 列挙型	値	説明
cclSignoff	0	切断が要求されたか、またはユーザーが端末をサインオフしました。
cclShutdown	1	CICS サーバーがシャットダウンされました。
cclOutOfService	2	端末は使用停止に切り替えられました。
cclUnknown	3	認知できない状態が生じました。
cclFailed	4	端末の切断に失敗しました。
cclNotDiscon	5	端末は切断されていません。

CclTerminal サインオン・タイプ

表 12. CclTerminal サインオン・タイプ

VB 列挙型	値	説明
cclSignonCapable	0	端末はサインオン・トランザクションをサポートします。
cclSignonIncapable	1	端末はサインオン・トランザクションをサポートしません。
cclSignonUnknown	2	端末サインオン機能が認知できません。

CclScreen AID キー・コード

表 13. CclScreen AID キー・コード

VB 列挙型	値	説明
cclEnter	0	Enter (入力) キー
cclClear	1	Clear (消去) キー
cclPA1	2	プログラム・アテンション・キー 1
cclPA2	3	プログラム・アテンション・キー 2
cclPA3	4	プログラム・アテンション・キー 3
cclPF1	5	プログラム機能キー 1
cclPF2	6	プログラム機能キー 2
cclPF3	7	プログラム機能キー 3
cclPF4	8	プログラム機能キー 4
cclPF5	9	プログラム機能キー 5
cclPF6	10	プログラム機能キー 6
cclPF7	11	プログラム機能キー 7
cclPF8	12	プログラム機能キー 8
cclPF9	13	プログラム機能キー 9
cclPF10	14	プログラム機能キー 10
cclPF11	15	プログラム機能キー 11
cclPF12	16	プログラム機能キー 12
cclPF13	17	プログラム機能キー 13
cclPF14	18	プログラム機能キー 14
cclPF15	19	プログラム機能キー 15
cclPF16	20	プログラム機能キー 16

表 13. CclScreen AID キー・コード (続き)

VB 列挙型	値	説明
cclPF17	21	プログラム機能キー 17
cclPF18	22	プログラム機能キー 18
cclPF19	23	プログラム機能キー 19
cclPF20	24	プログラム機能キー 20
cclPF21	25	プログラム機能キー 21
cclPF22	26	プログラム機能キー 22
cclPF23	27	プログラム機能キー 23
cclPF24	28	プログラム機能キー 24

CclField 保護状態属性

表 14. CclField 保護状態属性

VB 列挙型	値	説明
cclProtect	0	保護フィールド (変更不可能)
cclUnprotect	1	非保護 (入力) フィールド

CclField 数値属性

表 15. CclField 数値属性

VB 列挙型	値	説明
cclAlphanumeric	0	英数字入力フィールド
cclNumeric	1	数字入力フィールド

CclField 輝度属性

表 16. CclField 輝度属性

VB 列挙型	値	説明
cclNormal	0	標準表示
cclIntense	1	強調表示
cclDark	2	非表示フィールド

CclField 変更属性

表 17. CclField 変更属性

VB 列挙型	値	説明
cclUnmodified	0	フィールドに修正は加えられていません。
cclModified	1	フィールドに修正が加えられました。

CclField 強調表示属性

表 18. CclField 強調表示属性

VB 列挙型	値	説明
cclHltDefault	0	デフォルト・フィールド・テキストの強調表示
cclHltNormal	1	3270 基本属性の指定によるフィールド・テキストの強調表示
cclHltBlink	2	テキストの明滅
cclHltReverse	3	反転表示テキスト
cclHltUnderscore	4	下線付きテキスト
cclHltIntense	5	高輝度表示テキスト

CclField 透過性属性

表 19. CclField 透過性属性

VB 列挙型	値	説明
cclTrnDefault	0	デフォルト (不透明) フィールド・バックグラウンド
cclTrnOr	1	透過フィールド・バックグラウンド (OR)
cclTrnXor	2	透過フィールド・バックグラウンド (XOR)
cclTrnOpaque	3	不透明フィールド・バックグラウンド

CclField カラー属性

表 20. CclField カラー属性

VB 列挙型	値	説明
cclDefaultColor	0	
cclBlue	1	
cclRed	2	
cclPink	3	
cclGreen	4	
cclCyan	5	
cclYellow	6	
cclNeutral	7	
cclBlack	8	
cclDarkBlue	9	
cclOrange	10	
cclPurple	11	
cclPaleGreen	12	
cclPaleCyan	13	
cclGray	14	
cclWhite	15	

付録D. エラー・コード参照

一覧	値	説明	ECI	EPI
cclNoError	0	エラーは生じなかった。	O	O
cclBufferOverflow	1	拡張可能ではない CclBuf オブジェクトを拡張しようとした。	O	
cclMultipleInstance	2	複数の ECI オブジェクトを作成しようとした。	O	
cclActiveFlow	3	現行の流れはまだアクティブであり、この流れが非アクティブになるまでこの流れは使用できない。	O	
cclActiveUOW	4	現行の UOW はまだアクティブであり、バックアウトまたはコミットが必要である。	O	
cclSyncType	5	メソッド呼び出しの同期タイプが誤り。	O	O
cclDataLength	9	CommArea > 32768 バイト、またはインバウンド 3270 データ・ストリームが端末バッファ・サイズに対して大き過ぎる。	O	O
cclNoCICS	10	クライアントが選択不可能である、またはサーバー・インプリメンテーションが選択不可能である、または作業論理単位を開始しようとしたが、指定した CICS サーバーが選択不可能である。リソースは更新されていない。	O	O

一覧	値	説明	ECI	EPI
cclCICS Died	11	作業論理単位が開始または継続されるはずであったが、その CICS サーバーはもはや選択不可能である。これがアクティブ UOW を使用したリンク呼び出しであれば、変更はバックアウトされる。これが UOW コミットであったか、または変更がコミットされたかバックアウトされたかをアプリケーションで判別できない場合は、今後の手作業によるリカバリーのためにこの条件をログに記録しなければならない。	O	
cclNoReply	12	未解決の応答はない。	O	
cclTransaction	13	ECI プログラムの異常終了。	O	
cclSystemError	14	認識できない内部エラーが生じた。	O	O
cclResource	15	サーバー・インプリメンテーションまたはクライアントが、要求を完了するのに十分なリソースを持っていなかった。すなわち不十分な SNA セッション。	O	O
cclMaxUOWs	16	作業論理単位を作成しようとしたが、アプリケーションはすでに構成がサポートする数と同じだけの未解決の作業論理単位を持っている。	O	
cclUnknownServer	17	要求されたサーバーは探し出せない。	O	O
cclSecurity	18	ユーザーは、サーバーが予期している有効な組み合わせのユーザー ID とパスワードを提供しなかった。	O	O
cclMaxServers	19	ユーザーの構成で許可されている限度を超える数のサーバーに要求を開始しようとした。ユーザーのクライアントまたはサーバー用の資料で、使用できるサーバー数を制御する方法を調べてください。	O	O

一覧	値	説明	ECI	EPI
cclMaxRequests	20	要求を満足する十分な通信リソースがない。ユーザーのクライアントまたはサーバー用の資料で、通信リソースを制御する方法を調べてください。	O	O
cclRolledBack	21	作業論理単位をコミットしようとしたが、サーバーはその変更をコミットできず、その代わりにバックアウトした。	O	
cclParameter	22	指定したパラメーターの誤り	O	O
cclInvalidState	23	メソッドを呼び出すのにオブジェクトは正しい状態ではない。すなわち端末オブジェクトはまだサーバー状態で、データ送信の試みがなされた。	O	O
ccltransId	24	疑似変換トランザクションにヌルのトランザクション ID が提供されたかまたは戻された。		O
cclInitEPI	25	EPI オブジェクトがない、または正しい初期設定に失敗した。		O
cclConnect	26	端末の追加時に予期しないエラーが生じた。		O
cclDataStream	27	サポートされないデータ・ストリーム		O
cclInvalidMap	28	マップ定義と画面が一致しない。		O
cclClass	29	不明な内部クラス・エラーが生じた。	O	O
cclStartTranFailure	30	トランザクションの開始が失敗。		O
cclTimeout	31	サーバーからの応答が返る前にタイムアウトが発生した。	O	O
cclNoPassword	32	オブジェクトのパスワードがヌルである。	O	O
cclNoUserid	33	オブジェクトのユーザー ID がヌルである。	O	O
cclNullNewPassword	34	提供されたパスワードがヌルである。	O	O

一覧	値	説明	ECI	EPI
cclPemNotSupported	35	CICS サーバーはパスワード満了管理機能をサポートしない。このメソッドを使用することはできない。	0	0
cclPemNotActive	36	パスワード満了管理機能がアクティブではない。	0	0
cclPasswordExpired	37	パスワードが期限切れになった。情報が戻らない。	0	0
cclPasswordInvalid	38	パスワードが無効である。	0	0
cclPasswordRejected	39	パスワードが定義された標準に準拠していないので、パスワード変更が失敗した。	0	0
cclUseridInvalid	40	ユーザー ID が不明。	0	0
cclInvalidTermid	41	無効な端末 ID		0
cclInvalidModelId	42	無効なモデル / タイプ		0
cclnot3270	43	3270 デバイスではない		0
cclinvalidCCSid	44	無効な CCSid		0
cclServerBusy	45	CICS サーバーが使用中		0
cclSignonNotPoss	46	端末をサインオン可能としてインストールすることをサーバーが許可しない。		0

特記事項

本書はアメリカ合衆国で提供されている製品およびサービス用に作成されたものであり、本書に記載の製品、サービス、またはフィーチャー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、およびフィーチャーについては、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プログラムまたは製品を使用することができます。ただし、IBM 以外の製品、プログラムまたはサービスの操作性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。実施権の許諾については、下記の宛先に書面にてご照会ください。

〒106-0032 東京都港区六本木 3 丁目 2-31

AP 事業所

IBM World Trade Asia Corporation

Intellectual Property Law & 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ます。

本書は定期的に見直され、必要な変更（たとえば、技術的に不適確な表現や誤植など）は、本書の次版に組み込まれます。IBM は、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。

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 United Kingdom Laboratories, MP151,
Hursley Park, Winchester, Hampshire,
England, SO21 2JN

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、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。

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。また、IBM 以外の製品に関するパフォーマンスの正確性、互換性、またはその他の要求は確認できません。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。

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、IBM 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。お客様は、IBM 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。

商標

以下は、IBM Corporation の商標です。

CICS

IBM

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。

UNIX は、The Open Group がライセンスしている米国およびその他の国における登録商標です。

参照文献

CICS Transaction Gateway および CICS ユニバーサル・クライアントのライ ブラリー

この章では、すべての CICS Transaction Gateway、CICS ユニバーサル・クライアント、および関連資料をリストし、それらが使用できるさまざまな形式を説明します。

この章で扱う見出し項目は、次のとおりです。

- 『CICS Transaction Gateway 資料』
- 118ページの『CICS ユニバーサル・クライアント資料』
- 118ページの『CICS ファミリー資料』
- 118ページの『資料のファイル名』
- 119ページの『サンプル構成資料』
- 119ページの『その他の資料』
- 120ページの『オンライン資料の表示』

CICS Transaction Gateway 資料

- *CICS Transaction Gateway: Windows[®] Gateway 管理*、SC88-8984
この資料では、CICS[®] Transaction Gateway (Windows[®] 版) の管理について説明します。
- *CICS Transaction Gateway: AIX[®] Gateway 管理*、SC88-8985
この資料では、CICS[®] Transaction Gateway (AIX[®] 版) の管理について説明します。
- *CICS Transaction Gateway: Solaris Gateway 管理*、SC88-8986

この資料では、CICS[®] Transaction Gateway (Solaris 版) の管理について説明します。

- *CICS Transaction Gateway: Linux Gateway 管理*、SC88-8989
この資料では、CICS[®] Transaction Gateway (Linux 版) の管理について説明します。
- *CICS Transaction Gateway: HP-UX Gateway 管理*、SC88-8988
この資料では、CICS[®] Transaction Gateway (HP-UX 版) の管理について説明します。
- *CICS Transaction Gateway: OS/390[®] Gateway 管理*、SC88-8987

この資料では、CICS[®] Transaction Gateway (OS/390[®] 版) の管理について説明します。

- *CICS Transaction Gateway: ゲートウェイ・メッセージ*

このオンライン・ブックでは、CICS Transaction Gateway が生成するエラー・メッセージをリストおよび説明します。この資料をオーダーすることはできません。

- *CICS Transaction Gateway: ゲートウェイ・プログラミング*、SC88-8990

この資料では、CICS Transaction Gateway を使用した Java[™] プログラミングの概要を提供します。

プログラミング参照情報を含む追加の HTML ページもあります。

CICS ユニバーサル・クライアント資料

- *CICS Transaction Gateway: Windows®* クライアント管理、SC88-8991

この資料では、CICS ユニバーサル・クライアント (Windows 版) の管理について説明します。
- *CICS Transaction Gateway: AIX®* クライアント管理、SC88-8992

この資料では、CICS ユニバーサル・クライアント (AIX 版) の管理について説明します。
- *CICS Transaction Gateway: Solaris* クライアント管理、SC88-8993

この資料では、CICS ユニバーサル・クライアント (Solaris 版) の管理について説明します。
- *CICS Transaction Gateway: Linux* クライアント管理、SC88-8995

この資料では、CICS ユニバーサル・クライアント (Linux 版) の管理について説明します。
- *CICS Transaction Gateway: HP-UX* クライアント管理、SC88-8994

この資料では、CICS ユニバーサル・クライアント (HP-UX 版) の管理について説明します。
- *CICS Transaction Gateway: クライアント・メッセージ*

このオンライン・ブックでは、CICS ユニバーサル・クライアントが生成するエラー・メッセージとトレース・メッセージをリストおよび説明します。

この資料をオーダーすることはできません。
- *CICS Transaction Gateway: C++* プログラミング、SC88-8996

この資料では、C++ 言語を使用して ECI と EPI 用のオブジェクト指向プログラムを作成する方法を説明します。

- *CICS Transaction Gateway COM* オートメーション・プログラミング、SC88-8997

この資料では、コンポーネント・オブジェクト・モデル (COM) 規格に従って、ECI と EPI 用のオブジェクト指向プログラムを作成する方法を説明します。

CICS ファミリー資料

- *CICS®* ファミリー: クライアント / サーバー プログラミング、SC88-8998

この資料では、CICS クライアント / サーバー・プログラミング – すなわち外部呼び出しインターフェース (ECI)、外部表示インターフェース (EPI)、および外部セキュリティ・インターフェース (ESI) と関連したプログラミング・インターフェースを説明します。この資料は CICS サーバー・システムと通信する目的で、クライアント・アプリケーションを開発するアプリケーション設計者とプログラマーを対象としています。

資料のファイル名

表21 は、CICS Transaction Gateway および CICS ユニバーサル・クライアントの資料のソフトコピー・ファイル名を示しています。

表 21. *CICS Transaction Gateway* と *CICS* ユニバーサル・クライアントの資料名およびファイル名

資料タイトル	ファイル名
CICS Transaction Gateway: クライアント・メッセージ	CCLIAB
CICS Transaction Gateway: AIX® クライアント管理	CCLIAD

表 21. CICS Transaction Gateway と CICS ユニバーサル・クライアントの資料名およびファイル名 (続き)

資料タイトル	ファイル名
CICS Transaction Gateway: Windows® クライアント管理	CCLIAF
CICS Transaction Gateway: Solaris クライアント管理	CCLIAG
CICS Transaction Gateway: Linux クライアント管理	CCLIAR
CICS Transaction Gateway: HP-UX クライアント管理	CCLIAT
CICS Transaction Gateway: OS/390® Gateway 管理	CCLIAI
CICS Transaction Gateway: ゲートウェイ・メッセージ	CCLIAJ
CICS Transaction Gateway: ゲートウェイ・プログラミング	CCLIAK
CICS Transaction Gateway: Windows® Gateway 管理	CCLIAL
CICS Transaction Gateway: AIX® Gateway 管理	CCLIAN
CICS Transaction Gateway: Solaris Gateway 管理	CCLIAO
CICS Transaction Gateway: Linux Gateway 管理	CCLIAS
CICS Transaction Gateway: HP-UX Gateway 管理	CCLIAU
CICS Transaction Gateway: C++ プログラミング	CCLIAP
CICS Transaction Gateway COM オートメーション・プログラミング	CCLIAQ
CICS® ファミリー: クライアント / サーバー プログラミング	DFHZAD
注: このテーブルのファイル名には二桁の接尾部を含んでいません。	

サンプル構成資料

多くのサンプル構成資料が PDF 形式で使用可能です。

これらの資料では、ユーザーを支援するためのステップバイステップのガイドが提供されています。たとえば、これらの資料は、さまざまなプロトコルを使用して、CICS サーバーと通信するために CICS ユニバーサル・クライアントを構成する際に役立ちます。これらの資料は、CICS Transaction Gateway および CICS ユニバーサル・クライアントに提供されている情報をより詳細に説明します。

サンプル構成文書が選択可能になる場合は、Web サイトからそれらをダウンロードすることができます。以下をアクセスし、

www.ibm.com/software/ts/cics/

Library リンクに従ってください。

その他の資料

次の国際技術サポート機関 (ITSO)

Redbook™ の出版物には、クライアント / サーバー構成の多くの例が記載されています。

- *Revealed! CICS Transaction Gateway with more CICS Clients Unmasked, SG24-5277*
この資料は以下の資料を置き換えます。
- *CICS Clients Unmasked, GG24-2534*

多くのソースから ITSO レッドブックを入手することができます。最新の情報については、以下を参照してください。

www.ibm.com/redbooks/

以下で CICS プロダクトの情報を見ることができます。

www.ibm.com/software/ts/cics/

オンライン資料の表示

オンライン資料の表示

CICS Transaction Gateway と CICS ユニバーサル・クライアントに提供されているすべての資料に弊社のオンライン・ライブラリーでアクセスすることができます (英語のみ)。オンライン・ライブラリーを使用するためには適切な Web ブラウザーと Adobe Acrobat Reader が必要です (また、これらを構成する必要もあります)。

オンライン・ライブラリーを使用すると、以下にリンクできます。

- PDF 形式の CICS Transaction Gateway および CICS ユニバーサル・クライアント資料。
- ハイパーテキスト・マークアップ言語 (HTML) ファイル形式のプログラミング解説書資料 (CICS Transaction Gateway 用) のみ提供)。
- README ファイル。
- PDF 形式のサンプル構成資料。
- CICS Web サイト。

Acrobat Reader の使用に関する情報も提供されています。

資料の更新バージョンが適時提供されます。以下の弊社の Web サイトをチェックして、

www.ibm.com/software/ts/cics/

Library リンクに進みます。

注: 資料には他の言語に翻訳されているものもあります。これらの資料は、オンライン・ライブラリーに含まれていません。ただし、上記 Web サイトでスタンドアロンの PDF ファイルとして使用可能です。

PDF 資料の表示

PDF 情報は以下の強力なファンクションを提供します。

- 情報のナビゲート。 PDF 資料内、および他の PDF 資料および Web ページにはハイパーテキスト・リンクがあります。
- 特定情報の検索。
- PostScript プリンターでの PDF 資料の全印刷または部分印刷。

Adobe Web サイトで Acrobat Reader についてさらに知ることができます。

www.adobe.com/acrobat/

用語集

AID. アテンション ID (Attention Identifier)

ATI. 自動トランザクション開始 (Automatic Transaction Initiation)

BMS. 基本マッピング・サポート

COMMAREA. CICS 連絡域 (CICS Communications Area)

COM. コンポーネント・オブジェクト・モデル (Component Object Model)

ECI. 外部呼び出しインターフェース (External Call Interface)

EPI. 外部表示インターフェース (External Presentation Interface)

ESI. 外部セキュリティー・インターフェース (External Security Interface)

ISC. システム間連絡 (Inter-Systems Communication)

OLE. オブジェクト・リンキング & エンベッディング (Object Linking and Embedding)

UOW. 作業単位 (Unit of Work)

索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。

[ア行]

アパートメント

作成、COM オブジェクトの
オートメーション・コンパチブル
オブジェクト指向サポート、CICS
クライアントでの

CICS オブジェクト指向プログラ
ミングの紹介 3

オンライン資料、HTML 120

オンライン文書の表示 120

オンライン・ブック、PDF 120

[カ行]

開始メソッド

ATI サポートの 28

画面

BMS マップ・データの EPI

COM クラスでの使用 26

CICS 3270 セッションの実行 22
クライアント初期設定ファイル

Connect 84

Details 41

ServerCount の 49, 53

ServerDesc 54

ServerName 54, 90

ServerName の 43

構文表記法 vii

コンポーネント・オブジェクト・モ
デル

作業環境の確立 5

[サ行]

サーバー 5

作業単位内での ECI リンク呼び出し

CICS への ECI リンク呼び出し

COM インターフェースの使用

18

サポートされる環境

作業環境の確立 5

自動トランザクション開始

(Automatic Transaction

Initiation) 27

商標 115

処理、例外 12

資料 117

印刷 120

オンライン 120

CICS Transaction Gateway および

CICS ユニバーサル・クライア

ントのライブラリー 117

PDF 120

資料、CICS Transaction Gateway お

よび CICS ユニバーサル・クライ

アントのライブラリー 117

新規作成

CICS への ECI リンク呼び出し

15

セッション

CICS 3270 セッションの実行 22

Send の 89

Session COM クラスの 79

SetSyncType 80

Start 92

State の 92

接続、EPI を使用した CICS 3270

アプリケーションへの

BMS マップ・データの EPI

COM クラスでの使用 25

CICS 3270 セッションの実行 22

CICS サーバー情報 24

COM クラスの使用 21

EPI 呼び出し同期タイプ 23

送信メソッド

ATI サポートで 28

ソフトコピー・ブック、PDF 120

[タ行]

端末

CICS 3270 セッションの実行 22

EPI COM クラスの 51

Screen COM クラスの 71

ServerName 90

端末 COM クラス

メソッド

AlterSecurity 83

CCSID 84

ChangePassword 84

Connect 84

Devtype 85

Diagnose 85

Disconnect 85

DisconnectWithPurge 85

DiscReason 86

ExCode 86

ExCodeText 86

Install 87

MakeSecurityDefault 87

NetName 87

Password 87

Poll 87

PollForReply 88

QueryATI 88

ReadTimeout 89

ReceiveATI 89

Screen 89

Send 89

ServerName 90

SetATI 90

SetTermDefns 90

SignonCapability 91

Start 92

State 92

TermId 92

TransId 93

Userid 93

端末 COM クラス (続き)

- VerifyPassword 93
- 端末クラス 27
- 遅延同期
 - ATI サポートで 28
- 鉄道ダイアグラム vii
- 同期
 - ATI サポートで 28
- 登録 6
- トランザクション開始、自動 27

[ハ行]

- ハードコピー・ブック 120
- ハイパーテキスト・マークアップ言語 (HTML) 120
- パブリック・メソッド
 - SecAttr COM クラスの 75
 - SecTime COM クラスの 77
- ビジネス・オブジェクト 16
- フィールド
 - Field COM クラスの 57
 - Screen COM クラスの 71
- フロー
 - CICS サーバー情報および接続状況 17
 - SetSyncType 65
- プログラミングの概要
 - COM インターフェースの使用 11
- 文書 117
 - HTML 120
 - PDF 120

[マ行]

- マップ
 - BMS マップ・データの EPI COM クラスでの使用 25
- メソッド
 - AbendCode 63
 - AlterSecurity 40, 83
 - AppendString 36
 - AppendText 57
 - BackgroundColor 58
 - BackOut 95
 - BaseAttribute 58

メソッド (続き)

- Buffer COM クラスの 36
- CallType 64
- CallTypeText 64
- Cancel 40
- CCSID 84
- Changed 40
- ChangePassword 41, 84
- Column 58
- Commit 96
- Connect 84
- Connect COM クラスの 40
- CursorCol 71
- CursorRow 71
- Data 36
- DataTag 58
- Depth 72
- Details 41
- Devtype 85
- Diagnose 51, 64, 79, 85
- Disconnect 85
- DisconnectWithPurge 85
- DiscReason 86
- ECI COM クラスの 47
- EPI COM クラスの 51
- ErrorFormat 47, 52
- ErrorOffset 48, 52
- ErrorWindow 48, 52
- ExCode 48, 53, 68, 86
- ExCodeText 49, 53, 86
- ExtractString 36
- Field COM クラスの 57
- FieldByIndex 72
- FieldByName 68
- FieldByPosition 72
- FieldCount 72
- Flow COM クラスの 63
- Flowid 64
- ForceReset 64, 96
- ForegroundColor 59
- Highlight 59
- InputProt 59
- InputType 60
- InsertString 37
- Install 87
- Intensity 60

メソッド (続き)

- Length 37, 60
- Link 42
- MakeSecurityDefault 43, 87
- Map COM クラスの 67
- MapName 72
- MapSetName 73
- NetName 87
- Overlay 37
- Password 43, 87
- Poll 64, 87
- PollForReply 88
- Position 61
- QueryATI 88
- ReadTimeout 89
- ReceiveATI 89
- ResetDataTag 61
- Row 61
- Screen 89
- Screen COM クラスの 71
- Send 89
- ServerCount 49, 53
- ServerDesc 49, 54
- ServerName 43, 50, 54, 90
- ServerStatus 43
- ServerStatusText 44
- Session COM クラスの 79
- SetAID 73
- SetATI 90
- SetBaseAttribute 61
- SetCursor 73
- SetData 37
- SetErrorFormat 50, 54
- SetExtAttribute 61
- SetLength 38
- SetString 38
- SetSyncType 65, 79
- SetTermDefns 90
- SetText 62
- SetTimeout 65
- SignonCapability 91
- Start 92
- State 54, 80, 92
- Status 44
- String 38
- SyncType 66

メソッド (続き)

TermId 92
Terminal COM クラスの 83
Terminate 55
Text 62
TextLength 62
Timeout 66
TranDetails 44
TransId 81, 93
Transparency 62
UnpaddedPassword 45
UnpaddedServerName 45
UnpaddedUserId 45
UOW COM クラスの 95
UowId 96
UserId 45
Userid 93
Validate 68
VerifyPassword 45, 93
Wait 66
Width 74

[ラ行]

例外処理 12

[数字]

3270 データ・ストリーム 21

A

AbendCode
メソッド
Flow COM クラスの 63
AlterSecurity
メソッド
Connect COM クラスの 40
Terminal COM クラスの 83
AppendString
メソッド
Buffer COM クラスの 36
AppendText
メソッド
Field COM クラスの 57
array (パラメーター)
SetData 37

ATI 27
attachTran (パラメーター)
TranDetails 44
Attribute (パラメーター)
SetBaseAttribute の 61
SetExtAttribute の 61

B

BackgroundColor
メソッド
Field COM クラスの 58
BackOut
メソッド
UOW COM クラスの 95
Backout
作業単位内での ECI リンク呼び出し 19
Poll 65
BaseAttribute
メソッド
Field COM クラスの 58
BMS マップ・データの EPI COM クラスでの使用
EPI を使用した CICS 3270 アプリケーションへの接続
COM クラスの使用 25
Buffer
AppendString 36
Buffer COM クラスの 35
Link 42
Poll 65
String 38
Buffer COM クラス
メソッド
AppendString 36
Data 36
ExtractString 36
InsertString 37
Length 37
Overlay 37
SetData 37
SetLength 38
SetString 38
String 38
Buffer COM クラスの 36

C

CallType
メソッド
Flow COM クラスの 64
CallTypeText
メソッド
Flow COM クラスの 64
Cancel
メソッド
Connect COM クラスの 40
Poll 65
Ccal Screen.fieldbyPosition メソッド
Field COM クラスの 57
Ccl フィールド
CICS 3270 セッションの実行 22
Ccl マップ
BMS マップ・データの EPI COM クラスでの使用 26
cclActive
State の 55, 80
cclAlphanumeric
InputType の 60
cclATIDisabled
QueryATI の 88
SetATI の 90
cclATIEnabled
QueryATI の 88
SetATI の 90
cclAvailable
ServerStatus 43
cclClient
State の 80
cclDark
Intensity 60
cclDiscon
State の 55, 80
cclDSync
ECI 呼び出し同期タイプ 17
EPI 呼び出し同期タイプ 23, 24
SetSyncType 65, 80
cclError
State の 55, 80
cclIntense
Intensity 60

cclModified
 DataTag 58

cclNoError
 ExCode の 53

cclNormal
 Intensity 60

cclNumeric
 InputType の 60

CCLOECL.EXE
 COM クラスの使用 9

CCLOEPL.EXE
 COM クラスの使用 9

CclOSecTime
 Ccl SecAttr インターフェースの
 76

cclProtect
 InputProt の 59

cclServer
 State の 80

cclSync
 ECI 呼び出し同期タイプ 16
 EPI 呼び出し同期タイプ 23
 SetSyncType 65, 80

cclSystemError
 ExCode の 53

cclUnavailable
 ServerStatus 43

cclUnknown
 ServerStatus 43

cclUnknownServer
 ExCode の 53

cclUnmodified
 DataTag 58
 ResetDataTag 61

cclUnprotect
 InputProt の 59

Ccl.Field
 FieldByName 68

Ccl.Screen
 Send の 90

CCSIId
 メソッド
 Terminal COM クラスの 84

CCSid (パラメーター)
 SetTermDefns の 90

Changed
 メソッド
 Connect COM クラスの 40
 Cancel 40
 Changed 40
 Details 42
 Poll 65

changed
 ServerStatus 43
 ServerStatusText 44

ChangePassword
 メソッド
 Connect COM クラスの 41
 Terminal COM クラスの 84

CICS 3270 セッションの実行
 EPI を使用した CICS 3270 アプリケーションへの接続
 COM クラスの使用 22

CICS サーバー情報
 EPI を使用した CICS 3270 アプリケーションへの接続
 COM クラスの使用 24

CICS サーバー情報および接続状況
 CICS への ECI リンク呼び出し
 COM クラスの使用 17

CICS への ECI リンク呼び出し
 作業単位内での ECI リンク呼び出し 18

CICS サーバー情報および接続状況 17

COM クラスの使用 14

ECI 呼び出し同期タイプ 16

colPos (パラメーター)
 FieldByPosition 72
 SetCursor 73

Column
 メソッド
 Field COM クラスの 58

COM クラスの使用
 接続、EPI を使用した CICS 3270 アプリケーションへの
 BMS マップ・データの EPI
 COM クラスでの使用 25

CICS 3270 セッションの実行
 22

CICS サーバー情報 24

COM クラスの使用 (続き)
 接続、EPI を使用した CICS 3270 アプリケーションへの (続き)
 EPI 呼び出し同期タイプ 23

CICS への ECI リンク呼び出し
 作業単位内での ECI リンク呼び出し 18

CICS サーバー情報および接続状況 17

ECI 呼び出し同期タイプ 16

commArea (パラメーター)
 AbendCode の 63
 Link 42
 Poll 64, 65

Commit
 作業単位内での ECI リンク呼び出し 19

メソッド
 UOW COM クラスの 96

Poll 65

Connect
 メソッド
 Terminal COM クラスの 84

CICS サーバー情報および接続状況 17

Connect COM クラスの 39

ServerName の 43

Terminal COM クラスの 83

UOW COM クラスの 95

UserId の 45

Connect COM クラス
 メソッド
 AlterSecurity 40
 Cancel 40
 Changed 40
 ChangePassword 41
 Details 41
 Link 42
 MakeSecurityDefault 43
 Password 43
 ServerName 43
 ServerStatus 43
 ServerStatusText 44
 Status 44
 TranDetails 44

- Connect COM クラス (続き)
 - UnpaddedPassword 45
 - UnpaddedServerName 45
 - UnpaddedUserid 45
 - Userid 45
 - VerifyPassword 45
- Connect COM クラスの 39
- Connect.Link
 - 作業単位内での ECI リンク呼び出し 18
 - ECI 呼び出し同期タイプ 16, 17
- CreateObject 36, 39, 47, 51, 63, 67, 79, 83, 95
 - EPI を使用した CICS 3270 アプリケーションへの接続 22
 - VBScript を使用した CICS への ECI リンク呼び出し 16
- CursorCol
 - メソッド
 - Screen COM クラスの 71
- CursorRow
 - メソッド
 - Screen COM クラスの 71
- D**
- Data
 - メソッド
 - Buffer COM クラスの 36
- DataTag
 - メソッド
 - Field COM クラスの 58
- Day
 - SecTime COM クラスの 77
- DBCS 21
- Delphi 5
- Depth
 - メソッド
 - Screen COM クラスの 72
- Details
 - メソッド
 - Connect COM クラスの 41
 - CICS への ECI リンク呼び出し 15
 - Connect COM クラスの 39
- Devtype
 - メソッド
 - Terminal COM クラスの 85
- devType (パラメーター)
 - Connect 84, 85
 - SetTermDefns の 90
- Diagnose
 - メソッド
 - EPI COM クラスの 51
 - Flow COM クラスの 64
 - Session COM クラスの 79
 - Terminal COM クラスの 85
 - EPI COM クラスの 51
 - State の 55, 80
- Disconnect
 - メソッド
 - Terminal COM クラスの 85
 - Disconnect 85
- DisconnectWithPurge
 - メソッド
 - Terminal COM クラスの 85
- DiscReason
 - メソッド
 - Terminal COM クラスの 86
 - DiscReason の 86
- display (パラメーター)
 - ErrorWindow 48, 52
- E**
- ECI
 - CICS サーバー情報および接続状況 17
- ECI COM クラス
 - メソッド
 - ErrorFormat 47
 - ErrorOffset 48
 - ErrorWindow 48
 - ExCode 48
 - ExCodeText 49
 - ServerCount 49
 - ServerDesc 49
 - ServerName 50
 - SetErrorFormat 50
 - ECI COM クラスの 47
- ECI 呼び出し同期タイプ
 - CICS への ECI リンク呼び出し COM インターフェースの使用 16
- EPI
 - CICS サーバー情報 24
 - EPI COM クラスの 51
 - EPI COM クラス
 - メソッド
 - Diagnose 51
 - ErrorFormat 52
 - ErrorOffset 52
 - ErrorWindow 52
 - ExCode 53
 - ExCodeText 53
 - ServerCount 53
 - ServerDesc 54
 - ServerName 54
 - SetErrorFormat 54
 - State 54
 - Terminate 55
 - EPI COM クラスの 51
 - EPI 呼び出し同期タイプ
 - EPI を使用した CICS 3270 アプリケーションへの接続 COM クラスの使用 23
 - Err オブジェクト 12
 - ErrorFormat
 - メソッド
 - ECI COM クラスの 47
 - EPI COM クラスの 52
 - ErrorOffset
 - メソッド
 - ECI COM クラスの 48
 - EPI COM クラスの 52
 - ErrorWindow 12
 - メソッド
 - ECI COM クラスの 48
 - EPI COM クラスの 52
 - ExCode 12, 48, 52
 - メソッド
 - ECI COM クラスの 48
 - EPI COM クラスの 53
 - Map COM クラスの 68
 - Terminal COM クラスの 86
 - EPI COM クラスの 51

ExCode 12, 48, 52 (続き)

ErrorWindow 48, 52

State の 55, 80

ExCodeText 12

メソッド

ECI COM クラスの 49

EPI COM クラスの 53

Terminal COM クラスの 86

ExCode の 49

ExpiryTime

SecAttr COM クラスの 75

ExtractString

メソッド

Buffer COM クラスの 36

F

FALSE

Validate 69

False

Poll 65, 88

false

ErrorWindow 48, 52

Field COM クラス

メソッド

AppendText 57

BackgroundColor 58

BaseAttribute 58

Column 58

DataTag 58

ForegroundColor 59

Highlight 59

InputProt 59

InputType 60

Intensity 60

Length 60

Position 61

ResetDataTag 61

Row 61

SetBaseAttribute 61

SetExtAttribute 61

SetText 62

Text 62

TextLength 62

Transparency 62

Field COM クラスの 63, 67, 79, 83,
95

FieldByIndex

メソッド

Screen COM クラスの 72

BMS マップ・データの EPI

COM クラスでの使用 26

FieldByName

メソッド

Map COM クラスの 68

Validate 69

FieldByPosition

メソッド

Screen COM クラスの 72

BMS マップ・データの EPI

COM クラスでの使用 26

FieldCount

メソッド

Screen COM クラスの 72

Flow

Cancel 40

Changed 40

Commit 96

Connect COM クラスの 39

Flow COM クラスの 63

Link 42

SetSyncType 65

Status 44

Flow COM クラス

メソッド

AbendCode 63

CallType 64

CallTypeText 64

Diagnose 64

Flowid 64

ForceReset 64

Poll 64

SetSyncType 65

SetTimeout 65

SyncType 66

Timeout 66

Wait 66

flow (パラメーター)

BackOut 95

Cancel 40

Changed 40

flow (パラメーター) (続き)

Commit 96

Link 42

Status 44

Flowid

メソッド

Flow COM クラスの 64

ForceReset

メソッド

Flow COM クラスの 64

UOW COM クラスの 96

ForegroundColor

メソッド

Field COM クラスの 59

format (パラメーター)

SetErrorFormat の 50, 54

Form_Load

CICS への ECI リンク呼び出し
15

EPI を使用した CICS 3270 アプリケーションへの接続 22

G

GetDate

SecTime COM クラスの 77

H

Highlight

メソッド

Field COM クラスの 59

Hours

SecTime COM クラスの 77

HTML 資料の表示 120

HTML (ハイパーテキスト・マークアップ言語) 120

Hunderdths

SecTime COM クラスの 78

I

index (パラメーター)

ExCode の 53

FieldByIndex の 72

ServerDesc 49, 54

index (パラメーター) (続き)

ServerName 50, 54

InputProt

メソッド

Field COM クラスの 59

InputType

メソッド

Field COM クラスの 60

InsertString

メソッド

Buffer COM クラスの 37

Install

メソッド

Terminal COM クラスの 87

Intensity

メソッド

Field COM クラスの 60

K

key (パラメーター)

SetAID の 73

L

LastVerifiedTime

SecAttr COM クラスの 76

Length

メソッド

Buffer COM クラスの 37

Field COM クラスの 60

length (パラメーター)

ExtractString 36

SetLength 38

Link

メソッド

Connect COM クラスの 42

CICS への ECI リンク呼び出し
15

Details 42

Poll 65

UOW COM クラスの 95

link

SetSyncType 65

M

MakeSecurityDefault

メソッド

Connect COM クラスの 43

Terminal COM クラスの 87

Map

Map COM クラスの 67

Map COM クラス

メソッド

ExCode 68

FieldByName 68

Validate 68

MapName

メソッド

Screen COM クラスの 72

mapname (パラメーター)

Validate 68

MapSetName

メソッド

Screen COM クラスの 73

Map.FieldByName

BMS マップ・データの EPI

COM クラスでの使用 26

Map.Validate

BMS マップ・データの EPI

COM クラスでの使用 26

Minutes

SecTime COM クラスの 78

Month

SecTime COM クラスの 78

N

name (パラメーター)

FieldByName 68

NetName

メソッド

Terminal COM クラスの 87

New

Buffer COM クラスの 36

Connect COM クラスの 39

ECI COM クラスの 47

EPI COM クラスの 51

Field COM クラスの 63, 67, 79,

83, 95

newPassword (パラメーター)

AlterSecurity の 40, 83

ChangePassword の 41

newUserId (パラメーター)

AlterSecurity の 40, 83

Nothing

Poll 65

nworkName (パラメーター)

Connect 84, 85

SetTermDefns の 90

O

offset (パラメーター)

ExtractString 36

InsertString 37

Overlay 37

On Error Goto 12

On Error Resume 12

Overlay

メソッド

Buffer COM クラスの 37

P

Password

メソッド

Connect COM クラスの 43

Terminal COM クラスの 87

password (パラメーター)

ChangePassword の 84

Details 41

SetTermDefns の 90

PDF (Portable Document

Format) 120

PDF ブックの表示 120

Poll

メソッド

Flow COM クラスの 64

Terminal COM クラスの 87

ECI 呼び出し同期タイプ 17

EPI 呼び出し同期タイプ 23

SetSyncType 65, 80

poll メソッド

ATI サポートで 28

PollForReply
メソッド
Terminal COM クラスの 88

pollForReply メソッド
ATI サポートで 28

Portable Document Format
(PDF) 120

Position
メソッド
Field COM クラスの 61

PostScript ブック 120

programName (パラメーター)
Link 42

Q

QueryATI
メソッド
Terminal COM クラスの 88

queryATI 27

R

ReadTimeout
メソッド
Terminal COM クラスの 89

ReadTimeout (パラメーター)
SetTermDefns の 90

ReceiveATI
メソッド
Terminal COM クラスの 89

receiveATI メソッド
ATI サポートで 28

ResetDataTag
メソッド
Field COM クラスの 61

ResetDataTag 61

Row
メソッド
Field COM クラスの 61

rowPos (パラメーター)
FieldByPosition 72
SetCursor 73

runTran (パラメーター)
TranDetails 44

S

Screen
メソッド
Terminal COM クラスの 89

Field COM クラスの 57

Screen 89

Screen COM クラスの 71

Validate 68

Screen COM クラス
メソッド
CursorCol 71
CursorRow 71
Depth 72
FieldByIndex 72
FieldByPosition 72
FieldCount 72
MapName 72
MapSetName 73
SetAID 73
SetCursor 73
Width 74

screenRef (パラメーター)
Validate 68

Screen.fieldbyIndex メソッド
Ccl Field COM クラスの 57

Seconds
SecTime COM クラスの 78

Send
メソッド
Terminal COM クラスの 89

Poll 88

SetSyncType 80

ServerCount
メソッド
ECI COM クラスの 49
EPI COM クラスの 53

ServerDesc
メソッド
ECI COM クラスの 49
EPI COM クラスの 54

ExCode の 53

ServerName
メソッド
Connect COM クラスの 43
ECI COM クラスの 50

ServerName (続き)
メソッド (続き)
EPI COM クラスの 54
Terminal COM クラスの 90

Details 41

ExCode の 53

serverName (パラメーター)
Details 41

ServerStatus
メソッド
Connect COM クラスの 43

ServerStatusText
メソッド
Connect COM クラスの 44

servName (パラメーター)
Connect 84
SetTermDefns の 90

Session COM クラス
メソッド
Diagnose 79
SetSyncType 79
State 80
TransId 81

session (パラメーター)
ReceiveATI の 89
Send の 89
Start 92

Session.SetSyncType
EPI 呼び出し同期タイプ 23

SetAID
メソッド
Screen COM クラスの 73

SetATI
メソッド
Terminal COM クラスの 90

setATI 27

SetBaseAttribute
メソッド
Field COM クラスの 61

SetCursor
メソッド
Screen COM クラスの 73

SetData
メソッド
Buffer COM クラスの 37

SetErrorFormat 13

SetLastError 13 (続き)
 メソッド
 ECI COM クラスの 50
 EPI COM クラスの 54
 SetExtAttribute
 メソッド
 Field COM クラスの 61
 SetLength
 メソッド
 Buffer COM クラスの 38
 Buffer COM クラスの 35
 SetString
 メソッド
 Buffer COM クラスの 38
 SetSyncType
 メソッド
 Flow COM クラスの 65
 Session COM クラスの 79
 ECI 呼び出し同期タイプ 17
 SetTermDefns
 メソッド
 Terminal COM クラスの 90
 SetText
 メソッド
 Field COM クラスの 62
 SetTimeout
 メソッド
 Flow COM クラスの 65
 SignonCapability
 メソッド
 Terminal COM クラスの 91
 signonCapability (パラメーター)
 SetTermDefns の 90
 Start
 メソッド
 Terminal COM クラスの 92
 Poll 88
 SetSyncType 80
 startData (パラメーター)
 Start 92
 State
 メソッド
 EPI COM クラスの 54
 Session COM クラスの 80
 Terminal COM クラスの 92
 EPI COM クラスの 51
 State (続き)
 State の 92
 stateVal (パラメーター)
 SetATI の 90
 Status
 メソッド
 Connect COM クラスの 44
 Details 42
 Poll 65
 status
 ServerStatus 43
 ServerStatusText 44
 SetSyncType 65
 String
 メソッド
 Buffer COM クラスの 38
 string (パラメーター)
 AppendString 36
 InsertString 37
 Overlay 37
 SetString 38
 SyncType
 メソッド
 Flow COM クラスの 66
 syncType (パラメーター)
 SetSyncType 65, 79, 80
 SyncType の 66
T
 TermId
 メソッド
 Terminal COM クラスの 92
 Terminal.Connect
 Screen COM クラスの 71
 Terminal.Poll
 EPI 呼び出し同期タイプ 24
 Terminal.Screen
 Screen COM クラスの 71
 Terminal.Send
 EPI 呼び出し同期タイプ 23
 Terminal.Start
 CICS 3270 セッションの実行 22
 EPI 呼び出し同期タイプ 23
 Terminate
 メソッド
 EPI COM クラスの 55
 Terminate 55
 Text
 メソッド
 Field COM クラスの 62
 TextLength
 メソッド
 Field COM クラスの 62
 textString (パラメーター)
 AppendText 57, 58
 SetText 62
 Timeout
 メソッド
 Flow COM クラスの 66
 timeout (パラメーター)
 Install の 87
 tranCode (パラメーター)
 Start 92
 TranDetails
 メソッド
 Connect COM クラスの 44
 Connect COM クラスの 39
 TransId
 メソッド
 Session COM クラスの 81
 Terminal COM クラスの 93
 Transparency
 メソッド
 Field COM クラスの 62
 TRUE
 Validate 69
 True
 Poll 65, 88
 true
 ErrorWindow 48, 52
U
 unitOfWork (パラメーター)
 Link 42
 UnpaddedPassword
 メソッド
 Connect COM クラスの 45

UnpaddedServerName
メソッド
 Connect COM クラスの 45

UnpaddedUserid
メソッド
 Connect COM クラスの 45

UOW
作業単位内での ECI リンク呼び出し 18, 19
Link 42
UOW COM クラスの 95

UOW COM クラス
メソッド
 BackOut 95
 Commit 96
 ForceReset 96
 UowId 96

UowId
メソッド
 UOW COM クラスの 96

UserId
メソッド
 Connect COM クラスの 45

Userid
メソッド
 Terminal COM クラスの 93

userID (パラメーター)
Details 41

userId (パラメーター)
Details 41

userid (パラメーター)
SetTermDefns の 90

V

Validate
メソッド
 Map COM クラスの 68

Value (パラメーター)
SetExtAttribute の 61

VBScript 5

VerifyPassword
メソッド
 Connect COM クラスの 45
 Terminal COM クラスの 93

Visual Basic 5

W

Wait
メソッド
 Flow COM クラスの 66
Wait の 66

Width
メソッド
 Screen COM クラスの 74

Y

Year
SecTime COM クラスの 78



プログラム番号: 5724-A75

Printed in Japan

SC88-8997-00



日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12