

DB2 照会報告書作成プログラム
バージョン 12 リリース 1

DB2 QMF アプリケーション
開発の手引き

IBM

DB2 照会報告書作成プログラム
バージョン 12 リリース 1

DB2 QMF アプリケーション
開発の手引き

IBM

お願い

本書および本書で紹介する製品をご使用になる前に、この情報の終わりにある『特記事項』に記載されている情報をお読みください。

このエディションは、IBM DB2 12 for z/OS (5650-DB2) および IBM DB2 11 for z/OS (5615-DB2) のフィーチャーである IBM DB2 照会報告書作成プログラム (QMF) Classic Edition および Enterprise Edition バージョン 12 リリース 1 に適用されます。また、独立型 IBM DB2 for z/OS ツールである IBM DB2 QMF for z/OS (5697-QM2) のバージョン 12 リリース 1 にも当てはまります。この情報は、新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

原典： SC27-8878-00
DB2 Query Management Facility
Version 12 Release 1
Developing DB2 QMF applications

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 1982, 2016.

目次

本書について	vii
前提知識	vii
サービス更新およびサポート情報	viii
強調表示規則	viii
構文図の読み方	viii
ご意見の送付方法	x
第 1 章 QMF アプリケーション開発の概要	1
QMF でのアプリケーション開発とは	1
ユーザーがアプリケーションを使用する方法	1
アプリケーションとの主要な対話	1
QMF セッションからのアプリケーションの開始	2
使用可能な QMF アプリケーション開発ツール	3
QMF プロシージャ	3
QMF へのアプリケーション・プログラミング・インターフェース	4
各国語機能情報に関する規則	5
第 2 章 アプリケーションとしてのプロシージャ	7
初期プロシージャ	7
プロシージャ内の QMF CONNECT	9
プロシージャ内の置換変数	10
RUN コマンドでの値の指定	10
「RUN コマンド・プロンプト」パネルでの値の指定	11
ロジックを持つプロシージャ内の REXX 変数	12
ロジックを持つプロシージャへの引数の受け渡し	12
ロジックを持つプロシージャ内の REXX エラー処理ステートメント	13
エラー処理サブルーチンへの分岐	13
REXX EXIT ステートメントによるメッセージ	13
ロジックを持つプロシージャからの REXX プログラムの呼び出し	14
置換変数を含まない REXX プログラムの呼び出し	15
置換変数を含む REXX プログラムの呼び出し	15
第 3 章 呼び出し可能インターフェースおよび QMF アプリケーション	17
呼び出し可能インターフェースとは	17
QMF 呼び出し可能インターフェースの使用に関する考慮事項	17
インターフェース連絡域 (DSQCOMM)	18
戻りコード	20
呼び出し可能インターフェースを使用するためのコマンド	21
アプリケーションからの QMF の開始	22
呼び出し可能インターフェース・アプリケーションの実行	22
QMF 内での呼び出し可能インターフェース	22
エラー処理	23
CICS のもとでの呼び出し可能インターフェース・プログラムの実行	23
第 4 章 ISPF ダイアログからの QMF コマンドの発行	25
コマンド・インターフェースを使用するプログラムの作成: 例	25
コマンド・インターフェースの呼び出し	26
END コマンド	27
コマンド・インターフェースでの変数	27
コマンド・インターフェースの戻りコード	27

第 5 章 ADDRESS QRW および QMF コマンド環境	31
第 6 章 ISPF サービスを使用する QMF アプリケーションの作成	33
ISPF アプリケーションからの QMF の開始および実行	33
変数を含む照会の実行	34
QMF 内から ISPF サービスを使用するプログラムの開始	35
ロジックを持つプロシージャ内の ISPF サービス	35
ISPF での EDIT コマンド	36
ISPF およびアプリケーションのデバッグ	37
第 7 章 2 か国語使用のアプリケーションの作成	39
英語環境と NLF 環境の比較	39
2 か国語使用のアプリケーションで使用するオブジェクトの作成	41
コマンド言語変数	41
2 か国語使用のアプリケーション内の初期プロシージャ	42
英語のみのコマンド	43
複数言語環境	44
変換可能なアプリケーションの作成	44
第 8 章 アプリケーションでの QMF コマンド	45
アプリケーション用に設計されたコマンド	45
CONNECT	45
END	46
EXIT	47
GET GLOBAL (拡張構文)	47
INTERACT	48
MESSAGE	51
SET GLOBAL (拡張構文)	54
START	56
TRACE	63
RUN QUERY 報告書ミニ・セッションで使用できるコマンド	68
コマンド同義語	71
第 9 章 オブジェクトのエクスポートとインポート	73
エクスポートした UNIX ファイル、TSO データ・セット、または CICS データ・キューを使用して実行できる内容	73
データのエクスポートと保管	74
データ・オブジェクトおよびデータベース表のエクスポート	74
QMF 形式でのデータまたは表のエクスポート	75
IXF 形式でのデータまたは表のエクスポート	81
XML 形式でのデータまたは表のエクスポート	98
CSV 形式でのデータまたは表のエクスポート	101
データ・オブジェクトと表のエクスポートおよびインポートに関する規則と情報	102
書式、報告書、および指示照会のエクスポート	103
エクスポート・ファイルの汎用形式	103
書式のエクスポート	113
アプリケーション内の QMF 書式オブジェクトに関する考慮事項	122
標準的な報告書のエクスポート	124
HTML 形式内の報告書のエクスポート	128
制御情報なしでの報告書のエクスポート	129
横方向スタイル報告書のエクスポート	130
指示照会のエクスポート	132
エクスポート後の指示照会が有効な形式であることの確認	140
書式および指示照会のインポート	141
プロシージャおよび SQL 照会	142
エクスポートされた書式ベースの図表および QBE 照会	143

外部化オブジェクトのサイズの仕様	143
記憶域についての考慮事項	146
CICS データ・キュー	146
TSO データ・セット	147
第 10 章 QMF アプリケーションのデバッグ	149
呼び出し可能インターフェース・アプリケーションのデバッグ	149
トレース用の L オプション	149
トレース用の A オプション	150
トレースをオフにする	150
QMF トレース・データ出力の割り振り	151
トレース用の QMF MESSAGE コマンド	151
START コマンドまたはその他の QMF コマンドのエラー	152
第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定	153
アセンブラ言語インターフェース	153
アセンブラ用のインターフェース連絡域マッピング (DSQCOMMA)	153
アセンブラ 言語用の関数呼び出し	154
アセンブラ・プログラミングの例	156
アセンブラ用の DSQCOMM	163
CICS での アセンブラ プログラムの実行	164
TSO での アセンブラ プログラムの実行	165
C 言語インターフェース	167
C 言語用のインターフェース連絡域マッピング (DSQCOMM C)	167
C 言語用の関数呼び出し	168
C 言語プログラミングの例	170
C 用の DSQCOMM	173
CICS での C プログラムの実行	175
TSO での C プログラムの実行	175
COBOL 言語インターフェース	178
COBOL 用のインターフェース連絡域マッピング (DSQCOMMB)	178
COBOL 用の関数呼び出し	179
COBOL での ISPF LIBDEF サービス	180
COBOL・プログラミングの例	181
COBOL 用の DSQCOMM	182
COBOL 呼び出し可能インターフェース・プログラムの実行に関する考慮事項	184
CICS での COBOL プログラムの実行	185
TSO での COBOL プログラムの実行	185
Fortran 言語インターフェース	188
Fortran 用のインターフェース連絡域マッピング (DSQCOMMF)	188
Fortran 用の関数呼び出し	189
Fortran プログラムの例	191
Fortran 用の DSQCOMM	194
Fortran プログラムの実行	195
PL/I 言語インターフェース	198
PL/I 用のインターフェース連絡域マッピング (DSQCOMML)	198
PL/I 用の関数呼び出し	199
PL/I プログラムの例	201
PL/I 用の DSQCOMM	203
CICS でのプログラムの実行	205
TSO でのプログラムの実行	205
REXX 言語インターフェース	208
REXX 用のインターフェース連絡変数	208
REXX 用の関数呼び出し	209
REXX プログラムの例	210
REXX プログラムの実行	211

INTERACT ループを使用する REXX の例	212
付録 A. プロダクト・インターフェース・マクロ	215
付録 B. QMF 大域変数	217
QMF 大域変数の命名規則	217
大域変数の値の設定と表示	218
プロファイル関連でない状態情報の大域変数	218
プロファイル関連状態情報の大域変数	224
CICS 関連の大域変数	225
最後に発行されたコマンドで生成されたメッセージに関連する大域変数	226
表エディター関連の大域変数	227
さまざまな表示を制御する大域変数	229
コマンドとプロシージャーの実行を制御する大域変数	240
CONVERT QUERY の結果を保管する大域変数	250
RUN QUERY エラー・メッセージ情報を示す大域変数	251
パネル入力値を保管する大域変数	251
特記事項	263
プログラミング・インターフェース情報	264
商標	265
用語集	267
索引	283

本書について

IBM® DB2® 照会報告書作成プログラム (QMF) for TSO/CICS® は、堅固に統合された強力で信頼性の高いツールであり、照会および報告書作成の機能を提供し、以下のリレーショナル・データベースのデータへのアクセスおよびデータの提示を支援します。

- DB2 for z/OS®
- DB2 for Linux, UNIX, and Windows
- DB2 for iSeries
- DB2 Server (VSE および VM 版)

本書は、QMF™ 機能を使用するアプリケーションの開発を担当しているアプリケーション・プログラマーを対象としています。これらのトピックでは、以下の内容について説明します。

- アプリケーション・プログラミング設計意思の決定
- さまざまなプログラミング手法の中からの選択
- QMF コマンドおよび呼び出し可能インターフェースの使用法の理解
- 2 か国語使用のアプリケーションの作成

アセンブラー、C、FORTRAN、COBOL、PL/I、および REXX に固有のプログラミング例が用意されています。

前提知識

QMF のアプリケーション・プログラミングを開始する前に、特定の環境を構成するコンポーネントに加え、いくつかの概念や用語に精通している必要があります。

製品

QMF 用アプリケーションを開発するには、お客様の環境およびビジネス要件に応じて、次の製品の一部またはすべてに精通している必要があります。

- z/OS オペレーティング・システム。
- DB2。QMF 用のデータベース・マネージャーです。
- タイム・シェアリング・オプション(TSO)。DB2 QMF およびそれに関連した製品をサポートする環境です。
- 対話式システム生産性向上機能 (ISPF)。DB2 QMF 用のダイアログ・マネージャーです。
- 顧客情報管理システム (CICS)。汎用のデータ通信およびオンライン・トランザクション処理システムです。CICS は、DB2 QMF と z/OS の間のインターフェースを提供します。
- 図形データ表示管理プログラム (GDDM) 製品。パネルの表示および図表の作成に必要となります。QMF の印刷サービスを使用するために、GDDM を使用することもできます。

- アセンブラー、C、COBOL、FORTRAN、PL/I または REXX。QMF で呼び出し可能インターフェースを作成する場合に使用します。

概念

QMF アプリケーションでは、QMF がサポートするいずれかの言語で書かれたアプリケーション・プログラム内から QMF オブジェクトを処理し、QMF 機能を実行することになります。本書では、読者が照会とプロシージャの作成方法、報告書の形式設定方法、およびデータベースの変更方法に関する知識を持っていることを前提としています。

関連情報:

 [IBM Publications Center](#)

これらの製品について説明した資料を検索します。

サービス更新およびサポート情報

サービス更新およびサポート情報 (ソフトウェア・フィックスパック、PTF、よくある質問 (FAQ)、技術情報、トラブルシューティング情報、およびダウンロードなど) を検索するには、次の Web ページを参照してください。

IBM ソフトウェア・サポート Web サイト

強調表示規則

本書では、以下のような強調表示の規則を使用しています。

- 太字体タイプは、コマンドまたはユーザー・インターフェース・コントロール (フィールド名、フォルダー名、アイコン名、メニュー選択名など) を示します。
- モノスペース・タイプは、ユーザーが示されたとおり正確に入力するテキストの例を示します。
- イタリック は、他の資料の表題、または重要な用語の強調を示します。ユーザーが値を置き換える必要のある変数を示すためにも使用されます。

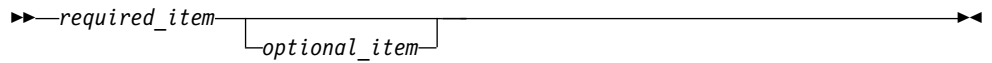
構文図の読み方

本書で使用されている構文図には、以下の規則が適用されます。

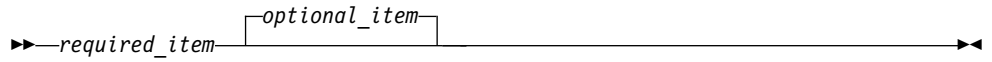
- 構文図は線の経路に沿って、左から右、上から下へ読んでいきます。以下の規則が使用されます。
 - >>--- 記号は、構文図の始まりを示します。
 - ---> 記号は、構文図が次の行に続くことを示します。
 - >--- 記号は、構文図が前の行から続いていることを示します。
 - --->< 記号は、構文図の終わりを示します。
- 必須項目は、水平線 (主経路) 上に表示されます。

▶—*required_item*————▶

- オプション項目は、主経路の下に表示されます。

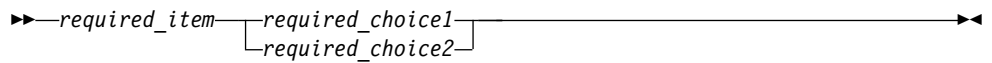


オプション項目が主経路より上に表示されている場合、その項目は構文エレメントの実行には影響を与えず、読みやすさのためにのみ使用されています。

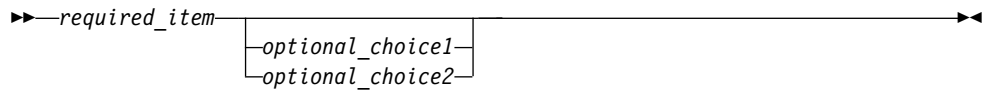


- 複数の項目から選択できる場合、項目は縦に積み重ねて表示されます。

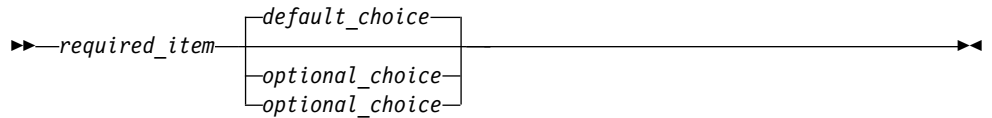
項目の 1 つを選択することが必須である場合は、積み重ねられた項目の 1 つが主経路上に表示されます。



項目の 1 つを選択することが任意である場合は、積み重ねられた項目全体が主経路より下に表示されます。



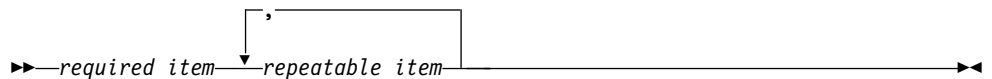
項目の 1 つがデフォルトである場合、その項目は主経路より上に表示され、残りの項目は下に表示されます。



- 主線より上の、左側に戻る矢印は、繰り返し可能な項目を示します。



繰り返し矢印にコンマが含まれている場合は、繰り返される項目をコンマで区切る必要があります。



積み重ねられた項目より上にある繰り返し矢印は、積み重ねの中の項目が繰り返し可能であることを示します。

- キーワードとその最も短い省略形 (該当する場合) は、大文字で表示されます。これらは、示されているとおり正確にスペルを入力する必要があります。変数は

すべて小文字のイタリック体 (例えば *column-name*) で表示されます。これらは、ユーザー指定の名前または値を表します。

- 構文図でキーワードとパラメーターの間に句読点がない場合は、キーワードとパラメーターを 1 つ以上のスペースで区切ってください。
- 句読記号、括弧、算術演算子、およびその他の記号は、構文図に示されているとおり正確に入力してください。
- 脚注は、例えば (1) のように、括弧内の番号で示されます。

ご意見の送付方法

IBM お客様のご意見をお寄せください。本マニュアルまたは他の資料に関するご意見やご感想は、次の方法でお送りください。

- 次の場所にあるオンラインの読者ご意見フォームを使用する。

<http://www.ibm.com/software/data/rcf>

- ご意見を E メールで comments@us.ibm.com に送信する。本書の表題、本書の部品番号、製品のバージョン、およびご意見の対象となるテキストの位置 (該当する場合。例えばページ番号や表番号など) を記入してください。

第 1 章 QMF アプリケーション開発の概要

アプリケーション開発とは、QMF でプロシージャまたはアプリケーションを作成する過程を意味します。

ユーザーは、自分のアプリケーション中で QMF の多くの機能を使用することができます。例えば、次に挙げることを行うアプリケーションを作成することができます。

- 照会またはプロシージャの実行
- QMF オブジェクトおよび表のエクスポートまたはインポート
- 報告書または図表の表示または印刷
- ユーザーによるデータベースへのデータの入力またはデータベース内のデータの変更
- ユーザーによるいくつかのオブジェクトへの 1 回での一括変更

QMF のユーザーに便利な機能を提供するアプリケーションを作成することもできます。例えば、リモート・ロケーションで QMF 報告書を印刷するコマンド、または週ごとの売上結果の図表を生成するファンクション・キーを作成します。

QMF でのアプリケーション開発とは

「アプリケーション」という言葉には多くの意味があります。QMF では、アプリケーションは QMF コマンドを発行し、QMF サービスを使用して特定のビジネス・タスクを達成するプロシージャまたはプログラムです。

アプリケーション開発では、以下を行う必要があります。

- プロシージャまたはアプリケーションが解決する必要がある問題の理解
- プロシージャまたはアプリケーションの設計
- コード、関連するメッセージ、およびヘルプ・パネルの作成

ユーザーがアプリケーションを使用する方法

QMF アプリケーションには 2 つの主要なタイプがあります。1 つはユーザーがアプリケーションと対話するタイプで、もう 1 つはアプリケーションを QMF 内から開始できるタイプです。

アプリケーションとの主要な対話

アプリケーションが QMF に慣れていないユーザーを対象にしている場合、ユーザーが主としてアプリケーションと対話するようにした方がよいでしょう。

QMF がアクティブになっていることをユーザーに知らせないようにすることができます。この場合、アプリケーションは QMF サービスを使用しますが、QMF の外で実行されます。プログラムは必要な場合にだけ、QMF コマンドを発行します。

QMF サービスを使用するアプリケーションを作成するとします。このアプリケーションは、ユーザーにメニュー方式のインターフェースを提供します。次の例では、アプリケーションが QMF を制御します。ユーザーは、ユーザー・インターフェースとだけ対話し、QMF がアクティブであることに気付きません。

ユーザーがオプション 1 を選択すると、アプリケーションは、照会を実行し、結果の報告書を印刷する QMF プロシージャールを実行します。

```
J & H Supply Company
Information System

Please select one of the following:

1. Print the monthly sales report
2. Create a new report
3. Modify information in the database
4. End the application

====> 1
```

図 1. アプリケーション定義パネルの例

QMF セッションからのアプリケーションの開始

ユーザーが QMF に精通している場合は、ユーザーがアプリケーションを QMF の拡張部分またはカスタマイズしたものと見なすようにすることもできます。この場合には、アプリケーションを QMF 内で実行するようにセットアップする必要があります。

あるユーザーから別のユーザーに QMF 報告書を送信する SEND_TO というアプリケーションを作成するとします。

ユーザーが、そのアプリケーションを QMF 環境内から実行するものと想定します。ユーザーはコマンド行を使用して、SEND_TO (新たに作成します) という QMF コマンド同義語を発行することができます。あるいは、アプリケーションを自動的に実行するファンクション・キーにアプリケーションを割り当てることもできます。

報告書を生成した後に、ユーザーはカスタマイズした QMF コマンド SEND_TO SMITH を QMF コマンド行に入力することによって、Smith 氏にこの報告書を送信することができます。

```

REPORT                                     LINE 1   POS 1   79

NAME      DEPT  JOB      SALARY      COMM
-----
DANIELS   10  MGR      19260.25    -
JONES     10  MGR      21234.00    -
LU        10  MGR      20010.00    -
MOLINARE  10  MGR      22959.20    -
HANES     15  MGR      20659.80    -
KERMISCH  15  CLERK    12258.50    110.10
NGAN      15  CLERK    12508.20    206.60
ROTHMAN   15  SALES    16502.83    1152.00
JAMES     20  CLERK    13504.60    128.20
PERNAL    20  SALES    18171.25    612.45
SANDERS   20  MGR      18357.50    -
SNEIDER   20  CLERK    14252.75    126.50
ABRAHAMS  38  CLERK    12009.75    236.50
MARENGHI  38  MGR      17506.75    -
1=Help    2=      3=End    4=Print   5=Chart   6=Query
7=Backward 8=Forward 9=Form   10=Left  11=Right  12=
OK, here is your report.
COMMAND ==>> SEND_TO SMITH

```

図 2. カスタマイズした QMF コマンドを入力するユーザーの例

使用可能な QMF アプリケーション開発ツール

QMF プロシージャを使用するアプリケーション、および QMF へのアプリケーション・プログラミング・インターフェースを使用するアプリケーションを作成できます。

コマンド同義語を作成して、プロシージャおよびアプリケーションを呼び出すことができます。コマンド同義語は、QMF、TSO、または CICS コマンドを実行するコマンドです。コマンドおよびその定義をコマンド同義語表に入力することにより、コマンド同義語を作成します。初期化中に、QMF は、QMF を開始したユーザーの QMF プロファイルで指定されているコマンド同義語表をロードします。

QMF プロシージャ

QMF プロシージャは、QMF 内で実行されるものであり、QMF コマンドを発行する QMF オブジェクトです。QMF プロシージャは、対話式または非対話式に実行できます。プロシージャを非対話式に実行するには、バッチ・モードで QMF を開始します。QMF for TSO を使用する場合は、DB2 for z/OS ストアード・プロシージャとして QMF を開始することもできます。

QMF プロシージャは、サイトで使用可能なすべての QMF コマンドを実行することができます。QMF には、線形手順とロジックを持つプロシージャという 2 つのタイプのプロシージャがあります。

- 線形手順 には QMF コマンドとコメントだけが入っています。QMF がサポートするすべての環境で、線形手順を使用することができます。
- ロジックを持つプロシージャ は、QMF コマンドと REXX ロジックを結合して、より強力なプログラムを作成できるようにします。CICS を除く QMF でサポートされているすべての環境でロジックを持つプロシージャを使用できま

す。ロジックを持つプロシージャーには、システム・コマンドを含め、QMF コマンド、および REXX プログラムで有効なすべてのステートメントを入れることができます。

QMF はシステム初期化プロシージャーおよびその他の方法を提供します。これらの方法では、ユーザーに向けて QMF ホーム・パネルを表示する前に、コマンドを実行して大域変数を設定できます。

関連概念:

7 ページの『第 2 章 アプリケーションとしてのプロシージャー』
アプリケーションをすべてプロシージャーとして作成することができます。CICS 環境で QMF を使用している場合、線形プロシージャーを作成できます。これは、QMF コマンド、または QMF コマンドを発行する同義語のみを含むプロシージャーです。TSO 環境で QMF を使用している場合、線形プロシージャーに加えて、ロジックを持つプロシージャーを作成することができます。ロジックを持つプロシージャーは、QMF コマンドおよびコマンド同義語に加えて REXX ステートメントと機能を含むことができます。

関連資料:

240 ページの『コマンドとプロシージャーの実行を制御する大域変数』
DSQEC 大域変数は、コマンドおよびプロシージャーの実行方法を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

QMF へのアプリケーション・プログラミング・インターフェース

QMF へ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は、呼び出し可能インターフェースとコマンド・インターフェースの 2 つがあります。

呼び出し可能インターフェース

QMF 呼び出し可能インターフェースを使用して、QMF 外で呼び出されるアプリケーションを作成できます。その後、このアプリケーションは QMF セッションを開始し、コマンドを実行するために QMF に送信します。

呼び出し可能インターフェースは、QMF サービスへのアクセスを可能にするプログラミング・インターフェースです。呼び出し可能インターフェースは、次の表で示されているプログラミング言語および環境に使用できます。

表 1. 呼び出し可能インターフェースのサポート

	CICS	TSO	ネイティブ z/OS パッチ
アセンブラー	×	×	×
C	×	×	×
COBOL	×	×	×
Fortran		×	×
PL/I	×	×	×
REXX		×	×

QMF は、DB2 for z/OS でサポートされているこれらのプログラミング言語の全バージョンをサポートします。

コマンド・インターフェース (TSO のみ)

QMF コマンド・インターフェースを使用して、ISPF ダイアログから QMF にコマンドを実行要求するアプリケーションを作成することができます。QMF は、このコマンド・インターフェースを使用し、ISPF 変数プールを介して ISPF ダイアログと通信します。QMF は、ISPF アプリケーションが開始される前に、開始される必要があります。

コマンド・インターフェースは、ISPF が使用可能な場合のみ使用することができます。CICS では使用できません。ISPF でサポートされている任意のプログラミング言語で、コマンド・インターフェース・アプリケーションを作成できます。

関連概念:


17 ページの『第 3 章 呼び出し可能インターフェースおよび QMF アプリケーション』

それぞれのプログラム言語で、QMF コマンドを実行するために QMF 呼び出し可能インターフェースを使用することができます。

25 ページの『第 4 章 ISPF ダイアログからの QMF コマンドの発行』

QMF で実行されている ISPF ダイアログから QMF コマンドを発行するには、QMF コマンド・インターフェースを使用します。

関連情報:

 アプリケーション・プログラムを開発するためのプログラミング言語と方式 DB2 for z/OS でサポートされているプログラミング言語に関する情報を参照してください。

各国語機能情報に関する規則

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

NLF を使用すると、ユーザーは英語以外の言語で、QMF コマンドを入力でき、ヘルプを表示でき、QMF タスクを実行できます。NLF は、DB2 QMF の個別機能としてインストールされます。

ここで説明しているすべてのタスクは、QMF の基本製品 (英語版) および任意の NLF で実行できます。基本セッションと NLF セッションのプロシーチャーはどちらも同じです。ただし、NLF ユーザー向けに特別な考慮事項がある場合は、それが明記されています。

ここに記載するプログラムまたはデータ・セットの名前に、変数 n が使用されていることがあります。これは、名前の中でこの文字がさまざまな文字に置き換えられることを示しています。変数 n は、使用している言語機能に対応する 1 文字の各国語 ID (NLID) に置き換えます。この ID を次の表に示します。下表には、QMF が各国語を識別する名前も示されています。

表 2. QMF NLF とその識別情報

各国語機能 (National Language Feature)	ID (NLID)	QMF でこの NLF に使用される名前
英語	E	ENGLISH
大文字英語	U	UPPERCASE
カナダ・フランス語	C	FRANCAIS CANADIEN
デンマーク語	Q	DANSK
フランス語	F	FRANCAIS
ドイツ語	D	DEUTSCH
イタリア語	I	ITALIANO
日本語漢字	K	NIHONGO
韓国語 (ハングル)	H	HANGEUL
ブラジル・ポルトガル語	P	PORTUGUES
スペイン語	S	ESPANOL
スウェーデン語	V	SVENSKA
スイス・フランス語	Y	FRANCAIS (SUISSE)
スイス・ドイツ語	Z	DEUTSCH (SCHWEIZ)

大文字英語機能では、英語を使用しますが、テキストはすべて英大文字に変換されます。カタカナ用のディスプレイ装置で作業しているユーザーは大文字を使用することによって、製品を使用でき、英語版のオンライン・ヘルプとメッセージを表示できます。

第 2 章 アプリケーションとしてのプロシージャー

アプリケーションをすべてプロシージャーとして作成することができます。CICS 環境で QMF を使用している場合、線形プロシージャーを作成できます。これは、QMF コマンド、または QMF コマンドを発行する同義語のみを含むプロシージャーです。TSO 環境で QMF を使用している場合、線形プロシージャーに加えて、ロジックを持つプロシージャーを作成することができます。ロジックを持つプロシージャーは、QMF コマンドおよびコマンド同義語に加えて REXX ステートメントと機能を含むことができます。

QMF 一時記憶域内のプロシージャーに対する操作を行うアプリケーションを作成する場合、アプリケーションをプロシージャーとして作成することはできません。プロシージャーを実行すると、それが QMF 一時記憶域内の現在のプロシージャーになるからです。

関連概念:

35 ページの『ロジックを持つプロシージャー内の ISPF サービス』
ISPF のもとで実行されるロジックを持つ QMF プロシージャーから ISPF コマンドを実行するには、QMF プログラム・ダイアログから ISPF コマンド・ダイアログに移動する必要があります。

初期プロシージャー

初期プロシージャーは、QMF セッションの開始直後に実行されるプロシージャーです。DSQSRUN パラメーターを使用して、このプロシージャーの名前を指定し、特定の状態における初期プロシージャーの動作について理解してください。

DSQSRUN パラメーターは、以下で使用できます。

- QMF を対話式で開始する場合の DSQQMF n コマンド (n は、使用している言語機能に対応する 1 文字の各国語 ID)。
- 呼び出し可能インターフェースを介して QMF を開始した場合、QMF START コマンド

TSO、ISPF、およびネイティブ z/OS バッチでは、アプリケーションは REXX プログラムを使用してプログラム・パラメーターを設定することもできます。このプログラムは、QMF START コマンドの DSQSCMD パラメーターで指定します。QMF for CICS では REXX はサポートされないため、START コマンドで DSQSMODE=I を使用してすべてのプログラム・パラメーターを指定する必要があります。DSQSMODE のこの値は、CICS での対話式操作を指定します。呼び出し可能インターフェースからのデフォルト・モードは B です (バッチ操作の場合)。

初期プロシージャー作成に関する考慮事項

初期プロシージャーを作成し、使用するときは、以下の点を考慮してください。

- デフォルトでは、DSQQMF n により開始された QMF の対話式セッションでユーザーが END コマンドを発行すると、必ず QMF によって初期プロシージャーが再実行されます。(変数 n は、使用している言語機能に対応する 1 文字の各

国語 ID です。) DSQEC_RERUN_IPROC 大域変数が、初期プロシージャーを再実行するかどうかを指定します。この変数のデフォルト値は 1 で初期プロシージャーを再実行することを示しています。0 を設定すると、初期プロシージャーを再実行しません。

呼び出し可能インターフェース・プログラムでは初期プロシージャーは再実行されないため、この大域変数は呼び出し可能インターフェース・プログラムには影響を与えません。

- 対話式 QMF セッションで使用する初期プロシージャーを作成する場合は、このプロシージャーの最後の現行パネルとしてホーム・パネルを使用しないでください。この場合、QMF はプロシージャーの最後にパネルを対話式に表示しません。重大エラーがなく、DSQEC_RERUN_IPROC が 1 に設定されている場合、QMF は、ユーザーと対話することなく初期プロシージャーを再実行します。この結果、割り込み不能ループになり、QMF が開始していないかのように見えます。

割り込み不能ループが発生しないようにするには、以下のいずれかのオプションを考慮してください。

- プロシージャーの最後の現行パネルがホーム・パネルでないようにする。
- プロシージャーに QMF EXIT コマンドまたは INTERACT コマンドを入れる。
- DSQEC_RERUN_IPROC をゼロ (0) に設定する。
- 初期プロシージャー内で置換変数の名前の前に使用する必要があるアンパーサンド (&) の数は、ご使用の環境に応じて異なる可能性があります。例えば、DSQSRUN を次のように指定することができます。

```
DSQSRUN=INITPROC (&VAR1 = value)
```

VAR1 とともに指定する必要があるアンパーサンドの数は、QMF の実行元が CICS であるか、TSO であるか、またはネイティブ z/OS バッチであるかに応じて異なります。この数は、ISPF が存在する場合、および QMF を開始するプログラムが REXX で記述されている場合は、異なります。

初期プロシージャーおよびリモート作業単位

初期プロシージャーは QMF を開始するシステム (ローカル・システム) に保管する必要があります。

初期プロシージャーで QMF CONNECT コマンドを使用するときには、リモート・ロケーションから切断する必要があります。つまり、アプリケーションをコード化して元のロケーションに再接続してからでないと、END コマンドをコード化して初期プロシージャーを再び呼び出すことができません。リモート・ロケーションに接続したままでは、エラーになります。初期プロシージャーでセットアップされた対話式セッション中に、コマンド行から QMF CONNECT を発行する場合も、切断する必要があります。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

57 ページの『START コマンドのキーワード』
START コマンドでキーワードを指定できます。

プロシージャー内の QMF CONNECT

別のユーザー ID またはリモート DB2 データベースに接続して、リモート作業単位サポートを使用するには、QMF CONNECT コマンドを発行します。このコマンドは、線形プロシージャーまたはロジックを持つプロシージャー内で使用できません。

QMF CONNECT コマンドを使用してリモート・データベースにアクセスするプロシージャーを作成する場合は、以下の状況に注意してください。

- リモート・データベースに接続して、RUN PROC コマンドを出す場合、そのプロシージャーおよびそのプロシージャーで使用するすべてのオブジェクトがリモート・データベースに保管されていなければなりません。
- プロシージャー内のすべての QMF コマンドは、QMF を実行中のシステム (ローカル・システム) の QMF 一時記憶域で実行されます。しかし、これらの QMF コマンドが使用するすべてのオブジェクト (例えば、照会、プロシージャー、または書式) は、現行ロケーション (リモート・システム) のデータベースに定義されていなければなりません。
- データベース (SQL ステートメント、QMF 照会、または EDIT TABLE の更新) に影響を与えるコマンドは、現行ロケーションで実行されます。
- プロシージャーに、システム固有のコマンド (CICS または TSO) が含まれている場合は、これらのコマンドは QMF が稼働しているシステム (ローカル・システム) で実行される。プロシージャーに、QMF を実行中のシステムで実行できないシステム固有のコマンドが含まれている場合は、そのプロシージャーを正常に実行できません。
- システム固有のコマンドで使用するすべてのデータ・セットまたはデータ・キューは、QMF を実行中のシステム (ローカル・システム) に存在していなければなりません。
- ご使用のサイトで TSO を使用し、大/小文字混合パスワードに対し RACF[®] サポートを利用している場合は、QMF プロファイルの CASE オプションを MIXED に設定してください。MIXED に設定しないと、QMF はすべての入力を大文字に変換するため、CONNECT コマンドが失敗します。QMF では大文字のコマンドのみが認識されるため、CASE=MIXED と設定した場合は必ず、QMF アプリケーション・ユーザーに対し、すべて大文字で入力するように指示してください。
- プロシージャーが、ストアード・プロシージャー・インターフェースを使用して QMF for TSO を開始する CALL ステートメントのパラメーターとして渡される場合、そのプロシージャーはリモート・データベースにアクセスできません。リモート・データベースにアクセスを試みるプロシージャーのコマンドは、プロシージャーがそのインターフェースを使用する前に、削除するか、コメント化する必要があります。

プロシージャー内の置換変数

線形プロシージャーおよびロジックを持つプロシージャーの中で、QMF 置換変数を使用できます。

置換変数は、QMF コマンドで使用できる任意の変数です。置換変数の前には必ずアンパーサンド (&) が置かれます。置換変数に値を割り当てるには、以下の方法があります。

- SET GLOBAL コマンドを使用して大域変数を設定
- RUN コマンドでの値の指定
- 「RUN コマンド・プロンプト」パネルでの値の指定。

関連資料:

54 ページの『SET GLOBAL (拡張構文)』

独自の大域変数を作成し、それらを QMF コマンドで置換変数として使用するには、SET GLOBAL コマンドを発行します。また、SET GLOBAL コマンドを使用すると、QMF の事前定義大域変数の値も設定できます。これは先頭が「DSQ」になります。

RUN コマンドでの値の指定

RUN コマンドを使用して、置換変数に値を割り当てるためには、次のようにします。

プロシージャーが線形プロシージャーの場合は、次のように、RUN PROC コマンドの変数値を割り当てます。

```
RUN PROC SCHEDULE (&&TYPE='VACATION')
```

プロシージャーがロジックを持つプロシージャーの場合、次のように、RUN PROC コマンドの変数値を割り当てます。

```
"RUN PROC SCHEDULE (&&TYPE='VACATION')"
```

&TYPE の値は、SCHEDULE と呼ばれるプロシージャーでのみ使用できます。

この例では、

- 変数値 VACATION は、値が文字ストリングなので、単一引用符で囲まれています。
- TYPE の前には 2 つのアンパーサンド (&&) が付いていて、RUN ステートメント上に設定されている値を SCHEDULE という名前のプロシージャーに渡すべきことを示しています。RUN ステートメントに &TYPE が指定されている場合には、このステートメントが入っているプロシージャーは、ユーザーに値を入力するようにプロンプトで指示します。

置換変数としてのこの値は、この値を定義しているプロシージャー内でのみ有効となります。この値は、定義するプロシージャーから呼び出されたプロシージャー、またはモジュールでは活動状態ではありません。

「RUN コマンド・プロンプト」パネルでの値の指定

置換変数を使用して照会またはプロシージャを実行するときには、RUN コマンドまたは大域変数で値を割り当てることができます。ただし、照会またはプロシージャ内の変数に値が含まれていない場合は、QMF によって RUN コマンド・プロンプト・パネルが表示されます。その後、このパネルで変数の値を指定できます。

置換変数としてのこの値は、この値を定義しているプロシージャ内でのみ有効となります。この値は、定義するプロシージャから呼び出されたプロシージャまたはモジュールでは、活動状態ではありません。

線形手順内の変数値の入力プロンプト

線形プロシージャでは、QMF は置換変数についてプロシージャを走査し、コマンドを処理する前に解決します。ユーザーは、プロシージャの実行前に、すべての変数について値を入力するようにプロンプト指示されます。

ロジックを持つプロシージャ内の変数値の入力プロンプト

ロジックを持つプロシージャでは、REXX が変数を含んでいるステートメントを検出するまで変数値の入力するようプロンプト指示されません。ユーザーに向けて 1 回のみプロンプトを出すには、変数の入力を求めるプロンプトを出すためのプロシージャを別個に実行します。

例えば、置換変数 LASTNAME および DEPT_NUM の入力を求めるプロンプトを 1 回のみ出すことができます。これらの変数は、ロジックを持つプロシージャ内で 2 行に分かれます。

```
/* This procedure runs two queries, displaying the report after each */
/* query has run. */

"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";
"INTERACT"
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";
```

図 3. ロジックを持つプロシージャで変数値の入力を求めるプロンプトを出す

以下の行を、コメント行の直後の、ロジックを持つプロシージャの先頭に追加します。

```
"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";
```

重要: ロジックを持つプロシージャは、少なくとも 1 つのコメント行で開始する必要があります。

このコマンドで、PROMPT_ME は次のような命令のない 1 コメント行を含むロジックを持つプロシージャです。

```
/* This procedure is a dummy procedure that provides prompting */
```

完全なプロシージャには、変数の入力を求めるプロンプトを出す PROMPT_ME プロシージャの RUN PROC コマンドが含まれています。

```

/* This procedure runs two queries, displaying the report after each */
/* query has run */

"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";
"INTERACT"
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";

```

図 4. 変数値の入力をプロンプト指示するロジックを持つプロシージャ

別の方法として、次の例のように SET GLOBAL コマンドを使用してプロシージャ内のすべての値の入力を一度にプロンプトで指示することもできます。

```
"SET GLOBAL (LASTNAME=&LASTNAME,DEPTNUM=&DEPT_NUM";
```

ロジックを持つプロシージャ内の REXX 変数

ロジックを持つプロシージャで REXX 変数を使用できます。これらの変数の値は、定義したプロシージャ内でのみ認識されます。

以下の方法によって、ロジックを持つプロシージャで REXX 変数を使用できます。

- SET GLOBAL コマンドを使用して REXX 変数を QMF 変数にコピーする。
- GET GLOBAL コマンドを使用して大域変数を REXX 変数にコピーする。
- REXX ステートメントで REXX 変数を使用する。

ロジックを持つプロシージャへの引数の受け渡し

ロジックを持つプロシージャの場合、QMF によって RUN PROC コマンドで ARG オプションを指定できます。このオプションを使用して、ロジックを持つプロシージャに引数または値を渡します。

次の例に示すように、REXX PARSE ARG または ARG ステートメントが入っているプロシージャを実行する場合、ARG オプションを使用してください。

```

PROC                                WILDE.SHOW_ARGS                                MODIFIED    LINE 1

/*****/
/* This procedure shows you how to use the 'ARG=' option on the RUN */
/* PROC command. */
/*****/
parse upper arg query_name form_name
"RUN QUERY" query_name "(FORM="form_name

```

図 5. RUN PROC コマンドの ARG オプションを使用した変数値の引き渡し

このプロシージャ用の RUN コマンドを次に示します。

```
RUN PROC SHOW_ARGS (ARG=(query_name form_name)
```

このコマンドで、*query_name* および *form_name* は、ロジックを持つプロシージャに渡されるパラメーターを記述する REXX 変数名です。照会および書式のオブジェクト名が含まれているこれらの変数を使用して、ロジックを持つプロシージャに渡されたパラメーターを参照します。

ロジックを持つプロシージャ内の REXX エラー処理ステートメント

REXX SIGNAL 命令などの REXX エラー処理技法をロジックを持つプロシージャで使用することができます。さらに、REXX EXIT 命令とともに QMF コマンドおよび変数を使用すると、ゼロ以外の戻りコードの解明に役立ちます。

エラー処理サブルーチンへの分岐

REXX SIGNAL ON ERROR 命令により、REXX は、非ゼロの戻りコードが検出されたときに、現在行からエラーのマークが付いたラベルへ分岐するように指示されます。

このステートメントには以下の 2 つの部分が必要です。

- SIGNAL ON ERROR 命令

コマンドが実行されるたびに REXX は、そのコマンドの戻りコードを *rc* と呼ばれる変数に入れます。

コマンドの戻りコードが非ゼロである場合、REXX はエラー ・ ラベルに分岐します。

SIGNAL ON ERROR を設定すると、QMF REXX プロシージャ (ADDRESS QRW) のコマンド環境からエラーが返されますが、REXX 呼び出し可能インターフェースからは返されません。

- エラー ・ ラベル

SIGNAL ON ERROR 命令を使用するときには、非ゼロの戻りコードが検出された場合にプロシージャが分岐できる先のラベルを指定する必要があります。このラベルの後に、ユーザーのエラー処理コードが続きます。戻りコードは、変数 *rc* に入っています。この変数を使用して、別のサブルーチンに分岐することができます。あるいは、次の例のように、この変数を EXIT 命令内で使用することができます。

```
/* error handling code for a procedure with logic */
error:
  exit rc
```

REXX EXIT ステートメントによるメッセージ

REXX EXIT 命令を使用して、ロジックを持つプロシージャを終了できます。

QMF は、ロジックを持つプロシージャの実行を終了するときに、必ずメッセージを出します。

EXIT 命令を使用する場合、発行されるメッセージは、下記の要因により決まりません。

- 最後の QMF コマンドがエラーを検出したかどうか
- 戻りコードがゼロかどうか

次の表に、発行されるメッセージとその条件を示します。

表 3. QMF から戻されるメッセージ

最後の QMF コマンドからの戻りコード	プロシージャー戻りコード	プロシージャー完了時のメッセージの例
0	0	OK。手順が実行されました
0	非ゼロ	プロシージャーからの戻りコードは 8
非ゼロ	0	QMF で提供されるエラー・メッセージ
非ゼロ	非ゼロ	QMF で提供されるエラー・メッセージ

QMF コマンドが正しくなく、かつ、戻りコードがゼロではない場合、QMF エラー・メッセージが戻りコード・メッセージより優先します。

最後のコマンドで発生したエラー・メッセージを表示し、終了して QMF 戻りコードを返すには、MESSAGE コマンドおよび EXIT DSQ_RETURN_CODE 命令を使用します。例えば、次のようにします。

```

:
"MESSAGE (TEXT='dsq_message_text'"
exit dsq_return_code

```

変数 dsq_message_text および dsq_return_code は、QMF から提供される REXX 変数です。MESSAGE コマンドおよび dsq_message_text 変数を使用すると、次の例に示すように、メッセージを保管でき、後から表示できます。

```

/* Monthly report */
signal on error
"DISPLAY TABLE JUNE_INFO"
"PRINT REPORT"
exit(0);
error:  original_msg = dsq_message_text
/* Saves error message. */
"RUN PROC GENERAL_RECOVERY"
/* This proc generates */
/* new dsq_message_text. */
"MESSAGE (TEXT=' original_msg '"
/* Display original error msg. */
exit dsq_return_code;

```

関連資料:

208 ページの『REXX 用のインターフェース連絡変数』

インターフェース連絡変数は、複数の REXX 変数で構成されます。これらの変数は、各呼び出しの完了後に設定され、呼び出し側プログラムで変更してはいけません。

51 ページの『MESSAGE』

アプリケーションを作成するときには、ユーザーに向けて表示する情報または次に実行するアクションに関する特定のメッセージをユーザーに送信することがよくあります。MESSAGE コマンドを使用すれば、独自のメッセージを作成して QMF パネルに表示することができます。ISPF では、ISPF エラー・メッセージに関するメッセージ・ヘルプを QMF で表示することも選択できます。

ロジックを持つプロシージャーからの REXX プログラムの呼び出し

置換変数を使用する場合と使用しない場合では、それぞれ異なる方法で REXX プログラムを呼び出します。

置換変数を含まない REXX プログラムの呼び出し

REXX プログラムに、置換変数を含む組み込み RUN コマンドがない場合は、プログラムを呼び出すために、ADDRESS 命令または CALL 命令を使用します。また、プログラムを関数として呼び出すこともできます。

以下のいずれかのコマンドを使用して、プログラムを呼び出します。

- ADDRESS 命令

この命令により、コマンド環境が設定されます。例えば、プログラムの名前が PANDA で、TSO 環境内から呼び出したい場合、コマンドは次のようになります。

```
ADDRESS TSO "PANDA"
```

- CALL 命令

この命令はプログラムを呼び出します。例えば、PANDA という名前のプログラムの場合、コマンドは次のようになります。

```
CALL PANDA
```

- 関数

PANDA という名のプログラムを、関数として呼び出すこともできます。

```
answer = PANDA()
```

いずれかの REXX 呼び出しを使用してプログラムを呼び出す場合、RUN コマンドから置換変数を除去することができます。その場合に QMF は、ユーザーに対して変数をプロンプトで指示します。

関連概念:

31 ページの『第 5 章 ADDRESS QRW および QMF コマンド環境』
QMF を TSO、ISPF、またはネイティブ z/OS で開始すると、QMF は QRW と呼ばれる REXX コマンド環境を作成します。REXX プログラムを実行している場合は、REXX ADDRESS コマンド ADDRESS QRW を発行することで、デフォルトのコマンド環境を QRW に設定できます。ADDRESS QRW を使用した場合、別の ADDRESS コマンドを出すまで、QMF はデフォルトのコマンド環境にとどまります。

置換変数を含む REXX プログラムの呼び出し

REXX アプリケーションに、置換変数を含む QMF RUN コマンドがある場合は、それを呼び出すために、TSO *program_name* コマンドを使用します。

ロジックを持つプロシージャーを実行する場合でも、またはプロシージャーによって呼び出される呼び出し可能インターフェース・プログラムを実行する場合でも、コマンドは同様に QMFに入ります。この文脈では、呼び出し可能インターフェース・プログラムはプロシージャー自体の論理的拡張となります。

例えば、次のようなコマンドを検討します。

```
RUN QUERY WEEKLY_Q (&DEPT=58)
```

ロジックを持つプロシージャーでは、照会に変数を渡すために置換変数に 2 つのアンパーサンドを使用します。

```
"RUN QUERY WEEKLY_Q (&&DEPT=58"
```

置換変数にアンパーサンドが 1 つしかない場合、QMF はその変数をそのプロシージャ自体のために解決し、照会に渡すことはできません。

ロジックを持つプロシージャから REXX 呼び出し可能インターフェース・アプリケーションを呼び出すときに、そのアプリケーションにコマンド RUN QUERY WEEKLY_Q (&DEPT=58 が含まれている場合は、QMF によって、呼び出し側プロシージャの場合と同様に変数が解決されます。アンパーサンドが 1 つしか使用されていないので、この変数は照会に渡されません。

ロジックを持つプロシージャによって呼び出された REXX 呼び出し可能インターフェース・アプリケーションから QMF に変数を渡す場合、以下の 3 つの選択肢があります。

- TSO コマンドを使用して、アプリケーションを呼び出す。

このようにアプリケーションを呼び出した場合、QMF は検出した置換変数を処理しません。上記の RUN QUERY コマンドの &DEPT=58 は、置換変数が解決された時点で照会に渡されます。

- アプリケーション内のすべての置換変数を、ロジックを持つプロシージャ内で使用しているかのように扱う。

すべての置換変数にアンパーサンドを加えて、ロジックを持つプロシージャがこれを解決しないようにします。

- 大域変数を使用する。

アプリケーションの開始時に大域変数を定義して、QMF セッションの間、これらを使用します。DSQEC_USERGLV_SAV 大域変数は、あるセッションから別のセッションまで大域変数値を保持するように設定できます。

関連資料:

240 ページの『コマンドとプロシージャの実行を制御する大域変数』

DSQEC 大域変数は、コマンドおよびプロシージャの実行方法を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

第 3 章 呼び出し可能インターフェースおよび QMF アプリケーション

それぞれのプログラム言語で、QMF コマンドを実行するために QMF 呼び出し可能インターフェースを使用することができます。

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

呼び出し可能インターフェースとは

QMF 呼び出し可能インターフェースは、様々なプログラム言語用の標準インターフェースを提供し、共通記憶域およびプログラム変数へのアクセスを提供します。

アプリケーション・プログラムは、QMF コマンドを実行する必要がある場合に、プログラムと QMF の間の通信を開始する必要があります。この通信を行うには、QMF インターフェース・ルーチンへの呼び出しを行います。QMF には、サポートする言語ごとにルーチンが用意されています。

アプリケーション・プログラムは、最初の START 呼び出しの後に 1 つまたは複数の QMF コマンドを出すことができます。アプリケーション・プログラムは、ルーチンを呼び出してから、各 QMF コマンドを出します。

QMF コマンドの処理が終了すると、QMF は QMF の状況を示す戻りコードを提供します。呼び出し可能インターフェースは、コマンドの処理に関する他の情報を収集し、この情報を QMF とアプリケーション・プログラムの両方からアクセス可能な変数に保管します。これらの変数は、変数プール またはインターフェース連絡域 にあります。呼び出し可能インターフェースが呼び出し側アプリケーション・プログラムに制御を返すときに、アプリケーションはこれらの変数を参照できますが、変更することはできません。

アプリケーション・プログラムは、QMF を使用する必要がなくなった場合、プログラムと QMF の間の通信を終了するための呼び出しを行います。この呼び出しは、QMF ルーチンに対して行います。

QMF 呼び出し可能インターフェースの使用に関する考慮事項

呼び出し可能インターフェースを使用したアプリケーションと QMF の間の制御の流れは、特定の規則に従います。

QMF 呼び出し可能インターフェースで使用するアプリケーション・プログラムを作成するときは、以下の点に留意してください。

- QMF に対する呼び出しの後は、QMF が QMF コマンドの処理を終了した場合のみ、制御が呼び出し側のアプリケーション・プログラムに戻されます。

- QMF は、呼び出しを処理していない間は、非アクティブ状態です。
- アプリケーション・プログラムと QMF は、戻りコードと、変数プールまたはインターフェース連絡域に保管されている変数データを使用して通信します。
- すべての QMF コマンドは、英大文字でコーディングする必要があります。

QMF の各国語機能 (NLF) を使用している場合、QMF コマンドは、主要 NLF 言語で作成する必要があります。また、コマンドは大文字で書くか、QMF によって大文字に変換する必要があります。QMF プロファイルの CASE オプションが UPPER に設定されている場合、コマンドは QMF によって大文字に変換されます。QMF を始動するときに、START コマンドの DSQALANG パラメーターに値を指定して、主要言語を設定します。この値は、DSQEC_NLFCMD_LANG 大域変数に記録されています。

- 受け渡すコマンドの最大長は、REXX プログラムの場合は 2,048 バイト、それ以外の言語の場合は 32,768 バイトです。

次の図は、アプリケーションが呼び出し可能インターフェースを介して QMF にコマンドを渡す方法を示しています。

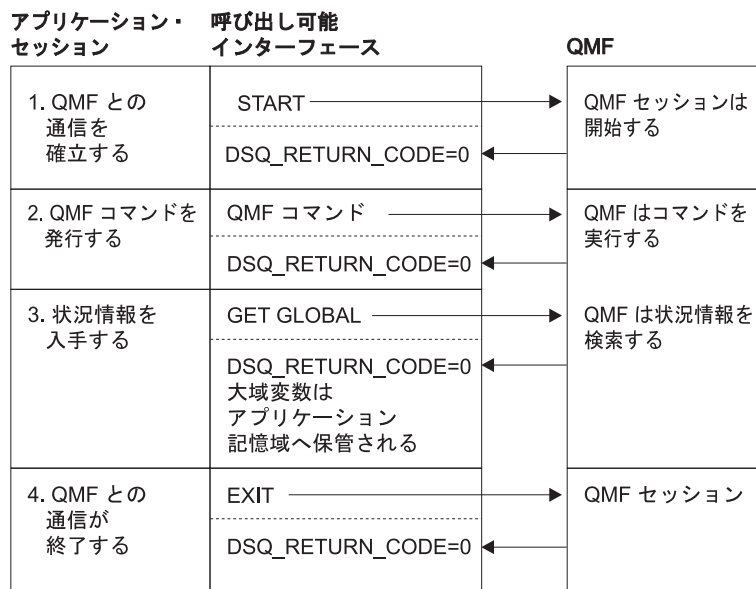


図 6. アプリケーションが QMF 呼び出し可能インターフェースを使用して、QMF と連絡する方法

関連資料:

57 ページの『START コマンドのキーワード』
START コマンドでキーワードを指定できます。

インターフェース連絡域 (DSQCOMM)

QMF には、サポートしているプログラム言語ごとにインターフェース連絡域が用意されています。この領域には、戻りコードと理由コードの定義、および QMF に対する関数呼び出しの定義が含まれています。

インターフェース連絡域は、インターフェース連絡変数の記憶域を定義します。この領域に保管されている変数には、QMF と呼び出し可能インターフェース・アプリ

セッションの両方からアクセスできます。ただし、QMF でのみ値を変更できるようにしてください。アプリケーション・プログラムではこれらの変数が読み取り専用として扱われることを確認してください。

REXX 呼び出し可能インターフェースは、連絡域を使用するのではなく、QMF が提供するインターフェース連絡変数を使用します。

QMF 呼び出し可能インターフェース連絡域は、すべての呼び出し可能インターフェース呼び出しに必要です。呼び出し可能インターフェース連絡域用の記憶域は、QMF を使用するプログラムが割り振ります。

START コマンドは、QMF セッションの固有のインスタンスまたはオカレンスを確立します。START コマンドは、以下の条件のもとで 1 つの QMF セッションしか確立できません。

- TSO アドレス・スペース内
- 単一の CICS トランザクションから

START コマンドの実行中に、QMF はインターフェース連絡域内で変数を更新します。

これらの変数は、アプリケーション・プログラムが変更してはなりません。ただし、以下の例外があります。

DSQ_COMM_LEVEL

DSQ_COMM_LEVEL を DSQ_CURRENT_COMM_LEVEL の値に設定して、DSQCOMM のレベルを識別します。この例外は、REXX には適用されません。

DSQ_INSTANCE_ID

QMF 内から呼び出し可能インターフェース・プログラムを呼び出す場合は、最初の呼び出しで DSQ_INSTANCE_ID をゼロ (0) に設定する必要があります。この設定を行うと、QMF によって、最初の START コマンドで設定された値に変数がリセットされます。

START コマンドに続くすべての呼び出しで、QMF インスタンスに対応するインターフェース連絡域のアドレスを渡す必要があります。アプリケーション・プログラムは、正しいインターフェース連絡域を指していなければなりません。

サポートされているプログラミング言語にはそれぞれ、固有のインターフェース連絡域があります。アプリケーション・プログラムは、移植可能であるためには、値ではなく変数名によって変数を参照する必要があります。値は、システムによって異なる可能性があるからです。

インターフェース連絡域内の変数には、次の表に示す情報が格納されます。

表 4. 変更してはならない DSQCOMM フィールド

変数により提供される情報	説明
戻りコード	QMF がコマンドを処理した後の、QMF 処理の状況を示す
インスタンス ID	START コマンドによって開始された QMF のインスタンスを示す

表 4. 変更してはならない DSQCOMM フィールド (続き)

変数により提供される情報	説明
完了メッセージ ID	QMF が表示するメッセージのメッセージ ID が入る このフィールドは、各 QMF コマンドの完了時に設定される。このフィールドには、コマンドの終了時に QMF が表示するメッセージが入る。
照会メッセージ ID	RUN QUERY コマンドの完了時に照会パネルに表示されるメッセージのメッセージ ID が入る このフィールドは、照会の実行時にエラーが発生すると設定される。このフィールドには、コマンドの終了時に照会オブジェクト内で QMF が表示するメッセージが入る。
エラーがある START コマンドのパラメーター	パラメーターのエラーによって START コマンドが失敗した場合に、エラーがあるパラメーターの名前が入る
取り消し標識	QMF がコマンドを実行中に、ユーザーが処理を取り消したかどうかを示す
完了メッセージ	QMF が表示する完了メッセージが入る
照会メッセージ	RUN QUERY コマンドの完了時に照会パネルに表示されるメッセージの照会メッセージ・テキストが入る 例えば、エラーがある照会オブジェクトを実行すると、QMF は、照会の実行を妨げたエラーについて記述するメッセージを表示する。照会メッセージ・フィールドには、このエラー・メッセージ・テキストが入る。

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

戻りコード

QMF 呼び出し可能インターフェースに対するそれぞれの呼び出しの後に、戻りコードが戻されます。戻りコード値は、QMF と一緒に出荷されるインターフェース連絡域によって記述されます。

戻りコードの値は、システムによって異なる可能性があります。アプリケーションをシステム間で移植可能にする場合、アプリケーションでは、これらのコードの値を変数名で参照する必要があります。インターフェース連絡域内の戻りコード変数の名前は、プログラミング言語の指定に記載されています。

この表は、呼び出し可能インターフェースの状態についての可能性のある戻りコードを示しています。

表 5. 呼び出し可能インターフェースの戻りコード

値	説明
0	実行は正常に終了した
4	QMF セッションは EXIT または END コマンドによって終了とマークされた
8	実行は失敗したが、セッションはエラーによって終了とマークされなかった
16	重大エラー、セッションは終了とマークされた

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

呼び出し可能インターフェースを使用するためのコマンド

呼び出し可能インターフェースを使用して、プロシージャ内で使用する任意の QMF コマンドを発行できます。ただし、一部のコマンド (START、GET GLOBAL、SET GLOBAL、および TRACE) では、呼び出し可能インターフェースの構文は特殊です。

START コマンドと TRACE コマンドは、呼び出し可能インターフェースでしか機能しません。REXX 以外の言語で作成した呼び出し可能インターフェース・アプリケーションで GET GLOBAL および SET GLOBAL を使用するには、これらのコマンドに拡張構文 を使用してください。

プログラミング言語の START コマンドおよび SET GLOBAL コマンドの例については、その言語の指定を参照してください。

関連概念:

45 ページの『第 8 章 アプリケーションでの QMF コマンド』

特定のコマンドはアプリケーション内で使用するよう設計されています。また、独自のコマンド同義語を作成することができます。

関連資料:

47 ページの『GET GLOBAL (拡張構文)』

GET GLOBAL コマンドを使用すると、アプリケーションの QMF 大域変数にアクセスできます。REXX 以外の言語のために、QMF は GET GLOBAL コマンドの拡張構文を用意しています。

54 ページの『SET GLOBAL (拡張構文)』

独自の大域変数を作成し、それらを QMF コマンドで置換変数として使用するには、SET GLOBAL コマンドを発行します。また、SET GLOBAL コマンドを使用すると、QMF の事前定義大域変数の値も設定できます。これは先頭が「DSQ」になります。

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

アプリケーションからの QMF の開始

アプリケーションから他のすべてのコマンドを実行する場合には、まずその前に QMF を開始する必要があります。呼び出し可能インターフェースを使用している場合、アプリケーションで START コマンドを出して QMF を開始します。QMF セッションは、一度に 1 つしか開始できません。

アプリケーションは、QMF が開始されているかどうかをテストするために START コマンドを出すことができます。QMF は、開始されていなければ、開始されます。QMF が開始されている場合、非ゼロの戻りコードが返され、以下のメッセージ番号およびメッセージが表示されます。

DSQ50720 QMF already active; secondary session not permitted.

START コマンドの結果が重大でないエラー (戻りコードが 4 または 8) の場合は、QMF が開始し、セッションが確立されます。この場合、QMF EXIT コマンドを出して QMF を停止する必要があります。エラーの原因については、インターフェース連絡域の内容または、QMF トレース・データ出力を調べてください。

QMF にパラメーターを渡すには、START コマンドで必要なコマンド・キーワードを指定します。

関連資料:

56 ページの『START』

呼び出し可能インターフェースを介して QMF を開始する場合、START コマンドを使用する必要があります。

呼び出し可能インターフェース・アプリケーションの実行

呼び出し可能インターフェース・アプリケーションを実行する場合は、対話式で QMF を実行する場合と同様に環境をセットアップする必要があります。

環境のセットアップおよび呼び出し可能インターフェース・アプリケーションのコンパイルと実行については、該当する言語の指定にあるコーディング・サンプルを参照してください。

関連資料:

153 ページの『第11章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

QMF 内での呼び出し可能インターフェース

ユーザー・プログラムから QMF オブジェクトを変更する必要がある場合は、QMF 内から呼び出し可能インターフェースを使用できます。例えば、対話式 QMF セッション中に呼び出し可能インターフェースを介して、オブジェクトをエクスポートまたはインポートできます。TSO コマンドを使用してアプリケーションを呼び出すことで、QMF 内から呼び出し可能インターフェースを使用できます。アプリケーションからは、任意の有効な QMF コマンドを実行できます。

制約事項: CICS 環境にいる間は、QMF 内から呼び出し可能インターフェースを使用することはできません。

QMF を最初に呼び出す前に、DSQCOMM インスタンス ID (DSQ_INSTANCE_ID) をゼロ (0) に設定する必要があります。QMF は、現在のインスタンスを判別し、以後の QMF 呼び出しで使用するために DSQ_INSTANCE_ID を更新します。

エラー処理

QMF コマンドが完了するたびに、必ず DSQCOMM 連絡域の dsq_message_text 変数にメッセージ・テキストが入り、dsq_return_code 変数には戻りコードが入ります。

戻りコードには、次の値のいずれかが割り当てられています。

dsq_success

コマンドが正常に終了

dsq_warning

警告を伴って正常に終了

dsq_failure

コマンドが正しく実行されなかった

dsq_severe

重大エラー、QMF セッションは終了

各 DSQCOMM 域の変数およびフィールドは、プログラミング言語の指定に記載されています。

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

CICS のもとでの呼び出し可能インターフェース・プログラムの実行

QMF 呼び出し可能インターフェースを使用するプログラムを実行するには、CICS プログラムの通常のインストール方法を使用して、CICS にこれらのプログラムをインストールしてください。

通常の CICS 要件の他に、CICS で実行するすべての QMF 呼び出し可能インターフェース・プログラムには以下の考慮事項が適用されます。

環境 プログラムは、QMF プロダクトを呼び出すと、対話式 QMF プロダクトと同じ特性を持つようになります。すなわち、大きな会話型プログラムになります。

QMF は、CICS コマンドが入っているアセンブラー言語プログラムです。これは、他のアセンブラー言語プログラム、または呼び出し可能アプリケーション・プログラミング・インターフェースでサポートされるプログラムでリンク・エディットできます。高水準言語を使用して QMF を呼び出す場

合、まずプログラムをリンク・エディットする必要があります。また、オンライン・リソース定義 (RDO) プログラムの定義によってその高水準言語を指定する必要があります。高水準プログラムごとに、CICS に関する特定の考慮事項と制約があります。

CICS では、任意のデフォルトの QMF プログラム・パラメーターをオーバーライドする場合は、START コマンドでオーバーライド値をパラメーターとして指定する必要があります。例えば、呼び出し可能インターフェースからの操作のデフォルト・モードは batch モードです。対話式 QMF セッションを実行する場合には、DSQSMODE=I を使用して、START コマンドを出す必要があります。

CICS 領域に関する考慮事項

QMF インターフェース連絡モジュールおよびメイン QMF モジュールが入っているユーザー・プログラムは、同じ領域または区画で実行する必要があります。QMF インストール時に記述する QMF リソースも、QMF を実行する CICS の領域または区画に割り振る必要があります。

データベース

プログラムを呼び出す CICS トランザクションは、DB2 に記述しておく必要があります。

関連概念:

4 ページの『QMF へのアプリケーション・プログラミング・インターフェース』
QMF へ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は、呼び出し可能インターフェースとコマンド・インターフェースの 2 つがあります。

第 4 章 ISPF ダイアログからの QMF コマンドの発行

QMF で実行されている ISPF ダイアログから QMF コマンドを発行するには、QMF コマンド・インターフェースを使用します。

制約事項: QMF コマンド・インターフェースには ISPF の実行が必要ですが、ISPF は CICS 環境では実行しません。したがって、CICS のもとでのアプリケーション開発には、QMF 呼び出し可能インターフェースを使用する必要があります。

QMF は次の図のように、QMF コマンド・インターフェースを使用して、ISPF 変数プール経由でダイアログと通信します。

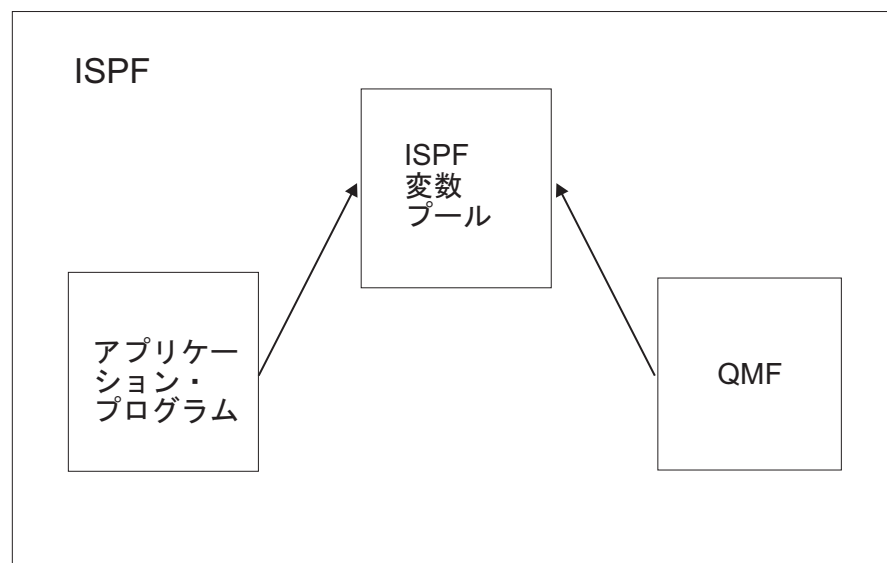


図 7. QMF と対話する QMF コマンド・インターフェース・アプリケーション

コマンド・インターフェースを効果的に使用するには、ISPF サービスおよび変数プールについての理解も必要です。

コマンド・インターフェースを使用するプログラムの作成: 例

この例では、ユーザーに照会名の指定、指定した照会の実行、および報告書の表示を求める、ISPF パネルを表示するプログラムを作成します。

手順

1. コマンド・インターフェース REXX プログラムを作成する。
 - a. DISPLAY サービスを使用して ISPF パネルを表示する。この例では、パネル名は QRYNAME です。

```
ADDRESS ISPEXEC "DISPLAY PANEL(QRYNAME)"
```
 - b. 直前の DISPLAY サービスからのユーザー入力に基づいて、QMF 照会を実行する。ISPF 変数 QNAME には、QMF 照会の名前が入っている。

```
ADDRESS ISPEXEC "SELECT PGM(DSQCCI) PARM(RUN QUERY" QNAME ")"
```

c. 照会の結果を表示する。

```
ADDRESS ISPEXEC "SELECT PGM(DSQCCI) PARM(INTERACT)"
```

2. QMF コマンド行から TSO コマンドを使用して、プログラムを呼び出す。例えば、プログラムの名前が GETINFO である場合、以下のコマンドを使用します。

```
TSO GETINFO
```

コマンド・インターフェースの呼び出し

コマンド・インターフェースは DSQCCI という名前のプログラムです。このプログラムは、ISPF SELECT サービスを介してプログラムから呼び出すことができます。

コマンド・インターフェースを開始するには、まず ISPF を開始します。その後、ISPF SELECT サービスを使用して QMF を開始し、QMF コマンド・インターフェース (DSQCCI) を呼び出します。ISPF SELECT PGM コマンドの PARM オプションを使用して、QMF コマンドを渡すことができます。

コマンド・インターフェースを開始した後は、ISPF SELECT PGM コマンドの PARM パラメーターを使用して、実行したい QMF コマンドを渡します。

```
SELECT PGM(DSQCCI) PARM(qmf_command)
```

コマンド・インターフェースに対するパラメーターとして指定するすべての QMF コマンドは、QMF プロファイルの設定値に関係なく大文字でなければなりません。ISPF では、コマンドが小文字から大文字に変換されません。QMF コマンドを小文字で指定すると、QMF で認識されません。

ISPF アプリケーションから QMF コマンド・プロンプトを使用するには、INTERACT コマンドを発行します。INTERACT コマンドの後に、プロンプトを表示させる QMF コマンドを続けて、そのステートメントの最後に疑問符 (?) を入力します。例えば、RUN QUERY プロンプト・パネルを表示するには、次のコマンドを発行します。

```
SELECT PGM(DSQCCI) PARM(INTERACT RUN QUERY ?)
```

SELECT サービスを使用するには、RUN QUERY コマンドで 2 つのアンパーサンドを使用する必要があります。2 つのアンパーサンドを使用すると、ISPF でその変数が ISPF 自体のものであると解釈されません。

呼び出し時に、NEWPOOL および NEWAPPL オプションを指定してはなりません。NEWPOOL および NEWAPPL オプションを省略すると、コマンド・インターフェースは、アプリケーションの変数に確実にアクセスできます。コマンド・インターフェースは、QMF とアプリケーションの間の通信に共用プールを使用します。

END コマンド

コマンド・インターフェースの実行中に、END コマンドを実行すると、DSQCCI プログラムが終了し、制御が呼び出し側アプリケーションに戻ります。QMF セッションはアクティブのままです。

QMF では、EXIT コマンドが発行された場合、またはコマンド・インターフェース呼び出し中に重大エラーが発生した場合に、セッションに終了のマークを付けるために、大域変数 DSQCSESC が設定されます。DSQCCI を呼び出したプログラムが終了し、制御が QMF に戻ると QMF セッションが終了します。

コマンド・インターフェースでの変数

STATE コマンドを実行すると、QMF から提供された各変数の現行値が返されます。STATE コマンドは、コマンド・インターフェースでのみ使用できます。QMF 変数は、VPUT コマンドを使用して QMF 変数を ISPF 変数プールに配置することができます。

関連資料:

217 ページの『付録 B. QMF 大域変数』

QMF には、QMF セッションのさまざまな特性や、QMF コマンド、およびパネル表示を制御するために役立つ、多数の大域変数があります。大域変数を使用すると、プロシージャおよびアプリケーションでの QMF 関数の動作を制御することもできます。

コマンド・インターフェースの戻りコード

コマンド・インターフェースの戻りコードは正数またはゼロです。ゼロという値は、正常な実行を示します。正の値は、実行が失敗したか、異常終了したことを示します。戻りコードは、アプリケーションの言語に関係なく同じです。

戻りコードは、ユーザーの EXEC または CLIST 内の変数に表示されます。REXX exec を実行すると、戻りコードは RC と呼ばれる REXX 変数に置かれます。CLIST を実行すると、戻りコードは CLIST 変数 &LASTCC に入れられます。

次の例は、戻りコードを調べる EXEC を示しています。この例では、REXX 変数 RC を使用して、エラーがあるか照会およびテストを実行する方法を示しています。

```
ADDRESS ISPEXEC SELECT PGM(DSQCCI) PARM(RUN QUERY (FORM=FORMA))
Select
  When (RC = 0) Then nop
  When (RC = 64) Then
    Say "You must run QMF with ISPF to use command interface."
  When (RC = 100) Then
    Say "You need to start QMF before you begin your application"
  Otherwise
    Say "Unexpected error ("RC") from QMF command interface."
End
```

エラー処理用のコードは、EXEC または CLIST だけでなく、プログラム・モジュールにも置くことができます。

戻りコード 0 から 16

戻りコード 0 ~ 16 は、コマンド・インターフェースによって渡されたコマンドの QMF 処理について記述します。コマンド・インターフェースは、コードだけでなく、QMF コマンド・メッセージ変数の値も、アプリケーションの ISPF 共用プールに返します。

表 6. 戻りコード 0 から 16

値	説明
0	実行は正常に終了した
4	QMF セッションは EXIT または END コマンドによって終了とマークされた
8	実行は失敗したが、セッションはエラーによって終了とマークされなかった
16	重大エラー、セッションは終了とマークされた

戻りコード 4 は、セッションを終了とマークする原因となったコマンドによるのみ戻されます。次にアプリケーションが別のコマンドの実行を試みると、QMF は別の戻りコード値をアプリケーションに戻します。

20 以上の戻りコード

これらのコードは、通常、コマンド・インターフェース (DSQCCI) でのなんらかの失敗を示しています。この失敗によって、インターフェースは変数をアプリケーション共用プールにコピーできなくなります。その結果、QMF 変数は無効であるか、または設定されていない可能性があります。プログラムが STATE コマンドを使用している場合は、STATE 変数についても同じ結果となる可能性があります。(変数は、アプリケーション共用プールにコピーされる場合に、設定されます。)

これらの戻りコードは通常、0 から 16 までの範囲の戻りコードより重大なエラーを示します。そのうちのいくつかの戻りコードについては、IBM ソフトウェア・サポートのサービスが必要となる場合があります。

この表内の共用変数は、QMF 変数 (および現行コマンドが STATE コマンドである場合は STATE 変数) を指します。いくつかのコードは、コマンドは実行されたが共用変数が設定されなかったことを示しています。QMF は STATE コマンドを適切に実行しましたが、コマンド・インターフェースは、更新された共用 QMF 変数および STATE 変数の設定に失敗しました。この失敗の理由については、エラー・コードの説明を参照してください。

表 7. 20 以上の戻りコード

値	説明
20	ユーザー出口ルーチンがコマンド・インターフェースを呼び出した。これらの呼び出しは、常に無効である。コマンド・インターフェースに渡されたコマンドは実行されていない。共用変数は設定されていない。
24	ISPF VCOPY コマンドでエラーが発生した。コマンド・インターフェースに渡されたコマンドは実行された。共用変数は設定されていない。
32	ISPF VREPLACE コマンドでエラーが発生した。コマンド・インターフェースに渡されたコマンドは実行された。共用変数は設定されていない。

表 7. 20 以上の戻りコード (続き)

値	説明
36	ISPF VPUT コマンドでエラーが発生した。コマンド・インターフェースに渡されたコマンドは実行された。共用変数は設定されていない。
40	ISPF VREPLACE コマンドでエラーが発生した。このコードは、STATE コマンドの実行にのみ適用される。コマンド・インターフェースに渡されたコマンドは実行されたが、共用変数は設定されていない。
44	ISPF VPUT コマンドでエラーが発生した。このコードは、STATE コマンドの実行にのみ適用される。 QMF 変数は設定されていたが、STATE 変数が設定されていなかった。
60	コマンド・インターフェースに無効な呼び出しが行われた。コマンド・インターフェースに渡されたコマンドは実行されていない。共用変数は設定されていない。
64	このエラーは、DSQCCI が実行されたが、ISPF がアクティブでない場合に発生する。例えば、ユーザーが ISPF SELECT PGM コマンドを使用せずに DSQCCI を呼び出した場合など。
100	このエラーは、QMF がアクティブでないときに、アプリケーションが QMF コマンドを実行しようとしたときに発生する。アプリケーションを実行する前に、QMF を開始する。コマンド・インターフェースに渡されたコマンドは実行されていない。共用変数は設定されていない。
104	アンカーが見つからない。コマンド・インターフェースに渡されたコマンドは実行されていない。共用変数は設定されているが、無効である。

第 5 章 ADDRESS QRW および QMF コマンド環境

QMF を TSO、ISPF、またはネイティブ z/OS で開始すると、QMF は QRW と呼ばれる REXX コマンド環境を作成します。REXX プログラムを実行している場合は、REXX ADDRESS コマンド ADDRESS QRW を発行することで、デフォルトのコマンド環境を QRW に設定できます。ADDRESS QRW を使用した場合、別の ADDRESS コマンドを出すまで、QMF はデフォルトのコマンド環境にとどまります。

制約事項: ADDRESS QRW は CICS 環境ではサポートされません。

また、次のように REXX ADDRESS の後に QMF コマンドを発行することで、単一コマンドを QRW 環境で実行するように指示できます。

```
ADDRESS QRW qmf_command
```

この場合、QMF は、ADDRESS QRW ステートメントが後に続くコマンドのみのコマンド環境です。

ロジックを持つ QMF プロシージャを使用している場合、QRW はデフォルトのコマンド環境になります。

次の例は、QMF コマンド環境の使用方法を示しています。

```
⋮  
call dsqcix "START (DSQSMODE=INTERACTIVE"  
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure  
  then exit dsq_return_code  
  
ADDRESS QRW  
"RUN PROC MONDAY_P"  
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure  
  then exit dsq_return_code  
  
"EXIT"  
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure  
  then exit dsq_return_code  
⋮
```

第 6 章 ISPF サービスを使用する QMF アプリケーションの作成

独自のユーザー・インターフェースを持つアプリケーションを作成することで、「QMF」パネルをう回することができます。呼び出し可能インターフェースまたはコマンド・インターフェースのいずれかを使用して、ISPF を使用するアプリケーションを作成できます。

制約事項: ISPF は CICS 環境で実行できないので、ISPF サービスを CICS のもとで使用することはできません。

関連概念:

17 ページの『第 3 章 呼び出し可能インターフェースおよび QMF アプリケーション』

それぞれのプログラム言語で、QMF コマンドを実行するために QMF 呼び出し可能インターフェースを使用することができます。

25 ページの『第 4 章 ISPF ダイアログからの QMF コマンドの発行』
QMF で実行されている ISPF ダイアログから QMF コマンドを発行するには、QMF コマンド・インターフェースを使用します。

ISPF アプリケーションからの QMF の開始および実行

ISPF アプリケーションから呼び出し可能インターフェースを使用して QMF を開始および実行するときには、特定の規則に従う必要があります。

ISPF を使用する呼び出し可能インターフェース・アプリケーションを作成する場合は、以下の要件に従う必要があります。

- 呼び出し可能インターフェース・アプリケーションは ISPF ダイアログの言語と一致しなければならない。

例えば、ISPF ダイアログが PL/I プログラムの場合は、PL/I 用の QMF 呼び出し可能インターフェースを使用する必要があります。

- 正しい各国語 ID を使用しなければならない。

DSQ n という ID を使用して ISPF アプリケーションを開始する必要があります。 n は各国語機能 (NLF) ID です。このアプリケーション ID によって、QMF は ISPF 環境 (ファンクション・キーの設定やラベルなど) を指定変更できなくなります。また、この ID によって、QMF が開始された後でも、ISPF 環境はそのまま保持されます。例えば、このステートメントによって PL/I プログラム MYPROG が開始され、このプログラムによって、呼び出し可能インターフェースの START コマンドを使用して QMF が開始されます。

```
SELECT PGM(MYPROG) NEWAPPL(DSQ $n$ )
```

- 変数の値を設定および取り出すには、STATE コマンドの代わりにアプリケーションで GET GLOBAL または SET GLOBAL コマンドを使用します。

STATE コマンドは、状態情報が含まれている変数に対してのみ機能します。
GET GLOBAL コマンドおよび SET GLOBAL コマンドは、すべての QMF 大域変数に対して機能します。ただし、読み取り専用の大域変数を設定することはできません。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

57 ページの『START コマンドのキーワード』

START コマンドでキーワードを指定できます。

218 ページの『プロファイル関連でない状態情報の大域変数』

DSQAO 大域変数には、状況情報、またはパラメーターやフラグの設定が格納されます。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

224 ページの『プロファイル関連状態情報の大域変数』

DSQAP 大域変数は、QMF プロファイル設定に関連する情報を保管します。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

変数を含む照会の実行

ISPF サービスを使用するアプリケーションは、変数を含む照会を実行できます。

以下の 3 つのうちのいずれかの方法で、ISPF サービスを使用するアプリケーションからこのような照会を実行することができます。

- ISPF ファイル調整サービスを使用する。

この技法を使用する場合、ISPF ファイル調整スケルトンによって照会を表します。このスケルトンでは、変更可能な照会の部分が ISPF ダイアログ変数として表示されます。これらの変数に適切な値を指定したら、プログラムは特定の ISPF ファイル調整サービスを開始します。結果は、照会が入っている順次ファイルとなります。

次に、プログラムは、この照会を QMF 一時記憶域にインポートし、QMF に実行させます。必要な IMPORT および RUN コマンドは、呼び出し可能インターフェースまたはコマンド・インターフェースを介して実行することができます。

この技法を使用するには、プログラムで ISPF ダイアログ・サービスを使用して ISPF ダイアログ変数を定義する方法を知っている必要があります。

- プログラム開発機能 (PDF) エディターを使用して QMF オブジェクトを作成する。

PDF 編集マクロで PDF エディターを使用して、照会、プロシージャ、書式、およびプロファイルに対するデータ入力項目を設計して制御することができます。REXX を使用して、PDF マクロを作成できます。

- ISPF ダイアログを使用して照会を作成する。

プログラムでは、ISPF 表示サービスを使用して画面を表示し、ユーザーからの入力に基づいてファイルを作成できます。このファイルに SQL 照会が格納されたら、ファイルが QMF にインポートされて実行されます。

QMF 内から ISPF サービスを使用するプログラムの開始

QMF 内から ISPF プログラムを開始する場合は、線形手順またはロジックを持つプロシージャからプログラムを呼び出す必要があります。

プログラムを呼び出すには、プロシージャに次のコマンドを組み込むことで、ISPF SELECT PGM サービスを使用します。

```
ADDRESS ISPEXEC "SELECT PGM(programname)"
```

プログラムを ISPF ダイアログ関数として実行していることを ISPF に伝えるために、CMD キーワードを使用します。このコマンドの構文は次のとおりです。

```
ADDRESS ISPEXEC "SELECT CMD(cmdname)"
```

または

```
ADDRESS ISPEXEC "SELECT CMD(cmdname parameters)"
```

このステートメントで、*cmdname* は呼び出し可能インターフェース・コマンドの名前です。

ロジックを持つプロシージャ内の ISPF サービス

ISPF のもとで実行されるロジックを持つ QMF プロシージャから ISPF コマンドを実行するには、QMF プログラム・ダイアログから ISPF コマンド・ダイアログに移動する必要があります。

正しい ISPF 環境を設定し、ISPF コマンドが入っているプログラムを実行するには、次に示すように、CMD キーワードを指定して ISPF SELECT CMD ステートメントを使用してください。

```
ADDRESS ISPEXEC "SELECT CMD(userprogram)"
```

このステートメントで、*userprogram* は ISPF コマンドが入っているプログラムです。

例えば、ISPF コマンドが入っているプログラムの名前が DIALOG である場合、ロジックを持つプロシージャに次のコマンドを組み込んでください。

```
ADDRESS ISPEXEC "SELECT CMD(DIALOG)"
```

また、QMF TSO コマンドを使用して、ISPF コマンドを含むプログラムを実行できます (例: TSO DIALOG)。この場合、QMF が ISPF SELECT CMD ステートメントを出してくれます。

ISPF で QMF を実行するときには、ISPF サービスを必要とするプログラムを開始するロジックを持つプロシージャで、ISPF SELECT CMD 環境を使用する必要があります。例えば、ISPF のもとで QMF を実行中であり、ロジックを持つプロシ

ージャーが DB2 コマンド DSN を発行するとします。DSN コマンドは ISPF サービスを使用するため、以下のいずれかのコマンドを使用して DSN コマンドを発行します。

```
ADDRESS ISPEXEC "SELECT CMD(DSN)"
```

または

```
ADDRESS ISPEXEC "SELECT CMD(DSNEXEC)"
```

2 番目のステートメントで、DSNEXEC は ADDRESS TSO DSN ステートメントを含むプログラムです。

ISPF での EDIT コマンド

ISPF 下で QMF アプリケーションを実行するとき、QMF SQL 照会またはプロシージャーを編集するには、EDIT QUERY コマンドまたは EDIT PROC コマンドを使用します。

PROC パネルまたは QUERY パネルから QMF EDIT コマンドを発行する場合は、PROC または QUERY のオブジェクト・タイプを指定する必要はありません。EDIT コマンドは、各パネルから呼び出された際に、その値を想定します。デフォルトで、QMF EDIT コマンドはプロシージャーまたは照会を PDF エディター・セッションの状態にします。QMF は、QMF アプリケーション ID の DSQ n を使用して PDF エディターを開始します。 n は NLF ID です。さらに、QMF はファンクション・キーおよびコマンド行の場所を QMF アプリケーションに一致するように設定します。

デフォルト・エディターをオーバーライドするには、次のように EDIT QUERY および EDIT PROC コマンドを使用してください。

```
EDIT QUERY (E=name)  
EDIT PROC (E=name)
```

これらのステートメントでは、*name* に以下のいずれかの値を設定できます。

- 使用可能なエディター
- デフォルト以外のアプリケーション ID を指定する REXX プログラムの名前。デフォルトのアプリケーション ID は、DSQ n (n は使用している NLF の各国語 ID) です。

QMF で提供されるキーとは異なるファンクション・キーを使用するには、QMF のデフォルトのアプリケーション ID とは異なるアプリケーション ID を使用します。

PDF PROFILE データ・セット・メンバーを必要とする PDF EDIT オプションを使用する場合、このようなメンバーを作成する必要があります。例えば、PDF EDIT RECOVERY オプションには、DSQ n EDRT PROFILE データ・セット・メンバー (n は適切な NLF 文字) が必要です。このコマンドを発行する前に、NLF がインストールされていることが必要です。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語

には、各国語機能 (NLF) が提供されます。

ISPF およびアプリケーションのデバッグ

QMF トレース機能は、様々な詳細レベルで QMF アクティビティをトレースするのに役立ちます。ISPF を使用するアプリケーションをより効率的にデバッグするには、ISPF ログ・サービスおよび PDF ダイアログ・テスト・サービスも使用します。これらのサービスは、QMF トレース機能を補完します。

ISPF ログ・サービス

ISPF ログ・サービスを使用して、メッセージを ISPF ログ・ファイルに書き込みます。例えば、REXX では、メッセージを ISPF ログに書き込むための ISPF コマンドは、次のようになります。

```
ADDRESS ISPEXEC LOG MSG (message-id)
```

このステートメントで、*message-id* はメッセージ・ライブラリーから取り出されてログに書き込まれるメッセージの ID コードです。

PDF ダイアログ・テスト・サービス

サイトに PDF がある場合は、PDF ダイアログ・テスト・サービスを使用して、ISPF アプリケーション・サービス呼び出しを ISPF ログ・ファイルに記録できます。さらに、PDF ダイアログ・テスト・サービスのログ・オプションを使用して、ログ・ファイルまたはデータ・セットの内容をブラウズできます。ISPF を終了するときに、ログ・ファイルまたはデータ・セットを印刷することもできます。

ダイアログ・テスト・サービスには、アプリケーションをデバッグするための他の多くの有用なオプションがあります。例えば、対話式にデバッグできます。アプリケーションのすべてまたは一部を実行し、結果を調査し、アプリケーションを変更し、再実行できます。また、ダイアログ・テスト・サービスを使用して、以下の目標を達成することもできます。

- 選択パネル、コマンド・プロシージャ、およびプログラムの開始
- パネルの表示
- 変数の追加および変数値の変更
- ISPF ダイアログ・サービスの実行
- ブレークポイント定義の追加、変更、および削除
- 関数および変数のトレース定義の追加、変更、および削除

トレース定義を作成、変更、および削除するには、ダイアログ・テスト・サービスのトレース (TRACES) オプションを使用します。さらにこのオプションは、ダイアログ・サービス呼び出しおよびダイアログ変数の使用をモニターする目的でも使用します。処理中に、いずれかのトレース定義が満たされると、トレース出力が ISPF ログに書き込まれます。ダイアログ・テスト・サービスの LOG オプションを使用すると、ISPF ログをブラウズしたり、ISPF の終了時に印刷出力を調べたりすることができます。

関連概念:

149 ページの『第 10 章 QMF アプリケーションのデバッグ』
エラー処理に加えて、QMF には呼び出し可能インターフェース・アプリケーション
に対するデバッグ機能も用意されています。

第 7 章 2 か国語使用のアプリケーションの作成

多くのビジネス・アプリケーションは、いくつかの異なる国語で実行する必要があります。英語版のアプリケーションを作成し、それを QMF がサポートする任意の国語で実行することができるようになりました。

QMF がサポートする各国語は、各国語機能 (NLF) と呼ばれます。NLF は、特定の各国語に合わせて作成された QMF セッションを提供します。

QMF は、コマンドおよび書式に関して 2 か国語サポートを提供しています。任意の NLF を用いて英語の QMF コマンドを実行し、英語の書式を表示することができます。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

英語環境と NLF 環境の比較

使用中の各国語機能 (NLF) に関係なく、QMF セッション環境の特性は同じですが、いくつかの相違点があります。NLF をインストールしていない場合、使用可能な唯一の QMF セッション環境は、英語環境です。

環境の類似点

使用中の NLF に関係なく、類似点は QMF セッション環境において最も重要です。

機能 一般に、NLF セッションでは、英語セッションで実行できるすべてのことを実行できます。一時記憶域にオブジェクトを作成してデータベースに保管し、報告書を形式設定して印刷し、SQL コマンドを出すことができます。さらに、指示照会、SQL 照会および QBE 照会、および QMF プロシージャを実行することもできます。環境間の相違点は、実行できる内容ではなく、入力方法および表示言語です。

SQL および QBE

SQL 言語および QBE 言語の動詞、演算子、およびキーワードは変換されません。

書式の取扱コード

コードは同じであるため、変換されません。

システム・コマンド

TSO コマンドおよび CICS コマンドは、TSO コマンドまたは CICS コマンドを使用して QMF から発行できます。これらのコマンドは変換によって影響を受けないため、TSO、CICS の後に実行するコマンドを入力します。このコマンドは、QMF の外で実行する場合とまったく同様に入力してください。

環境の相違点

NLF 環境と英語環境の間の相違点をいくつか示します。

QMF コマンド言語

すべての NLF には、変換された QMF 動詞とキーワードの完全なセットがあります。変換されたこれらの動詞およびキーワードは、NLF の言語環境での操作中に、QMF コマンドで表示されます。特定の NLF では、これらの語が変換されている場合があります。例えば、ドイツ語の NLF で、動詞 DISPLAY およびキーワード PROC が、それぞれ ANZEIGEN および PROZEDUR に変換されているとします。ドイツ語の NLF セッションでは、コマンド ANZEIGEN PROZEDUR を発行しますが、DISPLAY PROC を発行できません。

QMF 言語のいくつかの要素はコマンド同義語で、変換が可能です。その結果、各 NLF には、固有の名前が付いたコマンド同義語表があります。NLF をインストールすると、そのコマンド同義語表が作成され、NLF のプロファイルはその NLF に関するコマンド同義語表名を示します。

QMF パネルおよびメッセージ

各 NLF には QMF メッセージとパネルの完全なセットがあります。これらのメッセージとパネルは、QMF コマンドの動詞およびキーワードと同様に、変換されていない場合もあります。多くの場合は変換されます。パネルとメッセージの中で、テキストの固定部分は変換可能です。照会など、各パネルまたはメッセージ内で異なる可能性のある情報は変換されていません。

許可されるパネル入力

ユーザー入力を必要とする多くの QMF パネルでは、一部の入力の範囲が、少数の変換済みのキーワードに制限されています。このようなパネルの例としては、プロンプト・パネルや書式パネルなどが挙げられます。例えば、英語の YES および NO 応答は、ドイツ語では JA および NEIN になります。

プロファイルのパラメーター値

複数言語環境では、QMF セッションに使用できる NLF ごとに別個のプロファイルを持つことができます。これらの各プロファイルではパラメーターとその意味は同じですが、プロファイル名が変換されています。いくつかのパラメーター値も変換されています。

例えば、英語のプロファイルでは、CASE パラメーターの値は UPPER、STRING、または MIXED のいずれかです。ドイツ語のプロファイルでは、CASE パラメーターは SCHRIFT パラメーターとなり、有効な値は、GROSS、KETTE、および GEMISCHT になります。

エクスポートおよび保管された書式オブジェクト

書式オブジェクトを保管する言語を指定するには、SAVE コマンド、EXPORT コマンド、および IMPORT コマンドを使用します。書式オブジェクトは英語で保管することも、現行セッションの主要言語で保管することもできます。

サンプル表およびサンプル照会

IBM では、英語版のサンプル表およびサンプル照会の翻訳バージョンを提供しています (ただし、スウェーデン語および大文字機能は除きます)。サンプル表は、これらの機能には提供されていません。

2 か国語使用のアプリケーションで使用するオブジェクトの作成

2 か国語使用のアプリケーション内のオブジェクトは、他の QMF オブジェクトと同様です。重要な点は、オブジェクトを英語で作成するか、保管することです。

2 か国語使用のアプリケーションを英語で作成または保管する方法は、特定のオブジェクトに応じて異なります。

照会 QMF NLF によりサポートされている言語で指示照会や QBE 照会を作成できます。また、英語で SQL 照会を作成することもできます。

書式 書式は常に主要言語で作成します。SAVE コマンドのデフォルト言語 (ENGLISH) または主要言語を使用して保管します。

大域変数 DSQEC_FORM_LANG は、書式が保存、インポート、またはエクスポートされる言語を制御します。デフォルト値は英語を示す 1 です。値 0 は、主要セッションの言語で書式を保存、インポート、またはエクスポートするように指定します。

プロシージャ

プロシージャは、英語または主要言語のどちらを使用しても作成できます。

分析 分析オブジェクトは、英語または主要言語のどちらを使用しても作成できます。

SAVE コマンドを発行することによって、NLF で作成し保管した書式を英語に変換することができます。例えば、フランス語では SEMAINE_F と呼ばれる書式を WEEKLY_F という名前の英語の書式で保存するためのコマンドは、次のようになります。

```
SAUVER FORMAT SEMAINE_F EN WEEKLY_F (LANGUE=ANGLAIS)
```

このコマンドは、NLF 書式を、2 か国語使用のアプリケーションで使用できる英語の書式に変換します。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

コマンド言語変数

NLF セッションで英語コマンドを使用するには、主要言語変数 DSQEC_NLFCMD_LANG を英語に設定します。この変数を使用して、英語と NLF セッションの主要言語の間で切り替えます。

例えば、ご使用のアプリケーションが WEEKLY_P という名前のプロシージャであるとしたら、ここに記載するコマンドは、英語と主要 NLF 言語との間で切り替える方法を示しています。

```
"GET GLOBAL (CURR_LANG=DSQEC_NLFCMD_LANG" 1  
"SET GLOBAL (DSQEC_NLFCMD_LANG='1'"      2  
"RUN PROC WEEKLY_P"                        3  
"SET GLOBAL (DSQEC_NLFCMD_LANG=CURR_LANG" 4
```

上記のコマンドは、初期プロシージャから高水準言語プログラムへの、有効な QMF アプリケーションの一部になりますが、この並び順でなければなりません。

1 プロシージャのこの行は、変数内の主要言語の値を保管します。

GET GLOBAL コマンドは、CURR_LANG という変数内の主要言語の値を保管します。

2 プロシージャのこの行は、主要言語をアプリケーションが書かれた言語に設定します。

この例の WEEKLY_P アプリケーションは、英語コマンドを使用して作成されました。そのため、SET GLOBAL コマンドは、DSQEC_NLFCMD_LANG 変数を 1 に設定して、主要言語を英語に設定します。

3 プロシージャのこの行は、アプリケーションを実行します。

QMF セッションを英語に設定した後に、例のアプリケーションを実行できます。ユーザー・コマンドには、英語を使用する必要があります。ただし、ファンクション・キーを押した場合、対応するコマンドは主要言語を使用するものと想定されます。

QMF は、プロンプト・パネルがユーザーの主要言語を使用するものと想定します。「EXPORT コマンド・プロンプト」および「IMPORT コマンド・プロンプト」パネルでは、デフォルトのデータ・セット・タイプ、データ・キュー・タイプ、またはパス名も主要言語で表示されます。

セッションで有効な QMF プロファイルは、アプリケーションが開始した後に設定された NLF の下でのユーザーのプロファイルです。主要言語の QMF プロファイルは有効なプロファイルではありません。例えば、英語とドイツ語の両方で QMF を実行するユーザーは、英語とドイツ語の両方の QMF プロファイルを所有します。ユーザーがドイツ語 NLF の下で QMF セッションを開始すると、ドイツ語 QMF プロファイル内のオプションが有効になります。その後ユーザーが、DSQEC_NLFCMD_LANG 変数を英語に設定して、英語コマンドで作成されたプロシージャを実行します。この場合、ドイツ語の QMF プロファイル内のオプションは、そのセッションが終了するまで有効なままになります。

4 プロシージャのこの行は、主要言語に戻します。

アプリケーションが終了した後、例に示すようにコマンド言語変数を元の値にリセットします。

2 か国語使用のアプリケーション内の初期プロシージャ

アプリケーションが QMF を開始し、初期プロシージャを実行すると、QMF はユーザーが END コマンドを出すたびにそのプロシージャを実行します。このプロシージャがエラーを検出すると、QMF は終了します。

例えば、英語で QMF を実行するユーザーは、その主要言語で END コマンドを発行します。QMF は、そのコマンドをエラーとして解釈して終了します。

このような状態は、以下の 2 つのうちのいずれかの方法で回避できます。

- 2 か国語使用のアプリケーションを処理するように、初期プロシージャを変更する。

2 か国語使用の初期プロシージャには、次に示すコマンドが含まれます。

```
"GET GLOBAL (CURR_LANG=DSQEC_NLFCMD_LANG"  
"SET GLOBAL (DSQEC_NLFCMD_LANG=0"  
:  
/* QMF commands in the presiding language */  
:  
"SET GLOBAL (DSQEC_NLFCMD_LANG=CURR_LANG"
```

- END コマンドの後に、初期プロシージャを実行しないようにする。

変数 DSQEC_RERUN_IPROC を 0 に設定すれば、ユーザーが END コマンドを出した後で QMF が初期プロシージャを実行しないようにすることができます。

英語のみのコマンド

ほとんどの QMF コマンドは、コマンドを英語で実行する前に主要言語変数を変更する必要があります。ただし、主要言語変数が英語に設定されていない場合でも、一部のコマンドは英語で発行する必要があります。

例えば、英語で作成し NLF で実行する対話式アプリケーションがあるとします。ユーザーによってカスタマイズされたメッセージを送信するには、MESSAGE コマンドを使用する必要があります。さらに、メッセージを表示するには、次の例のように (この例はフランス語の NLF セッションで実行できる)、INTERACT コマンドを使用する必要があります。

```
proceed_text = 'Continue...'  
"RUN WEEKLY_Q" /* Use the English RUN command */  
"SET GLOBAL (DSQEC_NLFCMD_LANG=0" /* switch back to French */  
"MESSAGE (TEXT='proceed_text'" /* message in French */  
"INTERACT" /* show the report with message */
```

図 8. メッセージを表示するための MESSAGE および INTERACT コマンドの使用

以下の英語のコマンドは、どの NLF でも機能します。

- GET GLOBAL
- INTERACT
- MESSAGE
- SET GLOBAL
- START
- TRACE

複数言語環境

1 つまたは複数の NLF をインストールすると、複数言語環境が作成されます。適切な許可がある場合は、QMF セッションごとに、主要言語を選択できます。

例えば、あるセッションに英語を選択した場合、ドイツ語の NLF がインストール済みであれば、別のセッションにドイツ語を選択することができます。QMF セッション中は言語を切り替えることができませんが、コマンド言語変数を切り替えることはできます。適切な言語環境を獲得するために、現行セッションを終了して、別のセッションを開始してください。

変換可能なアプリケーションの作成

できるだけ多くの言語依存のオブジェクトに変数を使用することによって、新しい言語にアプリケーションを適合させるための時間を節約することができます。変数を使用して、複数の NLF で同じプログラムを実行します。

このような変数には、以下のものがあります。

- QMF コマンドの `verb`、オブジェクト名、およびオプション
- ユーザー定義パネル名

アプリケーション用の独自の ISPF パネルを作成する場合は、アプリケーションの実行時に使用する言語ごとに、変換された一連のパネルが必要です。これらのパネルに固有の名前を与え、アプリケーション・ユーザーが使用できるようにします。そうしておけば、アプリケーションは、パネル名に変数を使用できます。

- ユーザー定義メッセージ ID

独自の ISPF パネルを作成する場合は、ISPF から発行されるメッセージも作成します。このようなメッセージ・パネルには固有の ID が与えられ、変数を使用してそれらを参照できます。メッセージ・テキストを適切な NLF 言語に変換してください。アプリケーションは、メッセージ名に変数を使用することができます。

第 8 章 アプリケーションでの QMF コマンド

特定のコマンドはアプリケーション内で使用するよう設計されています。また、独自のコマンド同義語を作成することができます。

アプリケーション用に設計されたコマンド

特定の環境において QMF コマンド行上で有効なコマンドは、すべてアプリケーション内で有効です。ただし特定のコマンドは、アプリケーション内で使用するよう、特別に設計されています。

アプリケーション用に設計されたコマンドは、呼び出し可能インターフェース・アプリケーションとコマンド・インターフェース・アプリケーションの両方で使用できますが、2 つの例外があります。START コマンドと TRACE コマンドは、呼び出し可能インターフェースでしか使用できません。

CONNECT

QMF CONNECT コマンドを使用して、リモート・サーバー上のデータおよびオブジェクトにアクセスできます。リモート・システムに接続すると、そのシステムが現行ロケーションになります。

アプリケーションを作成するときは、以下から CONNECT コマンドを発行できます。

- 呼び出し可能インターフェース
- コマンド・インターフェース
- プロシージャ (線形手順またはロジックを持つプロシージャ)

ただし、プロシージャが、DB2 for z/OS ストアード・プロシージャとして QMF for TSO を開始する CALL ステートメントのパラメーターである場合は、CONNECT コマンドを発行できません。QMF がストアード・プロシージャとして開始されている場合は、リモート・サーバーを使ったコネクティビティはサポートされていません。

QMF CONNECT コマンドを使用してリモート・サーバーにアクセスすると、アプリケーションの特定の側面が影響を受ける可能性があります。以下の考慮事項があります。

- アプリケーションが新しいロケーションに接続すると、QMF プロファイル、コマンド同義語、およびファンクション・キーが、新しい (現行の) ロケーションにおける値に再初期化されます。
- QMF が開始されると、プログラムは QMF CONNECT コマンドを発行して、リモート・サーバーに接続できます。データベース・オブジェクトに影響を与える以後のすべての QMF コマンドまたは SQL ステートメントは、リモート・サーバーで実行されます。
- コマンドのタイプによって、リモート作業単位での動作は異なります。アプリケーションでリモート作業単位が使用される場合は、システム固有のコマンドすべ

ておよびほとんどの QMF コマンドが、QMF を実行中のシステム (ローカル・システム) で実行されます。ただし、QMF コマンドが次のいずれかを実行する場合、このコマンドはリモート・サーバーのデータベースに影響を与えます。

- SQL コマンドをデータベースに送信する場合
- データベースに保管されている QMF オブジェクトおよびデータを使用または変更する場合
- ご使用のサイトで TSO を使用し、大/小文字混合パスワードに対し RACF サポートを利用している場合は、QMF プロファイルの CASE オプションを MIXED に設定してください。MIXED に設定しないと、QMF によってすべての入力が大文字に変換されるため、パスワードの大/小文字が正しくないことが原因で、CONNECT コマンドが失敗します。QMF では大文字のコマンドのみが認識されるため、CASE=MIXED と設定した場合は必ず、QMF アプリケーション・ユーザーに対し、すべて大文字で入力するように指示してください。

次のサンプル・ステートメントは、REXX 呼び出し可能インターフェース・プログラムでの QMF CONNECT コマンドの使用方を示しています。プログラムはリモート・サーバーに接続し、定義済みのタスクを実行し、QMF を終了します。

1. 次のステートメントは、QMF セッションを開始します。

```
CALL DSQCIX "START"
```

2. 次のステートメントは、リモート DB2 データベース (DALLAS) に接続します。

```
CALL DSQCIX "CONNECT TO DALLAS"
```

3. このステートメントは、リモート・サーバーにデータを照会し、データを書式設定し、報告書を印刷するために、ロジックを持つプロシージャ (EARNINGS) を実行します。

```
CALL DSQCIX "RUN PROC EARNINGS"
```

プロシージャ EARNINGS には次のロジックが含まれています。

```
⋮  
"RUN QUERY EARNQ (FORM=EARNF"  
"PRINT REPORT"  
⋮
```

4. このステートメントは、QMF セッションを終了します。

```
CALL DSQCIX "EXIT"
```

END

END コマンドを使用して、QMF ホーム・パネルが現行パネルと同じになるよう設定できます。

例えば、QMF 報告書が現在の QMF パネルであるとし、呼び出し可能インターフェース・プログラムまたはコマンド・インターフェース・プログラムから END コマンドを発行すると、QMF ホーム・パネルが現行画面に設定されます。QMF ホーム・パネルが現行画面である場合は、END コマンドを発行しても、QMF セッションには影響しません。

EXIT

EXIT コマンドは、QMF セッションの開始方法に関係なく同じように機能します。このコマンドは、ユーザーのすべてのセッションを終了とマークします。

コマンド行に EXIT を入力すると、これを入力しているセッションがただちに終了します。INTERACT コマンドによって開始した各セッションは、そのセッションを開始したアプリケーションが完了すると終了します。アプリケーションで EXIT コマンドを出した場合、セッションは、元の QMF セッションが終了すると終了します。INTERACT コマンドによって開始したすべての対話式セッションは、QMF が終了する前に終了させる必要があります。

呼び出し可能インターフェース・プログラムでは、アプリケーションで QMF が不要になったときに、QMF EXIT コマンドを組み込むことが重要です。このコマンドを組み込まないと、QMF セッションは、ログオフするまで、またはバッチ・ジョブが完了するまでアクティブのままになります。

ユーザーまたはアプリケーションが EXIT コマンドを出すと、QMF は DSQAO_TERMINATE を 1 (終了とマークする) に設定します。QMF 内で実行中のアプリケーションだけが、この大域変数をテストし使用することができます。QMF がメイン QMF セッションに戻ったとき、DSQAO_TERMINATE が 1 に設定されていると、QMF はただちに終了してリソースを解放します。

関連資料:

218 ページの『プロファイル関連でない状態情報の大域変数』

DSQAO 大域変数には、状況情報、またはパラメーターやフラグの設定が格納されます。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

GET GLOBAL (拡張構文)

GET GLOBAL コマンドを使用すると、アプリケーションの QMF 大域変数にアクセスできます。REXX 以外の言語のために、QMF は GET GLOBAL コマンドの拡張構文を用意しています。

▶▶GET Global—(—| 変数定義 |—————▶▶

変数定義:

|—number of varnames—,—varname lengths—,—varnames—,—▶
▶—value lengths—,—values—,—value type—————|

GET GLOBAL コマンドを使用すると、QMF 大域変数の値を、アプリケーションが使用するアプリケーション定義の変数にコピーできます。GET GLOBAL コマンドで指定するパラメーターによって、アプリケーション変数が定義されます。

number of varnames

要求する変数の数。

varname lengths

指定する各変数名の長さのリスト。

変数名の長さ。後書きブランクで埋め込まれた 18 文字の区域が許可されます。

varnames

QMF 変数の名前のリスト。

大域変数名には、末尾ブランクを指定しないでください。末尾ブランクは、QMF によって削除されます。

value lengths

変数の値の長さのリスト

次の規則が変数値に適用されます。

- 指定した値の長さが、QMF に保管されている値の長さより短い場合は、QMF によって右側が切り捨てられ、切り捨て後の値が返されます。
- 指定した値の長さが、QMF に保管されている値の長さより長い場合は、QMF によって、末尾ブランクが埋め込まれた値が返されます。

values

変数値のリスト。

value type

値が入る記憶域のデータ・タイプ。データ・タイプは文字または整数でなければなりません。

INTERACT

INTERACT コマンドを使用すると、現行 QMF パネルを表示して、アプリケーションの異なる時点で QMF と対話できます。INTERACT コマンドには、セッションとコマンドという 2 つの形式があります。

QMF パネルから END コマンドを発行すると、QMF はアプリケーションに制御を返します。QMF パネルから EXIT コマンドを発行すると、QMF セッションに終了のマークが付き、QMF はアプリケーションに制御を返します。

セッション形式の INTERACT

INTERACT コマンドを実行すると、QMF はユーザーを現行パネルに置いて、ユーザーが QMF コマンドを対話式に実行できるようにします。INTERACT コマンドは、現行セッション内で別の QMF セッションを作りだします。

INTERACT コマンドは、ユーザーを対話式 QMF セッションまたは対話式 GDDM ICU セッションに置くことができます。

- 対話式 QMF セッションの場合

INTERACT コマンドは、通常であれば QMF パネルを表示する QMF コマンドの後に発行します。このセッションでは、ユーザーは対話式 QMF に有効なすべてのコマンドを入力することができます。

- 対話式 GDDM ICU セッションの場合

INTERACT コマンドは、通常は QMF に GDDM ICU を開始させて ICU パネルを表示させる、コマンドの後に発行します。このセッションで、ユーザーは ICU に有効なすべてのコマンドを入力することができます。

シナリオ

次に、報告書を作成するために 1 つのステップしか必要としないプロシーチャーの例を示します。

```
/* This procedure prints the weekly sales report. */  
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"  
"PRINT REPORT"
```

図 9. *INTERACT* コマンドがない単純なプロシーチャー

QMF によって、形式設定済みデータおよび「OK. 手順が実行されました」というメッセージが含まれた *REPORT* パネルが表示されます。

複数のステップからなるプロシーチャーを作成し、プロシーチャーの中間結果を確認できます。例えば、複数の照会を実行するプロシーチャーの中間結果を確認できます。コマンドの結果を確認するプロシーチャー内のポイントで *INTERACT* コマンドを使用します。この場合は、次に示すプロシーチャー内の最初の *RUN QUERY* コマンドの直後に *INTERACT* コマンドを挿入します。

```
/* This procedure generates a report showing annual sales. */  
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"  
"INTERACT"  
"RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"
```

図 10. プロシーチャーでの *INTERACT* の使用

「ホーム」パネルからこのプロシーチャーを実行すると、QMF は形式設定済みデータが入っている「*REPORT*」パネルを表示します。次に、「*REPORT*」パネルから *END* コマンドを入力します。このプロシーチャーは、2 次照会を実行し、最終報告書を表示します。 *INTERACT* コマンドを省略すると、QMF は最初の照会の結果を表示せずに最終報告書だけを表示します。

報告書の表示の抑止

QMF 呼び出し可能インターフェース・アプリケーションで照会を実行すると、QMF は結果の報告書を表示します。ただし、*DSQDC_DISPLAY_RPT* 大域変数をゼロ (0) に設定することによって、報告書を自動的に表示しないように QMF に指示することができます。 *START* コマンドで *DSQADPAN=0* を指定しても、この大域変数を設定することができます。

この大域変数は、アプリケーションから *RUN QUERY* コマンドを出した場合のみ有効です。この大域変数は、QMF コマンド行から *RUN QUERY* コマンドを出した場合には、報告書の表示に影響を与えません。

INTERACT コマンドによって開始されたセッションの終了

ユーザーが *END* コマンドを出すと、制御は *INTERACT* コマンドを出したプロセスに戻ります。ただし、2 つのセッションは独立していません。 *INTERACT* セッション中に実行したすべてのことが、古いセッションの再開時に有効のまま残ります。例えば、ユーザーが新しい対話式セッションで現行の書式オブジェクトを変更するとします。この場合、この新しいセッションが終了するときに、古いセッションでの現行の書式オブジェクトにこれらの変更が組み込まれます。

ユーザーが QMF オブジェクト・パネルから END コマンドを発行した後で QMF ホーム・パネルを表示するように、アプリケーションに指示できます。INTERACT ループのロジックを追加します。

関連資料:

212 ページの『INTERACT ループを使用する REXX の例』

対話式セッションでの END コマンドの動作を、対話式 QMF での END の動作方法と同様にすることができます。

コマンド形式の INTERACT

コマンド・インターフェース (DSQCCI) は、コマンド・インターフェース・アプリケーションがコマンド形式の INTERACT を使用し、QMF が対話式セッション (DSQSMODE=I) を実行している場合にのみ、QMF コマンドを対話式に実行します。

コマンド形式の INTERACT は、呼び出し可能インターフェースを介して出したコマンドには影響を与えません。呼び出し可能インターフェースにおいて、コマンドを対話式に実行するかどうかを制御する唯一の方法は、START コマンドでキーワード DSQSMODE=I を設定することです。

指定したコマンドの対話式実行を要求するには、次のコマンド構文を使用してください。

INTERACT *command*

このステートメントで、*command* は対話式に実行するコマンドです。このダイアログには、各種の QMF プロンプトおよび状況パネルが現れます。

例えば、次のコマンドは、RUN QUERY コマンドのオプションに関するコマンド・プロンプト・パネルを表示します。

```
INTERACT RUN QUERY ABC ?
```

対話式実行が許可されていない場合は、コマンド形式の INTERACT はこの後のコマンドに影響を与えません。QMF バッチ・セッションの場合、または QMF for TSO が DB2 for z/OS ストアード・プロシージャーとして開始された場合は、対話式セッションが許可されません。

現行セッションで対話式実行が許可されるかどうかを確認するには、DSQAO_INTERACT という名前の大域変数を調べます。DSQAO_INTERACT 大域変数の値が 1 であるときには、INTERACT が許可されます。

関連資料:

57 ページの『START コマンドのキーワード』

START コマンドでキーワードを指定できます。

218 ページの『プロファイル関連でない状態情報の大域変数』

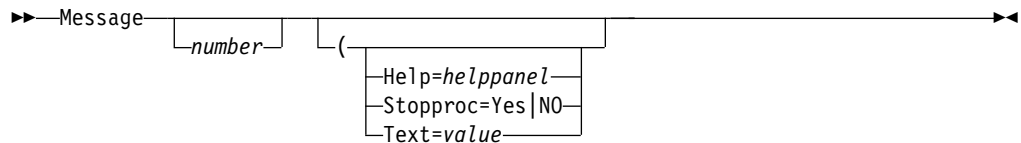
DSQAO 大域変数には、状況情報、またはパラメーターやフラグの設定が格納されます。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

MESSAGE

アプリケーションを作成するときには、ユーザーに向けて表示する情報または次に実行するアクションに関する特定のメッセージをユーザーに送信することがよくあります。MESSAGE コマンドを使用すれば、独自のメッセージを作成して QMF パネルに表示することができます。ISPF では、ISPF エラー・メッセージに関するメッセージ・ヘルプを QMF で表示することも選択できます。

構文

MESSAGE コマンド構文は次のとおりです。



number (番号:ISPF の場合のみ)

number (番号) は、ISPF のもとでのみ有効です。このパラメーターは、ISPF メッセージ・ライブラリーのメッセージ定義の識別番号です。

HELP

このパラメーターを使用して、この状態で通常表示されるメッセージに定義されているヘルプ・パネル以外のヘルプ・パネルを指定できます。*helppanel* を適切なパネル ID で置換します。

QMF パネルの定義が DSQPNLE にある場合、そのパネルを変更して、表示されるようにすることはできません。

ISPF で、独自のパネルを作成し、表示するには、そのパネルの定義が ISPF パネル・ライブラリーに含まれている必要があります。このライブラリーは、ISPLIB データ・セットに連結されている必要があります。このパネルは、メニューまたはデータ入力パネルではなく、ヘルプ・パネルでなければなりません。

ISPF で *number* を指定した場合、デフォルトで *helppanel* は、*number* によって指定したメッセージ定義を示すヘルプ・パネル標識になります。

ISPF において、*number* (番号) によって指定したメッセージ定義がヘルプ・パネル標識を参照していない場合、MESSAGE コマンドはメッセージ・ヘルプを提供しません。代わりに、ユーザーがヘルプを要求すると、オブジェクト・パネルに関する QMF ヘルプがユーザーの画面に表示されます。

STOPPROC

プロシージャ終了スイッチを設定することにより、**Stopproc** を使用して、線形プロシージャの実行を抑止します。次のコマンドは、プロシージャ終了スイッチを設定します。

```
Message (Stopproc=Yes
```

Stopproc=Yes の場合、プロシージャ終了スイッチがオンになります。デフォルト値は No (オフ) です。このスイッチは、線形プロシージャにのみ影響を与えます。

このスイッチがオンの間、制御を受けたすべての QMF プロシージャは、実行をただちに終了します。このスイッチがオフの間、プロシージャは通常どおり実行されます。

スイッチがオフの場合、オンに戻すことができるのは MESSAGE コマンドだけです。スイッチがオンの場合、以下のいずれかが発生するまでオンのままです。

- 他の QMF コマンドが発行される。このコマンドとは、スイッチをオンにするオプションが指定された MESSAGE コマンドを除く、すべての QMF コマンドです。
- アプリケーションの終了時に、制御がユーザーに戻される。ユーザーは、QMF プロシージャを実行するオンライン・コマンドをいつでも出すことができます。

DSQCM_MESSAGE 変数を調べると、プロシージャ終了スイッチがオンになっているかどうかを確認できます。終了オプションが有効である場合、この変数には終了スイッチをオンにした MESSAGE コマンドに関するメッセージが入っています。

TEXT

TEXT オプションを使用すれば、メッセージを定義したり、ISPF メッセージ定義内のテキストを指定変更することができます。value (値) をメッセージに使用する文字ストリングで置換します。空白文字を含む値は、区切り文字で囲む必要があります。メッセージ値に使用できる有効な区切り文字は、単一引用符、括弧、および二重引用符です。区切り文字が二重引用符である場合、引用符はメッセージの一部として表示されます。メッセージ値の最大長は、1 バイト文字で 360 文字です。メッセージを表示できる量は、ご使用のディスプレイ装置の表示幅により決まります。78 文字を超えるメッセージ値は、最初の 78 文字を残し切り捨てられます。QMF によってテキストは大文字に変換されませんが、コマンド・インターフェースを介して MESSAGE を発行した場合は、ISPF によってテキストが大文字に変換される場合があります。

メッセージに引用符が含まれている場合は、TEXT= の指定で二重引用符を使用する必要があります。

ISPF において、デフォルトは、number (番号) によって指定された ISPF メッセージの長メッセージ・テキストであり、これが生成されるメッセージになります。テキストはそのままの状態が残ります。ユーザーの QMF プロファイルの CASE 設定値に関係なく、大文字への変換は行われません。

プロシージャを使用して、2 つの照会を実行し、2 つの報告書を表示するアプリケーションを作成するものとします。QMF によって最初の報告書が表示される際には、2 番目の報告書の表示方法をユーザーに指示できます。次に示されているように、MESSAGE コマンドで定義したメッセージ (REPORT パネルに表示される) を含む線形プロシージャを作成できます。REPORT パネルにメッセージを表示するには、次のように、INTERACT コマンドの直前に MESSAGE コマンドを配置します。


```

:
  RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F
  MESSAGE (TEXT='OK, press END when you are finished viewing this report.'
  INTERACT
  RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F
:

```

図 11. MESSAGE コマンドの使用例

ロジックを持つプロシージャーを使用する場合は、次に示すように、MESSAGE コマンドで指定するテキスト・ストリングの代わりに REXX 変数を使用できます。REXX 変数を使用する場合、メッセージ・テキスト・ストリング内の変数名を二重引用符で囲む必要があります。

```

oktext = 'OK, press END when you are finished viewing this report.'
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
"MESSAGE (TEXT='"oktext"'"
"INTERACT"
"RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"

```

図 12. プロシージャーにおける MESSAGE コマンドでの REXX 変数の使用

次のメッセージは、REXX 継続文字を使用して、複数行にわたっています。

```

/* QMF REXX PROCEDURE */
MSGTEXT="You entered a data value incompatible with "||",
"the column data type; check the data type of the "||",
"column and try again."
"MESSAGE(TEXT=(MSGTEXT))"
EXIT

```

例

さまざまな条件下で MESSAGE コマンドを発行する方法の例を以下に示します。

- QMF 線形プロシージャーから MESSAGE コマンドを発行する例は次のとおりです。

このメッセージは、線形プロシージャーの継続文字を使用して、複数行にわたっています。

```

MESSAGE(TEXT='You entered a data value incompatible with
+the column data type; check the data type of the
+column and try again.')
```

- ISPF で MESSAGE コマンドを使用する例:

ISPF を使用するアプリケーションを構築している場合の MESSAGE コマンドの使用法の例を以下に示します。

– MESSAGE MSG011X

- メッセージ・テキストは、MSG011X 内の長メッセージです。
- メッセージ・ヘルプ・パネルは、MSG011X 内に (もしあれば) 指定されているパネルです。
- QMF がコマンドを処理した後にプロシージャー終了スイッチを設定するかどうかは、MSG011X のプロシージャー終了スイッチによって決定されます。

- MESSAGE MSG011X (HELP=ANELX STOPPROC=YES)
 - メッセージ・テキストは、MSG011X 内の長メッセージです。
 - メッセージ・ヘルプ・パネルは、ANELX という名前のパネルです。
 - プロシージャ終了スイッチはオンに変更され、アプリケーション内の QMF 線形プロシージャの実行を抑止します。

SET GLOBAL (拡張構文)

独自の大域変数を作成し、それらを QMF コマンドで置換変数として使用するには、SET GLOBAL コマンドを発行します。また、SET GLOBAL コマンドを使用すると、QMF の事前定義大域変数の値も設定できます。これは先頭が「DSQ」になります。

制約事項: SET GLOBAL コマンドを使用して、読み取り専用として定義されている大域変数を設定することはできません。

SET GLOBAL の構文 (拡張構文)

SET GLOBAL コマンドの拡張構文を使用すると、REXX 以外の呼び出し可能インターフェイス言語で変数の値を変更できます。他の言語の例としては、アセンブラー、C、COBOL、Fortran、および PL/I があります。

呼び出し可能インターフェイス・アプリケーションで使用される変数の変数名の長さは最大 18 文字です。変数が置換変数として使用されている場合、名前の長さは最大 17 文字です。コマンドの最大長 (コマンド構文を含む) は 2,000 バイトです。

コマンドの構文は次のとおりです。

▶▶—SET GLOBAL—(—| 変数定義 |—————▶▶

変数定義:

|—number of varnames—,—varname lengths—,—varnames—,—————▶

▶—value lengths—,—values—,—value type—————|

number of varnames

要求する変数の数。

varname lengths

指定する各変数名の長さのリスト。

大域変数名の長さが、プログラム内の大域名の実際の長さと同じことを確認します。後書き空白で埋め込まれた 18 文字の区域が許可されます。

varnames

QMF 変数の名前のリスト。

value lengths

変数値の長さのリスト。指定した値の長さが、記憶域に保管されている値の長さより短い場合、QMF に保管されるときに値の右側が切り捨てられます。

QMF はプログラム内の、ユーザーが割り当てたアドレスから開始して、ユーザーが割り当てた長さの値を使用します。長さが長すぎる場合、QMF は異常終了する可能性があります。

values

変数値のリスト。

value type

値が入る記憶域のデータ・タイプ。これは、文字または整数でなければなりません。

SET GLOBAL コマンドの拡張構文の使用例については、プログラミング言語の指定に記載されています。

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

大域変数の定義および使用に関するガイドライン

大域変数名は特定の規則に従います。

大域変数を定義および使用する場合は、次の規則に留意してください。

- SET GLOBAL コマンドでは、RUN コマンドや CONVERT コマンドと異なり、変数名の先頭にアンパーサンドが付きません。
- 書式大域変数を書式変数または集約変数と同じ名前で作成すると、QMF では、書式でその大域変数が使用されません。QMF では、書式に書式変数 (または集約変数) の値が使用され、大域変数の値は使用されません。
- QMF 書式は、名前に疑問符がある大域変数を認識しません。
- 大域変数名は、変数が置換変数として使用されない限り、18 文字に制限されています。置換変数名は 17 文字に制限されています。
- 大域変数名は数字を含んでいてかまいませんが、最初の文字だけは数字であってはなりません。
- 大域変数名は DSQ で開始してはなりません。これらの文字は、QMF の事前定義大域変数として QMF によって予約されています。
- 大域変数名の最初の文字は英字 (A から Z) か、または以下のいずれかの特殊文字でなければなりません。

`¢ ! $ ~ { } ? @ # % ¥`

- 大域変数名には、空白または次の文字が含まれてはなりません。

`* () - + ~ | : ; " ' < > / . , = &`

- QMF は、大域変数名から末尾空白を除去します。
- デフォルトでは、大域変数の値は、リセットするまで、または QMF セッションを終了するまで保持されます。ただし、あるセッションから別のセッションまで大域変数値を保持するように DSQEC_USERGLV_SAV 大域変数を設定できます。

関連資料:

240 ページの『コマンドとプロシージャーの実行を制御する大域変数』
DSQEC 大域変数は、コマンドおよびプロシージャーの実行方法を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

217 ページの『付録 B. QMF 大域変数』

QMF には、QMF セッションのさまざまな特性や、QMF コマンド、およびパネル表示を制御するために役立つ、多数の大域変数があります。大域変数を使用すると、プロシージャーおよびアプリケーションでの QMF 関数の動作を制御することもできます。

START

呼び出し可能インターフェースを介して QMF を開始する場合、START コマンドを使用する必要があります。

汎用構文

START コマンドの構文は、呼び出し可能インターフェース・アプリケーションに使用するプログラミング言語に応じて異なります。

一度に 1 つの QMF セッションしかアクティブにできません。アプリケーションから QMF を開始するときは、QMF が開始されているかどうかをテストするために START コマンドを発行します。

各プログラミング言語の構文の例は、プログラミング言語の指定に記載されています。

START コマンドの汎用構文は次のとおりです。

▶—START—(— | キーワード定義 | —————▶

 キーワード定義:

 |—number of keywords—,—keyword lengths—,—keywords—,—————▶

▶—value lengths—,—values—,—value type—————|

アセンブラー、C、COBOL、Fortran、および PL/I は、START コマンドについて以下の仕様を使用します。

number of keywords

START コマンドで使用する START コマンド・キーワードの数

keyword lengths

指定する各 START コマンド・キーワードの長さ

キーワード

START コマンド・キーワードの名前

value lengths

各 START コマンド・キーワードの値の長さが入っているリスト

values

このコマンドに指定する START コマンド・キーワードの値のリスト

value type

値のデータ・タイプ。データ・タイプは START コマンドの文字でなければなりません。

関連概念:

22 ページの『アプリケーションからの QMF の開始』

アプリケーションから他のすべてのコマンドを実行する場合には、まずその前に QMF を開始する必要があります。呼び出し可能インターフェースを使用している場合、アプリケーションで START コマンドを出して QMF を開始します。QMF セッションは、一度に 1 つしか開始できません。

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

『START コマンドのキーワード』

START コマンドでキーワードを指定できます。

START コマンドのキーワード

START コマンドでキーワードを指定できます。

START コマンドでは以下のキーワードを使用できます。

- DSQADPAN
- DSQALANG
- DSQSBSTG
- DSQSCMD (TSO のみ)
- DSQSDBCS
- DSQSDBNM
- DSQSDBQN (CICS のみ)
- DSQSDBQT (CICS のみ)
- DSQSDEBUG
- DSQSIROW
- DSQSMODE
- DSQSMRFI
- DSQSPILL
- DSQSPLAN (TSO のみ)
- DSQSPRID (TSO のみ)
- DSQSPTYP (TSO のみ)
- DSQSRSTG (TSO のみ)
- DSQSRUN
- DSQSSPQN (CICS のみ)
- DSQSSUBS (TSO のみ)

次の表で、これらのキーワードについて説明します。

START コマンド・キーワードは、以下の規則に従って指定できます。

- START コマンドで任意のキーワードを指定することができます。TSO では、DSQSCMD キーワードで指定した REXX プログラム中で任意のキーワードを指定することができます。QMF for CICS は REXX をサポートしないので、START コマンドですべてのキーワードを指定する必要があります。
- アプリケーションまたは初期プロシージャ (DSQSRUN キーワードで指定される) で、特定の環境でサポートされないキーワードを指定すると、このようなキーワードは無視されます。この規則により、環境固有のキーワードを変更せずに、複数の QMF 環境で実行可能な単一のプログラムをコンパイルできます。
- いずれかのキーワードを指定しないと、DSQSCMD キーワードで指定したプログラムに現れる START コマンド・キーワードの値が、QMF によって使用されます。この表に示すように、このプログラムを使用していない場合、QMF により各キーワードのデフォルト値が使用されます。

表 8. START コマンドのキーワード、説明、およびデフォルト値

START コマンドのキーワード	説明	デフォルト値
DSQADPAN	この START コマンドのパラメーターは呼び出し可能インターフェースでのみ使用するために用意されており、DSQDC_DISPLAY_RPT 大域変数を設定する。この変数は、照会をアプリケーション・プログラム内から実行した場合に、QMF が報告書を表示するかどうかを制御する。値 1 は、照会の実行時に報告書を表示する。報告書を表示しないようにするには、値を 0 に設定する。	1 (報告書の表示)
DSQALANG	開始するセッションの主要各国語を指定する。ユーザーが各国語で QMF コマンドを入力または指定できるように、アプリケーションにこのパラメーターを指定できる。このパラメーターの値は、下に示す 1 文字の各国語の ID である。主要言語が英語以外の言語のときに英語のコマンドを入力したい場合、QMF の 2 国語使用のサポートを使用することができる。指定した言語の各国語機能もインストールする必要がある。 C - カナダ・フランス語 D - ドイツ語 E - 英語 F - フランス語 H - 韓国語 (ハングル文字) I - イタリア語 K - 日本語 (漢字) P - ブラジル・ポルトガル語 Q - デンマーク語 S - スペイン語 U - 大文字英語 V - スウェーデン語 Y - スイス・フランス語 Z - スイス・ドイツ語	E (英語)

表 8. START コマンドのキーワード、説明、およびデフォルト値 (続き)

START コマンド のキーワード	説明	デフォルト値
DSQSBSTG	<p>QMF 報告書を生成するために使用する仮想記憶域のユーザー当たりの最大量を指定します。有効な値は次のとおりです。</p> <p>0 - 2147483647 バイト単位で記憶域を指定します。</p> <p>0KB - 2097152KB KB 単位で記憶域を指定します。</p> <p>0MB - 2048MB MB 単位で記憶域を指定します。</p> <p>0GB - 2GB GB 単位で記憶域を指定します。</p> <p>1% - 100% 使用可能な記憶域のパーセントを指定します。パーセンテージは、TSO でのみ有効です。</p> <p>CICS を使用している場合は、記憶域の固定容量 (バイト、KB、MB、または GB 単位) を指定できます。</p> <p>TSO を使用している場合は、QMF 報告書を生成するために使用される記憶域の最大量が、デフォルトで、QMF によって判別されます。DSQSBSTG パラメーターを有効なゼロ以外の値に設定すると、このパラメーターに指定した記憶域の容量が代わりに使用されます。TSO を使用している場合は、DSQSBSTG の値として、記憶域の固定容量、または使用可能な仮想記憶域のパーセンテージを指定できます。DSQSBSTG パラメーターを 0 に設定し、DSQSRSTG パラメーターをゼロ以外の値に設定すると、報告書の記憶域の最大量を判別するために、DSQSRSTG パラメーターが使用されます。DSQSBSTG と DSQSRSTG の両方に有効なゼロ以外の値を指定した場合は、DSQSBSTG の値が使用されます。</p> <p>DSQSBSTG の値を報告書作成に必要な記憶域の最小サイズより少なく設定すると、QMF は必要な記憶域の最小容量を自動的に割り振る。TSO では、記憶域の最小量は 15,000 バイトです。</p> <p>記憶域の使用可能な量は、QMF セッションが終わるまで、再評価されます。</p> <p>制約事項:</p> <ul style="list-style-type: none"> • パーセント値にはパーセント記号 (%) を付ける必要がありますが、スペースは挿入しないでください。 • GB、MB、または KB 形式の値にスペースを含めることはできません (例えば、「2 GB」と入力することはできません)。 • GB、MB、または KB 形式の値では、単位の後に文字を付加することができません (例えば、「2GBxyz」と入力することはできません)。 • 数値を先に入力せずに GB、MB、または KB を入力すると、値 0 として処理されます。 	<p>CICS の場合: 500000</p> <p>TSO の場合: 0</p>

表 8. START コマンドのキーワード、説明、およびデフォルト値 (続き)

START コマンド のキーワード	説明	デフォルト値
DSQSCMD (TSO のみ)	<p>このキーワードは、QMF for TSO で QMF プログラム・パラメーターを設定する REXX プログラムを指定します。</p> <p>QMF は、呼び出し可能インターフェース・アプリケーションから START コマンドを受け取ると、TSO はこのキーワードによって指定されている REXX プログラムを呼び出します。この REXX プログラムは、START コマンドで直接値が指定されない限り、QMF プログラム・パラメーターの値を提供します。TSO で提供されるデフォルト・プログラムは DSQSCMDE で、この表で示されるすべてのキーワードのデフォルト英語プログラム・パラメーターを提供します。特定のパラメーターの NULL 値は、TSO がそのパラメーターのデフォルト値を使用することを示します。</p> <p>NLF を使用している場合は、デフォルトのプログラム名を DSQSCMDn に変更できます。この n 変数は、使用している言語の各国語 ID (NLID) です。</p> <p>この表で示されていない場合でも、DSQSDBLG パラメーターもまた、TSO で提供される DSQSCMDE プログラムのデフォルトにより設定されます。このパラメーターは、QMF for TSO を DB2 for z/OS ストアード・プロシージャーとして開始するときに設定されます。ストアード・プロシージャー・インターフェースのコンテキストの外部で設定することはできません。</p>	DSQSCMDE
DSQSDBCS	<p>ディスプレイ装置が 2 バイト文字セット (DBCS) をサポートしない場合に、QMF が 2 バイト文字を許可するかどうかを指定する。値は YES または NO。</p> <p>非 DBCS ディスプレイ装置から DBCS データを印刷する場合、または DBCS データを印刷する QMF バッチ・ジョブを実行する場合は、値を YES に設定します。それ以外の場合は、値を NO に設定します。</p>	NO
DSQSDBNM	<p>QMF セッションの開始時に、接続するリモート・サーバーを指定する。NULL 値は、QMF がデフォルト・データベース (リモート作業単位を使用せずに、通常接続するデータベース) に接続することを意味する。</p>	NULL
DSQSDBQN (CICS のみ)	<p>QMF トレース・データに使用する CICS 記憶域キューの名前を指定します。名前は、DSQSDBQT によって指定される CICS キューのタイプの CICS 名前指定に従っていなければならない。</p>	DSQD
DSQSDBQT (CICS のみ)	<p>QMF トレース・データとして使用する CICS 記憶域のタイプを指定する。</p> <p>値は次のとおり。</p> <p>TD CICS 一時データ・キューを使用する。</p> <p>TS CICS 補助一時記憶域キューを使用する。QMF は大量のトレース・データを生成する可能性があるため、一時記憶域を指定する場合、注意が必要である。</p>	TD

表 8. START コマンドのキーワード、説明、およびデフォルト値 (続き)

START コマンド のキーワード	説明	デフォルト値
DSQSDEBUG	<p>QMF 初期設定時に製品アクティビティーをトレースするかどうかを指定する。値は次のとおり。</p> <p>ALL 最大限の詳細 QMF トレースを指定する。</p> <p>NONE QMF トレースを指定しない。</p> <p>QMF をバッチ・モードで開始した場合は、DSQSDEBUG の設定値とは関係なく、すべてのメッセージおよびコマンドがトレースされる (L2 レベルのトレースと同等)。</p>	NONE
DSQSIROW	<p>RUN QUERY、IMPORT DATA、または DISPLAY コマンドで最初のデータ画面を表示する前に QMF が取り出す行数を示す。</p>	100
DSQSMODE	<p>QMF を開始するモードを指定します。</p> <p>I 対話モードを指定する。</p> <p>B バッチ・モードを指定する。</p> <p>DSQSMODE の値が B の場合、QMF がバックグラウンド・ジョブとして実行できるように、パネル表示は禁止される。</p>	B (バッチ)
DSQSMRFI	<p>開始する QMF セッションが DB2 複数行フェッチおよび挿入を使用するかどうかを指定します。DB2 複数行フェッチおよび挿入を使用すると、多くの QMF コマンド (例えば、DISPLAY TABLE、EXPORT DATA または EXPORT TABLE、IMPORT TABLE、PRINT REPORT または PRINT TABLE、RUN QUERY または RUN PROC) がデータを取得するときに、これらのパフォーマンスが向上します。DB2 複数行フェッチおよび挿入を使用すると、さらに、報告書ナビゲーション中のコマンド (例えば、SAVE DATA、DPRE、BOTTOM または FORWARD) のパフォーマンスも向上します。</p> <p>このパラメーターは、DSQAO_DSQSMRFI 大域変数を設定します。</p> <p>YES QMF が DB2 複数行フェッチおよび挿入を使用することを指定します。</p> <p>MR が YES に設定されているときに、3 部構成の名前が含まれた QMF コマンドを使用する場合は、サーバーで DB2 for z/OS を実行する必要があります。コマンドを開始したリクエストと、コマンド送信先のサーバーの両方を、このバージョン・レベルで実行する必要があります。3 部構成の名前が含まれたコマンドを、DB2 for VSE and VM サーバーへ送ることはできません。</p> <p>NO QMF が DB2 複数行フェッチおよび挿入機能を使用しないことを指定します。</p> <p>XML データまたは LOB データを取得するときに、DSQSMRFI パラメーターの設定に関係なく、QMF によって単一行フェッチが使用されます。</p>	NO

表 8. START コマンドのキーワード、説明、およびデフォルト値 (続き)

START コマンド のキーワード	説明	デフォルト値
DSQSPILL	<p>報告書用に追加の記憶域が必要なときに、QMF が予備記憶域を使用するかどうかを指定します。値は YES または NO です。</p> <p>DSQSPILL に YES の値を指定し、かつ QMF for TSO を使用している場合は、使用する予備記憶域のタイプに適合するように、DSQSPTYP パラメーターを設定してください。</p> <p>CICS を使用している場合は、予備データを保持しておく一時記憶域キューを指定する方法について、DSQSSPQN キーワードの説明を参照。</p>	<p>CICS の場合: NO</p> <p>TSO の場合: YES</p>
DSQSPLAN (TSO のみ)	QMF に割り当てる DB2 アプリケーション・プラン ID を指定する。	QMF12
DSQSPRID (TSO のみ)	Q.PROFILES から適切な行を選択し、Q.ERROR_LOG 項目を適格化するために、TSO ログオン ID または 1 次データベース許可 ID を使用するかどうかを指定する。許可される値は、PRIMEID または TSOID である。	PRIMEID
DSQSPTYP (TSO のみ)	DSQSPILL パラメーターを YES に設定するときは、予備データに使用する記憶域のタイプを指定してください。FILE の値はファイルを指定します。64BIT の値は拡張記憶域を指定します。拡張記憶域が指定される場合、QMF は、DSQEC_EXTND_STG 大域変数で指定された大きさで、拡張記憶域マネージャーへの呼び出しごとに記憶域を獲得します。	FILE
DSQSRSTG (TSO のみ)	<p>報告書に使用可能な仮想記憶域を動的に割り振る。</p> <p>または、DSQSBSTG キーワードを使用すると、記憶域の最大量として、固定量または使用可能な記憶域のパーセンテージを設定できます。DSQSBSTG をゼロ以外の値に設定すると、QMF によって DSQSRSTG の値が無視されます。</p>	0
DSQSRUN	<p>QMF の開始後に実行する QMF 初期プロシージャの名前を指定する。呼び出し可能インターフェースの場合、初期プロシージャは 1 回だけ実行される。</p> <p>このプロシージャでは、大域変数とプロファイル値を設定してユーザーのセッションをカスタマイズするためにコマンドを含めることができます。</p>	NULL
DSQSSPQN (CICS のみ)	QMF 予備データに使用する CICS 一時記憶域キューの名前を指定する。プログラム・パラメーター DSQSPILL の値が YES の場合、この予備域は、報告書データを入れるために使用される。	DSQSvid (id は CICS 端末 ID)
DSQSSUBS (TSO のみ)	QMF が開始される DB2 データベースの ID を指定します。このキーワードに指定したデータベース ID は、アプリケーション・リクエスターとして構成する必要があります。	DSN

関連概念:

39 ページの『第 7 章 2 か国語使用のアプリケーションの作成』
 多くのビジネス・アプリケーションは、いくつかの異なる国語で実行する必要があります。英語版のアプリケーションを作成し、それを QMF がサポートする任意の

国語で実行することができるようになりました。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

240 ページの『コマンドとプロシージャの実行を制御する大域変数』

DSQEC 大域変数は、コマンドおよびプロシージャの実行方法を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

TRACE

TRACE コマンドを使用すると、呼び出し可能インターフェース・アプリケーションから QMF トレース・データ出力ヘトレース情報を追加できます。このコマンドは、アセンブラー、C、COBOL、Fortran、および PL/I アプリケーションで使用できます。これを QMF コマンド行から発行することはできません。

TRACE コマンド構文は次のとおりです。

```
▶▶—TRACE—(—| Trace area definitions |—————▶▶
```

Trace area definitions:

```
|—number of trace areas—,—trace title lengths—,—trace title addresses—,—▶
```

```
▶—trace area lengths—,—trace area addresses—,—value type—————|
```

number of trace areas

TRACE コマンドで使用しているトレース域定義の数。この数は、1 から 10 までの範囲内である必要があります。

trace title lengths

指定された各トレース・タイトルの長さのリスト。

trace title addresses

各トレース域に使用されるトレース・タイトルを指すアドレスのリスト。トレース・タイトルには最大 40 文字の長さを指定できます。末尾ブランクは削除されます。最初のトレース・タイトルが SNAPREGS である場合は、他のトレース・タイトルおよびトレース域アドレスがすべて無視され、QMF レジスター値が QMF トレースに書き込まれます。

trace area lengths

対応するトレース域アドレス以降に表示される記憶域のバイト数を含むリスト。トレース域の長さは、FIXED(31) 整数値に含まれている必要があります。

trace area addresses

トレース出力に表示されるアドレスのリスト。各トレース域アドレスに表示される記憶域のバイト数は、トレース域の長さによって決まります。最初のトレース・タイトルが「SNAPREGS」である場合は、トレース域アドレスおよびトレース域の長さがすべて無視されます。トレース域アドレスが 0 である場合は、トレース域の長さも 0 であると見なされます。

value type

値「FINT」である必要があります。「FINT」は定数値で、各プログラミング言語のインターフェース連絡域 (DSQCOMM) で提供されます。この定数は、DSQ_VARIABLE_FINT に類似した名前です。変数値を確認するには、ご使用のプログラミング言語のインターフェース連絡域を調べます。

TRACE コマンドは、QMF のトレース設定に関係なく、トレース域定義を QMF トレース・データ出力に書き込みます。QMF のトレース設定がアクティブであるときにのみ、トレース出力を書き込むには、QMF 大域変数 DSQAO_APPL_TRACE または DSQAP_TRACE を使用します。

TRACE の PL/I コーディング例

PL/I アプリケーションで TRACE コマンドを使用すると、アプリケーションのトレース情報を QMF トレース出力に書き込むことができます。

次のコーディング例では、まず、ユーザーが QMF アプリケーションのアクティビティをトレースしていることを確認するため、大域変数 DSQAO_APPL_TRACE が「0」に設定されていないことを検査しています。アプリケーション・トレースがオンになっている場合は、TRACE コマンドが発行されます。TRACE コマンドは、QMF トレース出力に書き込む 3 つのトレース域定義を指定します。

次の例は、QMF から配布される DSQABFP ファイルには含まれていません。

```
DSQABFP: PROCEDURE OPTIONS(MAIN REENTRANT) REORDER;
/*****
/* Sample Program: DSQABFP
/* PL/I Version of the QMF Callable Interface
*****/

/*****
/* Include and declare query interface communications area
*****/
%INCLUDE SYSLIB(DSQCOMML);

/*****
/* Builtin function
*****/
DCL LENGTH BUILTIN;

/*****
/* Query interface command length and commands
*****/
DCL COMMAND_LENGTH FIXED BIN(31);
DCL START_QUERY_INTERFACE CHAR(5) INIT('START');
DCL SET_GLOBAL_VARIABLES CHAR(10) INIT('SET GLOBAL');
DCL GET_GLOBAL_VARIABLES CHAR(10) INIT('GET GLOBAL');
DCL RUN_QUERY CHAR(12) INIT('RUN QUERY Q1');
DCL PRINT_REPORT CHAR(22) INIT('PRINT REPORT (FORM=F1)');
DCL TRACE_COMMAND CHAR(5) INIT('TRACE');
DCL END_QUERY_INTERFACE CHAR(4) INIT('EXIT');

/*****
/* Query command extension, number of parameters and lengths
*****/
DCL NUMBER_OF_PARAMETERS FIXED BIN(31);/* number of variables
DCL KEYWORD_LENGTHS(10) FIXED BIN(31);/* lengths of keyword names*/
DCL DATA_LENGTHS(10) FIXED BIN(31);/* lengths of variable data*/

/*****
/* Trace command parameters
*****/
```

```

/*****/
DCL AREA_DESCRIPTION(10)      CHAR(40);
DCL AREA_DESCRIPTION_LENGTH(10) FIXED BIN(31);
DCL AREA_PTR(10)              PTR;
DCL AREA_LENGTH(10)          FIXED BIN(31); /* Length of area at */
                                           /* AREA_PTR to be */
                                           /* displayed. */

/*****/
/* Keyword parameter and value for START command */
/*****/
DCL 1 START_KEYWORDS,
    3 START_KEYWORDS_1      CHAR(8)  INIT('DSQSMODE'),
    3 START_KEYWORDS_2      CHAR(8)  INIT('DSQSDBUG'),
    3 START_KEYWORDS_3      CHAR(8)  INIT('DSQSSUBS'),
    3 START_KEYWORDS_4      CHAR(8)  INIT('DSQSPLAN');
DCL 1 START_KEYWORD_VALUES,
    3 START_KEYWORD_VALUES_1 CHAR(11) INIT('BATCH'),
    3 START_KEYWORD_VALUES_2 CHAR(3)  INIT('ALL'),
    3 START_KEYWORD_VALUES_3 CHAR(4)  INIT('DSNA'),
    3 START_KEYWORD_VALUES_4 CHAR(6)  INIT('QMFDEV');

/*****/
/* Keyword parameter and value for SET command */
/*****/
DCL 1 SET_KEYWORDS,
    3 SET_KEYWORDS_1      CHAR(7)  INIT('MYVAR01'),
    3 SET_KEYWORDS_2      CHAR(5)  INIT('SHORT'),
    3 SET_KEYWORDS_3      CHAR(7)  INIT('MYVAR03');

DCL 1 SET_VALUES,
    3 SET_VALUES_1      FIXED BIN(31),
    3 SET_VALUES_2      FIXED BIN(31),
    3 SET_VALUES_3      FIXED BIN(31);

/*****/
/* Keyword parameter and value for GET command */
/*****/
DCL 1 GET_TRACE_KEYWORDS,
    3 GET_TRACE_KEYWORDS_1 CHAR(16) INIT('DSQAO_APPL_TRACE');

DCL 1 GET_TRACE_VALUE,
    3 GET_TRACE_VALUE_1 CHAR(1);

/*****/
/* Main program */
/*****/
DSQCOMM = '';
DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL;

/*****/
/* Start a query interface session */
/*****/
NUMBER_OF_PARAMETERS = 4;
COMMAND_LENGTH = LENGTH(START_QUERY_INTERFACE);
KEYWORD_LENGTHS(1) = LENGTH(START_KEYWORDS_1);
KEYWORD_LENGTHS(2) = LENGTH(START_KEYWORDS_2);
KEYWORD_LENGTHS(3) = LENGTH(START_KEYWORDS_3);
KEYWORD_LENGTHS(4) = LENGTH(START_KEYWORDS_4);
DATA_LENGTHS(1) = LENGTH(START_KEYWORD_VALUES_1);
DATA_LENGTHS(2) = LENGTH(START_KEYWORD_VALUES_2);
DATA_LENGTHS(3) = LENGTH(START_KEYWORD_VALUES_3);
DATA_LENGTHS(4) = LENGTH(START_KEYWORD_VALUES_4);

CALL DSQCIPX(DSQCOMM,
            COMMAND_LENGTH,
            START_QUERY_INTERFACE,

```

```

        NUMBER_OF_PARAMETERS,
        KEYWORD_LENGTHS,
        START_KEYWORDS,
        DATA_LENGTHS,
        START_KEYWORD_VALUES,
        DSQ_VARIABLE_CHAR);

/*****
/* Find out current trace setting */
*****/
NUMBER_OF_PARAMETERS = 1;
COMMAND_LENGTH = LENGTH(GET_GLOBAL_VARIABLES);
KEYWORD_LENGTHS(1) = LENGTH(GET_TRACE_KEYWORDS_1);
DATA_LENGTHS(1) = LENGTH(GET_TRACE_VALUE_1);

CALL DSQCIPX(DSQCOMM,
            COMMAND_LENGTH,
            GET_GLOBAL_VARIABLES,
            NUMBER_OF_PARAMETERS,
            KEYWORD_LENGTHS,
            GET_TRACE_KEYWORDS,
            DATA_LENGTHS,
            GET_TRACE_VALUE,
            DSQ_VARIABLE_CHAR);

/*****
/* Set numeric values into query using SET command */
*****/
NUMBER_OF_PARAMETERS = 3;
COMMAND_LENGTH = LENGTH(SET_GLOBAL_VARIABLES);
KEYWORD_LENGTHS(1) = LENGTH(SET_KEYWORDS_1);
KEYWORD_LENGTHS(2) = LENGTH(SET_KEYWORDS_2);
KEYWORD_LENGTHS(3) = LENGTH(SET_KEYWORDS_3);
DATA_LENGTHS(1) = 4;
DATA_LENGTHS(2) = 4;
DATA_LENGTHS(3) = 4;
SET_VALUES_1 = 20;
SET_VALUES_2 = 40;
SET_VALUES_3 = 4;

CALL DSQCIPX(DSQCOMM,
            COMMAND_LENGTH,
            SET_GLOBAL_VARIABLES,
            NUMBER_OF_PARAMETERS,
            KEYWORD_LENGTHS,
            SET_KEYWORDS,
            DATA_LENGTHS,
            SET_VALUES,
            DSQ_VARIABLE_FINT);

/*****
/* Run a Query */
*****/
COMMAND_LENGTH = LENGTH(RUN_QUERY);

CALL DSQCIPL(DSQCOMM,
            COMMAND_LENGTH,
            RUN_QUERY);

/*****
/* Trace command */
*****/
IF GET_TRACE_VALUE_1 ^= '0' THEN DO;
    NUMBER_OF_PARAMETERS = 3;
    COMMAND_LENGTH = LENGTH(TRACE_COMMAND);
    AREA_DESCRIPTION(1) = 'DSQAO_APPL_TRACE: ';
    AREA_DESCRIPTION_LENGTH(1) = LENGTH(AREA_DESCRIPTION(1));

```

```

AREA_PTR(1) = ADDR(GET_TRACE_VALUE_1);
AREA_LENGTH(1) = LENGTH(GET_TRACE_VALUE_1);
AREA_DESCRIPTION(2) = 'DSQ_COMM_LEVEL: ';
AREA_DESCRIPTION_LENGTH(2) = LENGTH(AREA_DESCRIPTION(2));
AREA_PTR(2) = ADDR(DSQ_COMM_LEVEL);
AREA_LENGTH(2) = LENGTH(DSQ_COMM_LEVEL);
AREA_DESCRIPTION(3) = 'DSQ_CURRENT_COMM_LEVEL: ';
AREA_DESCRIPTION_LENGTH(3) = LENGTH(AREA_DESCRIPTION(3));
AREA_PTR(3) = ADDR(DSQ_CURRENT_COMM_LEVEL);
AREA_LENGTH(3) = LENGTH(DSQ_CURRENT_COMM_LEVEL);

CALL DSQCIPX(DSQCOMM,
             COMMAND_LENGTH,
             TRACE_COMMAND,
             NUMBER_OF_PARAMETERS,
             AREA_DESCRIPTION_LENGTH,
             AREA_DESCRIPTION,
             AREA_LENGTH,
             AREA_PTR,
             DSQ_VARIABLE_FINT);

END;

/*****
/* Print the results of the query */
*****/
COMMAND_LENGTH = LENGTH(PRINT_REPORT);

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             PRINT_REPORT);

/*****
/* End the query interface session */
*****/
COMMAND_LENGTH = LENGTH(END_QUERY_INTERFACE);

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             END_QUERY_INTERFACE);

END      DSQABFP;

```

ユーザーがトレースをオンに設定していることをプログラムが検出すると、次のトレース出力が QMF トレース出力に書き込まれます。

```

-----
          DSQDTRC :TRACE COMMAND OUTPUT                (14534)
          TRACE_AREA_NUMBER
341033B8: 00000001 *... *
          TRACE_AREA_TITLE
341036A4: C4E2D8C1 D66DC1D7 D7D36DE3 D9C1C3C5 *DSQA0_APPL_TRACE*
341036B4: 7A *: *
          TRACE_AREA_CONTENTS
340D0A14: F2 *2 *
-----
          DSQDTRC :TRACE COMMAND OUTPUT                (14535)
          TRACE_AREA_NUMBER
341033B8: 00000002 *... *
          TRACE_AREA_TITLE
341036A4: C4E2D86D C3D6D4D4 6DD3C5E5 C5D37A *DSQ_COMM_LEVEL: *
          TRACE_AREA_CONTENTS
340D0500: C4E2D8D3 6EF0F0F1 F0F0F24C *DSQL>001002< *
-----
          DSQDTRC :TRACE COMMAND OUTPUT                (14536)
          TRACE_AREA_NUMBER
341033B8: 00000003 *... *
          TRACE_AREA_TITLE

```

```

341036A4: C4E2D86D C3E4D9D9 C5D5E36D C3D6D4D4 *DSQ_CURRENT_COMM*
341036B4: 6DD3C5E5 C5D37A *_LEVEL: *
TRACE_AREA_CONTENTS
33F00C50: C4E2D8D3 6EF0F0F1 F0F0F24C *DSQL>001002< *
-----

```

RUN QUERY 報告書ミニ・セッションで使用できるコマンド

QMF アプリケーションで使用するコマンドの一部は、アプリケーションの実行中に報告書を強制的に表示します。この環境は、報告書のミニセッションと呼ばれます。報告書ミニ・セッションを使用することによって、ユーザーによる QMF へのアクセスを制限できます。報告書ミニ・セッションでは、報告書を表示中にユーザーが発行できるコマンドが QMF によって制限されます。

報告書ミニ・セッションは、ネスト・セッション (セッション内のセッション) として機能します。ミニ・セッション内で、最初の QMF セッションはそのまま残りますが、報告書の表示中は、一時的に使用不可になります。ミニ・セッションは、END コマンドを発行するまで (または、終了ファンクション・キーを押すまで) 現行のアクティブ・セッションになります。ミニ・セッションを終了すると、アプリケーションでの指示に従って、最初の QMF セッションまたは呼び出しアプリケーションに戻ります。アプリケーションは、報告書ミニ・セッションが終了するまで、後続のコマンドを発行できません。

QMF 大域変数 DSQDC_DISPLAY_RPT によって、QMF が報告書ミニ・セッションを開始するかどうかが決まります。これは、DSQDC_DISPLAY_RPT によって、照会の実行後に QMF が報告書を表示するかどうかが決まるためです。この変数が 1 に設定されている場合は報告書を表示し、0 に設定されている場合は表示を抑制します。

呼び出し可能インターフェースを使用して QMF を開始する場合、以下のようになります。

- 大域変数 DSQDC_DISPLAY_RPT のデフォルト値は 1 です。DSQQMF n を指定して (対話式またはバッチ・モードで) QMF が開始された場合、この大域変数のデフォルト値は 0 になります。この変数 n は、各国語 ID を表します。
- 照会を実行するプロシージャまたはアプリケーションを実行すると、QMF は報告書ミニ・セッションを開始します。QMF によって、照会から生成された報告書がこのミニ・セッションに表示されます。
- プロシージャまたはアプリケーションが照会を実行しない場合、または照会を SQL Query パネルから実行する場合、QMF は報告書ミニ・セッションを開始しません。

QMF で報告書ミニ・セッションを開始しないようにするには、以下のいずれかの操作を実行します。

- DSQDC_DISPLAY_RPT の値を 0 に変更する。
- 呼び出し可能インターフェースから QMF を開始するときに、DSQADPAN パラメーターを 0 に設定する。

報告書ミニ・セッションから、以下のコマンドおよびこれらのコマンドの同義語を発行することができます。制限は括弧で囲みます。

- BACKWARD
- BOTTOM
- CANCEL (ポップアップ・ウィンドウがアクティブの場合)
- CICS
- DISPLAY REPORT
- DISPLAY CHART
- END
- FORWARD
- GET GLOBAL
- HELP
- INTERACT
- ISPF
- LEFT
- MESSAGE
- PRINT REPORT
- PRINT CHART
- QMF
- RETRIEVE
- RIGHT
- SAVE DATA
- SET PROFILE
- SET GLOBAL
- SHOW REPORT
- SHOW CHART
- SWITCH (オンライン・ヘルプがアクティブの場合)
- TOP
- TSO

ミニ・セッションで無効な コマンドは次のとおりです。

- ADD
- CANCEL
- CHANGE
- CHECK
- CLEAR
- CONNECT
- CONVERT
- DELETE
- DESCRIBE
- DISPLAY (QUERY, PROC, PROFILE, FORM, ANALYTICS)
- DRAW

- EDIT
- ENLARGE
- ERASE
- EXIT
- EXPORT
- EXTRACT
- GETQMF
- IMPORT
- INSERT
- INTERACT
- LIST
- NEXT
- PREVIOUS
- PRINT (QUERY, PROC, PROFILE, FORM)
- REDUCE
- REFRESH
- RESET GLOBAL
- RESET (照会、プロシージャー、書式)
- RUN
- SAVE
- SEARCH
- SHOW
- SORT
- SPECIFY
- START
- SWITCH
- TRACE

無効なコマンドを発行する CLIST またはプロシージャーを実行すると、QMF はエラー・メッセージを戻します。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

54 ページの『SET GLOBAL (拡張構文)』

独自の大域変数を作成し、それらを QMF コマンドで置換変数として使用するには、SET GLOBAL コマンドを発行します。また、SET GLOBAL コマンドを使用すると、QMF の事前定義大域変数の値も設定できます。これは先頭が「DSQ」になります。

コマンド同義語

コマンド同義語を作成できます。コマンド同義語とは、QMF コマンドに似たコマンドで、各種機能を実行できます。

コマンド同義語には柔軟性があり、ユーザーにとって便利です。例えば、コマンド同義語は特定のコマンドの機能を実行したり、アプリケーションを開始したりすることができます。

第 9 章 オブジェクトのエクスポートとインポート

QMF 環境の外にオブジェクトを置くために、QMF EXPORT および IMPORT コマンドを発行するアプリケーションを作成できます。

アプリケーションでは、表および次の QMF オブジェクトをエクスポートできます。

- DATA
- QUERY
- プロシージャ
- FORM
- REPORT
- CHART

オブジェクト (CSV 形式の報告書、オブジェクトを除く) をエクスポートすると、QMF によってオブジェクトが外部形式に変換されます。QMF はオブジェクトの外部形式を UNIX ファイル (データまたは表の場合のみ)、TSO データ・セット、または CICS データ・キューに保管します。QMF オブジェクトの外部形式は、QMF アプリケーション開発の強力な要素です。

IMPORT コマンドは、外部形式を読み取り、オブジェクトを QMF 一時記憶域またはデータベースに置きます。その場所は、コマンドの発行方法に応じて異なります。

エクスポートした **UNIX** ファイル、**TSO** データ・セット、または **CICS** データ・キューを使用して実行できる内容

データ・オブジェクトに対して実行する QMF の EXPORT コマンドおよび IMPORT コマンドは、さまざまな状況下で役立ちます。

例えば、IMPORT コマンドおよび EXPORT コマンドを使用すると、以下の目標を達成できます。

- 照会結果をアプリケーションに提供する。

QMF EXPORT コマンドを使用すれば、データベースからデータを取り出し、それをアプリケーションに取り入れることができます。

- アプリケーション内でオブジェクトを作成し、QMF 内で使用する。

オブジェクトに関する適切な形式を使用して、QMF 環境の外側でオブジェクトを作成することができます。例えば、データまたは表の場合、UNIX ファイルを QMF にインポートすると、新規 QMF オブジェクトが作成されます。新規オブジェクトは、オブジェクトが含まれている TSO データ・セットまたは CICS データ・キューをインポートするときにも、作成されます。

報告書、図表、または CSV ファイルを QMF にインポートすることはできません。報告書および図表の場合は、代わりに、それらを作成するときに使用したデータおよび書式をインポートできます。

- QMF オブジェクトを他の環境またはプロダクトで使用できるようにする。

重要: エクスポートしたオブジェクトを、例えば、EBCDIC および ASCII システムといった異なる CCSID または文字セットを使用している環境間や、異なる NLF 環境間で転送する場合は注意してください。この方法でシステム間でオブジェクトを転送すると、オブジェクトを使用できなくなる可能性があります。

指示照会または QBE 照会を QMF 以外のプログラムにインポートする必要がある場合は、まず CONVERT QUERY コマンドを使用する必要があります。

CONVERT QUERY コマンドによって照会は SQL 照会に変換されるため、エクスポート可能になり、他の製品で使用できるようになります。

QMF オブジェクトは、TSO、ISPF、またはネイティブ z/OS バッチの下の QMF 間で転送できます。また、CICS 下で QMF オブジェクトを転送するには、CICS 区画外一時データ・キューを使用します。

- データベースの外にオブジェクトおよびデータを保管する。

例えば、プログラムの途中で外部プログラムが処理できるようにデータをエクスポートすることができます。

- 2 か国語使用のアプリケーションを作成する。

主要言語で QMF 書式を作成してから、EXPORT コマンドで LANGUAGE オプションを使用して英語に変換することができます。また、IMPORT および EXPORT コマンドで LANGUAGE オプションを使用して、英語の書式を主要言語に変換することもできます。

データのエクスポートと保管

EXPORT DATA コマンドと SAVE DATA コマンドの相違点は、オブジェクトを保管する場所と方法にあります。

オブジェクトの保管方法によって、結果を使用して実行できる操作が異なります。

- データ・オブジェクトをエクスポートすると、UNIX ファイル、TSO データ・セット、または CICS データ・キューが作成されます。QMF アプリケーション・プログラムまたは他の外部アプリケーションを介して、それぞれの項目を順次に読み取り、変更し、印刷できます。
- SAVE DATA コマンドはデータベース表を作成します。保管されたデータを使用する操作は、データベースから実行する必要があります。

データ・オブジェクトおよびデータベース表のエクスポート

データ・オブジェクトおよび表オブジェクトは、QMF 形式、統合交換形式 (IXF)、XML 形式、またはコンマ区切り値 (CSV) 形式でエクスポートできます。デフォルトは、QMF 形式です。

照会を実行すると、QMF は結果を報告書に表示します。報告書用の生データは、データ・オブジェクトとして一時記憶域に保管されます。データベースに保管されたリレーショナル表および視点は、表オブジェクトと呼ばれます。

データ・オブジェクトおよび表オブジェクトは、QMF の外にある記憶域にエクスポートできます。一時記憶域内の表 (DATA) とデータベースに保管されている表 (TABLE) のエクスポート形式は同じです。データとしてエクスポートされたオブジェクトは、表としてインポートできます。また、その逆も可能です。ただし、データが CSV 形式でない場合に限りです。

QMF の外で独自の表を作成できます。QMF 形式、IXF 形式、または XML 形式を使用し、表を含む UNIX ファイル、TSO データ・セット、または CICS データ・キューの内容をインポートします。必要に応じて、望ましいフィールドを組み込み、ユーザー固有のデータを追加します。その後、UNIX ファイル、TSO データ・セット、または CICS データ・キューを表オブジェクトとして QMF にインポートします。

関連概念:

102 ページの『データ・オブジェクトと表のエクスポートおよびインポートに関する規則と情報』

QMF は、データ・オブジェクトおよび表オブジェクトを一時記憶域にエクスポートします。その記憶域を割り振る方法には、いくつかの規則があります。QMF では、インポート・エラーも固有の方法で処理されます。

QMF 形式でのデータまたは表のエクスポート

DATAFORMAT=QMF 節を含む EXPORT コマンドを使用してエクスポートするデータ・ファイルは、レコード内のデータについて記述しているヘッダー・レコードと、データが入っているデータ・レコードの 2 つの部分から構成されます。

ヘッダー・レコード

ヘッダー・レコードは、データ・ファイル内のエクスポート後のデータについて記述します。

外部データ・ファイルのレコード長は、データ・レコードで説明しているように、データ行の長さです。データ・レコードの前にあるヘッダー・レコードは、データ・レコードと同じ長さです。ヘッダー情報がデータ・レコードの長さを超える場合は、複数のヘッダー・レコードが作成されます。

ヘッダー・レコードには 2 つの形式が使用されます。1 つは短い列名に使用し、もう 1 つは長い列名に使用します。下表に、各形式のヘッダー・レコードに含まれる情報を示します。

表 9. 短い列名のヘッダー・レコード情報

バイト位置	情報とタイプ
1-8	QMF オブジェクト形式レベル (8 文字のデータ) これらのバイト位置は、すべての列名が短い名前である場合に、DSQDC_SHORT_EXPT 大域変数が 1 に設定されると、REL 1.0 を示します。

表 9. 短い列名のヘッダー・レコード情報 (続き)

バイト位置	情報とタイプ
9-10	ヘッダー・レコード数 (ハーフワードの符号付き整数)
11-12	データ列数 (ハーフワードの符号付き整数)
13-30	列名 最大列幅は 18 バイトです。
31-32	データ・タイプ (ハーフワードの符号付き整数) データ・タイプ・コードについては、77 ページの表 11 参照。このフィールドは、各データ・タイプの表に示されている 10 進コードと同等の 16 進コードを格納します。
33-34	列幅 (ハーフワード符号付き整数)。大部分のデータ・タイプでは、バイト数で表した列幅です。以下に例外を示します。 <ul style="list-style-type: none"> DECIMAL 列の場合、ハーフワードの最初のバイトが精度を表し、2 番目のバイトが位取りを表す。 GRAPHIC 列と VARGRAPHIC 列の場合、この値は 2 バイト文字の幅を示す。 FLOAT 列の場合、この値は単精度浮動小数点を示す 4 であるか、倍精度浮動小数点を示す 8 である。 DECFLOAT 列では、この値は長形式の場合は 8、拡張形式の場合は 16 です。
35	NULL の許可。NULL が許可される場合は Y、許可されない場合は N (1 文字のデータ)
36	未使用バイト

バイト 13-36 により記述されたブロックは、データにある数と同じ数の列の数について、繰り返します。

表 10. 長い列名のヘッダー・レコード情報

バイト位置	情報とタイプ
1-8	QMF オブジェクト形式レベル (8 文字のデータ) これらのバイト位置は、DSQDC_SHORT_EXPT 大域変数が 0 に設定されると、REL 3.0 を示します。
9-10	ヘッダー・レコード数 (ハーフワードの符号付き整数)
11-12	データ列数 (ハーフワードの符号付き整数)
13-42	列名 名前のデフォルトの最大長は、30 バイトです。ただし、DSQDC_SHORT_EXPT 大域変数を使用して、データをエクスポートする前に最大列幅を 18 バイトに設定することができます。その場合は、短い列名のヘッダー・レコード形式が使用されます。
43-44	データ・タイプ (ハーフワードの符号付き整数) データ・タイプ・コードについては、77 ページの表 11 参照。このフィールドは、各データ・タイプの表に示されている 10 進コードと同等の 16 進コードを格納します。

表 10. 長い列名のヘッダー・レコード情報 (続き)

バイト位置	情報とタイプ
45-46	<p>列幅 (ハーフワード符号付き整数)。大部分のデータ・タイプでは、バイト数で表した列幅です。以下に例外を示します。</p> <ul style="list-style-type: none"> • DECIMAL 列の場合、ハーフワードの最初のバイトが精度を表し、2番目のバイトが位取りを表す。 • GRAPHIC 列と VARGRAPHIC 列の場合、この値は 2 バイト文字の幅を示す。 • FLOAT 列の場合、この値は単精度浮動小数点を示す 4 であるか、倍精度浮動小数点を示す 8 である。 • DECFLOAT 列では、この値は長形式の場合は 8、拡張形式の場合は 16 です。 <p>デフォルトの最大列幅は 30 バイトです。ただし、DSQDC_SHORT_EXPT 大域変数を使用して、データをエクスポートする前に最大列幅を 18 バイトに設定することができます。その場合は、短い列名のヘッダー・レコード形式が使用されます。</p>
47	NULL の許可。NULL が許可される場合は Y、許可されない場合は N (1 文字のデータ)
48	未使用バイト

バイト 13 から 48 に記述されたブロックは、データに含まれている数と同じ列の数だけ繰り返されます。

データ・タイプ・コードについては、この表に記載されています。最初の列に示されている 16 進コードは、3 列目に示された各データ・タイプを示すために使用されます。

表 11. データ・タイプ・コード

16 進数のコード	10 進数のコード	データ・タイプ	意味
X'180'	384	DATE	日付
X'184'	388	TIME	時刻
X'188'	392	TIMESTAMP	タイム・スタンプ
X'1C0'	448	VARCHAR	可変長文字
X'1C4'	452	CHAR	固定長文字
X'1D0'	464	VARGRAPHIC	可変長図形
X'1D4'	468	GRAPHIC	固定長図形
X'1E0'	480	FLOAT	浮動小数点
X'1E4'	484	DECIMAL	10 進数
X'1EC'	492	BIGINT	大整数
X'1F0'	496	INTEGER	整数
X'1F4'	500	SMALLINT	短整数
X'38C'	908	VARBINARY	可変長バイナリー
X'390'	912	BINARY	固定長バイナリー

表 11. データ・タイプ・コード (続き)

16 進数のコード	10 進数のコード	データ・タイプ	意味
X'3E4'	996	DECFLOAT(16) および DECFLOAT(34)	長形式の 10 進浮動小数点と拡張形式の 10 進浮動小数点
X'990'	2448	TIMESTAMP WITH TIME ZONE	タイム・スタンプと時間帯

DATE、TIME、TIMESTAMP、または TIMESTAMP WITH TIME ZONE データ・タイプを含む列は常に、ISO 形式でエクスポートされます。

データ・レコード

データ・レコードは固定長 (FB) 形式であり、エクスポートされるデータが入っています。

データ・レコードの最大許容長は、7,000 バイトです。データ・レコードの長さは、レコードに含まれている各データ・タイプの幅の合計です。各データ・タイプの幅を計算するには、次の表を使用します。

表 12. データ・レコードの幅： 各列にバイト数を追加することにより、特定のデータ・レコードの幅を計算します。

データ・タイプ	NULL 標識	長さフィールド	SO/SI	データ
文字 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			ヘッダーの長さ (LIH)
日付 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			LIH
浮動小数点 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			8
10 進浮動小数点	2			長形式の値の場合は 8、 拡張形式の値の場合は 16
大整数 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			LIH

表 12. データ・レコードの幅 (続き): 各列にバイト数を追加することにより、特定のデータ・レコードの幅を計算します。

データ・タイプ	NULL 標識	長さフィー ルド	SO/SI	データ
整数 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			LIH
短整数 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			LIH
時刻 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			LIH
タイム・スタンプ 長形式の値の場合は 8、 拡張形式の値の場合は 16	2			LIH
タイム・スタンプと時間 帯	2	2		LIH
10 進数	2			(精度 + 2) // 2
図形	2		2	(LIH × 2)
可変文字	2	2		LIH
可変図形	2	2	2	(2 × LIH)
2 進	2			LIH
2 進変数	2	2		LIH

重要: LIH は、ヘッダー・レコード内でその列に与えられた幅です。

すべてのデータ・レコードには、2 バイトの標識情報があります。この情報は、次の表に示す値と意味を持つことがあります。

表 13. データ・レコードの妥当性を示す 16 進値

値	意味
X'0000'	この列には、有効なデータが入っている
X'FFFF' または X'FFFE'	この列には、NULL 値が入っている。この列内のデータは無意味

QMF 形式のデータ・オブジェクトの解釈

データ・レコードの長さが分かれば、ヘッダー・レコードの長さを計算することができます。

例えば、Q.STAFF 表から次のデータをエクスポートすると仮定します。

```

      ID  NAME                COMM
-----
     10  SANDERS              -
     20  PERNAL              612.45
  
```

この例では、各データ・レコードの長さは 23 バイトです。76 ページの表 10 では、最初の 12 バイトにレベル情報と番号情報が入っていることを示しています。

次の表に示すように、各列の幅を計算します。

表 14. 列の幅の計算

列名	データ・タイプ	列の幅 (ヘッダーの長さ)	列の幅
ID	SMALLINT	2	2 + 2 = 4
NAME	VARCHAR	9	2 + 2 + 9 = 13
COMM	DECIMAL (7,2)	7	(7 + 1)/2 + 2 = 6
		データ・レコードの長さ:	23

データの列ごとに 24 バイトがあり、3 つの列があります。したがって、この 3 つの列から構成されるデータ・オブジェクトの場合、ヘッダーは 84 バイトになります。

$$(12 + (24 \times 3) = 84)$$

各ヘッダー・レコードの長さは、データ・レコードと同じ 23 バイトです。ヘッダーの 84 バイトは、4 つの 23 バイトのヘッダー・レコードに渡って分散され、最後のレコードには空白が埋め込まれます。

次のサンプルでは、報告書のヘッダーおよびその 16 進表記を示します。40 は、16 進数コード・空白文字を表します。反転表示の番号は、サンプルの下の注に関連付けられています。

```

      R E L      3 . 0                I D
      1
1   D9 C5 D3 40 F1 4B F0 40 0004 0003 C9 C4 40 40 40 40 40 40 40 40
      2 3 4
      N N A M E
2   40 40 40 40 40 40 40 01F4 0002 D5 00 D5 C1 D4 C5 40 40 40 40 40
      5 6 7
      Y C O M M
3   40 40 40 40 40 40 40 01C0 0009 E8 00 C3 D6 D4 D4 40 40 40 40 40
      Y
4   40 40 40 40 40 40 40 40 01E4 07 02 E8 00 40 40 40 40 40 40 40
  
```

図 13. QMF 形式でエクスポートされたデータ・オブジェクトのサンプル・ヘッダー・レコード：40 は、16 進数コード・空白文字を表します。

次のサンプルでは、報告書のデータおよびその 16 進表記を示します。各バイト位置の意味の詳細については、76 ページの表 10 を参照してください。

```

          10          S A N D E R S
1  00 00 00 0A 00 00 00 07 E2 C1 D5 C4 C5 D9 E2 00 00 FF FF 00 00 00 40 40
      8          9          10
          20          P E R N A L
2  00 00 00 14 00 00 00 06 D7 C5 D9 D5 C1 D3 00 00 00 00 00 00 61 24 5C

```

図 14. QMF 形式でエクスポートされたデータ・オブジェクトのサンプル・データ・レコード

1 REL 3.0

オブジェクト形式レベル: 3.0

オブジェクト形式レベルは、このオブジェクトが使用するオブジェクト形式のバージョンを QMF に指示します。QMF オブジェクト形式を変更するたびに、レベル番号が変更されます。オブジェクト形式は、新しいリリースごとに変更されるわけではありません。

2 X'0004'

ヘッダー・レコード数: 4

3 X'0003'

データ列数: 3

4 X'C9 C4'

列名: ID

5 X'1F4'

データ・タイプ: SMALLINT

6 X'0002'

列幅: 2

7 X'D5'

NULL の使用: N は NO を意味する

8 X'0A'

最初のデータ・レコードの最初の列の値: 10

9 X'07'

最初のデータ・レコードの 2 番目の列の名前の長さ: 7

10 X'FFFF'

標識情報: 列には NULL 値が入る

IXF 形式でのデータまたは表のエクスポート

DATAFORMAT=IXF オプションを指定した EXPORT コマンドを使用してデータ・オブジェクトまたは表をエクスポートすると、データは統合交換形式 (IXF) でエクスポートされます。QMF は、IXF のサブセットをサポートします。

エクスポート後のデータまたは表が含まれている TSO データ・セットまたは CICS データ・キューは、次のレコードで構成されています。

- ヘッダー・レコード (H)

- 表レコード (T)
- 列レコード (C)
- データ・レコード (D)

エクスポート後のデータ・セットまたは CICS データ・キューは、1 つの H レコードに続く、1 つの T レコードから構成されます。T レコードには、T レコードの後にある C レコードのカウントが入ります。表内の列ごとに、1 つの C レコードがあります。複数の C レコードの後に複数の D レコードが続きます。表内の行ごとに 1 つの D レコードがあります。表示されるレコードの配置は、次の図のようになります。

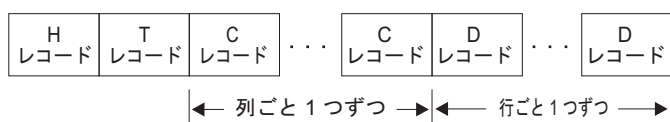


図 15. エクスポート後のデータ・セットまたは CICS データ・キュー内のレコードの配置 (IXF 形式)

ヘッダー・レコード (H)

IXF 形式のヘッダー・レコード (必須) は、データ・セットまたは CICS データ・キュー内の最初のレコードです。

ヘッダー・レコードは、文字データが含まれている 42 バイトのレコードです。H レコードの形式は、この表に示すとおりです。

表 15. エクスポート後のデータ・オブジェクトまたはデータベース表が含まれている IXF データ・セットまたはデータ・キュー内のヘッダー・レコードの部分

バイト位置	情報とタイプ
01	ヘッダー・レコード標識 (H)
02-04	TSO データ・セットまたは CICS データ・キューの ID
05-08	IXF バージョン。バージョンは、以下のいずれかのタイプです。 <ul style="list-style-type: none"> • 0000。短い列名 (18 文字以下) を含み、TIMESTAMP WITH TIME ZONE 列を含まないデータまたは表をサポートします。 • 0001。少なくとも 1 つの長い列名 (19 文字以上) を含み、TIMESTAMP WITH TIME ZONE 列を含まないデータまたは表をサポートします。 • 0002。短い列名 (18 文字以下) を含み、TIMESTAMP WITH TIME ZONE 列を少なくとも 1 つ含むデータまたは表をサポートします。 • 0003。少なくとも 1 つの長い列名 (19 文字以上) を含み、TIMESTAMP WITH TIME ZONE 列を少なくとも 1 つ含むデータまたは表をサポートします。
09-14	送り元のプロダクト名 (QMF)
15-20	送り元の製品リリース・レベル (VBR2M0)
21-28	データ・セットまたは CICS データ・キューの作成日付 (形式は YYYYMMDD)

表 15. エクスポート後のデータ・オブジェクトまたはデータベース表が含まれている IXF データ・セットまたはデータ・キュー内のヘッダー・レコードの部分 (続き)

バイト位置	情報とタイプ
29-34	データ・セットまたは CICS データ・キューの作成時刻 (形式は HHMMSS)
35-39	データ・セットまたは CICS データ・キュー内で最初の D (データ) レコードの前にあるレコード数 (文字形式で表される 5 桁の数値)
40	DBCS 標識。DBCS データの有無を示す。Y または N
41-42	ブランク

表レコード (T)

IXF 形式の表レコードは、ヘッダー・レコードの後にあります。IXF 形式のオブジェクトを含む各データ・セットまたはデータ・キューには、T レコードがなければなりません。表レコードには、EXPORT TABLE コマンドまたは EXPORT DATA コマンドを使用してエクスポートされたオブジェクトに関する表およびデータ情報を含みます。

IXF 形式でエクスポートされる表の名前は 18 文字で切り捨てられ、所有者名は 8 文字で切り捨てられます。照会を実行し、結果の DATA オブジェクトをエクスポートする場合、表レコードにはブランクの所有者と名前が含まれます。

次の表に、T レコードの形式を示します。

表 16. エクスポート後のデータ・オブジェクトまたはデータベース表が含まれている IXF データ・セットまたはデータ・キュー内の表レコードの部分

バイト位置	情報とタイプ
01	表レコード標識 (T)
02-03	データ名の長さ (18)
04-21	データを取り出す表の名前 (左寄せされ、右側にブランクが埋め込まれる) 表に名前がない場合、18 バイトのフィールド全体がブランクになる。
22-29	データ名修飾子。データを取り出すデータベース表の所有者の名前 表に所有者がない場合、8 バイトのフィールドがブランクになる。
30-41	データ・ソース (データベース)
42	データを記述するために使用される規則。列データの場合は C
43	データ形式。文字の場合は C (OUTPUTMODE=CHARACTER) 機械語の場合は M (OUTPUTMODE=BINARY)
44	データの場所。内部の場合は I
45-49	列 (C) レコードの数 (最初のデータ (D) レコードの前にある C レコードの数を示す文字形式の数値)
50-51	予約済み
52-81	ブランク

列レコード (C)

IXF 形式の列レコードは、列のデータ特性を記述します。表内の列ごとに 1 つの列レコードがあります。

18 文字より長い列名がある場合、列レコードの列名フィールドは 18 文字から 30 文字に増やす必要があります。ヘッダー・レコードで使用される IXF バージョン番号は、18 文字より長い列名があるかどうか、およびデータに **TIMESTAMP WITH TIME ZONE** 列が含まれているかどうかに応じて決まります。82 ページの表 15 は、各事例で使用される IXF バージョン番号を示します。

次の表に、**TIMESTAMP WITH TIME ZONE** 列を含まないデータまたは表の列レコードの形式 (IXF バージョン番号 0000 または 0001) を示します。**TIMESTAMP WITH TIME ZONE** 列が 1 つ以上含まれているデータまたは表 (IXF バージョン番号 0002 または 0003) については、85 ページの表 18 を参照してください。

表 17. **TIMESTAMP WITH TIME ZONE** 列を含まない IXF 形式 (IXF バージョン番号 0000 または 0001)

バイト位置	情報とタイプ
01	列レコード標識 (C)
02-03	列名の長さ
04-21	列名。データベースから入手されるか、または QMF によって生成される (列に、最初から名前がない場合) 名前は左寄せされ、必要に応じて、右側にブランクが埋め込まれます。
22 または 34	NULL が許可されるかどうかを示す標識 (Y または N)
23 または 35	列選択標識 (Y)
24 または 36	キー列標識 (Y)
25 または 37	データ・クラス (R)
26-28 または 38-40	データ・タイプ (データ・タイプ・コードについては、88 ページの表 20 を参照)
29-33 または 41-45	コード・ページ
34-38 または 46-50	予約済み
39-43 または 51-55	列データの長さ。文字形式の 10 進値 データ・タイプが DECIMAL の場合、最初の 3 バイトがデータの精度を表し、次の 2 バイトが位取りを表します。データ・タイプが BIGINT 、 INTEGER 、または SMALLINT の場合、長さはデータ・タイプに内在しているため、このフィールドはブランクです。

表 17. *TIMESTAMP WITH TIME ZONE* 列を含まない *IXF* 形式 (*IXF* バージョン番号 0000 または 0001) (続き)

バイト位置	情報とタイプ
44-49 または 56-61	列データの開始位置。文字形式の 10 進値 この値は、データ・レコードの開始から列のデータのオフセットを反映します。 列で NULL が許可される場合、このフィールドは NULL 標識を指す。列で NULL が許可されない場合、このフィールドはデータ自体を指す。列で NULL が許可されるかどうかに関係なく、NULL 標識用のスペースはレコード内に必ず存在する。開始位置は、データを含む最初のバイトからのオフセットである。したがって、データ (D) レコードの最初の 5 バイトは、実データの開始位置に関して考慮されない。(最初のデータ位置は、桁 0 ではなく、桁 1 である。)
50-79 または 62-91	該当する場合は、列ラベル情報 (該当しない場合、これらのバイト位置にはブランクが含まれる)
80-81 または 92-93	文字形式の 2 バイトのゼロ (00)

データまたは表が 1 つ以上の *TIMESTAMP WITH TIME ZONE* 列を含む場合、列レコードの形式は次のとおりです。

表 18. 1 つ以上の *TIMESTAMP WITH TIME ZONE* 列を含む *IXF* 形式 (*IXF* バージョン番号 0002 または 0003)

バイト位置	情報とタイプ
01	列レコード標識 (C)
02-03	列名の長さ
04-21	列名。データベースから入手されるか、または QMF によって生成される (列に、最初から名前がない場合) 名前は左寄せされ、必要に応じて、右側にブランクが埋め込まれます。
22 または 34	NULL が許可されるかどうかを示す標識 (Y または N)
23 または 35	列選択標識 (Y)
24 または 36	キー列標識 (Y)
25 または 37	データ・クラス (R)
26-29 または 38-41	データ・タイプ (データ・タイプ・コードについては、88 ページの表 20 参照)
30-34 または 42-46	コード・ページ
34-38 または 47-50	予約済み

表 18. 1 つ以上の *TIMESTAMP WITH TIME ZONE* 列を含む *IXF* 形式 (*IXF* バージョン番号 0002 または 0003) (続き)

バイト位置	情報とタイプ
39-43 または 51-55	列データの長さ。文字形式の 10 進値 データ・タイプが <i>DECIMAL</i> の場合、最初の 3 バイトがデータの精度を表し、次の 2 バイトが位取りを表します。データ・タイプが <i>BIGINT</i> 、 <i>INTEGER</i> 、または <i>SMALLINT</i> の場合、長さはデータ・タイプに内在しているため、このフィールドはブランクです。
44-49 または 56-61	列データの開始位置。文字形式の 10 進値 この値は、データ・レコードの開始から列のデータのオフセットを反映します。 列で <i>NULL</i> が許可される場合、このフィールドは <i>NULL</i> 標識を指す。列で <i>NULL</i> が許可されない場合、このフィールドはデータ自体を指す。列で <i>NULL</i> が許可されるかどうかに関係なく、 <i>NULL</i> 標識用のスペースはレコード内に必ず存在する。開始位置は、データを含む最初のバイトからのオフセットである。したがって、データ (<i>D</i>) レコードの最初の 5 バイトは、実データの開始位置に関して考慮されない。(最初のデータ位置は、桁 0 ではなく、桁 1 である。)
50-79 または 62-91	該当する場合は、列ラベル情報 (該当しない場合、これらのバイト位置にはブランクが含まれる)
80-81 または 92-93	文字形式の 2 バイトのゼロ (00)

データ・レコード (D)

IXF 形式のデータ・レコードは、変数ブロック (*VB*) 形式です。表内の行ごとに 1 つのデータ・レコードがあります。

この表では、データ・レコードの形式を示します。

表 19. エクスポート後のデータ・オブジェクトまたは表が含まれている *IXF* データ・セットまたはデータ・キュー内のデータ・レコードの形式

バイト位置	情報とタイプ
01	データ・レコード標識 (<i>D</i>)
02-04	予約済み
05	ブランク
06- レコード終わり	表レコードのバイト 43 が <i>M</i> (機械語) または <i>C</i> (文字) のいずれであるかによって、2 進数形式または文字形式である行データ バイト 6 は最初の列の行データの開始点 (桁 1) を示す。

列データ形式

次の図に示すように、 n 個の列用の D レコードのデータは、隣り合わせに配置されます。

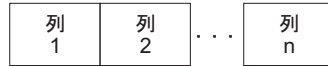


図 16. D レコード内の列データの形式

列ごとに、データは NULL 標識に続くデータ自体から構成されます。NULL が許可されている (C レコードのバイト 22 またはバイト 34 に値 Y がある) 場合、各 C レコードのバイト 44-49 またはバイト 56-61 は、その列のデータの前にある NULL 標識を指します。NULL が許可されていない (C レコードのバイト 22 または 34 に値 N がある) 場合、バイト 44-49 またはバイト 56-61 はデータ自体を指します。ただし、後者の場合でも、NULL 標識用のスペースはデータ・レコードに残されます。バイト 44-49 または 56-61 で、最初の位置は値 1 で示されます。これは、D レコードのバイト 6 を指します (バイト 1 から 5 は無視されます)。

NULL 標識の表示は、OUTPUTMODE に何を指定したか (文字または 2 進数) によって、異なります。OUTPUTMODE については、T レコードのバイト 43 に、文字の場合は C が、機械語 (2 進数) の場合は M が示されます。データ形式が文字の場合、1 バイトが NULL 標識に使用されます。

- ダッシュ (-) は、データが NULL であることを示す。
- ブランクは、データが NULL でないことを示す。

データ形式が 2 進数の場合、2 バイトが NULL 標識に使用されます。

- X'FFFF' はデータが NULL であることを示す。
- X'0000' はデータが NULL でないことを示す。

NULL 標識について詳しくは、IXF 形式のオブジェクトを解釈するために役立つ、以下の例を参照してください。

データ・タイプ別の列データの形式

D レコード内の IXF 列データの長さや形式は、OUTPUTMODE に応じて異なる可能性があります。

この表は、文字およびバイナリー・エクスポート形式の両方の各データ・タイプに対して D レコードの列データの長さや形式を示します。表では、IXFCLENG は C レコードのバイト 39-43 または 51-55 の内容 (列データの長さ) を示しています。

表 20. データ・タイプ別の IXF 列データの形式

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
384	DATE	<p>このデータ・タイプでは、IXFCLENG の値は意味がない。 長さ (10 バイト) はデータ・タイプによって決まる。</p> <p>形式は次のとおり。</p> <p style="text-align: center;"><i>yyyy-mm-dd</i></p> <p><i>yyyy</i> は年、<i>mm</i> は月、<i>dd</i> は日を示す。 <i>yyyy</i>、<i>mm</i>、および <i>dd</i> は数字でなければならない。先行ゼロは省略してはならない。 <i>yyyy</i> の有効範囲は 0001-9999、<i>mm</i> は 01-12 です。 <i>dd</i> の範囲は月によって異なる。例えば、以下の値は 2002 年 2 月 28 日を指定します。</p> <p>2002-02-28</p>	文字形式と同じ
388	TIME	<p>このデータ・タイプでは、IXFCLENG の値は意味がない。 長さ (8 バイト) はデータ・タイプによって決まる。</p> <p>形式は次のとおり。</p> <p style="text-align: center;"><i>hh.mm.ss</i></p> <p><i>hh</i> は 24 時間形式の時、<i>mm</i> は分、<i>ss</i> は秒を示す。 <i>hh</i>、<i>mm</i>、および <i>ss</i> はすべて数字でなければならない。先行ゼロは省略してはならない。可能な範囲は次のとおり。</p> <ul style="list-style-type: none"> • <i>hh</i> は 00 - 23 • <i>mm</i> は 00 - 59 • <i>ss</i> は 00 - 59 <p>真夜中を示す特殊値 24.00.00 も有効である。例:</p> <p>10.37.42 は 10:37:42 AM 08.00.00 は 8 AM ちょうど 23.30.00 は 11:30 PM</p>	文字形式と同じ

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
392	TIMESTAMP	<p>TIMESTAMP(0) の長さは 19 です。TIMESTAMP(<i>n</i>) の長さは 20+<i>n</i> です。 <i>n</i> は 1 から 12 までの数字です。例えば、TIMESTAMP(12) として定義された列の長さは 32 です。</p> <p>形式は次のとおり。</p> <p><i>yyyy-mm-dd-hh.mm.ss.nnnnnnnnnnn</i></p> <p><i>yyyy</i> は年、<i>mm</i> は月、<i>dd</i> は日、<i>hh</i> は 24 時間形式の時間、<i>mm</i> は分、<i>ss</i> は秒、<i>nnnnnnnnnnnn</i> は小数秒を示します。年、月、日、時、分、および秒の有効な範囲は、DATE と TIME の各データ・タイプと同じである。</p> <p>例:</p> <p>2010-12-31-23.59.59.999999999999 (2010 年の最後の小数秒)</p> <p>2010-01-01-00.00.00.000000000001 (2010 年の最初の小数秒)</p> <p>24.00.00.000000000000 は、有効なタイム・スタンプの時刻部分である。</p>	文字形式と同じ

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
448	VARCHAR LONG VARCHAR	IXFLENG は、文字ストリングの最大長である。データ長は、5 バイトの文字カウント・フィールドと、その後続く IXFLENG によって示される N バイトから構成される。(N の有効範囲は 0-32704 であり、カウント・フィールドの有効範囲は 0-N である)。カウント・フィールドによって示された文字数だけが有効で、残りの部分は無意味である。例えば、IXFLENG=00010 の場合、データ形式は次のようになる。 00005JONESxxxx この形式では、各 x は空白文字 (X'40') である。	IXFLENG は、文字ストリングの最大長である。データ長は、2 バイトの 2 進数カウント・フィールドと、その後続く IXFLENG によって示される N バイトから構成される。(N の有効範囲は 1-32704 であり、カウント・フィールドの有効範囲は 0-N である)。カウント・フィールドによって示された文字数だけが有効で、残りの部分は無意味である。例えば、IXFLENG=00010 の場合、データ形式は次のようになる。 nnJONESxxxx この形式では、nn=X'0005'、各 x は空白文字 (X'40') である。
452	CHAR	IXFLENG は、文字ストリングの長さである。データ長は、IXFLENG の N バイトによって示される。(N の有効範囲は 1-254 である)。例えば、IXFLENG=00005 の場合、データ形式は次のようになる。 JONES この形式で JONES は、C レコードのバイト 44-49 または 56-61 によって示される、5 バイトの文字ストリング	文字形式と同じ

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
464	VARGRAPHIC LONG VARGRAPHIC	<p>IXFLENG は 2 バイト文字の最大数 ($2 \times n$ バイト)。データ長は、5 バイトの文字カウント・フィールドと、IXFLENG によって示されたバイト数の 2 倍と 2 (シフト文字用) の合計である。カウント・フィールド内の 2 バイト文字数は、データの直前にシフトアウト (X'0E') があり、直後にシフトイン (X'0F') がある場合に有効である。残りの部分は無意味である。 (n の有効範囲は 1-16352 で、カウント・フィールドの場合、有効範囲は 0-n です。) 例えば、IXFLENG=00006 の場合、データ形式は次のようになる。</p> <p>00003oZZYXX<i>xxxxxx</i></p> <p>この形式で、文字「o」はシフトアウト、「i」はシフトイン、各 x はブランク文字 (X'40') である。</p>	<p>データ長は、2 バイトの 2 進カウント・フィールドと、その後続く IXFLENG によって示されたバイト数の 2 倍から構成される。</p> <p>(IXFLENG の有効範囲は 1-16352 であり、カウント・フィールドの有効範囲は 0-IXFLENG である。) カウント・フィールド内の 2 バイト文字数は有効である。周りを囲むシフトアウト文字とシフトイン文字はない。残りの部分は無意味である。例えば、IXFLENG=00008 の場合、データ形式は次のようになる。</p> <p>nnZZYXXWW<i>xxxxxxxx</i></p> <p>この形式では、$nn=X'0004'$、各 x はブランク文字 (X'40') である。</p>
468	GRAPHIC	<p>IXFLENG は 2 バイト文字の数 ($2 \times N$ バイト)。データ長は、$2 \times N$ バイトと、データの直前のシフトアウト (X'0E') と、データの直後のシフトイン (X'0F') から構成される。例えば、IXFLENG=00005 の場合、データ形式は次のようになる。</p> <p>oZZYXXWWV<i>i</i></p> <p>この形式で、文字「o」はシフトアウト、「i」はシフトインである。</p>	<p>データ・ストリングに、周りを囲むシフトイン文字とシフトアウト文字がないことを除いて、文字形式と同じである。</p> <p>例えば、IXFLENG=00005 の場合、データ形式は次のようになる。</p> <p>ZZYXXWWV</p>

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
480	FLOAT	<p>IXFCLENG の値は単精度の場合は 4、倍精度の場合は 8 です。データの長さや形式は、値の精度に応じて決まります。</p> <p>単精度値は、以下の形式で最大 14 文字から構成されます。 <i>sn.nnnnnnnEsee</i></p> <p>この形式では、</p> <ul style="list-style-type: none"> • <i>s</i> はオプションの符号 (正符号、負符号、または先頭の場合、数字が正の場合は空白) です。 • <i>n</i> は、最大 8 桁の仮数部の桁を表します。小数点は仮数部の 2 桁目でなければなりません。 • <i>E</i> は指数の先頭を表します。 • <i>e</i> は指数の桁を表します。1 桁または 2 桁になります。 <p>値は +/-5.4E-79 から +/-7.2E+75 の範囲内にある必要があります。</p> <p>例: -1.2345679E+07 6.2345679E-01 0.0000000E+00</p> <p>倍精度の値は、以下の形式で最大 23 文字から構成されます。 <i>sn.nnnnnnnnnnnnnnnnnEsee</i></p> <p>この形式では、</p> <ul style="list-style-type: none"> • <i>s</i> はオプションの符号 (正符号、負符号、または先頭の場合、数字が正の場合は空白) です。 • <i>n</i> は最大 18 桁の仮数部の桁を表します。小数点は仮数部の 2 桁目でなければなりません。 • <i>E</i> は指数の先頭を表します。 • <i>e</i> は指数の桁を表します。1 桁または 2 桁になります。 <p>値は +/-5.4E-79 から +/-7.2E+75 の範囲内にある必要があります。</p> <p>例: -1.2345678901234568E+14 6.23456789012345678E-01 0.0000000000000000E+00</p>	<p>IXFCLENG の値は単精度の場合は 4、倍精度の場合は 8 です。このデータは、単精度の 4 バイト浮動小数点値および倍精度の 8 バイト浮動小数点値から構成されます。</p>

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
484	DECIMAL	<p>C レコードのバイト 39-43 または 51-55 は、数値の精度 P (最初の 3 バイト) と位取り S (次の 2 バイト) を示す。P の有効範囲は 0-15 である。S は P 以下の任意の値である。</p> <p>データは、P+2 バイト (S=0 の場合は P+1 バイト) の文字値として形式設定され、右揃えされ、先頭のバイトは符号のために予約され、小数点 (S により暗黙指定されている桁) は S がゼロ以外の場合にのみ存在する。例えば、P=005 かつ S=00 の場合、データ形式は次のようになる。</p> <p>12345</p> <p>P=006 かつ S=02 の場合、データ形式は次のようになる。</p> <p>+2345.10</p> <p>P=004 かつ S=03 の場合、データ形式は次のようになる。</p> <p>-8.515</p>	<p>C レコードのバイト 39-43 または 51-55 は、数値の精度 P (最初の 3 バイト) と位取り S (次の 2 バイト) を示す。P の有効範囲は 0-15 である。S は P 以下の任意の値である。</p> <p>データは、パック 10 進数形式の (P+2)/2 バイト 10 進値からなる。最後のバイトは値の符号を示す。例えば、P=005 かつ S=00 の場合、データ形式は次のようになる。</p> <p>X'12345C'</p> <p>P=006 かつ S=02 の場合、データ形式は次のようになる。</p> <p>X'0234510D'</p>
492	BIGINT	<p>このデータ・タイプでは、IXFCLENG の値は意味がない。データの長さや形式は、データ・タイプによって決まる。データは、20 バイトの文字値で、右寄せされ、最初の文字が符号用に予約されている。例:</p> <p>0000000000000000033 +9223372036854775807 -9223372036854775808</p>	<p>IXFCLENG の値は意味がない。データの長さや形式は、データ・タイプによって決まる。データは 8 バイトの 2 進数値から構成される。</p>
496	INTEGER	<p>このデータ・タイプでは、IXFCLENG の値は意味がない。データの長さや形式は、データ・タイプによって決まる。</p> <p>データは、11 バイトの文字値で、右寄せされ、最初の文字が符号用に予約されている。例:</p> <p>0000000013 +1187642200 -0033588727</p>	<p>IXFCLENG の値は意味がない。データの長さや形式は、データ・タイプによって決まる。</p> <p>データは 4 バイトの 2 進数値から構成される。</p>

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
500	SMALLINT	<p>このデータ・タイプでは、IXFLENG の値は意味がない。データの長さや形式は、データ・タイプによって決まる。</p> <p>データは、6 バイトの文字値で、右寄せされ、最初の文字が符号用に予約されている。例:</p> <pre>00023 +00763 -21311</pre>	<p>IXFLENG の値は意味がない。データの長さや形式は、データ・タイプによって決まる。</p> <p>データは 2 バイトの 2 進数値から構成される。</p>
908	VARBINARY	適用外	<p>次のことを除いて、VARCHAR と同じ</p> <ul style="list-style-type: none"> IXFLENG は、バイナリー・ストリングの最大長 (バイト数) である。 N の有効範囲は 0-32704 である。
912	BINARY	適用外	<p>次のことを除いて、CHAR と同じ</p> <ul style="list-style-type: none"> IXFLENG は、バイナリー・ストリング・シーケンスの長さ (バイト数) である。 N の有効範囲は 1-255 である。

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
996	DECFLOAT	<p>IXFCLENG の値は長形式の場合は 16、拡張形式値の場合は 34 です。データの長さや形式は、値の精度に応じて決まります。</p> <p>長形式は、以下の形式で最大 23 文字から構成されます。</p> <p><i>sn.nnnnnnnnnnnnnnnEseee</i></p> <p>この形式では、</p> <ul style="list-style-type: none"> • <i>s</i> はオプションの符号 (正符号、負符号、または先頭の場合、数字が正の場合は空白) です。 • <i>n</i> は、最大 16 桁の仮数部の桁を表します。小数点は仮数部の 2 桁目でなければなりません。 • <i>E</i> は指数の先頭を表します。 • <i>e</i> は、最大 3 桁の指数部の桁を表します。 <p>この値は、 +/-1.0E-398 から +/-1.0E+384 の範囲である必要があります。</p> <p>例:</p> <p>-1.234567890123456E+003 1.234567890123456E-015 0.000000000000000E+000</p> <p>拡張形式値は、以下の形式で最大 42 文字から構成されます。</p> <p><i>sn.nnnEseeee</i></p> <p>この形式では、</p> <ul style="list-style-type: none"> • <i>s</i> はオプションの符号 (正符号、負符号、または先頭の場合、数字が正の場合は空白) です。 • <i>n</i> は、最大 34 桁の仮数部の桁を表します。小数点は仮数部の 2 桁目でなければなりません。 • <i>E</i> は指数の先頭を表します。 • <i>e</i> は、最大 4 桁の指数部の桁を表します。 <p>この値は、 +/-1.0E-6176 から +/-1.0E+6144 の範囲である必要があります。</p> <p>例:</p> <p>-1.234567890123456789012345678901234E+0033 4.321098765432109876543210987654321E-0001</p>	<p>IXFCLENG の値は長形式の値の場合は 8、拡張形式の値の場合は 16 です。このデータは、長形式の場合は 8 バイトの浮動小数点値で、拡張形式の場合は 16 バイトの 10 進浮動小数点値から構成されます。</p>

表 20. データ・タイプ別の IXF 列データの形式 (続き)

データ・タイプ・コード	データ・タイプ	データ長情報 (OUTPUTMODE = CHARACTER の時)	データ長情報 (OUTPUTMODE = BINARY の時)
2448	TIMESTAMP WITH TIME ZONE	<p>長さは TIMESTAMP(0) WITH TIME ZONE の場合は 147、TIMESTAMP(n) WITH TIME ZONE の場合は 148 + n (n = 1 から 12)。</p> <p>形式は次のとおり。</p> <p><i>yyyy-mo-dd-hh-mm-ss-nnnnnnnnnnzh:tm</i></p> <p><i>yyyy-mo-dd-hh-mm-ss</i> は、TIMESTAMP データと同じ方法でタイムスタンプを指定します。</p> <ul style="list-style-type: none"> • <i>nnnnnnnnnn</i> は 0 から 12 桁の数値で、小数秒の数を表します。 • <i>z</i> は、以前はグリニッジ標準時 (GMT) と呼ばれた協定世界時 (UTC) に相対的な時間帯オフセットを示す正符号 (+) または負符号 (-) です。 • <i>th</i> は時間帯の時を表す 2 桁の値です。 • <i>tm</i> は時間帯の分を表す 2 桁の値です。 <p>形式の時間帯部分の有効な範囲は、-24:00 から +24:00 です。UTC を指定するには、時間帯として -0:00 または +0:00 を指定するか、あるいは時間帯オフセットとその符号を大文字 Z で置き換えます。</p> <p>例えば、2010-09-30-13.08.36.123456654321-08:00 は、米国のカリフォルニア州サンホセでは、2010 年 9 月 30 日の 1:08 P.M. 36.123456654321 秒を示します。タイム・スタンプ 2010-09-30-13.08.36.123456654321Z は、UTC が有効な場所であれば必ず、時刻 1:08 P.M. 36.123456654321 秒を示します。</p>	文字形式と同じ

IXF 形式でエクスポートされたオブジェクトの解釈

次の例は、IXF 形式でエクスポートされたデータを解釈するために役立ちます。

QMF 形式のデータ・オブジェクトの例に示した表を、IXF 形式を使用 (OUTPUTMODE=CHARACTER を指定) してエクスポートするとします。エクスポートされた表は、次のようになります。

```

      ID  NAME          COMM
-----
     10  SANDERS        -
     20  PERNAL          612.45
    
```

エクスポートされたデータ・セットまたは CICS データ・キューは、次のように合計 7 つのレコード (1 つの H レコード、1 つの T レコード、3 つの C レコード、および 2 つの D レコード) で構成されます。

```

HIXF0000QMF  VAR1M02010120409560000005N
T18          database          CCI00003
C02ID        NYNR50000000      000002      00
    
```

```

C04NAME          YYNR44800000      00009000008      00
C04COMM          YYNR48400000      00702000023      00
D      00010 00007SANDERSxx -
D      00020 00006PERNALxxx 00612.45

```

印刷不可能な 2 進数文字は「x」の文字で示してあります。この図は、これらのレコードの詳細を示しています。

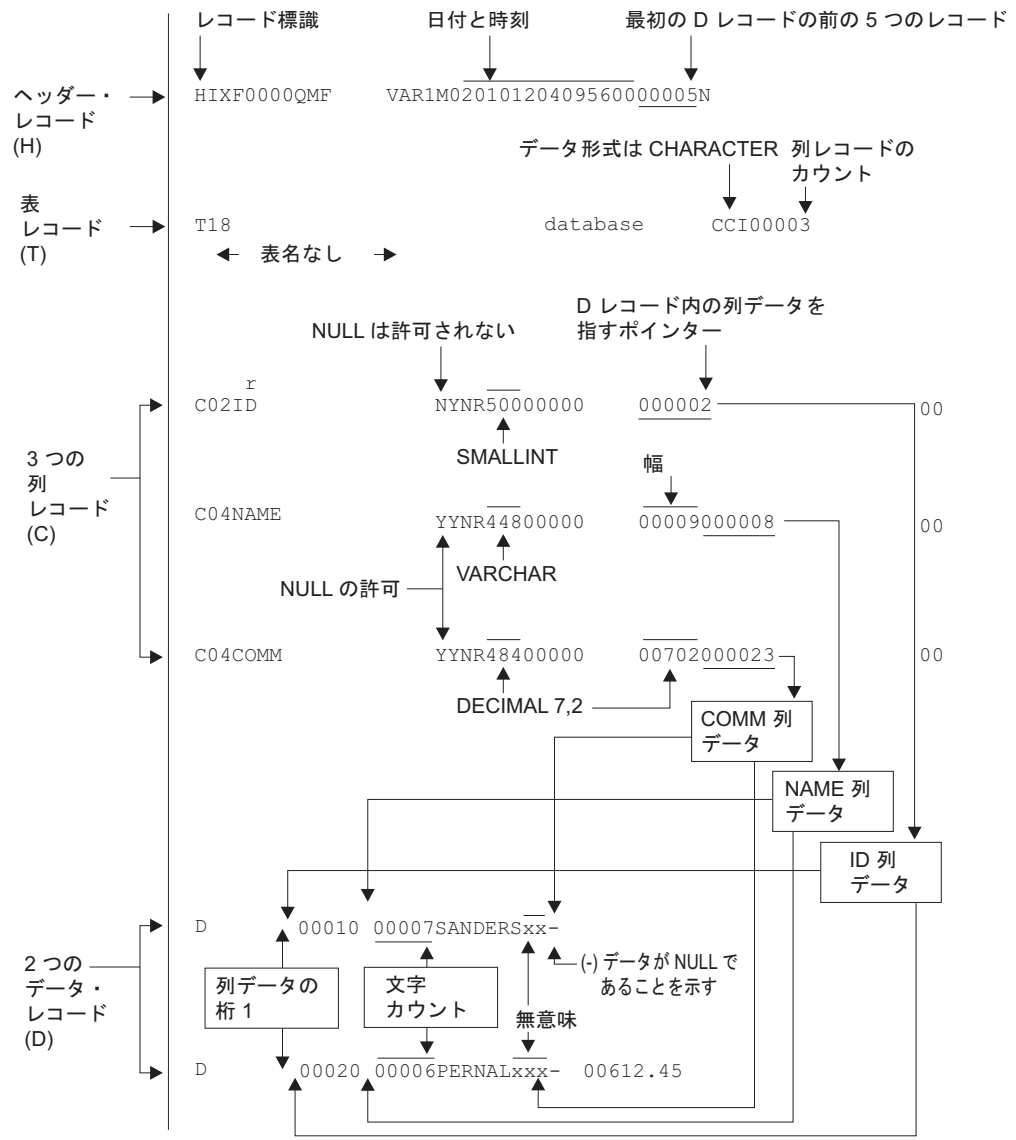


図 17. サンプル IXF レコードの形式 (OUTPUTMODE=CHARACTER)

IXF 形式を使用して同じ表をエクスポートしますが、OUTPUTMODE=BINARY を指定するとします。エクスポート後の データ・セットまたは CICS データ・キューは、この表に示されている 7 つのレコードから構成されます。

```

HIXF0000QMF  VAR1M02010120409565000005N
T18          database          CMI00003
C02ID          NYNR50000000      000003      00

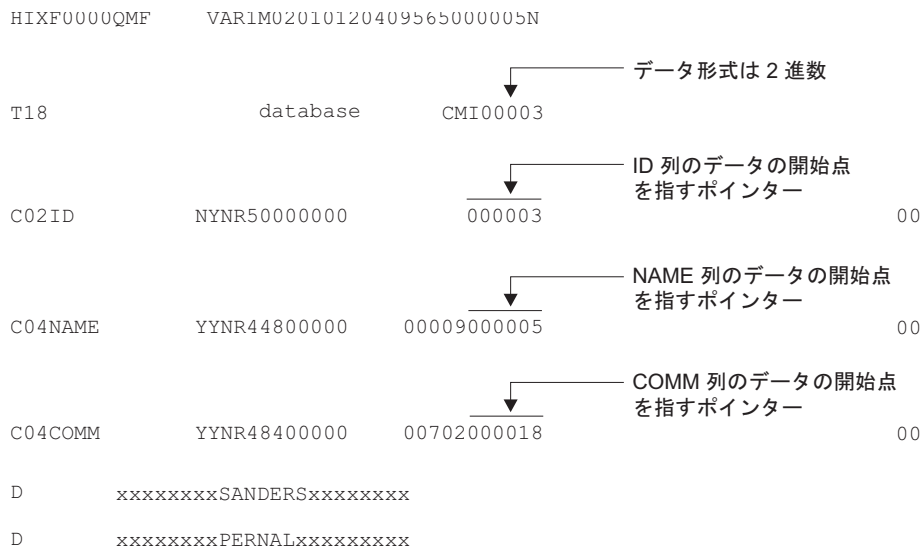
```

```

C04NAME          YYNR44800000      000090000005      00
C04COMM          YYNR48400000      007020000018      00
D   xxxxxxxxSANDERSxxxxxxxx
D   xxxxxxxxPERNALxxxxxxxx

```

バイト 44 から 49、または 56 から 61 (列データの開始位置) を除いて、H、T、および C レコード内の情報は本質的に同じです。しかし、D レコード内のデータは、大いに異なります。次の図は、エクスポートされたデータ・セットまたは CICS データ・キューのレコードの詳細を示しています。



2つのデータ (D) レコードを、各種フィールドの記述と共に、16進表記で以下に示す。

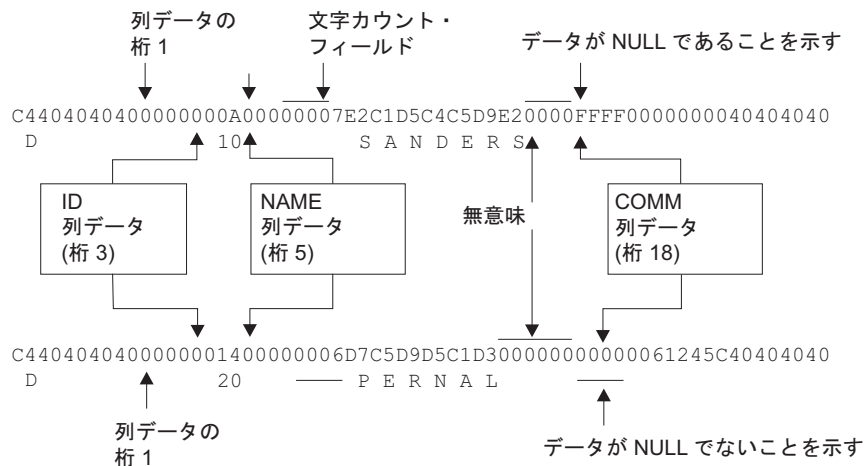


図 18. サンプル IXF レコードの形式 (OUTPUTMODE=BINARY)

XML 形式でのデータまたは表のエクスポート

データまたは表に XML 列または LOB データが含まれている場合は、EXPORT DATA または EXPORT TABLE コマンドで DATAFORMAT=XML 節を使用する

必要があります。この形式はまた、エクスポートされるデータまたは表が XML 列または LOB データを含まない時に使用することもできます。

制約事項: XML データ・タイプをサポートするデータベース・リリースに接続している場合にのみ、QMF は、XML データの操作をサポートします。

データまたは表を XML 形式でエクスポートすると、データはコマンドで指定した UNIX ファイル、TSO データ・セット、または CICS データ・キューにエクスポートされます。QMF は、データのインポートおよびエクスポート時の XML 1.0 指定 (エディション 4) を使用します。

QMF は、エクスポートまたはインポートで XML データを処理するときに、z/OS XML 構文解析サービスと z/OS Unicode 変換サービスを使用します。そのため、これらのサービスは構成済みでアクティブでなければなりません。

XML の例に示されているすべてのタグは、XML 列データがインポートされる前に存在している必要があります。これらのタグは、ファイルを構文解析するために QMF によって使用されるため、ファイル、データ・セット、または CICS データ・キューの内容に含まれている必要があります。QMF によってファイルの終わりで <extensions> タグが検出されると、カーソルが閉じ、インポートが終了します。このタグを変更または削除すると、データの読み取りが無期限に続行されることになります。

データは、CCSID 1208 の Unicode UTF-8 形式の XML 文書としてエクスポートされます。エクスポートされた XML データ・セットまたは CICS データ・キューは、ヘッダー・レコード、結果セットを定義するレコード、データまたは表内の各列のメタデータ・レコード、およびエクスポートされたデータまたは表内の各行のデータ・レコードで構成されます。

ヘッダー・レコード

エクスポートされた XML ファイル内のヘッダー・レコードには、使用されている XML のバージョン、エンコード・スキーム、およびエクスポートされた XML 文書のフォーマット設定に使用するスタイル・シートが含まれています。

次の例は、エクスポートされたファイルのヘッダー・レコードに含まれる情報のタイプを示しています。

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!-- ?xml-styleSheet type="text/xml" href="qmf.xmlt" ? -->
```

QMF には、エクスポートされた XML データを形式設定するために使用できるスタイル・シートが用意されています。デフォルトのスタイル・シート (デフォルト名は qmf.xmlt) は、QMF サンプル・データ・セットのメンバー DSQ1STSH として用意されています。このサンプル・データ・セットは QMF1210.SDSQSAP n (n は各国語 ID) という名前です。このデフォルト・スタイル・シートを、エクスポートされたファイルがある場所にコピーすると、その仕様に従って形式設定された状態で XML 文書を開くことができます。スタイル・シートに別の名前を使用する場合は、新しいスタイル・シート名を参照するように、エクスポートされたファイル内のヘッダーを変更します。

結果セットを定義するレコード

結果セットの定義には、XML ファイルで使用される QMF スキーマ・ファイルの名前空間定義およびスキーマ定義が含まれています。

この例には、7 列を含むエクスポート後の XML サンプル・ファイルに結果セットの定義のレコードが示されています。

```
<DataSet xmlns="http://www.ibm.com/QMF" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
<ResultSet>
<Metadata>
  <SourceDescription />
  <ColumnsAmount>7</ColumnsAmount>

  ....Definitions for each column go here.

</Metadata>
<Data>

  ....Data for each row goes here.

</Data>
</ResultSet>
<Extensions />
</DataSet>
```

メタデータ・レコード

エクスポートされた XML ファイル内の列のメタデータは、列数、列名、列ラベル (該当する場合)、データ・タイプ、データ長、データが NULL かどうかの指標、および形式で構成されています。

次に、「ID」という名前の列のメタデータの例を示します。エクスポート後の XML ファイルには、列ごとに 1 つ、列説明のブロックが含まれています。

```
<ColumnDescription id="1">
  <Name>ID</Name>
  <Label>ID</Label>
  <Type>smallint</Type>
  <Width>2</Width>
  <Nullable>false</Nullable>
  <Format>plain</Format>
</ColumnDescription>
```

データ・レコード

XML エクスポート・ファイルには、エクスポート・データの行ごとに 1 つ、行定義ブロックが含まれています。データ・レコードは、VBS (複数のブロックに渡って変数が格納される) 形式です。

Q.STAFF サンプル表の先頭行に示されているように、<cell> タグは行内の各列を番号で識別します。

```
<Row id="0">
  <Cell id="1">10</Cell>
  <Cell id="2">SANDERS</Cell>
  <Cell id="3">20</Cell>
  <Cell id="4">MGR</Cell>
  <Cell id="5">7</Cell>
  <Cell id="6">99999.99</Cell>
  <Cell id="7" null="1" />
</Row>
```


EXPORT DATA コマンドまたは EXPORT TABLE コマンドで DATAFORMAT=XML 節を使用するとき、データに XML データ・タイプで定義された列が含まれている場合は、QMF によって、その XML データが CDATA タグでラップされます。CDATA タグは、XML データがパーサーによって処理されないようにします。次に、エクスポートされたファイルに含まれている XML データの例を示します。

```
<Data>
<Row id="0">
<Cell id="1">Murphy</Cell>
<Cell id="2">1234</Cell>
<Cell id="3"[CDATA[xml version="1.0" encoding="utf-8?"]]Cell>
</Row>
</Data>
```

QMF での XML の検証方法

XML スキーマ文書には、XML 文書内の要素および属性を検証するために、XML 文書の構造が記述され、パラメーターが定義されています。

デフォルトのスキーマ・ファイルは、QMF サンプル・データ・セットのメンバー DSQ1SCHEM として QMF に用意されています。このサンプル・データ・セットは QMF1210.SDSQSAP n (n は各国語 ID) という名前です。このメンバーを、XML 文書が含まれているファイルの保管先のディレクトリーにコピーします。スキーマ文書に qmf_data.xsd という名前を付けます。これは、QMF for Workstation および QMF for WebSphere® 下のデフォルトのスキーマ文書の名前です。デフォルトのスキーマ・ファイルは、XML データの形式に対するお客様のビジネス要件に従って変更できます。qmf_data.xsd 以外の名前を使用する場合、または別のスキーマ・ファイルを使用する場合は、スキーマ文書を参照するすべてのファイル内に指定されているこの名前を変更してください。

CSV 形式でのデータまたは表のエクスポート

データまたは表は、CSV (コンマ区切り値) 形式でエクスポートできます。エクスポートしたデータはその後、ご使用のワークステーションにダウンロードできます。ワークステーションでは、CSV 形式のデータを、Microsoft Excel などのアプリケーションで使用できます。

データまたは表を CSV 形式でエクスポートするとき HEADER オプションを使用すると、データとともに列見出しをエクスポートするかどうかを指定できます。デフォルトでは、列見出しが組み込まれます。列見出しをエクスポートする場合は、DSQDC_COL_LABELS 大域変数の値によって、列ラベルまたは列名がエクスポートされるかどうか制御されます。DSQDC_COL_LABELS のデフォルトは 1 で、列ラベルがエクスポートされます。

エクスポートされたデータは、列区切り文字で区切られた列値の行として形式設定されます。HEADER=YES と指定した場合は、列区切り文字で区切られた列名からなる 1 行が、列値の行の前に配置されます。列区切り文字の値は、ユーザーが設定する Q.PROFILES.DECIMAL 値によって決まります。

CSV 形式でエクスポートされたデータは、DB2 for z/OS のローカルの SBCS コード化スキームでエンコードされます。

データ・オブジェクトと表のエクスポートおよびインポートに関する規則と情報

QMF は、データ・オブジェクトおよび表オブジェクトを一時記憶域にエクスポートします。その記憶域を割り振る方法には、いくつかの規則があります。QMF では、インポート・エラーも固有の方法で処理されます。

UNIX ファイル、TSO データ・セット、または CICS データ・キューの割り振り

QMF IMPORT DATA コマンドは、データを QMF 一時記憶域に保管してから、報告書を画面に表示するように見えます。実際には、データの一部だけが保管され、表示されます。UNIX ファイル、TSO データ・セット、または CICS データ・キューは開いた状態で、QMF に割り振られたままになります。ユーザーがデータをスクロールすると、QMF がレコードを読み取ります。

この接続は、データ・オブジェクトが置換またはリセットされるまで、あるいは QMF がすべてのレコードを読み取るまで保持されます。その後、UNIX ファイル、TSO データ・セット、または CICS データ・キューは閉じて、QMF への割り振りが解除されたものと見なされます。アプリケーションで IMPORT DATA コマンドを使用して、QMF に割り振られた UNIX ファイル、TSO データ・セット、または CICS データ・キューの削除または変更を試行してはなりません。アプリケーションは、別のデータ・ソースを使用するか、またはデータ・オブジェクト用の QMF 一時記憶域を空にする (RESET DATA コマンドを使用) 必要があります。その後、アプリケーションは、読み取っているファイル、データ・セット、またはデータ・キューの変更や削除を試行できます。

IMPORT コマンドの実行中、QMF は UNIX ファイル、TSO データ・セットまたは CICS データ・キューを読み取る際にロックしません。つまり、ファイル、データ・セット、またはキューを読み取る際に変更されないようにするステップを実行しません。QMF が読み取りを終了する前に、ファイル、データ・セット、またはキューが消去されるか変更されると、結果は予測不可能になり、システム・エラーが発生する可能性があります。

不完全データ・プロンプトは、エクスポートされたオブジェクト全体を取得するのに十分な一時記憶域がないときに表示される可能性があります。追加の記憶域が必要な場合は、DSQSPILL パラメーターを使用して予備記憶域の使用を指定することができます。TSO で QMF を使用している場合、予備データに拡張記憶域を使用するために、DSQSPILL パラメーターに加えて、DSQSPTYP パラメーターを使用します。

エクスポート・エラー

UNIX ファイル、TSO データ・セット、または CICS データ・キューから QMF ヘデータがインポートされた後、QMF によって REPORT パネルおよび確認メッセージが表示されます。

ファイル、データ・セット、またはデータ・キューに形式エラーが含まれていた場合は、QMF によって REPORT パネルが表示されません。形式エラーが含まれていた場合は、QMF で IMPORT コマンドが処理される前に現行パネルだったオブジェクト・パネルに、QMF から出されたエラー・メッセージが表示されます。ただし、

現行オブジェクト・パネルが REPORT パネルだった場合は、QMF にインポートされたデータでエラーが検出されると、QMF によってホーム・パネルおよびエラー・メッセージが表示されます。

書式、報告書、および指示照会のエクスポート

書式オブジェクトと指示照会オブジェクトは、エクスポートされ、表構造でオブジェクトを表すエンコード形式にインポートされます。報告書もエンコード形式でエクスポートされますが、インポートすることはできません。

このエンコード形式によって、オブジェクトの個々の部分を容易に処理することができます。エンコード形式を使用してオブジェクトをエクスポートするときには、以下の条件が適用されます。

- すべての表番号とフィールド番号は、4 桁の番号として書き込まれます。
- 表の各列は、オブジェクト内に通常現れる並び順に書き込まれます。ただし、最大長の列は表レコードおよび関連する行レコードの右端に移動されます。
- 数値の長さは (必要に応じた先行ゼロを含めて) 3 桁です。
- ブランク文字は、すべてのレコード内で区切り文字として使用されます。
- 各レコードの最後の文字の後には、区切り文字は書き込まれません。
- すべての予約フィールド内にはブランクが書き込まれます。
- E レコードが、出力ファイルに書き込まれる最後のレコードです。

関連資料:

143 ページの『外部化オブジェクトのサイズの仕様』

外部の表およびオブジェクトには、レコード・サイズとレコード形式の両方の仕様があり、これらはオブジェクトのタイプに応じて異なります。

エクスポート・ファイルの汎用形式

書式、報告書、または指示照会のエンコード形式は、固定形式のヘッダー・レコードおよび可変長形式のレコード (オブジェクトの構成要素) で構成されます。

ヘッダー・レコード

エクスポートされた書式、報告書、または指示照会のほとんどのレコードは、可変長形式です。しかし、ヘッダー・レコードは、レコードが入っているデータ・セットまたはデータ・キューが可変長形式の場合でも固定形式です。

ヘッダー・レコードは、エクスポート後の書式、報告書、または指示照会の内容を識別するために使用されます。ヘッダー・レコードは、エクスポート後のデータ・セットまたはデータ・キューにある最初のレコードです。このレコードは、オブジェクトの特性について記述します。

ヘッダー・レコードは、この表に記載されている情報を含みます。アスタリスクは、インポートにフィールドが必要であることを示しています。

表 21. ヘッダー・レコード情報

バイト位置	情報とタイプ
01*	ヘッダー・レコード標識 (H)

表 21. ヘッダー・レコード情報 (続き)

バイト位置	情報とタイプ
02	ブランク
03-05*	プロダクト ID (QMF)
06	ブランク
07-08	書式、報告書、および指示照会がエクスポートされた QMF のリリース・レベル。QMF バージョン 12 リリース 1 の場合、この番号は 19 になります。
09	ブランク
10*	オブジェクトのタイプ: <ul style="list-style-type: none"> • 書式の場合は F • 報告の場合は R • 指示照会の場合は T
11	ブランク
12-13*	QMF オブジェクト・レベル: <ul style="list-style-type: none"> • 報告の場合は 01 • 書式の場合は 04 • 期間指定が含まれていない指示照会の場合は 01 • 1 つ以上の表に期間指定が含まれている指示照会の場合は 02 <p>オブジェクト・レベルは、オブジェクトの形式の変更を示します。QMF リリースで形式が変更されるたびに、そのオブジェクト・レベルも変更されます。オブジェクト・レベルは、形式の変更によって、アプリケーションにエラーが生じる可能性がある場合のみ増加します。</p>
14	ブランク
15*	オブジェクトのデータ形式 (書式、報告書、および指示照会オブジェクトをエクスポートするために、エンコード形式が使用された場合は「E」)
16	ブランク
17	オブジェクトの状況: E - エラーを含んでいる (書式の場合のみ)、W - 警告を含んでいる、V - 有効
18	ブランク
19	全オブジェクトまたは部分オブジェクトの標識 (全オブジェクトの場合は「W」)
20	ブランク
21	オブジェクトのエクスポート時に使用された各国語 (英語の場合は E)
22	ブランク
23*	書式、報告書、または指示照会を QMF の外で適切な形式で作成し、QMF にインポートできます。QMF を使用して、一時記憶域内にあるオブジェクトをインポートするオブジェクトと置換する場合は、このバイト位置に R をコーディングします。
24	ブランク

表 21. ヘッダー・レコード情報 (続き)

バイト位置	情報とタイプ
25-26	各レコードの冒頭にある制御域の長さ: <ul style="list-style-type: none"> • 書式の場合は 01 • 報告の場合は 02 • 指示照会の場合は 01
27	ブランク
28-29	V レコードおよび T レコードに指定されている整数の長さフィールドの長さ (03)
30	ブランク
31-38	形式 <i>yy/mm/dd</i> の日付スタンプ
39	ブランク
40-44	<i>hh:mm:ss</i> という形式のタイム・スタンプ
45	ブランク
51	ブランク

関連資料:

113 ページの『書式のエクスポート』

書式オブジェクトには、QMF のすべての書式パネルに指定されている情報が入っています。書式をエクスポートすると、QMF はデフォルト値から派生した値を持つ書式パネルをエンコード形式に変換します。

132 ページの『指示照会のエクスポート』

エクスポート後の指示照会オブジェクトには、指示照会の基本パネルの確認域に表示される情報が入っています。

124 ページの『標準的な報告書のエクスポート』

QMF によって報告書を表示すると、書式と一時記憶域内のデータ・オブジェクトの対話の結果を見ることができます。報告書オブジェクトは、一時記憶域には存在しません。報告書をエクスポートすると、QMF によって、実際には書式とデータ・オブジェクトの対話がエクスポートされます。

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

エクスポートされたオブジェクトのレコード

ヘッダー・レコード (固定形式レコード) を除き、エクスポートされた書式、報告書、および指示照会のすべてのレコードは可変長形式レコードです。可変長フォーマット・レコードが入力時に受け入れられます。

可変長形式レコードには、次の図のような汎用書式があります。

制御域	レコード・データ域
-----	-----------

図 19. 可変長形式レコードの汎用書式

制御域の内容を以下に示します。

表 22. 可変長形式レコードの汎用書式

バイト位置	説明
01	レコード ID (H、V、T、R、E、*、L、C)
02	ブランク (省略される場合がある。特定のタイプの可変長形式レコードを参照)

レコード・データ域は、個々のレコードに関する情報が入っている可変長域です。この領域の各フィールドは、区切り文字 (ここに記載する例ではブランク文字) で区切られます。

データ値レコード (V):

エクスポートされた書式、報告書、または指示照会の値レコードは、オブジェクトの単一フィールドの値 (例えば、書式内でヘッダーの前に挿入されるブランク行) を提供するために使用されます。

V レコードには以下のデータが含まれています。

- オブジェクトに固有のフィールド番号
- フィールドの長さ
- フィールドの値

V レコードの制御域は、この表に示すとおりです。

表 23. V レコードの制御域

バイト位置	説明
01	値レコード ID (V)
02	ブランク (報告書の場合にのみ使用され、書式および指示照会の場合には省略される)

V レコードのレコード・データ域は、この表に示すとおりです。

表 24. V レコードのレコード・データ域

バイト位置	説明
01	ブランク
02-05	フィールド番号 (1001-9999)
06	ブランク
07-09	データ値の長さ (000-999) アスタリスク (*) に続く 2 つのブランクでもよい。アスタリスクは、データ値がレコードの終わりによって区切られていることを示す。
10	ブランク
11-終わり	データ

V レコードのレコード・データ域:

- レコード・データ域のバイト位置は、制御域の終わりからのオフセットであり、その長さはヘッダー・レコードに示されています。
- 省略されたデータ値 (長さフィールドの後がレコード終わりまたは空白のみ) は、フィールドに NULL 値が入っていることを示します。
- 長さフィールドがゼロの場合、フィールドのデフォルト値が適用され、警告メッセージが出されます。
- 指定された長さが、後に続く実際のデータと異なる場合、QMF によって警告が出されます。

データ表記レコード (T):

エンコード形式では、表にオブジェクトの大部分のデータが表示されます。これらの表はデータベースのリレーショナル表ではなく、エンコードされた形式内の情報をグループ化する方法です。各 T レコードは 1 つの表を定義し、各表はオブジェクトの特定の部分 (例えば、書式内の合計計算) に対応しています。したがって、エクスポート後の 1 つのファイルには、このような多くのエンコード表が入っている可能性があります。

T レコードの後には常に R レコードが続きます。T レコードは、その後続く R レコードについて記述します。T レコードの後に R レコードがない場合、その表は省略されます。

エクスポートされた書式、報告書、または指示照会の表の内容をアプリケーション・プログラムから参照していることを確認するため、T レコード内のエンコードを使用して、R レコード内での値の場所を正確に特定します。アプリケーション・プログラムで R レコード内の情報を見つけるために、固定オフセットを使用してはなりません。

T レコードの制御域は、この表に示すとおりです。

表 25. T レコードの制御域

バイト位置	説明
01	表レコード ID (T)
02	空白 (報告書の場合にのみ使用され、書式および指示照会の場合には省略される)

T レコードのレコード・データ域は、この表に示すとおりです。表のバイト位置は、制御域の終わりに続くオフセットです。制御域の長さはヘッダー・レコードに示されています。

表 26. T レコードのレコード・データ域

バイト位置	説明
01	空白
02-05	表番号 (1001-9999)
06	空白

表 26. T レコードのレコード・データ域 (続き)

バイト位置	説明
07-09	この表内の行数 (R レコード) 数値の代わりに使用されたアスタリスク (*) は、表が、後続のすべての R レコードから構成されることを意味する。
10	ブランク
11-13	レコード (000-999) 内の列数
14	ブランク
15-18, 24-27, ...	この列のフィールド番号 (反復フィールド)
19, 28, ...	ブランク (反復フィールド)
20-22, 29-31, ...	この列のデータ値の長さ (反復フィールド)

バイト 11 から 13 (列数) は、後続のフィールド番号とデータ値の長さのペアの個数を示します。そのため、バイト 15 以降の情報は列ごとに繰り返されます。

オブジェクトのエクスポートおよびインポート時に、T レコードに関する次の情報に留意してください。

- 書式または指示照会をインポートする場合、R レコード数が、T レコードのレコード・データ域のバイト 07-09 に指定されている行カウントと一致していなければなりません。さもないと、QMF は警告を出します。
- 書式または指示照会をインポートする場合、バイト 11-13 に示されている列数が、後続のバイトにあるフィールド番号 / 長さのペア数と一致していなければなりません。さもないと、QMF は警告を出します。
- フィールド番号 / 長さのペア数は、表内の列数に限定されます。また、その並び順は任意です。
- 長さがゼロの列は、一時記憶域内のオブジェクトの更新時にデフォルト値に設定され、警告が出されます。この表に組み込まれない列も、この方法で処理されます。ただし、指示照会がある場合は、可能な限りデフォルトが提供されます。それ以外の場合は、エラーが発生します。
- 列フィールドをブランクに設定するには、列の T レコードに正の長さがあり、R レコードにブランク値がなければなりません。

表行レコード (R):

エクスポートされた書式、報告書、または指示照会の R レコードには、エンコード表内の単一行の一連の値が含まれています。R レコードには、関連する T レコードによって記述されている順に並べられた値のリストが入っています。R レコードは、T レコードに指定されているデータ値の位置と長さの記述に一致しています。

R レコードの制御域は、この表に示すとおりです。

表 27. R レコードの制御域

バイト位置	説明
01	行レコード ID (R)

表 27. R レコードの制御域 (続き)

バイト位置	説明
02	ブランク (報告書の場合にのみ使用され、書式および指示照会の場合には省略される)

制御域に続いて、R レコードのデータ域は、区切り文字 (ブランク文字) によって分離されている一連の値から構成されます。形式は次のとおりです。

`_value..value..value..`

この形式の `value...` はこの行および列のデータ値を表し、`_` は区切り文字です。

R レコードを処理する場合は、次の情報に留意してください。

- R レコードは、別の R レコードまたは T レコードの直後に続いていなければなりません。
- データ値の数は、関連する T レコード内の記述に一致していなければなりません。
- 関連付けられている T レコード内の長さがゼロのデータ値は、オブジェクトのこの行と列に値が適用されないことを示します。つまり、行と列はそのデフォルト値に設定されます。ただし、T レコード内にフィールドが存在する場合は、R レコードにこのフィールド用の余分なブランクが含まれている必要があります。長さがゼロの値の場合は、R レコード内で 1 つのブランクの後に別のブランクが続きます。

オブジェクトの終わりレコード (E):

エクスポートされた書式、報告書、または指示照会の E レコードは、エクスポートされたオブジェクトの終わりを指定します。これは、エクスポートしたファイルの最後のレコードで、文字 E として現れます。エクスポート後の報告書の場合、E レコードの後にブランク文字が続いて、制御域を終了します。書式の場合、ブランクは省略されます。

E レコードに続くレコードはすべて無視されます。E レコードが、インポートするファイルに組み込まれていない場合、QMF では、ファイルの終わりがオブジェクトの終わりを暗黙指定していると想定されます。

アプリケーション・データ・レコード (*):

エクスポートされた書式、報告書、または指示照会のアプリケーション・データ・レコードによって、アプリケーション・プログラムは、独自のデータを外部ファイル内のオブジェクトに関連付けることができます。アプリケーション・プログラムは、これらレコードをコメント・レコードとして頻繁に使用して、ファイル内のオブジェクトをさらに詳細に記述します。

アスタリスクに続く情報は無視され、入力処理に影響を与えません。

アプリケーション・データ・レコードは、ヘッダー (H) レコードの前を除く外部ファイル内のどこにでも置くことができます。QMF はエクスポート時にアプリケーション・データ・レコードを書き込みません。しかし、これらのレコードを、作成

後のデータ・セットまたは CICS データ・キュー内で使用することができます。アプリケーション・データ・レコードの内容は、この表のとおりです。

表 28. アプリケーション・データ・レコードの内容

バイト位置	説明
01	アプリケーション・データ・レコード ID (*)
02- レコードの終わり	データ

以下は、エクスポートされた書式に表示されるアプリケーション・データ・レコードの例です。

*This is the form that groups by DEPT.

報告書行レコード (L):

報告書内の形式設定済みの各行は、L レコードによって記述されます。報告書内の行ごとに 1 つの L レコードがあります。

他の可変長形式レコード (V、T、および R) と同様に、L レコードは、制御域およびそれに続くレコード・データ域で構成されます。制御域の形式は、他のレコードに似ています。レコード・データ域は、固定域およびそれに続く形式設定済みの報告書行で構成されます。固定域は、その後続く報告書行に関する情報を提供します。L レコードの形式を次の図に示します。

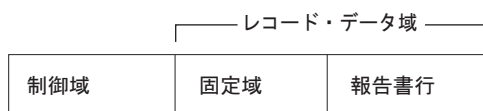


図 20. L レコードの形式

L レコードの制御域は、この表に示すとおりです。

表 29. L レコードの制御域

バイト位置	説明
01	値レコード ID (L)

表 29. L レコードの制御域 (続き)

バイト位置	説明
02	<p>継続標識</p> <p>現行レコードをデータ継続レコードに継続するかどうかを示す。</p> <ul style="list-style-type: none"> • C は継続を示す <p>C レコードは、制御域のバイト 2 にある継続文字によってマークされている L レコードの直後に続きます。</p> <ul style="list-style-type: none"> • D は、現行レコード終わりと次のレコードのデータ部分の先頭に、DBCS 区切り文字 S0 と SI を挿入して、継続することを示す <p>制御域内の継続標識として D が指定されている場合は、現行レコードが長すぎて単一の物理レコードに収まりません。レコードを分割するプロセスで、継続する DBCS データの整合性を保つために、現行レコードおよび次のレコードに、S0 (シフトアウト) 文字および SI (シフトイン) 文字が追加されました。</p> <ul style="list-style-type: none"> • ブランクは継続しないことを示す

L レコードのレコード・データ域は、この表に示すとおりです。バイト 6-13 は、行タイプ属性です。バイト 06 は常に 1 です。バイト 7 から 13 の各バイトは、形式設定済みの報告書の行に、対応する行タイプ属性があるかどうかを示します (1 = 属性がある、0 = 属性がない)。

表 30. L レコードのレコード・データ域

バイト位置	説明
01	ブランク
02-04	<p>報告書構成部分標識:</p> <p>110 = ページ・ヘッダー 120 = ページ後書き 13n = 切れ目ヘッダー (n は切れ目番号、1-6) 15n = 切れ目後書き (n は切れ目番号、1-6) 170 = 列ヘッダー 171 = 明細ヘッダー 180 = 明細行 181 = グループ合計行 190 = 最終後書き</p>
05	ブランク
06	1

表 30. L レコードのレコード・データ域 (続き)

バイト位置	説明
07	データ
08	Text
09	区切り文字
10	列の折り返し 列の折り返し (バイト 10) と行の折り返し (バイト 11) に関する属性は、複数の物理的な報告書行への、単一の論理報告書行の継続を示すために使用されます。L 形式レコード内にいずれかの属性がある場合、列データまたは折り返し行が、次の L 形式レコードに継続されることを意味します。
11	行の折り返し 列の折り返し (バイト 10) と行の折り返し (バイト 11) に関する属性は、複数の物理的な報告書行への、単一の論理報告書行の継続を示すために使用されます。L 形式レコード内にいずれかの属性がある場合、列データまたは折り返し行が、次の L 形式レコードに継続されることを意味します。
12	2 番目のデータ行 (横方向報告書のみ) パーセント列または累積合計列を含む横方向報告書には、グループ (および切れ目、最終) の合計ごとに 2 つのデータ行を含めることができます。最初の合計データ行には、固有の『横方向』値ごとに横方向に計算された、列の累積パーセント値または累積合計値が入ります。2 番目の合計データ行には、(報告書内または制御の切れ目内の) グループごとに下方向に計算された列の累積パーセント値または累積合計値が入ります。2 番目のデータ行 (バイト 12) の行タイプは、この特性を持つエクスポート後の報告書内の 2 番目のデータ行を示します。
13	予約済み
14	ブランク

次に、テキストおよびデータを含む報告書内の切れ目後書き行の L レコードの例を示します。

L 151 11100000 DEPARTMENT TOTALS 93,659.45

データ継続レコード (C):

エクスポートされた書式、報告書、または指示照会に含まれる C レコードは、値または値のセットを複数のレコードにわたって継続するために使用されます。C レコ

ードは、継続するレコードの直後に配置されます。C レコードの形式は、継続する元のレコードの形式に対応します。C レコードは、L レコードを継続する目的でのみ QMF によって使用されます。

C レコードの制御域は、この表に示すとおりです。

表 31. C レコードの制御域

バイト位置	説明
01	値レコード ID (C)
02	<p>継続標識。現行レコードを別の C レコードに継続するかどうかを示す。</p> <ul style="list-style-type: none"> • C は継続を示す <p>C レコードは、制御域のバイト 2 にある継続文字によってマークされている L レコードの直後に続きます。</p> <ul style="list-style-type: none"> • D は、現行レコード終わりと次のレコードのデータ部分の先頭に、DBCS 区切り文字 S0 と SI を挿入して、継続することを示す <p>制御域内の継続標識として D が指定されている場合は、現行レコードが長すぎて単一の物理レコードに収まりません。レコードを分割するプロセスで、継続する DBCS の整合性を保つために、現行レコードおよび次のレコードに、S0 (シフトアウト) 文字および SI (シフトイン) 文字が追加されました。</p> <ul style="list-style-type: none"> • ブランクは継続しないことを示す

C レコードのレコード・データ域は、この表に示すとおりです。示されているバイト位置は、制御域の終わりからのオフセットであり、その長さはヘッダー・レコードに示されています。

表 32. C レコードのレコード・データ域

バイト位置	説明
01	ブランク
02-終わり	継続する値または値のセット

書式のエクスポート

書式オブジェクトには、QMF のすべての書式パネルに指定されている情報が入っています。書式をエクスポートすると、QMF はデフォルト値から派生した値を持つ書式パネルをエンコード形式に変換します。

以下のパネルは、パネルを変更した場合にのみ、エンコード形式になります。

- FORM.BREAK n (n は 1 から 6)
- FORM.CALC

- FORM.CONDITIONS
- FORM.DETAIL が 1 より大きい、すべてのバリエーション・パネル

外部形式から未使用のパネルを除去することによって、システム上のスペースを節約することができます。

サンプルのエクスポート結果を表示するためのデフォルト書式の作成

デフォルト書式を作成するには、空の報告書を作成する照会を実行します。このような照会の例を以下に示します。

```
SELECT JOB
FROM Q.STAFF
WHERE NAME='NO_NAME'
```

QMF によって報告書が表示されたら、EXPORT FORM TO DEFAULT (CICS では QUEUETYPE=xx パラメーターを含めて) を入力してください。

エクスポート後の書式の外観

DEFAULT という名前のデータ・セットまたは CICS データ・キューには、以下の、エクスポートされた書式のサンプル形式の情報が含まれています。

```
H QMF 19 F 04 E V W E R 01 03 13/01/15 16:20

T 1110 001 011 1112 007 1113 040 1114 007 1115 006 1116 005 1117 005 1118 003 1119 008 1120 008
  1122 006 1121 050
R CHAR      JOB                2      5      C      1  DEFAULT
  DEFAULT NO
V 1201 001 0
V 1202 001 2
T 1210 001 003 1212 004 1213 006 1214 055
R 1      CENTER
V 1301 001 2
V 1302 001 0
T 1310 001 003 1312 004 1313 006 1314 055
R 1      CENTER
V 1401 002 NO
V 1402 001 1
V 1403 001 0
T 1410 001 003 1412 004 1413 006 1414 055
R 1      RIGHT
V 1501 001 1
V 1502 003 YES
V 1503 003 YES
V 1504 003 YES
V 1505 003 YES
V 1506 003 YES
V 1507 003 YES
V 1508 003 YES
V 1509 003 YES
V 1510 003 YES
V 1511 004 NONE
V 1512 002 NO
V 1513 007 DEFAULT
V 1514 002 NO
V 1515 004 NONE
V 2790 001 1
V 2791 003 YES
V 2805 003 YES
T 2810 001 003 2812 004 2813 006 2814 055
R 1      LEFT
V 2901 002 NO
V 2902 001 1
V 2904 001 0
V 2906 002 NO
V 2907 002 NO
T 2910 001 003 2912 004 2913 006 2914 055
R 1      LEFT
V 3080 001 1
V 3101 002 NO
```

```

V 3102 002 NO
V 3103 001 0
V 3104 001 0
T 3110 001 003 3112 004 3113 006 3114 055
R 1 LEFT
V 3201 002 NO
V 3202 001 1
V 3203 001 0
V 3204 001 1T 3210 001 003 3212 004 3213 006 3214 055
R 1 RIGHT
V 3080 001 2
V 3101 002 NO
V 3102 002 NO
V 3103 001 0
V 3104 001 0
T 3110 001 003 3112 004 3113 006 3114 055
R 1 LEFT
V 3201 002 NO
V 3202 001 1
V 3203 001 0
V 3204 001 1
T 3210 001 003 3212 004 3213 006 3214 055
R 1 RIGHT
E

```

初期プロシージャー内でコマンド `IMPORT FORM FROM DEFAULT` (CICS では `QUEUETYPE=xx` パラメーターを含めて) 出すことによって、ログオンするたびにデフォルトの データ・セットまたは CICS データ・キュー をインポートすることができます。

エクスポートされたデータ・セットまたはデータ・キュー内のヘッダー・レコードの解釈

次に、QMF 書式のヘッダー・レコードの例を示します。

```
H QMF 19 F 04 E V W E R 01 03 13/01/15 16:20
```

この例について、次の表で説明します。

表 33. 書式ヘッダー・レコードのサンプル

例からの値	説明
H QMF 19 F	このレコードは、バージョン 12.1 の QMF 書式ヘッダー・レコードです。
04	書式の構造のオブジェクト・レベルは 4。
E	形式はエンコードされます (エクスポート後の書式、報告書、および指示照会で使用される形式)。
V	エクスポート後の書式にエラーまたは警告が含まれていない。
W	ファイルに書式全体が含まれている。
E	オブジェクトのエクスポート時に使用された国語は英語。
R	インポート時に、一時記憶域のオブジェクトが置換される。
01	制御域の長さは 1 バイト。
03	整数長さフィールドの長さは 3 バイト。
13/01/15	この日付スタンプは January 15, 2013を示します。
16:20	タイム・スタンプは、4:20 PM を指定します。

英語以外のセッションから書式をエクスポートする場合、現行セッションの言語または英語のいずれかを指定できます。そのため、H レコード内の各国語 ID は、書

式のエクスポート元のセッションの言語を示していない可能性があります。

エクスポートされた書式のレコードの解釈

114 ページの『エクスポート後の書式の外観』では、エクスポート後のサンプルを示します。エクスポート後の書式には、エクスポートされた結果を解釈するのに役立つ、特別な意味がある関連コードを持つ V、T、および R レコードが含まれています。この表では、エクスポート後の書式内にある各フィールドおよびコードについて説明します。

V レコードのフィールド 3080 は、その後続く切れ目パネルの「トリガー」として機能します。このレコードは、書式内で切れ目パネルごとに 1 回現れます。このフィールドの値は、フィールド 3080 に続く各フィールドで記述されている切れ目パネルの番号を示します。

表 34. エクスポート後の FORM オブジェクトの表番号とフィールド番号

表番号または フィールド番号	レコード・ タイプ	説明	書式パネル
1110	T	列ヘッダー表	FORM.COLUMNS
1112	R	列データ・タイプ 列データ・タイプは、書式パネルに表示されませんが、外部形式の書式に関連しています。 書式のインポート時に列データ・タイプは不要です。インポート時に列データ・タイプがない場合、CICS は編集コードからデフォルトのデータ・タイプ情報を提供します。(詳細については、120 ページの『書式オブジェクトのインポート』を参照してください。) エクスポート時に QMF が提供する列データ・タイプは、指定された編集コードに基づいています。編集コード U、V、M または無効な編集コードについては、QMF はデータ・タイプ・キーワード UNKNOWN を指定します。 119 ページの表 35 では、QMF が使用するデータ・タイプ・キーワードを示します。	FORM.COLUMNS
1113	R	列見出し	FORM.COLUMNS
1114	R	列取扱コード	FORM.COLUMNS
1115	R	列字下げ	FORM.COLUMNS
1116	R	列幅	FORM.COLUMNS
1117	R	列編集コード	FORM.COLUMNS

表 34. エクスポート後の FORM オブジェクトの表番号とフィールド番号 (続き)

表番号または フィールド番号	レコード・ タイプ	説明	書式パネル
1118	R	列順序	FORM.COLUMNS
1119	R	列ヘッダー位置合わせ	FORM.COLUMNS
1120	R	列データ位置合わせ	FORM.COLUMNS
1121	R	列定義	FORM.COLUMNS
1122	R	列定義での NULL の受け渡し	FORM.COLUMNS
1180	T	合計計算表	FORM.CALC
1182	R	計算識別番号	FORM.CALC
1183	R	合計計算式	FORM.CALC
1184	R	合計計算幅	FORM.CALC
1185	R	合計計算編集コード	FORM.CALC
1186	R	計算での NULL の受け渡し	FORM.CALC
1201	V	ヘッダーの前のブランク行数	FORM.PAGE
1202	V	ヘッダーの後のブランク行数	FORM.PAGE
1210	T	ページ・ヘッダー表	FORM.PAGE
1212	R	ページ・ヘッダー行番号	FORM.PAGE
1213	R	ページ・ヘッダー位置合わせ	FORM.PAGE
1214	R	ページ・ヘッダー・テキスト	FORM.PAGE
1301	V	後書きの前のブランク行数	FORM.PAGE
1302	V	後書きの後のブランク行数	FORM.PAGE
1310	T	ページ後書き表	FORM.PAGE
1312	R	ページ後書き行番号	FORM.PAGE
1313	R	ページ後書き位置合わせ	FORM.PAGE
1314	R	ページ後書きテキスト	FORM.PAGE
1401	V	最終テキスト用の改ページ	FORM.FINAL
1402	V	最終合計行番号	FORM.FINAL
1403	V	最終テキストの前のブランク行数	FORM.FINAL
1410	T	最終テキスト表	FORM.FINAL
1412	R	最終テキスト行番号	FORM.FINAL
1413	R	最終テキスト位置合わせ	FORM.FINAL
1414	R	最終テキスト	FORM.FINAL
1501	V	明細行スペース	FORM.OPTIONS
1502	V	切れ目列の一括表示	FORM.OPTIONS
1503	V	省略時切れ目テキスト	FORM.OPTIONS
1504	V	グループ用の列ヘッダー内の機能名	FORM.OPTIONS
1505	V	ページの保持された列折り返し行数	FORM.OPTIONS
1506	V	横方向合計列	FORM.OPTIONS

表 34. エクスポート後の FORM オブジェクトの表番号とフィールド番号 (続き)

表番号または フィールド番号	レコード・ タイプ	説明	書式パネル
1507	V	列ヘッダーの区切り線	FORM.OPTIONS
1508	V	切れ目合計の区切り線	FORM.OPTIONS
1509	V	横方向ヘッダーの区切り線	FORM.OPTIONS
1510	V	最終合計の区切り線	FORM.OPTIONS
1511	V	折り返し報告書行の幅	FORM.OPTIONS
1512	V	切れ目での新しいページ番号付け	FORM.OPTIONS
1513	V	切れ目または最終テキストの幅	FORM.OPTIONS
1514	V	列の再配列	FORM.OPTION
1515	V	固定列	FORM.OPTIONS
2790	V	詳細バリエーション番号	FORM.DETAIL
2791	V	詳細バリエーション選択	FORM.DETAIL
2805	V	列ヘッダーの組み込み	FORM.DETAIL
2810	T	明細ヘッダー表	FORM.DETAIL
2812	R	明細ヘッダー・テキスト行	FORM.DETAIL
2813	R	明細ヘッダー位置合わせ	FORM.DETAIL
2814	R	明細見出しテキスト	FORM.DETAIL
2901	V	詳細テキスト用の改ページ	FORM.DETAIL
2902	V	列データの行番号	FORM.DETAIL
2904	V	詳細テキスト後にスキップする行数	FORM.DETAIL
2906	V	明細ヘッダーの反復	FORM.DETAIL
2907	V	一緒に保持する詳細テキスト行数	FORM.DETAIL
2910	T	詳細テキスト表	FORM.DETAIL
2912	R	詳細テキスト行番号	FORM.DETAIL
2913	R	詳細テキスト位置合わせ	FORM.DETAIL
2914	R	詳細テキスト	FORM.DETAIL
3080	V	切れ目パネル番号	FORM.BREAKn
3101	V	切れ目ヘッダー用の改ページ	FORM.BREAKn
3102	V	切れ目ヘッダーの反復	FORM.BREAKn
3103	V	切れ目ヘッダーの前にスキップする行数	FORM.BREAKn
3104	V	切れ目ヘッダーの後にスキップする行数	FORM.BREAKn
3110	T	切れ目ヘッダー・テキスト表	FORM.BREAKn
3112	R	切れ目ヘッダー行番号	FORM.BREAKn
3113	R	切れ目ヘッダー位置合わせ	FORM.BREAKn
3114	R	切れ目見出しテキスト	FORM.BREAKn
3201	V	切れ目テキスト用の改ページ	FORM.BREAKn
3202	V	切れ目テキスト合計行	FORM.BREAKn

表 34. エクスポート後の FORM オブジェクトの表番号とフィールド番号 (続き)

表番号または フィールド番号	レコード・ タイプ	説明	書式パネル
3203	V	切れ目テキストの前にスキップする行数	FORM.BREAKn
3204	V	切れ目テキストの後にスキップする行数	FORM.BREAKn
3210	T	切れ目テキスト表	FORM.BREAKn
3212	R	切れ目テキスト行	FORM.BREAKn
3213	R	切れ目テキスト位置合わせ	FORM.BREAKn
3214	R	切れ目テキスト	FORM.BREAKn
3310	T	条件表	FORM.CONDITIONS
3312	R	条件識別番号	FORM.CONDITIONS
3313	R	条件式	FORM.CONDITIONS
3314	R	条件パネルでの NULL の受け渡し	FORM.CONDITIONS

次の表に、書式で指定された編集コード用に QMF で生成されるデータ・タイプ・キーワードを示します。この表では、 x は、表示される小数点以下の桁数を示します。この x は、0 から 99 までの整数です。

表 35. QMF 書式パネルで指定された編集コードに対して生成されるデータ・タイプ・キーワード

指定された編集コード	データ・タイプ・キーワード
C、CW、CT、CD x	CHAR
B、BW、X、XW	BINARY
G、GW	GRAPHIC
E、D、I、J、K、L、P、EZ、DZ、IZ、JZ、KZ、LZ、PZ、DZC、D x 、I x 、J x 、K x 、L x 、P x	NUMERIC
文字 TD で開始する編集コード	DATE
文字 TT で開始する編集コード	TIME
TSI	TIMEST
TSZ	TSTMPTZ
M	UNKNOWN
U、V	UNKNOWN
入力された無効な編集コード	UNKNOWN

書式をエクスポートすると、QMF によって、デフォルトから変更された値を含むバリエーション・パネルのみがエクスポートされます。したがって、外部書式のバリエーションの合計数が、パネル上のバリエーション・カウント標識に示されている数より少ない場合があります。QMF は各バリエーション番号を変更して、そのバリエーションを連続する順序の中に戻すことができます。

書式オブジェクトのインポート

書式をインポートする場合、以下のフィールドは大文字でなければなりません。

- すべてのレコードのレコード ID
- ヘッダー・レコード内の以下のフィールド
 - プロダクト ID (QMF)
 - オブジェクトのタイプ (F)
 - オブジェクトの形式 (E)
 - オブジェクトに対するアクション (R)
- COLUMNS 表に関する R レコード内のデータ・タイプ値 (数値、文字、図形、または不明なデータ・タイプ)

サイトが日付 / 時刻データ・タイプをサポートしている場合、データ・タイプ値 DATE、TIME、TIMEST、および TSTMPTZ も大文字でなければなりません。

- 書式パネル内で使用されるすべての書式キーワードおよび置換変数

書式をインポートする場合、書式内のすべての入力そのまま残されます。書式キーワードが小文字の場合、書式パネル内のエラー標識がオンになります。エラーを訂正するには、フィールドに重ねて入力してください。データ・タイプ値が大文字でない場合、エラーが生じて IMPORT が終了します。

COLUMNS 表の T レコード (フィールド番号 1110) は、ヘッダー・レコードの直後にある必要があります。T レコードには、エンコード形式の行数の数値カウントも含まれている必要があります (行カウントとして * は使用できません)。

COLUMNS 表全体を読み取った場合、指定されていないフィールドはデフォルト値に設定されて、書式が表示されます。

- バリエーション・パネル

バリエーション番号フィールド (フィールド番号 2790) は、このフィールドに続くすべてのバリエーション・パネル情報によって、どのバリエーション・パネルが更新されるかを示します。この V レコードは、バリエーション・パネルの他のすべての V レコード、T レコード、および R レコードより前にある必要があります。

特定のバリエーション用の値が、エンコード形式で複数回現れた場合、後の値が前の値を置換します。書式内のバリエーションの数は、書式内の最高のバリエーション番号に等しい数です。インポート時には、バリエーション番号の並び順は無関係です。

- 変換パネル

英語の書式を英語以外のセッションにインポートするときに、H レコードの各国語 ID が E である場合は、QMF によって予約語が変換されます。予約語は、QMF によって現行セッションの言語に変換されます。予約語の例としては、FORM.COLUMNS 内の USAGE 列の値などが挙げられます。

- インポート書式でのデータ・タイプ、編集コード、および幅の省略

COLUMNS 表では、データ・タイプ (フィールド番号 1112)、編集コード (フィールド番号 1117)、および幅 (フィールド番号 1116) を以下の規則に従ってオプションで省略することができます。

- データ・タイプと幅を省略する場合には、編集コードを指定する必要があります。指定した編集コードに基づいて、データ・タイプと幅の適切なデフォルトが QMF によって挿入されます。
- 編集コードと幅を省略する場合、データ・タイプを指定する必要があります。編集コードと幅のデフォルト値が QMF によって提供されます。
- 幅は、データ・タイプまたは編集コードと共に指定する必要があります。

次の表に、列のデータ・タイプが格納されるフィールドの値に関する情報を示します。QMF では、U コード、V コード、または無効な編集コードへの応答に、この表に示したデータ・タイプ値に加えて、UNKNOWN データ・タイプ・キーワードが使用されます。

表 36. 列のデータ・タイプが格納されるフィールドの値

外部化された書式に表示されているデータ・タイプ	10 進数のコード	データ・タイプ	意味
DATE	384	DATE	日付
TIME	388	TIME	時刻
TIMEST	392	TIMESTAMP	タイム・スタンプ
NUMERIC	496 500 492 484 480 996	INTEGER SMALLINT BIGINT DECIMAL FLOAT DECFLOAT	整数 短整数 大整数 10 進 浮動小数点 10 進浮動小数点
CHAR	448 452 456 904	VARCHAR CHAR LONG VARCHAR ROWID	可変長文字 固定長文字 長可変長文字 行 ID
GRAPHIC	464 468 472	VARGRAPHIC GRAPHIC LONG VARGRAPHIC	可変長図形 固定長図形 長可変長図形
BINARY	908 912	VARBINARY BINARY	可変長バイナリー 固定長バイナリー
TSTMP TZ	2448	TIMESTAMP WITH TIME ZONE	タイム・スタンプと時間帯

- インポート中のエラーの検出

インポート中に書式ファイルの形式に関するエラーが QMF で検出されると、インポート機能が終了します。QMF から、エラーおよびファイル内でのその場所を記述するメッセージが出されます。

エラーがヘッダー・レコード内で検出されても、書式が一時記憶域内に存在している場合は、既存の書式が表示されます。書式が正常にインポートされた場合は、QMF によってその書式パネルが表示されます。

ヘッダー・レコードが読み取られた後でエラーが検出された場合、一時記憶域内の既存の書式が廃棄され、ホーム・パネルが表示されます。ただし、データ・オブジェクトが存在している場合、QMF によってデータのデフォルト書式が生成されますが、表示はされません。

特定の小さなエラーが QMF によって検出されても、インポートは終了しません。このような場合、QMF は警告メッセージを出し、可能であれば、デフォルト値を適用します。例を次に示します。

- V レコード
 - ゼロ長フィールド
 - 指定された長さフィールドが、提供されたデータの長さと一致しない場合
- T レコード
 - ゼロの列の長さ
 - 指定された列数が、後続のフィールド番号 / 長さのペアと一致しない場合

以下のようにして、エラーと警告に応答することができます。

- 一度に 1 つずつ問題を修正する
- プロファイルの TRACE オプションを L2 に設定し (SET PROFILE (TRACE=L2) コマンドを使用)、IMPORT FORM コマンドを実行する。L2 トレース・オプションを設定すると、メッセージおよびコマンドが最高の詳細レベルでトレースされます。このオプションを使用すると、IMPORT コマンドに関連するすべてのメッセージ・テキストを表示できます。

次のコマンドは、特定のメッセージ・コードに関連付けられたメッセージ・テキストを表示します。

```
HELP DSQnnnnn
```

nnnnn は 5 文字の固有メッセージ・コードです。

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

アプリケーション内の QMF 書式オブジェクトに関する考慮事項

アプリケーション内で QMF 書式を作成して使用する際には、いくつかのヒントが役に立ちます。

アプリケーション内で QMF 書式を使用している場合は、以下の点に留意してください。

QMF の外側での書式 データ・セットまたは CICS データ・キュー の作成

QMF の外で書式を作成する場合 (EXPORT FORM を使用していない場合)、QMF に正常にインポートするためには完全な書式オブジェクトは不要です。必要なのは、ヘッダー (H) レコードと、それに続く COLUMNS 表の T レコードおよび R レコードだけです。書式の残りの部分については、インポート時にデフォルト値が適用されます。

ユーザー固有の書式データ・セットまたは CICS データ・キューを作成すれば、EXPORT FORM を使用した場合に入手するデータ・セットまたはデータ・キューと正確に同じである必要はありません。例えば、QMF で書式をエクスポートするときに、値 (V) レコード内のすべてのデータ値の前に長さが付けられます。ただし、書式のインポート時には、データ値がレコード終わりによって区切られていることを示すアスタリスク (*) を使用できます。

インポート後の書式内の R レコード・カウントが、QMF によってデフォルト書式の関連領域に割り振られているデフォルト行数より少ない場合、余分な行は保持されます。

ヘッダー・レコード内のオブジェクト・レベルの検査

書式を含むデータ・セットまたはデータ・キューのヘッダー・レコード内のオブジェクト・レベルは、書式の生成時の形式構造のレベルを示します。(オブジェクト・レベルは、ヘッダー・レコードのバイト 12 および 13 に示されます。) 書式を含むデータ・セットまたはデータ・キューの内容が、アプリケーションによって適切に解釈されるようにしてください。アプリケーションがベースとしている形式を、オブジェクト・レベルが示しているかどうかを確認してください。

アプリケーション・データ・レコードの使用

アプリケーション・データ・レコードは、アプリケーション・プログラム内で役に立ちます。アプリケーション・データ・レコードを使用して、書式オブジェクトについて、データ・セットまたは CICS データ・キュー内に独自のコメントを組み込むことができます。このレコードは、データ・セットまたは CICS データ・キュー内のヘッダー・レコードの後の任意の場所に置くことができます。QMF は、このようなレコードを読み取ると、レコード内の * 文字の後にあるすべてのデータを無視します。したがって、このレコードは、インポート・プロセスに何の影響も与えません。

CICS で書式を使用する場合の制約

REXX は、QMF for CICS では使用できません。REXX に依存する QMF 書式上の区域は、CICS 環境でその書式を実行しようとしても機能しません。このような区域には、FORM.CALC パネル、FORM.CONDITIONS パネル、および定義指定ウィンドウでのすべての入力が含まれます。QMF for CICS ユーザーは、REXX の計算、条件付き行形式設定、および列定義を使用できません。

関連概念:

141 ページの『書式および指示照会のインポート』

書式または指示照会をインポートするための規則に注意してください。

関連資料:

113 ページの『書式のエクスポート』

書式オブジェクトには、QMF のすべての書式パネルに指定されている情報が入っています。書式をエクスポートすると、QMF はデフォルト値から派生した値を持つ書式パネルをエンコード形式に変換します。

109 ページの『アプリケーション・データ・レコード (*)』

エクスポートされた書式、報告書、または指示照会のアプリケーション・データ・レコードによって、アプリケーション・プログラムは、独自のデータを外部ファイル内のオブジェクトに関連付けることができます。アプリケーション・プログラムは、これらレコードをコメント・レコードとして頻繁に使用して、ファイル内のオブジェクトをさらに詳細に記述します。

103 ページの『ヘッダー・レコード』

エクスポートされた書式、報告書、または指示照会のほとんどのレコードは、可変長形式です。しかし、ヘッダー・レコードは、レコードが入っているデータ・セットまたはデータ・キューが可変長形式の場合でも固定形式です。

標準的な報告書のエクスポート

QMF によって報告書を表示すると、書式と一時記憶域内のデータ・オブジェクトの対話の結果を見ることができます。報告書オブジェクトは、一時記憶域には存在しません。報告書をエクスポートすると、QMF によって、実際には書式とデータ・オブジェクトの対話がエクスポートされます。

報告書は、データベースに保管できません。また、エクスポート後の報告書を QMF にインポートすることもできません。ただし、エクスポートされた報告書を使用して以下を行うことができます。

- 報告書からデータを取り出して、アプリケーションで使用する。
- 報告書の外観を印刷用に変更し、アプリケーションによって再表示する。

サンプル報告書 (エクスポート前)

次に、レベル 1 の切れ目がある表形式の報告書のサンプルを示します。

フィールド番号のリストについては、126 ページの『エクスポートされたデータ・セットまたはデータ・キュー内の報告書ヘッダー・レコードの解釈』を参照してください。

REPORT	LINE 1	POS 1	79
J & H SUPPLY COMPANY AVERAGE SALARIES (DEPTS 10, 15, 20) REPORT 17			
DEPT	JOB	AVERAGE	SALARY
-----	-----	-----	-----
10	MGR	20865.86	
	*	20865.86	
15	CLERK	12383.35	
	MGR	20659.80	
	SALES	16502.83	
	*	15482.33	
20	CLERK	13878.68	
	MGR	18357.50	
	SALES	18171.25	
	*	16071.53	
		=====	
		17473.24	
COMPANY NAME REPORT 17			

図 21. エクスポートする前の QMF の表報告書

エクスポート後の報告書の外観

以下は、サンプルの表形式報告書からエクスポートされた報告書の形式です。

```

H QMF 19 R 01 E V W E R 02 03 13/01/15 16:20
V 1001 006 PERIOD
V 1002 003 016
T 1010 003 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000008 000001 BREAK1
R C 000009 000011 000015 000001 GROUP
R L2 000016 000018 000027 000001 AVERAGE
L 110 10100000
L 110 10100000
L 110 10100000
L 110 10000000
L 110 10000000
L 170 10000000
L 170 10000000
L 170 10010000
L 181 11000000
L 151 10010000
L 151 11100000
L 151 10000000
L 181 11000000
L 181 11000000
L 181 11000000
L 151 10010000
L 151 11100000
L 151 10000000
L 181 11000000
L 181 11000000

```

J & H SUPPLY COMPANY			
AVERAGE SALARIES (DEPTS 10, 15, 20)			
REPORT 17			
DEPT	JOB	AVERAGE	SALARY
-----	-----	-----	-----
10	MGR	20865.86	
	*	20865.86	
15	CLERK	12383.35	
	MGR	20659.80	
	SALES	16502.83	
	*	15482.33	
20	CLERK	13878.67	
	MGR	18357.50	

```

L 181 11000000          SALES    18171.25
L 151 10010000          -----
L 151 11100000          *    16071.52
L 151 10000000
L 190 10010000          =====
L 190 11000000          17473.24
L 120 10000000
L 120 10000000
L 120 10100000          COMPANY NAME
L 120 10100000          REPORT 17
E

```

QMF は、報告書をエクスポートするとき、形式設定済み報告書の全テキストと報告書の内容を解釈するための追加情報を書き込みます。

ヘッダー・レコードは、エクスポート・ファイルの最初のレコードです。この後に適切な V、T、および R の各レコードが続きます。報告書が横方向スタイルの報告書の場合には、最初のグループの後に、V、T、および R の各レコードから構成される別のグループが続きます。

H、V、T、R、および E の各レコードの他に、エクスポート報告書には、以下の 2 つの追加のレコード・タイプも必要です。

- 報告書行、または L、レコード
- データ連結、または C、レコード

これらの 2 つのレコードは、V、T、および R の各レコードの最後のグループの後に続きます。

アプリケーションで報告書の形式設定済みデータのみを使用したい場合、QMF に印刷出力を データ・セットまたは CICS データ・キュー に送らせることができます。このデータ・セットまたは CICS データ・キューにはレイアウト情報がなく、形式設定済みデータのみが入ります。

エクスポートされたデータ・セットまたはデータ・キュー内の報告書ヘッダー・レコードの解釈

次に、QMF 報告書のヘッダー・レコードの例を示します。

```
H QMF 19 R 01 E V W E R 02 03 13/01/15 16:20
```

この例について、次の表で説明します。

表 37. 報告書用ヘッダー・レコードのサンプル

例からの値	説明
H QMF 19 R	このレコードは、バージョン 12.1 の QMF 報告書のヘッダー・レコードです。
01	報告書の構造のオブジェクト・レベルは 1 です。
E	形式はエンコードされます (エクスポート後の書式、報告書、および指示照会で使用される形式)。
V	エクスポート後の報告書にはエラーまたは警告が含まれていない。
W	ファイルに報告書全体が含まれている。
E	英語は、オブジェクトのエクスポート時に使用された各国語。
R	この標識は無視されます。

表 37. 報告書用ヘッダー・レコードのサンプル (続き)

例からの値	説明
02	制御域の長さは 2 バイトです。
03	整数長さフィールドの長さは 3 バイト。
13/01/15	この日付スタンプは January 15, 2013を示します。
16:20	タイム・スタンプは、4:20 PM を指定します。

エクスポートされた報告書のレコードの解釈

次の表では、125 ページの『エクスポート後の報告書の外観』に示したエクスポート後の報告書の、T レコードの表番号と V レコードのフィールド番号について説明します。

表 38. エクスポートされた報告書の表番号とフィールド番号

表番号または フィールド番号	レコード・ タイプ	説明
1001	V	プロファイル DECIMAL オプション
1002	V	L レコード制御域の長さ + 固定域
1010	T	形式設定済み報告書表
		報告書内の形式設定済みデータ列ごと
1012	T	OMIT を除くすべての取扱コードごと
1013	T	データの形式を設定する編集コード
1014	T	形式設定済みデータが入っているフィールド (字下げ域を含む) の開始位置
1015	T	形式設定済みデータが入っているフィールド (字下げ域を含まない) の開始位置
1016	T	形式設定済みデータが入っているフィールドの終了位置
1017	T	形式設定済み列値が現れる、論理報告書行内の相対的な物理報告書行数

エクスポート後の報告書内の表およびフィールド番号を処理する場合は、次の点に注意してください。

- 報告書行の位置 1 は、L レコード固定域の直後になります。
- 報告書の各ヘッダー (PAGE または BREAK) あるいは各フッター (PAGE, BREAK、または FINAL) 内のテキスト行用の R レコードは、書式のデフォルトへの変更を含んでいる最後の行まで (最後の行を含む) しか書き込まれません。

特定のヘッダーまたは後書きに関するすべてのフィールドに元の値がある場合でも、ヘッダーまたは後書きごとに少なくとも 1 つの R レコードが書き込まれます。

- 最大レコード長を超える報告書オブジェクトについては、継続レコードが書き込まれます。

関連資料:

130 ページの『横方向スタイル報告書のエクスポート』

エクスポートされた横方向スタイル報告書には、エクスポートされた標準報告書には含まれないフィールドが含まれます。

112 ページの『データ継続レコード (C)』

エクスポートされた書式、報告書、または指示照会に含まれる C レコードは、値または値のセットを複数のレコードにわたって継続するために使用されます。C レコードは、継続するレコードの直後に配置されます。C レコードの形式は、継続する元のレコードの形式に対応します。C レコードは、L レコードを継続する目的のみ QMF によって使用されます。

110 ページの『報告書行レコード (L)』

報告書内の形式設定済みの各行は、L レコードによって記述されます。報告書内の行ごとに 1 つの L レコードがあります。

HTML 形式内の報告書のエクスポート

報告書を HTML 形式でエクスポートすると、QMF によってその本文の前後に、必要な HTML タグが付加されます。その後、報告書を Web サーバーに配置すると、HTML 準拠の Web ブラウザーで表示できます。

次に、QMF によって報告書の前後に配置される HTML コーディングのサンプルを示します。これらのタグ・セットはそれぞれ、開始タグと終了タグで構成されます。終了タグは斜線 (/) で始まり、すべてのタグが大括弧で囲まれています。

これらのタグの詳細については、HTML の資料を参照してください。

```
<HTML>
<HEAD>
<TITLE>
報告書
</TITLE>
</HEAD>
<BODY>
<PRE>
```

```

                J & H SUPPLY COMPANY
AVERAGE SALARY (DEPTS 10, 15, 20)
                REPORT 17
```

DEPT	JOB	AVERAGE SALARY
10	MGR	20865.86
	*	20865.86
15	CLERK	12383.53
	MGR	20659.80
	SALES	16052.83
	*	15482.33
20	CLERK	13878.67
	MGR	18357.50
	SALES	18171.25

```

* 16071.52
=====
17473.52

COMPANY NAME
REPORT 17

```

```

</PRE>
</BODY>
</HTML>

```

この表では、この HTML コーディングについて簡単に説明します。

表 39. エクスポート後の HTML 報告書に使用される HTML タグ

タグ・セット	説明
<HTML></HTML>	これらのタグは、HTML 文書としてファイルを定義します。
<HEAD></HEAD>	これらのタグは、文書のヘッダーの境界をマークする。
<TITLE></TITLE>	これらのタグの間に、QMF が「報告書」の語を挿入する。これらのタグの間の内容は、HTML 文書のタイトルに含まれる。タイトルの配置は、ブラウザおよびプラットフォームによって異なる。これらのタグは、ヘッダー内に置かれる。
<BODY></BODY>	これらのタグは、ヘッダーの後に続き、文書の本文を含む。報告書出力は、文書の本文に置かれる。
<PRE></PRE>	これらのタグの間の内容は、そのまま表示される。これらの間では、HTML 形式設定は実行されない。QMF は、これらのタグの間の報告書出力を、HTML 文書の本文に置く。

制御情報なしでの報告書のエクスポート

報告書のエクスポート時に、QMF はデフォルトで、報告書のさまざまな箇所に制御情報を設定します。

エクスポートされた報告書は、例えば次のような形式になります。

```

H QMF 17 R 01 E V W E R 02 03 14/03/05 11:07
V 1001 006 PERIOD
V 1002 003 016
T 1010 005 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000010 000001
R C 000011 000013 000026 000001
R L 000027 000029 000035 000001
R C 000036 000038 000047 000001
R C 000048 000050 000062 000001
L 110 10000000
L 110 10000000
L 170 10000000 DEPTNUMB DEPTNAME MANAGER DIVISION LOCATION
L 170 10010000 -----
L 180 11000000 10 HEAD OFFICE 160 CORPORATE NEW YORK
L 180 11000000 15 NEW ENGLAND 50 EASTERN BOSTON
L 180 11000000 20 MID ATLANTIC 10 EASTERN WASHINGTON
L 180 11000000 38 SOUTH ATLANTIC 30 EASTERN ATLANTA
L 180 11000000 42 GREAT LAKES 100 MIDWEST CHICAGO
L 180 11000000 51 PLAINS 140 MIDWEST DALLAS
L 180 11000000 66 PACIFIC 270 WESTERN SAN FRANCISCO
L 180 11000000 84 MOUNTAIN 290 WESTERN DENVER
L 120 10000000
L 120 10000000
E

```

図 22. 制御情報付きでエクスポートされた報告書の例

EXPORT REPORT コマンドで DATAFORMAT=TEXT オプションを指定すると、次の例のように、制御情報なしで報告書をエクスポートすることができます。

```

DEPTNUMB DEPTNAME MANAGER DIVISION LOCATION
-----
10 HEAD OFFICE 160 CORPORATE NEW YORK
15 NEW ENGLAND 50 EASTERN BOSTON
20 MID ATLANTIC 10 EASTERN WASHINGTON
38 SOUTH ATLANTIC 30 EASTERN ATLANTA
42 GREAT LAKES 100 MIDWEST CHICAGO
51 PLAINS 140 MIDWEST DALLAS
66 PACIFIC 270 WESTERN SAN FRANCISCO
84 MOUNTAIN 290 WESTERN DENVER

```

図 23. 制御情報なしでエクスポートされた報告書の例

横方向スタイル報告書のエクスポート

エクスポートされた横方向スタイル報告書には、エクスポートされた標準報告書には含まれないフィールドが含まれます。

次に、エクスポートされた横方向スタイル報告書のサンプルを示します。

REPORT		LINE 1	POS 1	79
J & H SUPPLY COMPANY DEPT AVERAGE SALARIES REPORT 18 (ACROSS REPORT)				
<----- JOB ----->				
	<- CLERK -->	<-- MGR --->	<- SALES -->	<- TOTAL -->
DEPT	AVERAGE SALARY	AVERAGE SALARY	AVERAGE SALARY	AVERAGE SALARY
-----	-----	-----	-----	-----
10		20865.86		20865.86
15	12383.35	20659.80	16502.83	15482.33
20	13878.68	18357.50	18171.25	16071.53
38	12482.25	17506.75	17407.15	15457.11
	=====	=====	=====	=====
	12914.76	19998.21	17372.10	16880.26
COMPANY NAME REPORT 18 PAGE 1				

サンプルの横方向スタイル報告書をエクスポートすると、次のようなエンコード形式になります。

```

H QMF 19 R 01 E V W E R 02 03 13/01/15 16:20

V 1001 006 PERIOD
V 1002 003 016
T 1010 002 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000008 000001 GROUP
R L2 000003 000005 000014 000001 AVERAGE
V 2001 005 C
V 2002 003 001
V 2003 003 YES
T 2010 004 003 2012 006 2013 006 2014 006
R 000014 000018 000009
R 000029 000031 000023
R 000042 000046 000037
R 000056 000060 000051
L 110 10100000
L 110 10100000
L 110 10100000
L 110 10000000
L 110 10000000
L 170 10000000
L 170 11000000
L 170 10000000
L 170 10000000
L 170 10010000
L 181 11000000
L 181 11000000
L 181 11000000
L 181 11000000
L 181 11000000
L 190 10010000
L 190 11000000
ddL 120 10000000
L 120 10000000
L 120 10100000
L 120 10100000
L 120 10100000
E

```

<----- JOB ----->				
	<- CLERK -->	<-- MGR --->	<- SALES -->	<- TOTAL -->
DEPT	AVERAGE SALARY	AVERAGE SALARY	AVERAGE SALARY	AVERAGE SALARY
-----	-----	-----	-----	-----
10		20865.86		20865.86
15	12383.35	20659.80	16502.83	15482.33
20	13878.68	18357.50	18171.25	16071.53
38	12482.25	17506.75	17407.15	15457.11
	=====	=====	=====	=====
	12914.76	19998.21	17372.10	16880.26

```

L 120 10100000
L 120 10100000
L 120 10100000
L 120 10100000
E

```

127 ページの表 38 は、標準的な報告書と横方向スタイル報告書の両方に共通するフィールド番号について説明しています。次の表に、エクスポートされた横方向スタイル報告書に含まれる追加フィールドを示します。

表 40. エクスポート後の横方向スタイル報告書のフィールド番号

フィールド番号	レコード・タイプ	説明
2001	V	横方向の値の形式を設定する編集コード
2002	V	横方向グループごとのデータ行数
2003	V	横方向合計列の有無を示す
2010	T	横方向報告書表
		横方向の値ごと
2012	T	書式設定された横方向値の開始位置 (横方向値は列ヘッダ行に表示されます)
2013	T	形式設定済み横方向値の終了位置
2014	T	この横方向値に関連する報告書列のセットの、前にある字下げ域を含めた、開始位置

集約された列が横方向報告書に含まれている場合、フィールド 1014、1015、および 1016 は、横方向値の集約された列のセットに含まれている、フィールドの相対的な開始位置および終了位置を記述します。(表内のフィールド 2014 を参照してください。)

指示照会のエクスポート

エクスポート後の指示照会オブジェクトには、指示照会の基本パネルの確認域に表示される情報が入っています。

サンプル照会 (エクスポート前)

エクスポートされた指示照会を含むデータ・セットまたはデータ・キューは、2 つの方法で QMF にインポートできます。データ・セットまたはデータ・キューは、QMF 一時記憶域にインポートすることも、データベースに直接インポートすることもできます。指示照会をインポートするときは、QMF にインポートされる照会が、データベース内のデータと整合しているかどうかを検査されます。例えば、インポートする指示照会の表 XYZ に列 A、B、および C が含まれている場合、QMF では、列 A、B、および C が含まれている表 XYZ がデータベースに存在するかどうかを検査されます。

この例は、エクスポート前に「指示照会 (Prompted Query)」基本パネルに表示されるサンプル・エコー・テキストを示しています。


```

Tables:
  Q.STAFF(A)
  Q.ORG(B)
  Q.STAFF(C)

Join Tables:
  A.DEPT And B.DEPTNUMB
  And A.ID And C.ID

Columns:
  A.ID
  A.DEPT
  A.JOB
  A.SALARY
  DEPTNUMB
  C.SALARY
  C.SALARY+A.COMM

Row Conditions:
  If A.SALARY Is Greater Than 10000
  And A.DEPT Is Equal To 84 or 96

Sort:
  Descending by C.SALARY+A.COMM

Duplicate Rows:
  Keep duplicate rows

```

エクスポート後の照会の外観

次に、エクスポートされた指示照会の形式の例を示します。

```

H QMF 19 T 01 E V W E R 01 03 13/01/15 16:20
T 1110 003 002 1112 001 1113 050
R A Q.STAFF
R B Q.ORG
R C Q.STAFF
T 1150 002 002 1152 020 1153 020
R A.DEPT          B.DEPTNUMB
R A.ID            C.ID
T 1210 007 002 1212 001 1213 255
R C A.ID
R C A.DEPT
R C A.JOB
R C A.SALARY
R C B.DEPTNUMB
R C C.SALARY
R C C.SALARY+A.COMM
T 1310 009 003 1312 001 1313 008 1314 255
R 1 C          A.SALARY
R 2 IS        GT
R 3           10000
R 4 I
R 1 C          A.DEPT
R 2 IS        EQ
R 3           84
R 3           96
R 4 A
T 1410 001 002 1412 001 1413 255
R D C.SALARY+A.COMM
V 1501 001 K
E

```

エクスポートされたデータ・セットまたはデータ・キュー内のヘッダー・レコードの解釈

下表では、133 ページの『エクスポート後の照会の外観』に示したエクスポート後の指示照会に含まれている、次のヘッダー・レコードについて説明します。

H QMF 19 T 01 E V W E R 01 03 13/01/15 16:20

表 41. 指示照会ヘッダー・レコードのサンプル

例からの値	説明
H QMF 19 T	この指示照会ヘッダー・レコードは、QMF バージョン 12.1 を指定します。
01	指示照会の構造は、オブジェクト・レベル 1 です。エクスポートされた照会オブジェクトに期間指定が含まれている場合は、代わりにオブジェクト・レベル 2 が指定されます。
E	形式はエンコードされます (エクスポート後の書式、報告書、および指示照会で使用される形式)。
V	エクスポート後の指示照会にエラーまたは警告が含まれていない。
W	ファイルに指示照会全体が含まれている。
E	英語は、オブジェクトのエクスポート時に使用された各国語。
R	インポート時に、一時記憶域のオブジェクトが置換される。
01	制御域の長さは 1 バイト。
03	整数長さフィールドの長さは 3 バイト。
13/01/15	この日付スタンプは January 15, 2013 を示します。
16:20	タイム・スタンプは、4:20 PM を指定します。

指示照会エンコード形式の完全な例については、133 ページの『エクスポート後の照会の外観』を参照してください。

エクスポートされた指示照会のレコードの解釈

表定義 (フィールド番号 1110) は常にエクスポートされます。結合条件 (フィールド番号 1510) は、複数の表を選択した場合、常にエクスポートされます。

指示照会ファイルをインポートするには、そのファイル内で H レコードの後に、エンコードされた表の T レコードが続いている必要があります。表を指定しないと、空の照会がインポートされます。結合条件は、複数の表を選択しない場合、不要です。

表 42. エクスポートされた指示照会オブジェクトの表番号とフィールド番号

レコード・タイプ	表番号	フィールド番号	フィールドの説明
T	1110	-	<p>表定義の表</p> <p>133 ページの『エクスポート後の照会の外観』に示したエクスポート後の指示照会のこのセクションに含まれている T レコードは、このセクションを、照会と関係のある表名が含まれている部分として識別します。</p> <p>T 1110 003 002 1112 001 1113 050</p> <p>「003」は 3 つの表を指し、「002」は 2 つのフィールド番号 (1112 および 1113) を指します。ただし、エクスポートされた照会オブジェクトに期間指定が含まれている場合、値「003」は、3 つのフィールド番号 (1112、1113、および 1114) を示すために使用されます。</p> <p>それぞれの T レコードの後には R レコードが続きます。この例では、R レコードは指示照会結合に関係する表を示しています。</p> <p>R A Q.STAFF R B Q.ORG R C Q.STAFF</p> <p>エクスポート・ファイルのこの部分は、132 ページの『サンプル照会 (エクスポート前)』に示す指示照会の次の部分に対応します。</p> <p>Tables: Q.STAFF(A) Q.ORG(B) Q.STAFF(C)</p>
		1112	表 ID (有効な表 ID は A-Z、および #、\$、@)
		1113	表名 (最大 280 文字)
		1114	期間指定 (最大 560 文字)。このフィールド番号は、エクスポートされた照会オブジェクトに期間指定が含まれていた場合にのみ、組み込まれます。

表 42. エクスポートされた指示照会オブジェクトの表番号とフィールド番号 (続き)

レコード・ タイプ	表番号	フィールド番 号	フィールドの説明
T	1150	-	<p>結合条件表</p> <p>133 ページの『エクスポート後の照会の外観』に示したエクスポート後の指示照会のこのセクションに含まれている T レコードは、このセクションを、照会と関係のある結合条件が含まれている部分として識別します。各 T レコードの後には、結合する表を識別する R レコードが続きます。</p> <p>T 1150 002 002 1152 020 1153 020 R A.DEPT B.DEPTNUMB R A.ID C.ID</p> <p>エクスポート後のサンプル照会のこの部分は、132 ページの『サンプル照会 (エクスポート前)』に示すサンプル指示照会の次の部分に対応します。</p> <p>Join Tables: A.DEPT And B.DEPTNUMB And A.ID And C.ID</p>
		1152	列 1 の名前: 短い長さ (22) 拡張された長さ (34)
		1153	列 2 の名前: 短い長さ (22) 拡張された長さ (34)

表 42. エクスポートされた指示照会オブジェクトの表番号とフィールド番号 (続き)

レコード・ タイプ	表番号	フィールド番 号	フィールドの説明
T	1210	-	<p>リスト</p> <p>133 ページの『エクスポート後の照会の外観』に示したエクスポート後の指示照会のこのセクションに含まれている T レコードは、このセクションを、照会と関係のある列名が含まれている部分として識別します。それぞれの T レコードの後には、列名を示す R レコードが続きます。このセクションは、エクスポート後の照会で次のように表示されます。</p> <p>T 1210 007 002 1212 001 1213 255 R C A.ID R C A.DEPT R C A.JOB R C A.SALARY R C B.DEPTNUMB R C C.SALARY R C C.SALARY+A.COMM</p> <p>エクスポート後の照会のこのセクションは、132 ページの『サンプル照会 (エクスポート前)』に示すサンプル照会の次のセクションに対応します。</p> <p>Columns: A.ID A.DEPT A.JOB A.SALARY DEPTNUMB C.SALARY C.SALARY+A.COMM</p>
		1212	<p>列のタイプ</p> <ul style="list-style-type: none"> • C=列 • E=式 • S=式がある合計機能 • F=1 列しかない合計機能
		1213	<p>列名、式、または合計機能: 短い長さ (255) 拡張された長さ (560)</p>

表 42. エクスポートされた指示照会オブジェクトの表番号とフィールド番号 (続き)

レコード・ タイプ	表番号	フィールド番 号	フィールドの説明
T	1310	-	<p>行選択条件</p> <p>133 ページの『エクスポート後の照会の外観』のエクスポート後の指示照会のこのセクションにある T レコードは、エクスポート後の照会のこのセクションを、照会条件が含まれている部分として示します。それぞれの T レコードの後には、各条件の特性を示す R レコードが続きます。このセクションは、エクスポート後の指示照会で次のように表示されます。</p> <pre>T 1310 009 003 1312 001 1313 008 1314 255 R 1 C A.SALARY R 2 IS GT R 3 10000 R 4 I R 1 C A.DEPT R 2 IS EQ R 3 84 R 3 96 R 4 A</pre> <p>エクスポート後の照会のこのセクションは、132 ページの『サンプル照会 (エクスポート前)』に示す照会の次のセクションに対応します。</p> <p>Row Conditions: If A.SALARY Is Greater Than 10000 And A.DEPT Is Equal To 84 or 96</p>
		1312	<p>入力タイプ</p> <ul style="list-style-type: none"> • 1 - 演算子の左方 • 2 - 演算子 • 3 - 演算子の右方 • 4 - 結合子
		1313	<p>入力タイプが '1' の場合、列タイプを識別する</p> <ul style="list-style-type: none"> • C=列 • E=式 • S=合計機能 • F=合計機能 (列名だけを指定した場合)
			<p>入力タイプが '2' の場合、動詞を識別する</p> <ul style="list-style-type: none"> • 動詞 'is' を意味する IS (デフォルト) • 動詞 'is not' を意味する ISN
			<p>入力タイプが '3' の場合 (使用されていない)</p>
			<p>入力タイプが '4' の場合、コネクターを識別する</p> <ul style="list-style-type: none"> • '論理和 (or)' を意味する OR • '論理積 (and)' を意味する A (デフォルト)

表 42. エクスポートされた指示照会オブジェクトの表番号とフィールド番号 (続き)

レコード・ タイプ	表番号	フィールド番 号	フィールドの説明
		1314	入力タイプが '1' の場合、このフィールドは列名、式、または合計機能: 短い長さ (255) 拡張された長さ (560)
			入力タイプが '2' の場合、演算子を識別する <ul style="list-style-type: none"> • '等しい (equal to)' を意味する EQ • 'より小さい (less than)' を意味する LT • '以下 (less than or equal to)' を意味する LE • 'より大きい (greater than)' を意味する GT • '以上 (greater than or equal to)' を意味する GE • '間 (between)' を意味する BT • '開始点 (starting with)' を意味する SW • '終了点 (ending with)' を意味する EW • '包含 (containing)' を意味する CT • NULL を意味する NL
			入力タイプが '3' の場合、値を識別する
			入力タイプが '4' の場合、(使用されていない)
T	1410	-	ソート条件表 133 ページの『エクスポート後の照会の外観』に示したエクスポート後の指示照会のこのセクションに含まれている T レコードは、このセクションを、照会のソート条件が含まれている部分として識別します。それぞれの T レコードの後には、各ソート条件の特性を示す R レコードが続きます。このセクションは、エクスポート後の指示照会で次のように表示されます。 T 1410 001 002 1412 001 1413 255 R D C.SALARY+A.COMM エクスポート後の照会のこのセクションは、132 ページの『サンプル照会 (エクスポート前)』に示すサンプル照会の次のセクションに対応します。 Sort: Descending by C.SALARY+A.COMM
		1412	ソート方向 <ul style="list-style-type: none"> • '昇順 (ascending)' を意味する A • '降順 (descending)' を意味する D
		1413	列: 短い長さ (255) 拡張された長さ (560)

表 42. エクスポートされた指示照会オブジェクトの表番号とフィールド番号 (続き)

レコード・タイプ	表番号	フィールド番号	フィールドの説明
V		1501	<p>重複行の処理:</p> <ul style="list-style-type: none"> • '保持 (keep)' を意味する K • '廃棄 (discard)' を意味する D <p>例えば、133 ページの『エクスポート後の照会の外観』に示したエクスポート後のサンプル指示照会に含まれている次の行は、値 K のレコード長が 1 (「001」) であることを示しています。この行はさらに、照会を作成したユーザーが重複行を保持するように指定したこと (「K」) も示しています。</p> <p>V 1501 001 K</p> <p>エクスポート後の照会のこのセクションは、132 ページの『サンプル照会 (エクスポート前)』に示すサンプル照会の次のセクションに対応します。</p> <p>Duplicate Rows: Keep duplicate rows</p>

フィールド 1313 と 1314 の値の意味は、表番号 1310 の表のフィールド番号 1312 に示されている順序番号によって異なります。

関連資料:

103 ページの『ヘッダー・レコード』

エクスポートされた書式、報告書、または指示照会のほとんどのレコードは、可変長形式です。しかし、ヘッダー・レコードは、レコードが入っているデータ・セットまたはデータ・キューが可変長形式の場合でも固定形式です。

108 ページの『表行レコード (R)』

エクスポートされた書式、報告書、または指示照会の R レコードには、エンコード表内の単一行の一連の値が含まれています。R レコードには、関連する T レコードによって記述されている順に並べられた値のリストが入っています。R レコードは、T レコードに指定されているデータ値の位置と長さの記述に一致しています。

エクスポート後の指示照会が有効な形式であることの確認

アプリケーションによって変更された指示照会オブジェクトは、特定の規則に従ってインポートします。

アプリケーションによって変更された指示照会オブジェクトをインポートするには、以下の条件に注意してください。

- 指示照会ファイルをインポートする場合、インポートされるレコードは、ヘッダー (H) レコードの後に、以下の特定の順序で並んでいる必要があります。
 1. T レコード (表定義)
 2. R レコード (表名)
 3. T レコード (列定義)

4. R レコード (列)

- 行条件レコード (表番号 1310) は、各条件内で、項目タイプ順序番号 (フィールド番号 1312) に従った並び順でなければなりません。つまり、指示照会の確認域に行データが現れる順と同じでなければなりません。

残りのレコードの並び順は任意です。

- 表定義の表 (T レコード 1110) は、他の表または V レコードの前に現れなければなりません。
- 表の T レコード内の行カウントの値は、* または 0-15 の整数でなければなりません。行カウントのゼロ値によって、その照会のすべてが無視されます。これは空の照会がインポートされることを意味します。
- QMF は、指示照会インポートに関する警告を出しません。
- 2 番目の表に関する表 (表 1110) を指定すると、QMF はエラーを出し、その表の内容は無視されます。指示照会はインポート時にデフォルト値を提供しません。
- ソート表がある場合、その前に列表がなければなりません。
- QMF はインポート・ファイル内の重複レコードを受け入れます。そのレコードの最新値が使用されます。
- すべての列名は、インポート時に、表 ID によって修飾されていなければなりません。
- 指示照会を事前割り振りデータ・セットにエクスポートする場合、許可される最小論理レコード長 (LRECL) は 259 バイトです。
- 指示照会のエクスポート形式は、使用する言語に関係なく同じです。形式は言語によって異なることはありません。ヘッダー・レコード内の言語バイトは、インポート時に無視されます。指示照会をエクスポートするとき使用するコードについては、エクスポートされた指示照会オブジェクトの表番号とフィールド番号のリストで説明しています。

合計機能および合計式は変換されません。したがって、合計機能 COUNT、AVG、SUM、MIN、および MAX は変更されないままです。

書式および指示照会のインポート

書式または指示照会をインポートするための規則に注意してください。

書式または指示照会をインポートする場合は、以下に注意してください。

- ファイルの構成は、可変長レコードの場合も固定長レコードの場合もあります。
- レコード ID (H、V、T、R、E、*、L、または C) が各レコードの最初の位置になければなりません。
- 最初の 2 バイトは、制御情報 (制御域) 用に予約されています。
- すべてのデータ・フィールド (フィールド番号、長さ、および値を含む) の前後に区切り文字が 1 つずつ必要です。例外として、レコード内の最後のデータ・フィールドでは、レコード終わりが区切り文字のように機能するため、後に区切り文字を続ける必要はありません (本書の例では、区切り文字としてブランク文字を使用しています)。

- QMF は、インポート時に重複するデータ値または表を検出すると、前の値または表を置換します。しかし、特定のオブジェクトの規則に違反する重複は許可されません。例えば、書式用に提供された列数は、最初の COLUMNS 表の処理後に変更できません。
- 表番号、フィールド番号、および数値の長さに、先行ゼロまたは先行空白を組み込むことができます。ただし、末尾空白 (空白区切り文字を除く) は許可されません。フィールドは右寄せする必要があります。
- 長さまたはカウントの代わりに * を使用すると、左寄せされ、末尾空白で埋められます。
- データ入力フィールド用に提供された値がフィールドより短い場合は、末尾空白で埋められます。提供された値がフィールドより長い場合は、切り捨てられます。
- レコードが固定形式の長さより短い場合、指定されずに残されたフィールドは空白と想定されます。

関連資料:

113 ページの『書式のエクスポート』

書式オブジェクトには、QMF のすべての書式パネルに指定されている情報が入っています。書式をエクスポートすると、QMF はデフォルト値から派生した値を持つ書式パネルをエンコード形式に変換します。

143 ページの『外部化オブジェクトのサイズの仕様』

外部の表およびオブジェクトには、レコード・サイズとレコード形式の両方の仕様があり、これらはオブジェクトのタイプに応じて異なります。

プロシージャーおよび SQL 照会

これらのオブジェクトを表す TSO データ・セットまたは CICS データ・キューの形式は、すべての形式の中で最も単純です。データ・セットまたはデータ・キューの各レコードは、画面に表示される行のイメージ (79 バイトの固定長レコード) と本質的に同じです。

これらのオブジェクトの各行は 79 バイトですが、新規データ・セットおよび既存データ・セットの論理レコード長 (LRECL) は 79 から 32,760 バイトにすることができます。新規データ・セットにエクスポートすると、LRECL は、大域変数 DSQEC_DSLRECL1 で指定された値に設定されます。LRECL が 79 を上回る場合は、エクスポート中に QMF によって各オブジェクト・レコードに空白が埋め込まれます。

単純な SQL 照会の例を次に示します。

```
SQL query
SELECT *
FROM Q.STAFF
```

図 24. 単純な SQL 照会

次に、対応する外部形式の照会の例を示します。

```
SELECT *
FROM Q.STAFF
```

レコード形式が単純なので、QMF の外側での SQL 照会またはプロシーチャーの、作成あるいは編集は単純です。SQL 照会またはプロシーチャーは、列 1 から 79 の固定長データで構成されます。インポート中には、列 80 から 32,760 のデータは無視されます。結果のデータ・セットまたはからデータ・キューをインポートすると、照会またはプロシーチャーが QMF 一時記憶域に置かれ、実行が可能になります。

エクスポートされた書式ベースの図表および QBE 照会

書式ベースの図表および例示照会 (QBE) 照会オブジェクトを QMF 環境外で処理するために、エクスポートできます。

エクスポートされた書式ベースの図表

書式ベースの図表を、QMF オブジェクトとしてデータベースに保管したり、データベースから取得したりすることはできません。書式ベースの図表を QMF にインポートすることはできません。

図表を QMF にエクスポートすると、データが報告書形式から図形データ形式 (GDF) に変換されます。GDDM 形式の 1 つである GDF は、データ交換用の既存の標準です。エクスポート後の図表データを GDDM ユーティリティーを使用して印刷したり、文書に組み込んだりすることができます。

エクスポート後の図表オブジェクトを、GDF 形式に設定されたデータ・セットと同様に使用することができます。例えば、文書構成プログラム (DCF) を使用すると、アプリケーションは QMF 報告書 (印刷報告書またはエクスポートされた報告書を使用) と QMF 図表 (エクスポートされた図表を使用) を結合でき、形式設定された情報をプリンターへ送信できます。

エクスポートされた QBE 照会

QBE 照会オブジェクトは、QMF の内部形式を使用してエクスポートされます。この形式は、どのような方法でも変更できません。

外部化オブジェクトのサイズの仕様

外部の表およびオブジェクトには、レコード・サイズとレコード形式の両方の仕様があり、これらはオブジェクトのタイプに応じて異なります。

次の表に、TSO および CICS のインポート・ファイルとエクスポート・ファイルの両方の仕様を示します。CICS の場合は、レコード・サイズが適用されません。例えば、一時記憶域キューから、レコード・サイズが 32 KB の SQL 照会をインポートした場合、QMF はそれを 79 バイトに切り捨てます。

レコード形式は、CICS 一時記憶域または一時データ・キューの要素ではありません。一時記憶域キューは、レコードをその形式に関係なく保持します。一時データ・キューは、宛先管理テーブル (DCT) に対して定義され、レコード形式は無視されます。

EXPORT または IMPORT コマンドでは、データ・セットまたは CICS データ・キューの名前を指定する必要があります。キュー名には、デフォルトの接頭部または接尾部はありません。CICS 一時記憶域キュー名は 8 バイトです。一時データ・キュー名は 4 バイトです。

次の省略語は、表のレコード形式に使用されます。

- FB - 固定ブロック形式
- VB - 可変ブロック形式
- VBS - VBS (複数のブロックに渡って変数が格納される) 形式

表 43. ファイルおよびデータ・セットの属性

オブジェクト	レコード・サイズ	レコード形式
データまたは表 (QMF 書式)	最大サイズ: 7,000 バイト	固定長 (FB)
データまたは表 (IXF 形式)	最大サイズ: 32,756 定義列が入っているエクスポート後の書式の最小の LRECL は、161 バイトです。 インポート時に IXF データ・セットまたは CICS データ・キューとして QMF が受け入れる最小 LRECL は 49 バイト レコード・サイズは通常、エクスポート中の表内のデータ行の長さ (NULL 標識および DBCS 区切り文字のスペースを含む) に、IXF D タイプ・レコード・カウント・フィールドの長さ (5 バイト) を加算した値になります。	可変長 (VB)
データまたは表 (XML 形式)	最大サイズ: 2 GB	可変長 (VBS)

表 43. ファイルおよびデータ・セットの属性 (続き)

オブジェクト	レコード・サイズ	レコード形式
データまたは表 (CSV 形式)	<p>新規データ・セットにエクスポートされる最大 LRECL は、EXPORT コマンドの HEADER オプションに YES と NO のどちらが指定されているかに基づいて計算されます。</p> <ul style="list-style-type: none"> • HEADER=YES を指定すると、次の式が使用されます。 $\max(\text{Column_Names_Total_Length}, \text{Column_Data_Value_Total_Length}) + (\text{number of columns} * 3 - 1) + 4$ • HEADER=NO を指定すると、次の式が使用されます。 $(\text{length of column data values}) + (\text{number of columns} - 1) + (\text{number of columns} * 2) + 4$ <p>既存のデータ・セットにエクスポートする場合は、既存のデータ・セットの LRECL が、エクスポート可能な最大 LRECL になります。</p>	可変長 (VB)
指示照会	<p>最大: 7,290 バイト</p> <p>最小: EXPORT では 266 バイト、IMPORT では 41 バイト</p>	<p>EXPORT 時に可変長 (VB)</p> <p>IMPORT 時に固定長 (FB) または可変長 (VB)</p>
SQL 照会	<p>新規および既存のデータ・セットへの EXPORT の場合は 79 から 32,760 バイト、IMPORT の場合は任意のサイズが可能であるが、79 バイトに切り捨てられる</p>	<p>既存のデータ・セットへの EXPORT の場合は固定長 (FB) または可変長 (VB)、新規データ・セットへの EXPORT の場合は固定長 (FB)</p> <p>IMPORT 時に固定長 (FB) または可変長 (VB)</p>
QBE 照会	<p>1,024 バイトでなければならない。</p> <p>空の QBE 照会は 828 バイトです。</p>	可変長 (VB)
書式	<p>最大: 7,290 バイト</p> <p>最小: EXPORT では 161 バイト、IMPORT では 23 バイト</p>	<p>EXPORT 時に固定長 (FB)</p> <p>IMPORT 時に固定長 (FB) または可変長 (VB)</p>
Proc	<p>新規および既存のデータ・セットへの EXPORT の場合は 79 から 32,760 バイト、IMPORT の場合は任意のサイズが可能であるが、79 バイトに切り捨てられる</p>	<p>既存のデータ・セットへの EXPORT の場合は固定長 (FB) または可変長 (VB)、新規データ・セットへの EXPORT の場合は固定長 (FB)</p> <p>IMPORT 時に固定長 (FB) または可変長 (VB)</p>
報告書	<p>最大: 7,290 バイト</p> <p>最小: 65 バイト</p>	可変長 (VB)

表 43. ファイルおよびデータ・セットの属性 (続き)

オブジェクト	レコード・サイズ	レコード形式
HTML 報告書	最大: 32,000 バイト	可変長 (VB)

記憶域についての考慮事項

CICS データ・キューおよび TSO データ・セットとの間でオブジェクトをインポートおよびエクスポートするときには、QMF での記憶域の処理方法に注意してください。

CICS データ・キュー

CICS データ・キューにエクスポートされたオブジェクトについては、QMF でのキューの処理方法を理解してください。

オブジェクトを CICS データ・キューにエクスポートするときには、以下の条件に注意してください。

- CICS では、IMPORT コマンドと EXPORT コマンドのいずれの場合も、QUEUETYPE オプションを指定する必要があります。デフォルト値はありません。
- CICS では、一時データ (TD) キューからオブジェクトをインポートする場合、正しいオブジェクト・タイプを指定する必要があります。QMF がキューの内容を取得した後、キューは空になります。例えば、一時データ・キュー内のオブジェクト・タイプがプロシージャの場合に、「書式」を指定すると、QMF によってエラー・メッセージが出されます。しかし、そのキューは現在は空になっているので、(正しいオブジェクト・タイプを指定しても) 同じキューを使用して IMPORT コマンドをもう一度正常に出すことはできません。
- CICS では、IMPORT コマンドを出す前に、一時データ・キューまたは一時記憶域 (TS) キューに、単一の完全な QMF オブジェクトが入っていなければなりません。
- 一時データ・キューへエクスポートする場合、EXPORT コマンドを出す前に、そのキューがオープンされていて、使用可能で、かつ、空でなければなりません。

QMF は CICS 一時データ・キューを、一時記憶域キューとは異なる方法で処理します。

- 一時データ・キュー

QMF は、画面にオブジェクトを表示する前に、一時データ・キュー全体をインポートします。これは、キュー全体の内容が記憶域または予備域に収まらなければならないことを意味します。予備記憶域の使用を指定するためには、DSQSPILL パラメーターを使用できます。オブジェクトのサイズが大きく、予備データのファイルを使用している場合は、オブジェクトの表示に遅延が生じる可能性があります。

CICS 区画内一時データ・キューには、最大 32 KB の行データを保持できます。区画外一時データ・キューは、オブジェクトを保持するのに十分な大きにすることができます。

- 一時記憶域待ち行列

デフォルトで、QMF は、ユーザーに表示する前に、一時記憶域のうちの約 100 行を読み取ります。一時記憶域キューには 32 KB までの行データを保持できません。

QMF は、IMPORT コマンドおよび EXPORT コマンドで SUSPEND パラメーターを使用して、CICS によるコマンド実行時期の制御を可能にします。

IMPORT および EXPORT コマンドの SUSPEND パラメーターは、照会が使用中である場合に行うべきアクションを指定します。SUSPEND パラメーターを YES に設定した場合、QMF は CICS データ・キュー名に関して CICS ENQ (待機) を発行します。この設定では、キューが使用可能になるまで待機してから QMF オブジェクトをキューに書き込むように CICS に指示します。待機することで、QMF トランザクションが、キューで処理中の他のジョブを妨げることがなくなります。

SUSPEND パラメーターを NO に設定した場合、EXPORT コマンドは取り消され、メッセージが戻されます。SUSPEND のデフォルト値は NO です。

TSO データ・セット

TSO データ・セットにエクスポートされたオブジェクトの場合は、記憶域管理システムを必ず適切に構成してください。

標準 DASD デバイスを使用している場合は、使用している記憶域管理ソフトウェアが拡張データ・セットの動的割り振りを処理するように構成されていることを確認してください。これらのデータ・セットを構成するときには、デフォルトの記憶域クラスを指定します。ご使用の記憶域管理システムがこの方法で構成されている場合、オブジェクトをエクスポートすると、QMF によってデータ・セットが動的に割り振られます。データ・セットが存在しない場合、QMF では、EXPORT コマンドに指定された名前が使用されます。データを XML 形式でエクスポートする場合、データ・セットを適切に構成していないと、動的割り振りエラーを受け取ることがあります。拡張データ・セットの動的割り振りを構成する方法の詳細については、記憶域管理ソフトウェアから提供される情報を参照してください。

標準 DASD デバイスを使用していない場合、EXPORT コマンドを使用する前にデータ・セットを事前に割り振る必要があります。大域変数を使用して、エクスポートされるオブジェクトを含むことになる新規データ・セットのタイプとサイズを指定できます。

- 大域変数 DSQEC_PO を使用して、オブジェクトを新規データ・セットのメンバーにエクスポートしたときに作成される区分データ・セットのタイプを指定します。タイプとして指定できるのは、ご使用のサイトのデフォルト・タイプ、PDS データ・セット、または PDSE データ・セットです。
- 大域変数 DSQEC_DSALLOC_DIR を使用して、新規 PDS データ・セットのメンバーをエクスポートするときのディレクトリー・ブロックの数を指定します。デフォルトは 20 です。
- 大域変数 DSQEC_DSALLOC_PRI を使用して、トラック単位で 1 次スペース割り振りを指定します。デフォルトは、15 トラックです。
- 大域変数 DSQEC_DSALLOC_SEC を使用して、トラック単位で 2 次スペース割り振りを指定します。デフォルトは、105 トラックです。

関連資料:

240 ページの『コマンドとプロシージャの実行を制御する大域変数』
DSQEC 大域変数は、コマンドおよびプロシージャの実行方法を制御します。こ
れらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

第 10 章 QMF アプリケーションのデバッグ

エラー処理に加えて、QMF には呼び出し可能インターフェース・アプリケーションに対するデバッグ機能も用意されています。

REXX トレース機能を使用するには、REXX TRACE ステートメントを使用します。

関連概念:

33 ページの『第 6 章 ISPF サービスを使用する QMF アプリケーションの作成』

独自のユーザー・インターフェースを持つアプリケーションを作成することで、「QMF」パネルをう回することができます。呼び出し可能インターフェースまたはコマンド・インターフェースのいずれかを使用して、ISPF を使用するアプリケーションを作成できます。

呼び出し可能インターフェース・アプリケーションのデバッグ

QMF には、アプリケーションのデバッグのために、L と A という 2 つのトレース・オプション、および様々なレベルのトレース詳細が用意されています。

トレース用の L オプション

L オプションは、メッセージおよびコマンドを外部 TSO データ・セットまたは CICS データ・キューに書き込みます。

選択できる L オプションは 2 つあります。

- L1** 全ての QMF メッセージを QMF トレース・データ出力に書き込む。
- L2** 全ての QMF メッセージおよびコマンドを QMF トレース・データ出力に書き込む。

L オプションは以下のいずれかの方法で設定することができます。

- DISPLAY PROFILE コマンドを出し、PROFILE が表示されたら、TRACE オプションを L1 または L2 に変更する。
- 次のコマンドを出します。

```
SET PROFILE (TRACE=x
```

このステートメントで、*x* は L1 または L2 のいずれかです。

関連概念:

151 ページの『QMF トレース・データ出力の割り振り』

トレースを使用する場合、QMF を開始する前に、QMF トレース・データ出力を割り振る必要があります。

トレース用の A オプション

A オプションを使用して、QMF アプリケーション・サポート・サービスのトレース・レベルを指定することができます。

A オプションは、A0、A1、または A2 です。A0 はデフォルト値で、A トレースを全く実行しないことを示します。A1 および A2 はさらに詳細な結果を提供します。このパターンは、他の QMF トレース・オプションにも使用されます。

A オプションの指定は、L オプションの指定と同様に QMF SET PROFILE コマンドを介して、または DISPLAY PROFILE コマンドの実行後に画面上にこのオプションを入力することによって行います。例えば、デバッグするアプリケーションを呼び出す直前に、次のステートメントを入力することができます。

```
SET PROFILE (TRACE=L2A1)
```

アプリケーションを開始すると、L2 トレースと A1 トレースの両方が有効になっています。

現在の A オプションの設定値を判別するには、変数 DSQAO_APPL_TRACE を調べてください。設定値 A0、A1、または A2 の場合、この変数値は 0、1、または 2 です。DSQAO_APPL_TRACE の値を使用して、次に示す REXX アプリケーションのように、アプリケーションで必要なトレースを選択することができます。

```
/* REXX program to set tracing */
call dsqcix "GET GLOBAL(A_TRACE=DSQAO_APPL_TRACE"
if a_trace > 0 then
do
  /* trace code for both A1 and A2 */
  :
  if a_trace = 2 then
  do
    /* trace code for just A2 */
  :
  end
end
```

図 25. アプリケーション・サポート・サービスのトレースを設定するために使用できるサンプルの REXX プログラムの構造

トレースをオフにする

トレースをオフにするには、SET PROFILE コマンドを使用します。

何らかの理由でトレースをオフにする必要がある場合は、SET PROFILE (TRACE=NONE コマンドを発行します。

このコマンドによって、QMF セッションの残りの部分のトレースが中止されますが、永続 QMF プロファイルは影響を受けません。

QMF トレース・データ出力の割り振り

トレースを使用する場合、QMF を開始する前に、QMF トレース・データ出力を割り振る必要があります。

元の割り振りが要件に合わない場合には、データ・セットまたはデータ・キューの再割り振りが必要なことがあります。

TSO 用に QMF トレース・データ出力を割り振る方法の例については、使用している言語のプログラミング言語の指定を参照してください。

例では、コマンドは順次トレース・データ・セットまたはデータ・キューを割り振っています。QMF セッションの終了後に、それらを調べることができます。出力は固定長の 80 文字のレコードから構成されます。

CICS では、QMF がトレース・データを置くロケーションを、プログラム・パラメーター DSQSDBQT および DSQSDBQN を使用して指定することができます。QMF は大量のトレース・データを生成する可能性があるため、CICS 一時記憶域を使用する場合は注意が必要です。キュー・サイズを超えるトレース・データは廃棄されるため、CICS 一時記憶域は、メッセージまたは小規模アプリケーションからのトレース・データ用のみに使用します。

関連資料:

153 ページの『第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定』

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

57 ページの『START コマンドのキーワード』

START コマンドでキーワードを指定できます。

トレース用の QMF MESSAGE コマンド

QMF MESSAGE コマンドを、アプリケーションの終了時にメッセージの表示以外に使用することができます。すなわち、QMF トレース・データ出力にメッセージを記録するために使用することができます。

メッセージを記録するには、L トレース・オプションを L1 または L2 に設定して、アプリケーションを実行します。MESSAGE コマンドで処理されたすべてのメッセージが、他の QMF メッセージとともに、QMF トレース・データ出力に記録されます。L トレース・オプションが L2 に設定された場合は、コマンドも記録されます。

MESSAGE コマンドをプログラム内の重要なロケーションに置くことによって、有用な情報を QMF トレース・ファイルにログを記録することができます。その情報は、ディスプレイ装置または印刷出力で調べることができます。

以下の行は、トレースをオンにして、トレース出力に表示される意味のあるメッセージを発行する方法の例です。

```
call dsqcix "SET PROFILE (TRACE=L2"  
:  
:  
call dsqcix "MESSAGE (TEXT='QUERYA COMPLETED SUCCESSFULLY!'"
```

```
⋮  
call dsqcix "MESSAGE (TEXT='EXECB ENTERED WITH VALUE OF 7')"  
⋮
```

この例では、メッセージ「QUERYA COMPLETED SUCCESSFULLY」と「EXECB ENTERED WITH A VALUE OF 7」を含むレコードが、QMF トレース・データ出力に書き込まれています。

QMF メッセージは、リリースによって異なる可能性があるため、QMF トレース・データ出力をアプリケーションの入力として使用しないでください。

関連概念:

151 ページの『QMF トレース・データ出力の割り振り』

トレースを使用する場合、QMF を開始する前に、QMF トレース・データ出力を割り振る必要があります。

149 ページの『トレース用の L オプション』

L オプションは、メッセージおよびコマンドを外部 TSO データ・セットまたは CICS データ・キューに書き込みます。

START コマンドまたはその他の QMF コマンドのエラー

DSQCOMM は、レベルによってはメッセージ・テキストを含むことがあります。START コマンド (または任意の QMF コマンド) が失敗した場合、このメッセージ・テキストを使用して問題をトラブルシューティングすることができます。

第 11 章 呼び出し可能インターフェースの使用に関するプログラミング言語の指定

QMF アプリケーション・プログラミング・インターフェースは、複数のプログラミング言語に使用できます。

IBM では、プログラムをアセンブル (またはコンパイル) してリンク・エディットする方法、および呼び出し可能インターフェースを使用してプログラムを実行する方法について説明します。IBM は以下の例にある REXX EXEC、JCL、または CLIST を提供していませんが、これらをコピーして、お客様の要件に合うように変更できます。

アセンブラー言語インターフェース

QMF では、アセンブラー言語を呼び出し可能インターフェースで使用することができます。

アセンブラー用のインターフェース連絡域マッピング (DSQCOMMA)

DSQCOMMA は アセンブラー 言語用の DSQCOMM マッピングで、製品に用意されています。

この表は、DSQCOMMA の各値を示しています。

表 44. DSQCOMMA インターフェース連絡域の内容

構造名	データ・タイプ	説明
DSQ_RETURN_CODE	DS F	QMF コマンドの実行後の状況を示す値は次のとおり。 DSQ_SUCCESS 要求が正常に実行された DSQ_WARNING 警告を伴って正常に終了 DSQ_FAILURE コマンドが正しく実行されなかった DSQ_SEVERE 重大エラー、QMF セッションは終了
DSQ_INSTANCE_ID	DS F	START コマンドの実行時に、QMF によって設定される ID

表 44. DSQCOMMA インターフェース連絡域の内容 (続き)

構造名	データ・タイプ	説明
DSQ_COMM_LEVEL	DS CL12	DSQCOMM のレベルを識別する ご使用のアプリケーションに、この変数を DSQ_CURRENT_COMM_LEVEL の値に初期設定してから QMF START コマンドを出す命令を含めてください
DSQ_PRODUCT	DS CL2	使用中の IBM 照会プロダクトを識別する 先頭に DSQ_QMF が付く変数は、QMF for TSO バージョンおよび QMF for CICS バージョンを指定します。
DSQ_PRODUCT_RELEASE	DS CL2	使用中の QMF のリリース・レベル 変数 DSQ_QMF_V11R1 は QMF バージョン 12 リリース 1 を指定します。
DSQ_RESERVE1	DS XL28	将来の使用のために予約されている。
DSQ_MESSAGE_ID	DS CL8	完了メッセージ ID
DSQ_Q_MESSAGE_ID	DS CL8	照会メッセージ ID
DSQ_START_PARM_ERROR	DS CL8	パラメーター・エラーのために START コマンドが失敗した場合の、エラーが生じたパラメーターの名前
DSQ_CANCEL_IND	DS C	QMF コマンドの実行中にユーザーが QMF セッションを取り消したかどうかによって、以下の 2 つの値のいずれかが入る。 <ul style="list-style-type: none"> • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	DS XL23	将来の使用のために予約されている。
DSQ_RESERVE3	DS XL156	将来の使用のために予約されている。
DSQ_MESSAGE_TEXT	DS CL128	完了メッセージ・テキスト
DSQ_Q_MESSAGE_TEXT	DS CL128	照会メッセージ・テキスト

アセンブラー 言語用の関数呼び出し

QMF には、アセンブラー言語プログラム用に DSQCIA という 1 つの関数呼び出しが用意されています。この関数呼び出しには、通常の構文と拡張構文の 2 つの形式があります。

DSQCIA、通常の構文

この呼び出しは、アプリケーション・プログラム変数にアクセスする必要がない QMF コマンド用です。大部分の QMF コマンドに、この呼び出しを使用します。

```
CALL DSQCIA, (DSQCOMM, CMDLTH, CMDSTR), VL
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。FULLWORD パラメーター

CMDSTR

関数呼び出しで実行された QMF コマンド (CMDLTH で指定された長さの大文字ストリング)

VL は アセンブラー の VARIABLE LIST ステートメントです。

DSQCIA 拡張構文

拡張構文形式の DSQCIA 関数呼び出しは、アプリケーション・プログラム変数へのアクセスが必要な QMF コマンド (START、TRACE、および拡張形式の GET GLOBAL と SET GLOBAL) 用です。

```
CALL DSQCIA, (DSQCOMM,CMDLTH,CMDSTR,  
             PNUM,KLTH,KWORD,VLTH,VALUE,VTYP),VL
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。FULLWORD パラメーター。

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの大文字ストリング。

PNUM

コマンド・キーワードまたはトレース域の数 (FULLWORD パラメーター)。

KLTH

指定する各キーワードまたは各トレース・タイトルの長さ (FULLWORD パラメーターまたは FULLWORD パラメーターの配列)。

KWORD

QMF キーワード、キーワード、またはトレース・タイトルのアドレス (文字、文字の配列、または KLTH で指定された長さのトレース・タイトルのアドレスの配列)。

VLTH

キーワードまたはトレース・タイトルに関連付けられている各値の長さ (FULLWORD パラメーターまたは FULLWORD パラメーターの配列)。

VALUE

各キーワードに関連付けられている値、またはトレース・タイトルに関連付けられている値のアドレス。

この値のタイプは、文字、文字の配列、FULLWORD パラメーター、または FULLWORD パラメーターの配列として VTYP パラメーターに指定します。トレース・データの場合は、VTYP が FINT である必要があります。

VTTYPE

VALUE パラメーターの内容のデータ・タイプ。

このパラメーターには、インターフェース連絡域 DSQCOMMA で提供される以下の 2 つの値のいずれかを設定します。

- 文字値を示す DSQ_VARIABLE_CHAR

VTTYPE が DSQ_VARIABLE_CHAR である場合、VALUE は妥当性検査されません。

- 整数値を示す DSQ_VARIABLE_FINT。

VTTYPE が DSQ_VARIABLE_FINT であれば、VALUE は妥当性検査されます。VALUE は整数でなければなりません。

VALUE フィールドに指定したすべての値のデータ・タイプを VTTYPE に指定する必要があります。

VL はアセンブラーの VARIABLE LIST ステートメントです。

アセンブラー・プログラミングの例

IBM は、CICS および TSO 用のサンプルのアセンブラー・プログラムを提供しています。このサンプル・プログラムはライブラリー QMF1210.SDSQSAP n のメンバーです (n は各国語 ID です)。

CICS 用のサンプルのアセンブラー・プログラム

IBM は、DSQABFAC という名前の CICS 用のサンプルのアセンブラー・プログラムを提供しています。

この CICS 用のサンプルのアセンブラー・プログラムは、QMF1210.SDSQSAP n ライブラリーにあります (n は各国語 ID です)。

アセンブラー呼び出し可能インターフェースのこのサンプル・プログラムは、以下の機能を実行します。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

```
TITLE 'Sample HLASM Query Callable Interface'
*****
*
* Sample Program: DSQABFAC
* Assembler Version of the QMF Callable Interface for CICS
*
*****
DSQABFAC DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11)
DSQABFAC AMODE 31
DSQABFAC RMODE ANY
SPACE 1
```



```

*****
* Start a query interface session *
*****
LA R4,CICOMM ESTABLISH ACCESS TO DSQCOMM
USING DSQCOMM,R4
SPACE 1
MVC DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL
ST R4,QMFP1 Address of DSQCOMM
LA R1,STARTQIL Address of START command length
ST R1,QMFP2
LA R1,STARTQI Address of START command
ST R1,QMFP3
LA R1,1 One Start command parameter
ST R1,NUMPARMS
LA R1,NUMPARMS Address of number of parameters
ST R1,QMFP4
LA R1,STARTKYL Address of keyword lengths
ST R1,QMFP5
LA R1,STARTKY Address of keywords
ST R1,QMFP6
LA R1,STARTVL Address of value lengths
ST R1,QMFP7
LA R1,STARTV Address of values
ST R1,QMFP8
LA R1,DSQ_VARIABLE_CHAR Address of value data type
ST R1,QMFP9
OI QMFP9,X'80' Set end of parameter list
LA R1,QMFPLIST Address of parameter list
CALL DSQCIA
SPACE 1
*****
* Set numeric values into query using SET command *
*****
SPACE 1
LA R1,20 Set values for SET GLOBAL command
ST R1,VVAL1
LA R1,40
ST R1,VVAL2
LA R1,84
ST R1,VVAL3
LA R1,SETGL Addr of SET GLOBAL command length
ST R1,QMFP2
LA R1,SETG Address of SET GLOBAL command
ST R1,QMFP3
LA R1,3 Three SET GLOBAL variables
ST R1,NUMPARMS
LA R1,NUMPARMS Address of number of parameters
ST R1,QMFP4
LA R1,VNAME1L Address of variable name lengths
ST R1,QMFP5
LA R1,VNAME1 Address of variable names
ST R1,QMFP6
LA R1,VVAL1L Address of value lengths
ST R1,QMFP7
LA R1,VVAL1 Address of values
ST R1,QMFP8
LA R1,DSQ_VARIABLE_FINT Address of value data type
ST R1,QMFP9
OI QMFP9,X'80' Set end of parameter list
LA R1,QMFPLIST Address of parameter list
CALL DSQCIA
SPACE 1
*****
* Run a query *
*****
LA R1,QUERYL Addr of RUN QUERY command length
ST R1,QMFP2

```

```

        LA   R1,QUERY           Address of RUN QUERY command
        ST   R1,QMFP3
        OI   QMFP3,X'80'       Set end of parameter list
        LA   R1,QMFPLIST       Address of parameter list
        CALL DSQCIA
        SPACE 1
*****
* Print the result of the query *
*****
        LA   R1,REPTL          Addr of PRINT Report command lth
        ST   R1,QMFP2
        LA   R1,REPT           Address of PRINT Report command
        ST   R1,QMFP3
        OI   QMFP3,X'80'       Set end of parameter list
        LA   R1,QMFPLIST       Address of parameter list
        CALL DSQCIA
        SPACE 1
*****
* End the query interface session *
*****
        LA   R1,ENDQIL         Address of EXIT command length
        ST   R1,QMFP2
        LA   R1,ENDQI          Address of EXIT command
        ST   R1,QMFP3
        OI   QMFP3,X'80'       Set end of parameter list
        LA   R1,QMFPLIST       Address of parameter list
        CALL DSQCIA
        SPACE 1
*****
* Free Keyboard *
*****
        EXEC CICS SEND CONTROL FREEKB
        SPACE 1
*****
* Return *
*****
        SPACE 1
        XR   R15,R15           ZERO RETURN CODE
        DFHEIRET RCREG=15
*****
* Data Areas *
*****
        SPACE 1
* Query Interface commands
        SPACE 1
STARTQI DC   C'START'          START FUNCTION
SETG    DC   C'SET GLOBAL'    SET GLOBAL FUNCTION
QUERY   DC   C'RUN QUERY Q1'  RUN QUERY
REPT    DC   C'PRINT REPORT (FORM=F1,QUEUEN=DSQP,QUEUET=TS)'
ENDQI   DC   C'EXIT'          END INTERFACE
        SPACE 1
        DS   0F
STARTQIL DC AL4(L'STARTQI)     LENGTH OF START FUNCTION
SETGL   DC   AL4(L'SETG)      LENGTH OF SET GLOBAL FUNCTION
QUERYL  DC   AL4(L'QUERY)     LENGTH OF RUN QUERY COMMAND
REPTL   DC   AL4(L'REPT)      LENGTH OF PRINT REPORT COMMAND
ENDQIL  DC   AL4(L'ENDQI)     LENGTH OF END INTERFACE COMMAND
        SPACE 1
* START command keyword
        SPACE 1
STARTKY DC   C'DSQSMODE'
STARTV  DC   C'INTERACTIVE'
        DS   0F
STARTKYL DC AL4(L'STARTKY)
STARTVL DC AL4(L'STARTV)
        SPACE 1
* SET GLOBAL command variable names

```

```

SPACE 1
VNAME1 DC C'MYVAR01'
VNAME2 DC C'SHORT'
VNAME3 DC C'MYVAR03'
DS 0F
VNAME1L DC AL4(L'VNAME1)
VNAME2L DC AL4(L'VNAME2)
VNAME3L DC AL4(L'VNAME3)
SPACE 1
* SET GLOBAL command values
SPACE 1
VVAL1L DC AL4(L'VVAL1)
VVAL2L DC AL4(L'VVAL2)
VVAL3L DC AL4(L'VVAL3)
* Callable interface communications definition
DSQCOMMA
* Equates for registers 0-15
R0 EQU 00
R1 EQU 01
R2 EQU 02
R3 EQU 03
R4 EQU 04
R5 EQU 05
R6 EQU 06
R7 EQU 07
R8 EQU 08
R9 EQU 09
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
* Local variables located in CICS working storage
DFHEISTG DSECT
ORG DFHEIUSR
NUMPARMS DS F NUMBER OF KEYWORDS
* QMF SET GLOBAL command values
VVAL1 DS F
VVAL2 DS F
VVAL3 DS F
* QMF Callable interface parameter list
QMFP1 DS 0D
QMFP1 DS F
QMFP2 DS F
QMFP3 DS F
QMFP4 DS F
QMFP5 DS F
QMFP6 DS F
QMFP7 DS F
QMFP8 DS F
QMFP9 DS F
* Callable interface communications area
CICOMM DS CL(DSQCOMM_LEN)
CSECT
SPACE 1
END DSQABFAC

```

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

TSO 用のサンプルのアセンブラー・プログラム

IBM は、DSQABFA という名前の TSO 用のサンプルのアセンブラー・プログラムを提供しています。

この TSO 用のサンプルのアセンブラー・プログラムは、QMF1210.SDSQSAP n ライブラリーにあります (n は各国語 ID です)。

アセンブラー呼び出し可能インターフェースのこのサンプル・プログラムは、以下の機能を実行します。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

```
DSQABFA TITLE 'SAMPLE QMF CALLABLE INTERFACE'
DSQABFA CSECT
DSQABFA AMODE 31
DSQABFA RMODE ANY
*****
*                                                                 *
* Sample Program: DSQABFA                                       *
* Assembler Version of the QMF Callable Interface               *
*                                                                 *
*****
        SPACE 1
        STM  R14,R12,12(R13)          SAVE ENTRY REGISTERS
        BALR R12,0                    INITIALIZE BASE REGISTER
        USING *,R12
        LA   R2,SAVEAREA              CHAIN SAVE AREAS
        ST   R2,8(R13)
        ST   R13,SAVEAREA+4
        LR   R13,R2                   ESTABLISH SAVE AREA
        SPACE 1
*****
* Start a query interface session                                *
*****
        LA   R4,CICOMM                ESTABLISH ACCESS TO DSQCOMM
        USING DSQCOMM,R4
        SPACE 1
        MVC  DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL
        LA   R1,1                      1 PARAMETER
        ST   R1,NUMPARMS
        CALL DSQCIA,
             (CICOMM,                  QI COMMON AREA
              STARTQIL,                START COMMAND LENGTH
              STARTQI,                 START COMMAND
              NUMPARMS,                NUMBER OF KEYWORDS
              STARTKYL,                KEYWORD LENGTHS
              STARTKY,                 KEYWORDS
              STARTVL,                 VALUE LENGTHS
              STARTV,                  VALUES
              DSQ_VARIABLE_CHAR),VL    VALUES ARE CHARACTERS
        SPACE 1
*****
* Set numeric values into query using SET command                *
*****
```

```

SPACE 1
LA R1,20 SET VALUES TO BE MODIFIED
ST R1,VVAL1
LA R1,40
ST R1,VVAL2
LA R1,84
ST R1,VVAL3
LA R1,3 3 PARAMETERS
ST R1,NUMPARMS
SPACE 1
CALL DSQCIA,
(CICOMM,
SETGL, SET GLOBAL COMMAND LENGTH
SETG, SET GLOBAL COMMAND
NUMPARMS, NUM OF VARIABLES TO BE SET
VNAME1L, VARIABLE NAME LENGTHS
VNAME1, VARIABLE NAMES
VVAL1L, VALUE LENGTHS
VVAL1, VALUES
DSQ_VARIABLE_FINT),VL VALUES ARE INTEGERS
SPACE 1
*****
* Run a query *
*****
SPACE 1
CALL DSQCIA,
(CICOMM,
QUERYL, QUERY COMMAND LENGTH
QUERY),VL TEXT OF QUERY COMMAND
SPACE 1
*****
* Print the result of the query *
*****
SPACE 1
CALL DSQCIA,(CICOMM,REPTL,REPT),VL
SPACE 1
*****
* End the query interface session *
*****
SPACE 1
CALL DSQCIA,(CICOMM,ENDQIL,ENDQI),VL
SPACE 1
*****
* Return *
*****
SPACE 1
SR R15,R15 SET RETURN CODE
L R13,4(R13)
L R14,12(R13) RESTORE CALLER REGISTERS
LM R0,R12,20(R13)
BR R14
EJECT
*****
* Data Areas *
*****
SPACE 1
* Query Interface commands
SPACE 1
STARTQI DC C'START' START FUNCTION
SETG DC C'SET GLOBAL' SET GLOBAL FUNCTION
QUERY DC C'RUN QUERY Q1' RUN QUERY
REPT DC C'PRINT REPORT (FORM=F1)' PRINT REPORT
ENDQI DC C'EXIT' END INTERFACE
SPACE 1
DS 0F
STARTQIL DC AL4(L'STARTQI) LENGTH OF START FUNCTION
SETGL DC AL4(L'SETG) LENGTH OF SET GLOBAL FUNCTION

```

```

QUERYL  DC   AL4(L'QUERY)           LENGTH OF RUN QUERY COMMAND
REPTL   DC   AL4(L'REPT)           LENGTH OF PRINT REPORT COMMAND
ENDQIL  DC   AL4(L'ENDQI)         LENGTH OF END INTERFACE COMMAND
SPACE 1
* START command keyword
SPACE 1
STARTKY DC   C'DSQSMODE'
STARTV  DC   C'INTERACTIVE'
        DS   0F
STARTKYL DC  AL4(L'STARTKY)
STARTVL DC  AL4(L'STARTV)
SPACE 1
* SET GLOBAL command variable names
SPACE 1
VNAME1  DC   C'MYVAR01'
VNAME2  DC   C'SHORT'
VNAME3  DC   C'MYVAR03'
        DS   0F
VNAME1L DC  AL4(L'VNAME1)
VNAME2L DC  AL4(L'VNAME2)
VNAME3L DC  AL4(L'VNAME3)
SPACE 1
* SET GLOBAL command values
SPACE 1
VVAL1   DS   F
VVAL2   DS   F
VVAL3   DS   F
VVAL1L  DC  AL4(L'VVAL1)
VVAL2L  DC  AL4(L'VVAL2)
VVAL3L  DC  AL4(L'VVAL3)
SPACE 1
NUMPARMS DS   F                     NUMBER OF KEYWORDS
SPACE 1
* Callable interface communications area
SPACE 1
CICOMM  DS   CL(DSQCOMM_LEN)
SPACE 1
SAVEAREA DS   18F
        EJECT
        DSQCOMMA
SPACE 1
R0      EQU  00                     EQUATES FOR REGISTERS 0-15
R1      EQU  01
R2      EQU  02
R3      EQU  03
R4      EQU  04
R5      EQU  05
R6      EQU  06
R7      EQU  07
R8      EQU  08
R9      EQU  09
R10     EQU  10
R11     EQU  11
R12     EQU  12
R13     EQU  13
R14     EQU  14
R15     EQU  15
SPACE 1
END     DSQABFA

```

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

アセンブラー用の DSQCOMM

アセンブラーのインターフェース連絡域ファイルは、DSQCOMMA という名前のファイルです。

DSQCOMMA ファイルは QMF1210.SDSQSAP n ライブラリー内 (n は各国語 ID) にあります。次にファイルのコピーを示します。

```
MACRO
DSQCOMMA
*****
* Callable interface - variable constants *
*****
*
* Communications Level ID
*
DSQ_CURRENT_COMM_LEVEL DC CL12'DSQL>001002<'
*
* Query Product IDs
*
DSQ_QRW DC C'01'
DSQ_QMF DC C'02'
DSQ_QM4 DC C'03'
*
* Query Product Release IDs
*
DSQ_QRW_V1R2 DC C'01'
DSQ_QRW_V1R3 DC C'02'
DSQ_QMF_V2R4 DC C'01'
DSQ_QMF_V3R1 DC C'02'
DSQ_QMF_V3R1M1 DC C'03'
DSQ_QMF_V3R2 DC C'04'
DSQ_QMF_V3R3 DC C'05'
DSQ_QMF_V6R1 DC C'06'
DSQ_QMF_V7R1 DC C'07'
DSQ_QM4_V1R1 DC C'01'
DSQ_QMF_V7R2 DC C'08'
DSQ_QMF_V8R1 DC C'09'
DSQ_QMF_V9R1 DC C'10'
DSQ_QMF_V10R1 DC C'11'
*
* Extended parameter data types
*
DSQ_VARIABLE_CHAR DC C'CHAR'
DSQ_VARIABLE_FINT DC C'FINT'
*
* Return codes
*
DSQ_SUCCESS EQU 0
DSQ_WARNING EQU 4
DSQ_FAILURE EQU 8
DSQ_SEVERE EQU 16
*
* Instance ID values
*
DSQ_CONTINUE EQU 0
*
* Cancel indicator
*
DSQ_CANCEL_YES EQU C'1'
DSQ_CANCEL_NO EQU C'0'
*
*
DSQ_INTERACTIVE EQU C'1'
DSQ_BATCH EQU C'2'
*
```

```

DSQ_YES          EQU  C'1'
DSQ_NO           EQU  C'2'
*
*****
* Callable interface communications area *
*****
DSQCOMM          DSECT
DSQ_RETURN_CODE DS   F          FUNCTION RETURN CODE
DSQ_INSTANCE_ID DS   F          ID ESTABLISHED IN START CMD
DSQ_COMM_LEVEL  DS   CL12       COMMUNICATIONS LEVEL ID
DSQ_PRODUCT     DS   CL2        QUERY PRODUCT ID
DSQ_PRODUCT_RELEASE DS CL2      QUERY PRODUCT RELEASE ID
DSQ_RESERVE1    DS   CL28       RESERVED
DSQ_MESSAGE_ID  DS   CL8        COMPLETION MESSAGE ID
DSQ_Q_MESSAGE_ID DS CL8        QUERY MESSAGE ID
DSQ_START_PARM_ERROR DS CL8     START PARAMETER IN ERROR
DSQ_CANCEL_IND  DS   C          CMD CANCEL INDICATOR
DSQ_RESERVE2    DS   CL23       RESERVED
DSQ_RESERVE3    DS   CL156      RESERVED
DSQ_MESSAGE_TEXT DS CL128      COMPLETION MESSAGE
DSQ_Q_MESSAGE_TEXT DS CL128     QUERY MESSAGE
          SPACE 1
DSQCOMM_LEN     EQU  *-DSQCOMM  LENGTH OF DSQCOMM AREA
          MEND

```

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

CICS での アセンブラー プログラムの実行

プログラムは、作成した後、実行する前に、変換し、アセンブルし、リンク・エディットする必要があります。

これらの例の REXX JCL および CLIST は QMF に用意されていませんが、ここからこれらをコピーして、お客様の要件に合うように変更できます。

QMF 呼び出し可能インターフェースを使用するプログラムを変換、アセンブル、およびリンク・エディットするときには、以下の条件に注意してください。

- インターフェース連絡域 DSQCOMM がアセンブル・ステップで使用可能であるか、またはそれを DSECT としてプログラムにコピーする必要がある。
- プログラムのリンク・エディットのステップで QMF インターフェース・モジュール DSQCIA が使用可能でなければならない。

次に示す JCL は、CICS から提供されるプロシージャ DFHEBTAL の使用方法の例です。

```

//sampasm JOB
//          EXEC PROC=DFHEBTAL
//TRN.SYSIN DD *
*ASM      XOPTS(CICS translator options .....
          .
          .
          Your program or copy of QMF sample DSQABFA
          .
          .
          .
/*
//* Provide access to QMF communications macro DSQCOMM

```



```

//ASM.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Provide access to QMF interface module
/* Allocation for your CICS library
//LKED.CICSLOAD DD
/* Allocation for your target library
//LKED.SYSLMOD DD
/* Allocation for the QMF load module library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
        INCLUDE CICSLOAD(DFHEAI)
        INCLUDE CICSLOAD(DFHEAI0)
        INCLUDE QMFLOAD(DSQCIA)
        ORDER DFHEAI,DFHEAI0
        ENTRY sampasm
        MODE AMODE(31) RMODE(31)
        NAME sampasm(R)
/*

```

TSO での アセンブラー プログラムの実行

プログラムは、TSO で実行する前に、アセンブルし、リンク・エディットする必要があります。

TSO でのアセンブルおよびリンク・エディット

次のリストは、プログラムをアセンブルしてリンク・エディットするサンプル・ジョブを示しています。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

```

//sampasm JOB
//STEP1 EXEC PROC=ASMACL
/* Provide access to QMF communications macro DSQCOMM
//C.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//C.SYSIN DD *
        .
        .
        Your program or copy of QMF sample DSQABFA
        .
        .
/*
/* Provide access to QMF interface module
/* Allocation for your target library
//L.SYSLMOD DD
/* Allocation for the QMF load library
//L.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//L.SYSIN DD *
        INCLUDE QMFLOAD(DSQCIA)
        ENTRY sampasm
        MODE AMODE(31) RMODE(31)
        NAME sampasm(R)
/*

```

ISPF を使用する TSO での実行

プログラムのアセンブルが正常に終了したら、ISPF のもとでそのプログラムを実行することができます。

ISPF のもとの TSO でプログラムを実行するには、次に示す CLIST のようなプログラムを作成します。

```

PROC 0
CONTROL ASIS
/*****
/* Specify attribute list for dataset allocations */
/*****
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****
/* Datasets used by TSO */
/*****
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****
/* Datasets used by ISPF */
/*****
ALLOC FI(ISPLLIB) SHR REUSE +

        DA('QMF1210.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
        DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
        DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
        DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
        DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* QMF/GDDM Datasets */
/*****
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****
/* Datasets used by QMF */
/*****
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****
/* Start your program as the initial ISPF dialog */
/*****
ISPSTART PGM(sampasm) NEWAPPL(DSQE)
EXIT CODE(4)

```

EXIT CODE(4) は ISPF 後処理パネルを抑止します。

ISPF を使用しない TSO での実行

プログラムのアセンブルが正常に終了したら、ISPF なしでそのプログラムを実行することができます。

ISPF を使用しない TSO でプログラムを実行するには、次に示す CLIST のようなプログラムを作成します。

```

PROC 0
CONTROL ASIS
/*****
/* Note: QMF, DB2 and GDDM load libraries must be allocated */

```

```

/*      before executing this CLIST.          */
/*      Name of QMF load library is "QMF1210.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGMGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL sampasm
EXIT CODE(0)

```

C 言語インターフェース

QMF では、C 言語を呼び出し可能インターフェースで使用することができます。

C 言語用のインターフェース連絡域マッピング (DSQCOMMC)

DSQCOMMC は、QMF に用意されている C 言語プログラム用の DSQCOMM マッピングです。

次の表に、DSQCOMMC の各値を示します。

表 45. DSQCOMM のインターフェース連絡域

構造名	データ・タイプ	説明
DSQ_RETURN_CODE	符号付き長整数	QMF コマンドの実行後の状況を示す 値は次のとおり。 DSQ_SUCCESS 要求が正常に実行された DSQ_WARNING 警告を伴って正常に終了 DSQ_FAILURE コマンドが正しく実行されなかった DSQ_SEVERE 重大エラー、QMF セッションは終了
DSQ_INSTANCE_ID	符号付き長整数	START コマンドの実行時に、QMF によって設定される ID
DSQ_COMM_LEVEL	文字、長さ 12	DSQCOMM のレベルを識別する ご使用のアプリケーションに、この変数を DSQ_CURRENT_COMM_LEVEL の値に初期設定してか ら QMF START コマンドを出す命令を含めてください。
DSQ_PRODUCT	文字、長さ 2	使用中の IBM 照会プロダクトを識別する 先頭に DSQ_QMF が付く変数は、QMF for TSO バージ ョンおよび QMF for CICS バージョンを指定します。
DSQ_PRODUCT_RELEASE	文字、長さ 2	使用中の QMF のリリース・レベル 変数 DSQ_QMF_V11R1 は QMF バージョン 12 リリ ース 1 を指定します。
DSQ_RESERVE1	文字、長さ 28	将来の使用のために予約されている。
DSQ_MESSAGE_ID	文字、長さ 8	完了メッセージ ID
DSQ_Q_MESSAGE_ID	文字、長さ 8	照会メッセージ ID
DSQ_START_PARM_ERROR	文字、長さ 8	パラメーター・エラーのために START が失敗したとき の、エラーがあるパラメーター
DSQ_CANCEL_IND	文字、長さ 1	QMF コマンドの実行中にユーザーが取り消したかどうか によって、以下の 2 つの値のいずれかが入る。 • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	文字、長さ 23	将来の使用のために予約されている。
DSQ_RESERVE3	文字、長さ 156	将来の使用のために予約されている。
DSQ_MESSAGE_TEXT	文字、長さ 128	完了メッセージ・テキスト
DSQ_Q_MESSAGE_TEXT	文字、長さ 128	照会メッセージ・テキスト

C 言語用の関数呼び出し

QMF には、C 言語用に DSQCIC と DSQCICE という 2 つの関数呼び出しが用意されています。

DSQCIC

この呼び出しは、アプリケーション・プログラム変数にアクセスする必要がない QMF コマンド用です。大部分の QMF コマンドに、この呼び出しを使用します。構文は、次のとおりです。

```
DSQCIC (&DSQCOMM,&CMDLTH,&CMDSTR)
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。長タイプのパラメーター

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの、無符号文字タイプの配列として指定します。

QMF コマンドは大文字でなければなりません。

DSQICE

この呼び出しには、アプリケーション・プログラム変数へのアクセスが必要な QMF コマンド (START、TRACE、および拡張形式の GET GLOBAL と SET GLOBAL) 用の拡張構文があります。

```
DSQICE (&DSQCOMM,&CMDLTH,&CMDSTR,  
        &PNUM,&KLTH,&KEYWORD,  
        &VLTH,&VALUE,&VTYPE);
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。長整数パラメーター

CMDSTR

実行する QMF コマンド (無符号文字タイプの配列)。QMF コマンドは大文字でなければなりません。

PNUM

コマンド・キーワードまたはトレース域の数 (長整数パラメーター)。

KLTH

指定する各キーワード (KEYWORD) または各トレース・タイトルの長さ (長整数パラメーターまたは長整数パラメーターの配列)。

KEYWORD

QMF キーワード、キーワード、またはトレース・タイトルのアドレス (文字、文字の配列、またはトレース・タイトルのアドレスの配列)。

VLTH

キーワードまたはトレース・タイトルに関連付けられている各値の長さ (長整数パラメーターまたは長整数パラメーターの配列)。

VALUE

各キーワードに関連付けられている値、またはトレース・タイトルに関連付けられている値のアドレス。

この値のタイプは、無符号文字配列、長整数パラメーター、または長整数パラメーターの配列として VTYPE パラメーターに指定します。トレース・データの場合は、VTYPE が FINT である必要があります。

VTYPE

VALUE パラメーターの内容のデータ・タイプ。

このパラメーターには、インターフェース連絡域 DSQCOMM で提供される以下の 2 つの値のいずれかを設定します。

- 無符号文字タイプを示す DSQ_VARIABLE_CHAR
- 長整数を示す DSQ_VARIABLE_FINT

VALUE フィールドに指定したすべての値のデータ・タイプを VTYPE に指定する必要があります。

C 言語インターフェースでは、以下のパラメーターに関する考察事項があります。

- コマンド・ストリングと、START、GET、および SET のコマンド・パラメーターは、すべて入力文字ストリングです。これらのストリングを使用して、C では、終わりが NULL 値である記憶域を渡す必要があります。パラメーターの長さには NULL 値を含めます。QMF インターフェースに渡すパラメーター長を取得するには、コンパイル時長さ関数を使用します。
- ストリングが、ストリングの終わりに到達する前に NULL 値によって終了しない場合、QMF によってエラーが戻されます。NULL 値 (X'00') は、文字ストリングの終わりを示します。
- 出力文字ストリングである C パラメーター (GET コマンドで取得した値を含む) の場合は、QMF によって、QMF 記憶域からアプリケーションの記憶域にデータが移動されます。QMF によって、ストリングの終わりに NULL 標識も設定されます。文字ストリングがユーザーの記憶域に収まらない場合には、警告メッセージが出され、データの右側が切り捨てられます。NULL 標識は、常にデータ・ストリングの終わりに置かれます。

C 言語プログラミングの例

QMF には、サンプルの C プログラムである DSQABFC が用意されています。このサンプル・プログラムは、ライブラリー QMF1210.SDSQSAP n (n は各国語 ID) のメンバーです。

IBM C 言語呼び出し可能インターフェースのサンプル・プログラムは、以下の機能を実行します。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

```

/*****
/* Sample Program: DSQABFC
*/
/* C version of the callable interface
*/
*****/

/*****
/* Include standard and string "C" functions
*/
*****/
#include <string.h>
#include <stdlib.h>

/*****
/* Include and declare query interface communications area
*/
*****/
#include <DSQCOMM.C>

int main()
{

    struct dsqcomm communication_area;    /* DSQCOMM from include
    */

    /*****
    /* Query interface command length and commands
    */
    *****/
    signed long command_length;
    static char start_query_interface[] = "START";
    static char set_global_variables[] = "SET GLOBAL";
    static char run_query[] = "RUN QUERY Q1";
    static char print_report[] = "PRINT REPORT (FORM=F1";
    static char end_query_interface[] = "EXIT";

    /*****
    /* Query command extension, number of parameters and lengths
    */
    *****/
    signed long number_of_parameters;    /* number of variables
    */
    signed long keyword_lengths[10];    /* lengths of keyword names
    */
    signed long data_lengths[10];        /* lengths of variable data
    */

    /*****
    /* Variable data type constants
    */
    *****/
    static char char_data_type[] = DSQ_VARIABLE_CHAR;
    static char int_data_type[] = DSQ_VARIABLE_FINT;

    /*****
    /* Keyword parameter and value for START command
    */
    *****/
    static char start_keywords[] = "DSQSMODE";
    static char start_keyword_values[] = "INTERACTIVE";
    /*****
    /* Keyword parameter and values for SET command
    */

```

```

/*****
#define SIZE_VAL 8
char set_keywords [3][SIZE_VAL]; /* Parameter name array
*/
signed long set_values[3]; /* Parameter value array
*/

/*****
/* MAIN PROGRAM
*/
/*****

/*****
/* Start a query interface session
*/
/*****
    strncpy (communication_area.dsq_comm_level,
            DSQ_CURRENT_COMM_LEVEL,
            sizeof(communication_area.dsq_comm_level));
    number_of_parameters = 1;
    command_length = sizeof(start_query_interface);
    keyword_lengths[0] = sizeof(start_keywords);
    data_lengths[0] = sizeof(start_keyword_values);
    dsqcice(&communication_area,;

        &command_length,;

        &start_query_interface[0],

        &number_of_parameters,;

        &keyword_lengths[0],
        &start_keywords[0],
        &data_lengths[0],
        &start_keyword_values[0],
        &char_data_type[0]);

/*****
/* Set numeric values into query using SET command
*/
/*****
    number_of_parameters = 3;
    command_length = sizeof(set_global_variables);
    strcpy(set_keywords[0], "MYVAR01");
    strcpy(set_keywords[1], "SHORT");
    strcpy(set_keywords[2], "MYVAR03");
    keyword_lengths[0] = SIZE_VAL;
    keyword_lengths[1] = SIZE_VAL;
    keyword_lengths[2] = SIZE_VAL;
    data_lengths[0] = sizeof(long);
    data_lengths[1] = sizeof(long);
    data_lengths[2] = sizeof(long);
    set_values[0] = 20;
    set_values[1] = 40;
    set_values[2] = 84;
    dsqcice(&communication_area,;

        &command_length,;

        &set_global_variables[0],
        &number_of_parameters,;

        &keyword_lengths[0],
        &set_keywords[0][0],
        &data_lengths[0],
        &set_values[0],
        &int_data_type[0]);

```



```

/*****
/* Run a query
*/
/*****
    command_length = sizeof(run_query);
    dsqpic(&communication_area,&command_length,;

        &run_query[0]);

/*****
/* Print the results of the query
*/
/*****
    command_length = sizeof(print_report);
    dsqpic(&communication_area,&command_length,;

        &print_report[0]);

/*****
/* End the query interface session
*/
/*****
    command_length = sizeof(end_query_interface);
    dsqpic(&communication_area,&command_length,;

        &end_query_interface[0]);

    exit(0);
}

```

C 用の DSQCOMM

C 言語のインターフェース連絡域ファイルは、DSQCOMM という名前のファイルです。

QMF で提供される DSQCOMM インクルード・ファイルを次に示します。

```

/*****
/* C include for query callable interface
*/
/*****

/* Structure declare for communications area
*/

struct dsqcomm {
    long int dsq_return_code; /* Function return code */
    long int dsq_instance_id; /* ID established in START cmd*/
    char dsq_comm_level[12]; /* Communications level id */
    char dsq_product[2]; /* Query product id */
    char dsq_product_release[2]; /* Query product release */
    char dsq_reserve1[28]; /* Reserved */
    char dsq_message_id[8]; /* Completion message ID */
    char dsq_q_message_id[8]; /* Query message ID */
    char dsq_start_parm_error[8]; /* Start parameter in error */
    char dsq_cancel_ind[1]; /* Cmd cancelled indicator */
    /* 1 = cancelled, 0 = not cancelled*/
    char dsq_reserve2[23]; /* RESERVED AREAS */
    char dsq_reserve3[156];
    char dsq_message_text[128]; /* Message text */
    char dsq_q_message_text[128]; /* Query message text */
};

/* RETURN CODES
*/

#define DSQ_SUCCESS 0
#define DSQ_WARNING 4
#define DSQ_FAILURE 8

```

```

#define DSQ_SEVERE                16

/* Communications Level */

#define DSQ_CURRENT_COMM_LEVEL    "DSQL>001002<"

/* Query Product Codes */

#define DSQ_QRW                    "01"
#define DSQ_QMF                    "02"
#define DSQ_QM3                    "03"

/* Query Product Release Levels */

#define DSQ_QRW_V1R2                "01"
#define DSQ_QRW_V1R3                "02"
#define DSQ_QMF_V2R4                "01"
#define DSQ_QMF_V3R1                "02"
#define DSQ_QMF_V3R1M1              "03"
#define DSQ_QMF_V3R2                "04"
#define DSQ_QMF_V3R3                "05"
#define DSQ_QMF_V6R1                "06"
#define DSQ_QMF_V7R1                "07"
#define DSQ_QM4_V1R1                "01"
#define DSQ_QMF_V7R2                "08"
#define DSQ_QMF_V8R1                "09"
#define DSQ_QMF_V9R1                "10"
#define DSQ_QMF_V10R1               "11"

/* INSTANCE CODES */

#define DSQ_CONTINUE                0

/* CANCELLED INDICATOR */

#define DSQ_CANCEL_YES              "1"
#define DSQ_CANCEL_NO               "0"

/* VARIABLE TYPES */

#define DSQ_VARIABLE_CHAR           "CHAR"
#define DSQ_VARIABLE_FINT           "FINT"

#define DSQ_INTERACTIVE             "1"
#define DSQ_BATCH                   "2"

#define DSQ_YES                     "1"
#define DSQ_NO                      "2"

/* Call interface structure */

/* Calling format for normal call with 3 parameters */
#define dsqcic(parm1, parm2, parm3 )¥
        dsqcic( parm1, parm2, parm3)

/* Calling format for call with CMD_EXT area 9 parameters */
#define dsqcice(parm1, parm2, parm3,¥
        parm4, parm5, parm6, parm7, parm8, parm9 )¥
        dsqcic( parm1, parm2, parm3, ¥
        parm4, parm5, parm6, ¥
        parm7, parm8, parm9 )

/* DECLARE OS LINKAGE FORMAT */

#pragma linkage(dsqcicx, OS)

```

CICS での C プログラムの実行

C でプログラムを作成したら、変換、コンパイル、およびリンク・エディットしてから、実行する必要があります。

プログラムを変換、コンパイル、およびリンク・エディットするために必要なステップの例を以下に示します。これらの例の REXX JCL および CLIST は QMF に用意されていませんが、ここからこれらをコピーして、お客様の要件に合うように変更できます。

CICS 下で QMF 呼び出し可能インターフェースを使用するプログラムを変換、コンパイル、およびリンク・エディットするときには、以下の条件について考慮してください。

- インターフェース連絡域 DSQCOMM がコンパイル・ステップで使用可能であるか、またはそれをプログラムにコピーする必要がある。
- プログラムのリンク・エディットのステップで QMF インターフェース・モジュール DSQCICX が使用可能でなければならない。
- C で作成されたプログラムは、AMODE=31 を指定してリンク・エディットする必要があります。

次の例では、CICS から提供されるプロシージャ DFHYITDL を使用します。

```
//sampleC JOB
// EXEC PROC=DFHYITDL
//TRN.SYSIN DD *
.
.
.
Your program or copy of QMF sample DSQABFC
.
.
.
/*
/* Provide Access to QMF Communications Macro DSQCOMM
//C.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Allocation for target library
//LKED.SYSLMOD DD
/* Allocation for QMF load library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCICX)
NAME sampleC(R)
/*
```

TSO での C プログラムの実行

TSO で C プログラムを実行するには、プログラムをコンパイルし、リンク・エディットしてから、プログラムを実行します。その際、ISPF を使用するかどうかは任意です。

TSO のコンパイルおよびリンク・エディット

C プログラムは、TSO で実行する前に、コンパイルし、リンク・エディットする必要があります。

このジョブは、z/OS 用の IBM C コンパイラーを使用して、呼び出し可能インターフェース・アプリケーションをコンパイルしリンク・エディットします。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

```
//sampleC JOB
//STEP1 EXEC PROC=EDCCB,
// INFILE='name of dataset that contains source code',
// OUTFILE='name of dataset that contains executable'
/* Provide Access to QMF Communications Macro DSQCOMM
//COMPILE.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//BIND.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//BIND.SYSIN DD *
INCLUDE QMFLOAD(DSQCICX)
/*
```

ISPF を使用しない TSO でのプログラムの実行

C プログラムのコンパイルが正常に終了したら、ISPF なしでそのプログラムを実行することができます。

ISPF を使用しない TSO でプログラムを実行するには、次に示す CLIST のようなプログラムを作成します。

```
PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2, GDDM and C load libraries must be */
/* allocated before running this CLIST. */
/* Name of QMF load library is "QMF1210.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL sampleC
EXIT CODE(0)
```

ISPF のもとの TSO でのプログラムの実行

C プログラムのコンパイルが正常に終了したら、ISPF のもとでそのプログラムを実行することができます。

ISPF のもとの TSO でプログラムを実行するには、DSQABFC を実行するために、次に示す CLIST のようなプログラムを作成します。

```
PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD', +
        'EDC.SEDCLINK','PLI.SIBMLINK')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(sampleC) NEWAPPL(DSQE)
EXIT CODE(4)
```

EXIT CODE(4) は ISPF 後処理パネルを抑止します。

COBOL 言語インターフェース

QMF では、COBOL 言語を呼び出し可能インターフェースで使用することができます。

COBOL 用のインターフェース連絡域マッピング (DSQCOMMB)

DSQCOMMB は、QMF に用意されている COBOL 言語プログラム用の DSQCOMM マッピングです。

次の表に、DSQCOMMB の各値を示します。

表 46. COBOL (DSQCOMMB) のインターフェース連絡域

構造名	データ・タイプ	説明
DSQ-RETURN-CODE	PIC 9(8)	QMF コマンドの実行後の状況を示す 値は次のとおり。 DSQ-SUCCESS 要求が正常に実行された DSQ-WARNING 警告を伴って正常に終了 DSQ-FAILURE コマンドが正しく実行されなかった DSQ-SEVERE 重大エラー、QMF セッションは終了
DSQ-INSTANCE-ID	PIC 9(8)	START コマンドの実行時に、QMF によって設定される ID
DSQ-COMM-LEVEL	PIC X(12)	DSQCOMM のレベルを識別する ご使用のアプリケーションに、この変数を DSQ_CURRENT_COMM_LEVEL の値に初期設定してか ら QMF START コマンドを出す命令を含めてくださ い。
DSQ-PRODUCT	PIC X(2)	使用中の IBM 照会プロダクトを識別する DSQ-QMF が先頭に付く変数は、QMF for TSO and QMF for CICS のバージョンを指定します。
DSQ-PRODUCT-RELEASE	PIC X(2)	使用中の QMF のリリース・レベル 変数 DSQ_QMF_V11R1 は QMF バージョン 12 リリ ース 1 を指定します。
DSQ-RESERVE1	PIC X(28)	将来の使用のために予約されている。
DSQ-MESSAGE-ID	PIC X(8)	完了メッセージ ID
DSQ-Q-MESSAGE-ID	PIC X(8)	照会メッセージ ID
DSQ-START-PARM-ERROR	PIC X(8)	パラメーター・エラーのために START が失敗したとき の、エラーがあるパラメーター

表 46. COBOL (DSQCOMMB) のインターフェース連絡域 (続き)

構造名	データ・タイプ	説明
DSQ-CANCEL-IND	PIC X(1)	QMF コマンドの実行中にユーザーが取り消したかどうかによって、以下の 2 つの値のいずれかが入る。 <ul style="list-style-type: none"> • DSQ-CANCEL-YES • DSQ-CANCEL-NO
DSQ-RESERVE2	PIC X(23)	将来の使用のために予約されている。
DSQ-RESERVE3	PIC X(156)	将来の使用のために予約されている。
DSQ-MESSAGE-TEXT	PIC X(128)	完了メッセージ・テキスト
DSQ-Q-MESSAGE-TEXT	PIC X(128)	照会メッセージ・テキスト

COBOL 用の関数呼び出し

QMF には、COBOL 言語用に DSQCIB という 1 つの関数呼び出しが用意されています。この関数呼び出しには、DSQCIB と DSQCIB (拡張形式) という 2 つの形式があります。

DSQCIB

この呼び出しは、アプリケーション・プログラム変数にアクセスする必要がない QMF コマンド用です。大部分の QMF コマンドに、この呼び出しを使用します。

```
CALL DSQCIB USING DSQCOMM CMDLTH CMDSTR
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。整数パラメーター

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの大文字ストリング

DSQCIB (拡張形式)

この呼び出しには、アプリケーション・プログラム変数へのアクセスが必要な QMF コマンド (START、TRACE、および拡張形式の GET GLOBAL と SET GLOBAL) 用の拡張構文があります。

```
DSQCIB USING
  DSQCOMM CMDLTH CMDSTR
  PNUM KLTH KWORD VLTH VALUE VTYPE
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。整数パラメーター。

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの大文字ストリング。

PNUM

コマンド・キーワードまたはトレース域の数 (整数パラメーター)。

KLTH

指定する各キーワードまたは各トレース・タイトルの長さ (整数パラメーターまたは整数パラメーターの配列)。

KWORD

QMF キーワード、キーワード、またはトレース・タイトルのアドレス。

それぞれ、文字、文字の配列、または KLTH で指定された長さのトレース・タイトルのアドレスの配列として指定します。すべてのキーワードの長さが同じであれば、文字の配列を使用することができます。

VLTH

キーワードまたはトレース・タイトルに関連付けられている各値の長さ (整数パラメーターまたは整数パラメーターの配列)。

VALUE

各キーワードに関連付けられている値、またはトレース・タイトルに関連付けられている値のアドレス。

この値のタイプは、VTYPE パラメーターに指定します。この値は、文字、文字配列、整数パラメーター、または整数パラメーターの配列です。トレース・データの場合は、VTYPE が FINT である必要があります。

VTYPE

VALUE パラメーターの内容のデータ・タイプ。

このパラメーターには、連絡域 DSQCOMMB で提供される以下の 2 つの値のいずれかを設定します。

- 文字値を示す DSQ-VARIABLE-CHAR
- 整数値を示す DSQ-VARIABLE-FINT

VALUE フィールドに指定したすべての値のデータ・タイプを VTYPE に指定する必要があります。

COBOL での ISPF LIBDEF サービス

ご使用の QMF アプリケーションで LIBDEF 関数を使用するには、QMF インターフェース (DSQCIB) の動的呼び出しを静的呼び出しに変更します。

例えば、次の呼び出し ID ステートメントについて考えてみましょう。

```
CALL DSQCIB USING ...
```

このステートメントを次のような呼び出しリテラル書式に変更できます。

```
CALL "DSQCIB" USING ...
```


COBOL・プログラミングの例

QMF には、サンプルの COBOL プログラムである DSQABFCO が用意されています。このサンプル・プログラムは、ライブラリー QMF1210.SDSQSAPn (n は各国語 ID) のメンバーです。

COBOL 言語呼び出し可能インターフェースのサンプル・プログラムは、以下の機能を実行します。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

CICS の場合、STOP RUN ステートメントを GOBACK ステートメントに変更する必要があります。

```
*****
*   The following is a COBOL version of the query           *
*   callable interface *** DSQABFCO **.                    *
*****
IDENTIFICATION DIVISION.
PROGRAM-ID. DSQABFCO.
DATE-COMPILED.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*****
* Copy DSQCOMMB definition - contains query interface variables *
*****
COPY DSQCOMMB.

* Query interface commands
01 STARTQI PIC X(5) VALUE "START".
01 SETG PIC X(10) VALUE "SET GLOBAL".
01 QUERY PIC X(12) VALUE "RUN QUERY Q1".
01 REPT PIC X(22) VALUE "PRINT REPORT (FORM=F1 ".
01 ENDQI PIC X(4) VALUE "EXIT".

* Query command length
01 QICLTH PIC 9(8) USAGE IS COMP-4.
* Number of variables
01 QIPNUM PIC 9(8) USAGE IS COMP-4.
* Keyword variable lengths
01 QIKLTHS.
03 KLTHS PIC 9(8) OCCURS 10 USAGE IS COMP-4.
* Value Lengths
01 QIVLTHS.
03 VLTHS PIC 9(8) OCCURS 10 USAGE IS COMP-4.
* Start command keyword
01 SNAMES.
03 SNAME1 PIC X(8) VALUE "DSQSMODE".
* Start command keyword value
01 SVALUES.
03 SVALUE1 PIC X(11) VALUE "INTERACTIVE".
* Set GLOBAL command variable names to set
01 VNAMES.
03 VNAME1 PIC X(7) VALUE "MYVAR01".
```

```

03 VNAME2 PIC X(5) VALUE "SHORT".
03 VNAME3 PIC X(7) VALUE "MYVAR03".
* Variable value parameters
01 VVALUES.
03 VVALS PIC 9(8) OCCURS 10 USAGE IS COMP-4.

01 TEMP PIC 9(8) USAGE IS COMP-4.
PROCEDURE DIVISION.
*
* Start a query interface session
MOVE DSQ-CURRENT-COMM-LEVEL TO DSQ-COMM-LEVEL.
MOVE 5 TO QICLTH.
MOVE 8 TO KLTHS(1).
MOVE 11 TO VLTHS(1).
MOVE 1 TO QIPNUM.
CALL DSQCIB USING DSQCOMM, QICLTH, STARTQI,
QIPNUM, QIKLTHS, SNAMEs,
*
* Set numeric values into query variables using SET GLOBAL command
MOVE 10 TO QICLTH.
MOVE 7 TO KLTHS(1).
MOVE 5 TO KLTHS(2).
MOVE 7 TO KLTHS(3).
MOVE 4 TO VLTHS(1).
MOVE 4 TO VLTHS(2).
MOVE 4 TO VLTHS(3).
MOVE 20 TO VVALS(1).
MOVE 40 TO VVALS(2).
MOVE 84 TO VVALS(3).
MOVE 3 TO QIPNUM.
CALL DSQCIB USING DSQCOMM, QICLTH, SETG,
QIPNUM, QIKLTHS, VNAMEs,
QIVLTHS, VVALUES, DSQ-VARIABLE-FINT.
*
* Run a query
MOVE 12 TO QICLTH.
CALL DSQCIB USING DSQCOMM, QICLTH, QUERY.
*
* Print the results of the query
MOVE 22 TO QICLTH.
CALL DSQCIB USING DSQCOMM, QICLTH, REPT.
*
* End the query interface session
MOVE 4 TO QICLTH.
CALL DSQCIB USING DSQCOMM, QICLTH, ENDQI.

STOP RUN.

```

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

COBOL 用の DSQCOMM

COBOL 言語のインターフェース連絡域ファイルは、DSQCOMM.B という名前のファイルです。

QMF で提供される DSQCOMM.B インクルード・ファイルを次に示します。

```

*****
* COBOL INCLUDE FOR QUERY CALLABLE INTERFACE
*****

* STRUCTURE DECLARE FOR COMMUNICATIONS AREA

```

```

01 DSQCOMM.

    03 DSQ-RETURN-CODE PIC 9(8) USAGE IS COMP.
*      FUNCTION RETURN CODE      *
    03 DSQ-INSTANCE-ID PIC 9(8) USAGE IS COMP.
*      IDENTIFIER FROM START CMD *
    03 DSQ-COMM-LEVEL  PIC X(12).
*      COMMUNICATIONS LEVEL      *
    03 DSQ-PRODUCT     PIC X(2).
*      QUERY PRODUCT ID          *
    03 DSQ-PRODUCT-RELEASE PIC X(2).
*      QUERY PRODUCT RELEASE     *
    03 DSQ-RESERVE1    PIC X(28).
*      RESERVED AREA            *
    03 DSQ-MESSAGE-ID  PIC X(8).
*      COMPLETION MESSAGE ID    *
    03 DSQ-Q-MESSAGE-ID PIC X(8).
*      QUERY MESSAGE ID         *
    03 DSQ-START-PARM-ERROR PIC X(8).
*      START PARAMETER IN ERROR *
    03 DSQ-CANCEL-IND  PIC X(1).
*      1 = COMMAND CANCELLED    *
*      0 = COMMAND NOT CANCELLED *
    03 DSQ-RESERVE2    PIC X(23).
*      RESERVED AREA            *
    03 DSQ-RESERVE3    PIC X(156).
*      RESERVED AREA            *
    03 DSQ-MESSAGE-TEXT PIC X(128).
*      QMF MESSAGE TEXT         *
    03 DSQ-Q-MESSAGE-TEXT PIC X(128).
*      QMF QUERY MESSAGE TEXT   *
*      512 BYTES TOTAL          *

*  VALUES FOR DSQ-RETURN-CODE

    01 DSQ-SUCCESS PIC 9(8) USAGE IS COMP VALUE 0.
    01 DSQ-WARNING  PIC 9(8) USAGE IS COMP VALUE 4.
    01 DSQ-FAILURE  PIC 9(8) USAGE IS COMP VALUE 8.
    01 DSQ-SEVERE   PIC 9(8) USAGE IS COMP VALUE 16.

*  VALUES FOR DSQ-INSTANCE-ID

    01 DSQ-CONTINUE PIC 9(8) USAGE IS COMP VALUE 0.

*  VALUES FOR DSQ-COMM-LEVEL

    01 DSQ-CURRENT-COMM-LEVEL PIC X(12) VALUE "DSQL>001002<".

*  VALUES FOR DSQ-PRODUCT

    01 DSQ-QRW          PIC X(2) VALUE "01".
    01 DSQ-QMF          PIC X(2) VALUE "02".
    01 DSQ-QM4         PIC X(2) VALUE "03".

*  VALUES FOR DSQ-PRODUCT-RELEASE

    01 DSQ-QRW-V1R2     PIC X(2) VALUE "01".
    01 DSQ-QRW-V1R3     PIC X(2) VALUE "02".
    01 DSQ-QMF-V2R4     PIC X(2) VALUE "01".
    01 DSQ-QMF-V3R1     PIC X(2) VALUE "02".
    01 DSQ-QMF-V3R1M1   PIC X(2) VALUE "03".
    01 DSQ-QMF-V3R2     PIC X(2) VALUE "04".
    01 DSQ-QMF-V3R3     PIC X(2) VALUE "05".
    01 DSQ-QMF-V6R1     PIC X(2) VALUE "06".
    01 DSQ-QMF-V7R1     PIC X(2) VALUE "07".
    01 DSQ-QM4-V1R1     PIC X(2) VALUE "01".

```

```

01 DSQ-QMF-V7R2      PIC X(2) VALUE "08".
01 DSQ-QMF-V8R1      PIC X(2) VALUE "09".
01 DSQ-QMF-V9R1      PIC X(2) VALUE "10".
01 DSQ-QMF-V10R1     PIC X(2) VALUE "11".

*   VALUES FOR DSQ-CANCEL-IND

01 DSQ-CANCEL-YES     PIC X(1) VALUE "1".
01 DSQ-CANCEL-NO      PIC X(1) VALUE "0".

*   VALUES FOR MODE

01 DSQ-INTERACTIVE    PIC X(1) VALUE "1".
01 DSQ-BATCH           PIC X(1) VALUE "2".

*   VALUES YES AND NO

01 DSQ-YES             PIC X(1) VALUE "1".
01 DSQ-NO              PIC X(1) VALUE "2".

*   CALLABLE INTERFACE PROGRAM NAME

01 DSQCIB              PIC X(6) VALUE "DSQCIB".

*   VALUES FOR VARIABLE TYPE ON CALL PARAMETER

01 DSQ-VARIABLE-CHAR  PIC X(4) VALUE "CHAR".
01 DSQ-VARIABLE-FINT  PIC X(4) VALUE "FINT".

```

COBOL 呼び出し可能インターフェース・プログラムの実行に関する考慮事項

QMF 呼び出し可能インターフェースを使用する COBOL プログラムの実行に関する詳細情報に注意してください。

QMF 呼び出し可能インターフェースを使用するプログラムを変換、コンパイル、およびリンク・エディットするときには、以下の条件について考慮してください。

- 実行環境

QMF は、COBOL 環境ではアセンブラー・プログラムとして実行されます。COBOL プログラムから、COBOL 動的呼び出しを使用して、QMF インターフェース・プログラム DSQCIB を呼び出す必要があります。

- 引用符を使用するか、アポストロフィを使用するか

COBOL プログラム内では、リテラルを区切るために二重引用符 (") またはアポストロフィ (') を使用する必要があります。CICS変換プロセスおよび COBOL コンパイラーに対して、使用する区切り文字を示すために、QUOTE または APOST を指定してください。COBOL コンパイラーで有効な APOST または QUOTE オプションが、CICS 変換プログラムでも有効であることを確認してください。

QMF で配布されるような連絡域 (DSQCOMMB) およびサンプル COBOL プログラム (DSQABFCO) では、リテラルの区切り文字として引用符を使用します。ご使用のサイトまたはプログラムで引用符の代わりにアポストロフィを使用している場合は、DSQCOMMB を変更するか、または引用符をアポストロフィに変更して、構造をプログラムにコピーしてください。

- 連絡域 (DSQCOMMB) が使用可能であるか

連絡域 DSQCOMMB が COBOL コンパイル・ステップで使用可能であるか、またはそれを制御構造としてプログラムにコピーする必要があります。

- インターフェース・モジュール (DSQCIB) が使用可能であるか

プログラムのリンク・エディットのステップで、QMF インターフェース・モジュールが使用可能でなければなりません。

CICS での COBOL プログラムの実行

COBOL でプログラムを作成したら、変換、コンパイル、およびリンク・エディットしてから、実行する必要があります。

これらの例の JCL は、QMF には提供されていませんが、ここからコピーして、お客様の要件に合うように変更できます。

次の例では、CICS から提供されるプロシージャ DFHEBTVL (COBOL に対応) を示します。

```
//samCOBOL JOB
//          EXEC PROC=DFHEBTVL
//TRN.SYSIN DD *
*CBL      XOPTS(CICS translator options ...QUOTE COBOL2)
          .
          .
          .
          Your program or copy of QMF sample DSQABFCO
          .
          .
/*
/* Provide access to QMF communications macro DSQCOMMB
//COB.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Allocation for target library
//LKED.SYSLMOD DD
/* Allocation for QMF load library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE CICSLOAD(DFHECI)
          INCLUDE QMFLOAD(DSQCIB)
          ORDER DFHECI
          ENTRY samCOBOL
          MODE AMODE(31) RMODE(31)
          NAME samCOBOL(R)
/*
```

TSO での COBOL プログラムの実行

TSO で COBOL プログラムを実行するには、プログラムをコンパイルし、リンク・エディットしてから、プログラムを実行します。その際、ISPF を使用するかどうかは任意です。

TSO のコンパイルおよびリンク・エディット

COBOL プログラムは、TSO で実行する前に、コンパイルし、リンク・エディットする必要があります。

このジョブは、COBOL コンパイラーを使用して、呼び出し可能インターフェース・アプリケーションをコンパイルします。次に、アプリケーションをリンク・エディットしています。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

```
//samCOBOL JOB
//STEP1 EXEC PROC=IGYWCL
/* Provide access to QMF communications macro DSQCOMM
//COBOL.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//COBOL.SYSIN DD *
.
.
.
Your program or copy of QMF sample DSQABFCO
.
.
/* Provide access to QMF interface module
/* Allocation for target library
//LKED.SYSLMOD DD
/* Allocation for QMF load library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCIB)
ENTRY samCOBOL
MODE AMODE(31) RMODE(31)
NAME samCOBOL(R)
/*
```

ISPF を使用しない TSO でのプログラムの実行

COBOL プログラムのコンパイルが正常に終了したら、ISPF なしで JCL を使用してそのプログラムを実行することができます。

ISPF を使用しない TSO で COBOL コンパイラーおよびリンケージ・エディターを実行するには、次に示す CLIST のようなプログラムを作成します。

```
PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2, GDDM and COBOL load libraries must be */
/* allocated before running this CLIST. */
/* Name of QMF load library is "QMF1210.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
```

```

/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL samCOBOL
EXIT CODE(0)

```

ISPF のもとの TSO でのプログラムの実行

COBOL プログラムのコンパイルが正常に終了したら、ISPF のもとでそのプログラムを実行することができます。

ISPF のもとの TSO でプログラムを実行するには、次に示す CLIST のようなプログラムを作成します。

```

PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD',+
        'PRODUCT.COB2LIB')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)

```

```

ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(samCOBOL) NEWAPPL(DSQE)
EXIT CODE(4)

```

EXIT CODE(4) は、「ISPF 後処理」パネルの表示を抑止します。

Fortran 言語インターフェース

QMF for TSO では、Fortran 言語を呼び出し可能インターフェースで使用できません。

制約事項: CICS では Fortran を使用できないため、Fortran 用の QMF 呼び出し可能インターフェースは CICS では機能しません。

Fortran 用のインターフェース連絡域マッピング (DSQCOMMFM)

DSQCOMMFM は、QMF に用意されている Fortran 言語プログラム用の DSQCOMM マッピングです。

この表は、ユーザーが変更してはならない DSQCOMMFM の情報を示しています。

表 47. Fortran 用のインターフェース連絡域 (DSQCOMMFM)

構造名	データ・タイプ	説明
DSQ_RETURN_CODE	INTEGER	QMF コマンドの実行後の状況を示す 値は次のとおり。 DSQ_SUCCESS 要求が正常に実行された DSQ_WARNING 警告を伴って正常に終了 DSQ_FAILURE コマンドが正しく実行されなかった DSQ_SEVERE 重大エラー、QMF セッションは終了
DSQ_INSTANCE_ID	INTEGER	START コマンドの実行時に、QMF によって設定される ID
DSQ_COMM_LEVEL	CHARACTER(12)	DSQCOMM のレベルを識別する ご使用のアプリケーションに、この変数を DSQ_CURRENT_COMM_LEVEL の値に初期設定し てから QMF START コマンドを出す命令を含めてく ださい。
DSQ_PRODUCT	CHARACTER(2)	使用中の IBM 照会プロダクトを識別する 先頭に DSQ_QMF が付く変数は、QMF for TSO バ ージョンを指定します。

表 47. Fortran 用のインターフェース連絡域 (DSQCOMM) (続き)

構造名	データ・タイプ	説明
DSQ_PRODUCT_RELEASE	CHARACTER(2)	使用中の QMF のリリース・レベル 変数 DSQ_QMF_V11R1 は QMF バージョン 12 リリース 1 を指定します。
DSQ_RESERVE1	CHARACTER(28)	将来の使用のために予約されている。
DSQ_MESSAGE_ID	CHARACTER(8)	完了メッセージ ID
DSQ_Q_MESSAGE_ID	CHARACTER(8)	照会メッセージ ID
DSQ_START_PARM_ERROR	CHARACTER(8)	パラメーター・エラーのために START が失敗したときの、エラーがあるパラメーター
DSQ_CANCEL_IND	CHARACTER(1)	QMF コマンドの実行中にユーザーが取り消したかどうかによって、以下の 2 つの値のいずれかが入る。 <ul style="list-style-type: none"> • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	CHARACTER(23)	将来の使用のために予約されている。
DSQ_RESERVE3	CHARACTER(156)	将来の使用のために予約されている。
DSQ_MESSAGE_TEXT	CHARACTER(128)	完了メッセージ・テキスト
DSQ_Q_MESSAGE_TEXT	CHARACTER(128)	照会メッセージ・テキスト

Fortran 用の関数呼び出し

QMF には、Fortran 言語用に DSQCIF および DSQCIFE という 2 つの関数呼び出しが用意されています。2 つの呼び出しは、連絡域 (DSQCOMM) に記述されています。

DSQCIF

この呼び出しは、アプリケーション・プログラム変数にアクセスする必要がない QMF コマンド用です。大部分の QMF コマンドに、この呼び出しを使用します。

```
RC = DSQCIF(DSQCOMM,
+   CMDLTH,
+   CMDSTR)
```

各パラメーターには以下の値が入ります。

DSQCOMM

連絡域。

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。整数パラメーター。

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの大文字ストリング。

DSQCIFE

この呼び出しには、アプリケーション・プログラム変数へのアクセスが必要なコマンド (START、TRACE、および拡張形式の GET GLOBAL と SET GLOBAL) 用の拡張構文があります。

この呼び出しの構文は次のとおりです。

```
RC = DSQCIFE(DSQCOMM,  
+   CMDLTH,  
+   CMDSTR,  
+   PNUM,  
+   KLTH,  
+   KWORD,  
+   VLTH,  
+   VALUE,  
+   VTYPE)
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (**CMDSTR**) の長さ。整数パラメーター。

CMDSTR

実行する QMF コマンド。**CMDLTH** によって指定した長さの大文字ストリング。

PNUM

コマンド・キーワードまたはトレース域の数 (整数パラメーター)。

KLTH

指定する各キーワードまたは各トレース・タイトルの長さ (整数パラメーターまたはパラメーターの配列)。

KWORD

QMF キーワード、キーワード、またはトレース・タイトルのアドレス (文字、文字の配列、または **KLTH** で指定された長さのトレース・タイトルのアドレスの配列)。

すべてのキーワードの長さが同じ場合、文字の配列を使用することができます。キーワードは連続記憶域に含まれている必要があり、特殊な区切り文字で区切られてはなりません。

VLTH

キーワードまたはトレース・タイトルに関連付けられている各値の長さ (整数パラメーターまたはパラメーターの配列)。

VALUE

各キーワードに関連付けられている値、またはトレース・タイトルに関連付けられている値のアドレス。

この値のタイプを **VTYPE** パラメーターに指定します。この値は、文字、文字配列、整数パラメーター、またはパラメーターの配列のいずれかです。トレース・データの場合は、**VTYPE** が **FINT** である必要があります。文字値がある場合は、それらの値が連続記憶域に含まれている必要があり、特殊な区切り文字で区切られてはなりません。

VTYPE

VALUE パラメーターの内容のデータ・タイプ。

このパラメーターには、連絡域 **DSQCOMMF** で提供される以下の 2 つの値のいずれかを設定します。

- 文字値を示す **DSQ_VARIABLE_CHAR**

- 整数値を示す DSQ_VARIABLE_FINT。

VALUE フィールドに指定したすべての値のデータ・タイプを VTYPE に指定する必要があります。

Fortran プログラミングの例

QMF には、サンプル・プログラム DSQABFF が用意されています。このサンプル・プログラムは、ライブラリー QMF1210.SDSQSAPn (n は各国語 ID) のメンバーです。

Fortran 呼び出し可能インターフェースのサンプル・プログラムでは、以下の機能が実行されます。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

```

C*****
C Sample program: DSQABFF
C FORTRAN version of QMF manager callable interface
C
C Creation Date: 11/21/89
C
C ENVIRONMENT:    API IN FORTRAN
C*****
C
C Processing:
C     a. Start a query manager session using the callable interface.
C
C     b. Set global query manager numeric variables.
C
C     d. Run a query manager query using the callable interface.
C
C     e. Print a report using the callable interface.
C
C     f. Exit the query manager session.
C
C Prerequisites:1. Create the SAMPLE database.
C
C                 2. Create a prompted query, Q1, which has a SELECT state
C
C                 3. Create a form, F1, that displays data for query Q1.
C
C*****
C                 PROGRAM DSQABFF
C*****
C Include and declare query interface communications area
C*****
C                 INCLUDE (DSQCOMM)
C*****
C Query interface command lengths and commands
C*****
C                 INTEGER COMMAND_LENGTH
C                 CHARACTER START_QUERY_INTERFACE*5,
C                 +                 SET_GLOBAL_VARIABLES*10,

```

```

+          RUN_QUERY*12,
+          PRINT_REPORT*22,
+          END_QUERY_INTERFACE*4

C*****
C Query command extension, number of parameters and lengths
C*****
      INTEGER NUMBER_OF_PARAMETERS,
+       KEYWORD_LENGTHS(10),
+       DATA_LENGTHS(10)

C*****
C Variable data type constants
C*****
      CHARACTER CHAR_DATA_TYPE*4,
+       INT_DATA_TYPE*4

C*****
C Keyword parameter and value for START command
C*****
      CHARACTER*8  START_KEYWORDS(1)
      CHARACTER*11 START_KEYWORD_VALUES(1)

C*****
C Keyword parameter and values for SET command
C*****
      CHARACTER  SET_KEYWORDS(19)
      CHARACTER  SET_KEYWORD_1*7,
+       SET_KEYWORD_2*5,
+       SET_KEYWORD_3*7

      EQUIVALENCE (SET_KEYWORDS( 1), SET_KEYWORD_1),
+       (SET_KEYWORDS( 8), SET_KEYWORD_2),
+       (SET_KEYWORDS(13), SET_KEYWORD_3)

      CHARACTER  SET_VALUES(12)
      INTEGER*4  SET_VALUE_1,
+       SET_VALUE_2,
+       SET_VALUE_3

      EQUIVALENCE (SET_VALUES(1), SET_VALUE_1),
+       (SET_VALUES(5), SET_VALUE_2),
+       (SET_VALUES(9), SET_VALUE_3)

C*****
C Declare command length and return code variables
C*****
      INTEGER  LEN,
+       RC

C*****
C Initialization
C*****

      DATA START_QUERY_INTERFACE  /'START'           /
      DATA SET_GLOBAL_VARIABLES  /'SET GLOBAL'       /
      DATA RUN_QUERY              /'RUN QUERY Q1'    /
      DATA PRINT_REPORT           /'PRINT REPORT (FORM=F1)'/
      DATA END_QUERY_INTERFACE    /'EXIT'           /

      DATA CHAR_DATA_TYPE         /DSQ_VARIABLE_CHAR  /
      DATA INT_DATA_TYPE          /DSQ_VARIABLE_FINT  /

C*****
C Start query session
C*****
      DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL

```

```

NUMBER_OF_PARAMETERS = 1
COMMAND_LENGTH       = LEN(START_QUERY_INTERFACE)
KEYWORD_LENGTHS(1)  = LEN(START_KEYWORDS(1))
DATA_LENGTHS(1)     = LEN(START_KEYWORD_VALUES(1))
START_KEYWORDS(1)   = 'DSQSMODE'
START_KEYWORD_VALUES(1) = 'INTERACTIVE'

RC = DSQCIFE(DSQCOMM,
+          COMMAND_LENGTH,
+          START_QUERY_INTERFACE,
+          NUMBER_OF_PARAMETERS,
+          KEYWORD_LENGTHS,
+          START_KEYWORDS,
+          DATA_LENGTHS,
+          START_KEYWORD_VALUES,
+          CHAR_DATA_TYPE)

C*****
C  Set numeric values into query using SET command
C*****
NUMBER_OF_PARAMETERS = 3
COMMAND_LENGTH       = LEN(SET_GLOBAL_VARIABLES)
SET_KEYWORD_1       = 'MYVAR01'
SET_KEYWORD_2       = 'SHORT'
SET_KEYWORD_3       = 'MYVAR03'
KEYWORD_LENGTHS(1)  = LEN(SET_KEYWORD_1)
KEYWORD_LENGTHS(2)  = LEN(SET_KEYWORD_2)
KEYWORD_LENGTHS(3)  = LEN(SET_KEYWORD_3)
DATA_LENGTHS(1)     = 4
DATA_LENGTHS(2)     = 4
DATA_LENGTHS(3)     = 4
SET_VALUE_1         = 20
SET_VALUE_2         = 40
SET_VALUE_3         = 84

RC = DSQCIFE(DSQCOMM,
+          COMMAND_LENGTH,
+          SET_GLOBAL_VARIABLES,
+          NUMBER_OF_PARAMETERS,
+          KEYWORD_LENGTHS,
+          SET_KEYWORDS,
+          DATA_LENGTHS,
+          SET_VALUES,
+          INT_DATA_TYPE)

C*****
C  Run a query
C*****
COMMAND_LENGTH = LEN(RUN_QUERY)
RC = DSQCIF(DSQCOMM,
+          COMMAND_LENGTH,
+          RUN_QUERY)

C*****
C  Print the results of the query
C*****
COMMAND_LENGTH = LEN(PRINT_REPORT)
RC = DSQCIF(DSQCOMM,
+          COMMAND_LENGTH,
+          PRINT_REPORT)

C*****
C  End the query interface session
C*****
COMMAND_LENGTH = LEN(END_QUERY_INTERFACE)
RC = DSQCIF(DSQCOMM,

```

```
+          COMMAND_LENGTH,  
+          END_QUERY_INTERFACE)
```

```
END
```

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

Fortran 用の DSQCOMM

Fortran 言語のインターフェース連絡域ファイルは、DSQCOMM.F という名前のファイルです。

QMF で提供される DSQCOMM.F インクルード・ファイルを次に示します。

```
C*****  
C          FORTRAN include file for callable interface  
C*****  
C          Return codes  
          INTEGER DSQ_SUCCESS, DSQ_WARNING, DSQ_FAILURE, DSQ_SEVERE  
          PARAMETER(  
+             DSQ_SUCCESS = 0,  
+             DSQ_WARNING = 4,  
+             DSQ_FAILURE = 8,  
+             DSQ_SEVERE  = 16)  
  
C          Communications level  
          CHARACTER DSQ_CURRENT_COMM_LEVEL*12  
          PARAMETER(  
+             DSQ_CURRENT_COMM_LEVEL = 'DSQL>001002<')  
  
C          Query product IDs  
          CHARACTER DSQ_QRW*2, DSQ_QMF*2, DSQ_QM4*2  
          PARAMETER(  
+             DSQ_QRW = '01',  
+             DSQ_QMF = '02',  
+             DSQ_QM4 = '03')  
  
C          Query product release levels  
          CHARACTER DSQ_QRW_V1R2*2, DSQ_QRW_V1R3*2,  
+             DSQ_QMF_V2R4*2, DSQ_QMF_V3R1*2,  
+             DSQ_QMF_V3R1M1*2, DSQ_QMF_V3R2*2,  
+             DSQ_QMF_V3R3*2, DSQ_QMF_V6R1*2,  
+             DSQ_QMF_V7R1*2, DSQ_QM4_V1R1*2,  
+             DSQ_QMF_V7R2*2, DSQ_QMF_V8R1*2,  
+             DSQ_QMF_V9R1*2, DSQ_QMF_V10R1*2  
          PARAMETER(  
+             DSQ_QRW_V1R2 = '01',  
+             DSQ_QRW_V1R3 = '02',  
+             DSQ_QMF_V2R4 = '01',  
+             DSQ_QMF_V3R1 = '02',  
+             DSQ_QMF_V3R1M1 = '03',  
+             DSQ_QMF_V3R2 = '04',  
+             DSQ_QMF_V3R3 = '05',  
+             DSQ_QMF_V6R1 = '06',  
+             DSQ_QMF_V7R1 = '07',  
+             DSQ_QM4_V1R1 = '01',  
+             DSQ_QMF_V7R2 = '08',  
+             DSQ_QMF_V8R1 = '09',  
+             DSQ_QMF_V9R1 = '10',  
+             DSQ_QMF_V10R1 = '11')  
  
C          Host variable types
```

```

CHARACTER DSQ_VARIABLE_CHAR*4, DSQ_VARIABLE_FINT*4
PARAMETER(
+     DSQ_VARIABLE_CHAR = 'CHAR',
+     DSQ_VARIABLE_FINT = 'FINT')

C     Cancel indicator
CHARACTER DSQ_CANCEL_YES, DSQ_CANCEL_NO
PARAMETER(
+     DSQ_CANCEL_YES = '1',
+     DSQ_CANCEL_NO = '0')
CHARACTER DSQCOMM(512)
INTEGER DSQ_RETURN_CODE, DSQ_INSTANCE_ID
CHARACTER DSQ_COMM_LEVEL*12,
+     DSQ_PRODUCT*2,
+     DSQ_PRODUCT_RELEASE*2,
+     DSQ_RESERVE1*28,
+     DSQ_MESSAGE_ID*8,
+     DSQ_Q_MESSAGE_ID*8,
+     DSQ_START_PARM_ERROR*8,
+     DSQ_CANCEL_IND*1,
+     DSQ_RESERVE2*23,
+     DSQ_RESERVE3*156,
+     DSQ_MESSAGE_TEXT*128,
+     DSQ_Q_MESSAGE_TEXT*128

EQUIVALENCE (DSQCOMM( 1), DSQ_RETURN_CODE ),
+ (DSQCOMM( 5), DSQ_INSTANCE_ID ),
+ (DSQCOMM( 9), DSQ_COMM_LEVEL ),
+ (DSQCOMM(21), DSQ_PRODUCT ),
+ (DSQCOMM(23), DSQ_PRODUCT_RELEASE ),
+ (DSQCOMM(25), DSQ_RESERVE1 ),
+ (DSQCOMM(53), DSQ_MESSAGE_ID ),
+ (DSQCOMM(61), DSQ_Q_MESSAGE_ID ),
+ (DSQCOMM(69), DSQ_START_PARM_ERROR ),
+ (DSQCOMM(77), DSQ_CANCEL_IND ),
+ (DSQCOMM(78), DSQ_RESERVE2 ),
+ (DSQCOMM(101), DSQ_RESERVE3 ),
+ (DSQCOMM(257), DSQ_MESSAGE_TEXT ),
+ (DSQCOMM(385), DSQ_Q_MESSAGE_TEXT )

C     Callable interface normal and extended calls
EXTERNAL DSQCIF
EXTERNAL DSQCIFE

```

Fortran プログラムの実行

TSO で Fortran プログラムを実行するには、プログラムをコンパイルし、リンク・エディットしてから、プログラムを実行します。その際、ISPF を使用するかどうかは任意です。

プログラムのコンパイルおよびリンク・エディット

Fortran プログラムは、プログラムをコンパイルし、リンク・エディットしてからでないと、TSO で実行できません。

Fortran コンパイラおよびリンケージ・エディターを実行するための JCL では、z/OS 用の Fortran コンパイラが使用されます。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

この例の JCL は、QMF には提供されていませんが、ここからコピーして、お客様の要件に合うように変更できます。

```

//samFORT    JOB
//STEP1      EXEC PROC=VSF2CL
/* Provide access to QMF communications macro DSQCOMM
//FORT.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//FORT.SYSIN  DD *
.
.
.
Your program or copy of QMF sample DSQABFF
.
.
.
/*
/* Provide access to QMF interface module
/* Allocation for target library
//LKED.SYSLMOD DD
/* Allocation for QMF load library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN  DD *
        INCLUDE QMFLOAD(DSQCIF)
        INCLUDE QMFLOAD(DSQCIF)E)
        ENTRY samFORT
        MODE  AMODE(31) RMODE(31)
        NAME  samFORT(R)
/*

```

ISPF を使用しない TSO でのプログラムの実行

Fortran プログラムのコンパイルが正常に終了したら、ISPF なしで JCL を使用してそのプログラムを実行することができます。

この例の JCL は、QMF には提供されていませんが、ここからコピーして、お客様の要件に合うように変更できます。

次に示すプログラムは、Fortran コンパイラを使用して呼び出し可能インターフェース・アプリケーションを実行します。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

```

PROC 0
CONTROL ASIS
/*****
/* Note: QMF, DB2, GDDM and FORTRAN load libraries must be */
/*      allocated before running this CLIST.                */
/*      Name of QMF load library is "QMF1210.SDSQLOAD".    */
/*****
/* Specify attribute list for dataset allocations          */
/*****
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80)  RECFM(F B)   BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB  LRECL(79)  RECFM(F B A) BLKSIZE(4029)
/*****
/* Datasets used by TSO                                   */
/*****
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****
/* QMF/GDDM Datasets                                     */
/*****
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM)  DA('QMF1210.DSQCFRM')  SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM')     SHR REUSE
ALLOC FI(ADMGDF)   DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS)  DA('ADM.GDDM.NICKNAME') SHR REUSE

```



```

/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL samFORT
EXIT CODE(0)

```

ISPF のもとの TSO での実行

Fortran プログラムのコンパイルが正常に終了したら、ISPF のもとでそのプログラムを実行することができます。

この例の CLIST は、QMF には提供されていませんが、ここからコピーして、お客様の要件に合うように変更できます。

次に示す CLIST は、Fortran コンパイラを使用して呼び出し可能インターフェース・アプリケーションを実行します。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

```

PROC 0
CONTROL ASIS
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PRODUCT.VSF2LOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGMGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/

```

```

/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(samFORT) NEWAPPL(DSQE)
EXIT CODE(4)

```

EXIT CODE(4) は、「ISPF 後処理」パネルの表示を抑制します。

PL/I 言語インターフェース

QMF では、PL/I 言語を呼び出し可能インターフェースで使用できます。

PL/I 用のインターフェース連絡域マッピング (DSQCOMML)

DSQCOMML は PL/I 用の DSQCOMM マッピングであり、QMF に用意されています。

次の表に、DSQCOMML の各値を示します。

表 48. PL/I (DSQCOMML) のインターフェース連絡域

構造名	データ・タイプ	説明
DSQ_RETURN_CODE	FIXED BIN(31)	QMF コマンドの実行後の状況を示す 値は次のとおり。 DSQ_SUCCESS 要求が正常に実行された DSQ_WARNING 警告を伴って正常に終了 DSQ_FAILURE コマンドが正しく実行されなかった DSQ_SEVERE 重大エラー、QMF セッションは終了
DSQ_INSTANCE_ID	FIXED BIN(31)	START コマンドの実行時に、QMF によって設定される ID
DSQ_COMM_LEVEL	CHAR(12)	DSQCOMM 連絡域のレベルを識別する ご使用のアプリケーションに、この変数を DSQ_CURRENT_COMM_LEVEL の値に初期設定してから QMF START コマンドを出す命令を含めてください。
DSQ_PRODUCT	CHAR(2)	使用中の IBM 照会プロダクトを識別する。先頭に DSQ_QMF が付く変数は、QMF for TSO バージョンお よび CICS バージョンを指定します。

表 48. PL/I (DSQCOMML) のインターフェース連絡域 (続き)

構造名	データ・タイプ	説明
DSQ_PRODUCT_RELEASE	CHAR(2)	使用中の QMF のリリース・レベル。変数 DSQ_QMF_V11R1 は QMF バージョン 12 リリース 1 を指定します。
DSQ_RESERVE1	CHAR(28)	将来の使用のために予約されている。
DSQ_MESSAGE_ID	CHAR(8)	完了メッセージ ID
DSQ_Q_MESSAGE_ID	CHAR(8)	照会メッセージ ID
DSQ_START_PARM_ERROR	CHAR(8)	パラメーター・エラーのために START が失敗したときの、エラーがあるパラメーター
DSQ_CANCEL_IND	CHAR(1)	QMF コマンドの実行中にユーザーが取り消したかどうかによって、以下の 2 つの値のいずれかが入る。 <ul style="list-style-type: none"> • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	CHAR(23)	将来の使用のために予約されている。
DSQ_RESERVE3	CHAR(156)	将来の使用のために予約されている。
DSQ_MESSAGE_TEXT	CHAR(128)	完了メッセージ・テキスト
DSQ_Q_MESSAGE_TEXT	CHAR(128)	照会メッセージ・テキスト

PL/I 用の関数呼び出し

QMF には、PL/I 用に DSQCIPL と DSQCIPX という 2 つの関数呼び出しが用意されています。2 つの呼び出しは、連絡域 (DSQCOMML) に記述されています。

DSQCIPL の構文

この呼び出しは、アプリケーション・プログラム変数にアクセスする必要がない QMF コマンド用です。大部分の QMF コマンドに、この呼び出しを使用します。

```
CALL DSQCIPL(DSQCOMM,
             CMDLTH,
             CMDSTR)
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDSTR) の長さ。

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの大文字ストリング。

DSQCIPX の構文

この呼び出しは、アプリケーション・プログラム変数へのアクセスが必要なコマンド (START、TRACE、および拡張形式の GET GLOBAL と SET GLOBAL) 用です。

この呼び出しの構文は次のとおりです。

```
CALL DSQCIPX(DSQCOMM,  
             CMDLTH,  
             CMDSTR,  
             PNUM,  
             KLTH,  
             KWORD,  
             VLTH,  
             VALUE,  
             VTYPE)
```

各パラメーターには以下の値が入ります。

DSQCOMM

インターフェース連絡域

CMDLTH

コマンド・ストリング (CMDST) の長さ。整数 FIXED BIN(31) パラメーター。

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの大文字ストリング。

PNUM

コマンド・キーワードまたはトレース域の数 (整数 FIXED BIN(31) パラメーター)。

KLTH

指定する各キーワードまたは各トレース・タイトルの長さ (整数 FIXED BIN(31) パラメーターまたはパラメーターの配列)。

KWORD

QMF キーワード、キーワード、またはトレース・タイトルのアドレス。

それぞれ、文字、文字の配列、または KLTH で指定された長さのトレース・タイトルのアドレスの配列として指定します。すべてのキーワードの長さが同じ場合、文字の配列を使用することができます。キーワードは連続記憶域に含まれている必要があり、特殊な区切り文字で区切られてはなりません。

VLTH

キーワードまたはトレース・タイトルに関連付けられている各値の長さ (整数 FIXED BIN(31) パラメーターまたはパラメーターの配列)。

VALUE

各キーワードに関連付けられている値、またはトレース・タイトルに関連付けられている値のアドレス。

この値のタイプを VTYPE パラメーターに指定します。この値は、文字、文字配列、整数 FIXED BIN(31) パラメーター、またはパラメーターの配列のいずれかです。文字値がある場合は、それらの値が連続記憶域に含まれている必要があり、特殊な区切り文字で区切られてはなりません。

VTYPE

VALUE パラメーターの内容のデータ・タイプ。

このパラメーターには、連絡域 DSQCOMML で提供される以下の 2 つの値のいずれかを設定します。

- 文字値を示す DSQ_VARIABLE_CHAR
- 整数 FIXED BIN(31) 値を示す DSQ_VARIABLE_FINT。

VALUE フィールドに指定したすべての値のデータ・タイプを VTYPE に指定する必要があります。

PL/I プログラミングの例

サンプル・プログラム DSQABFP は、QMF から提供され、PL/I を使用します。このサンプル・プログラムは、ライブラリー QMF1210.SDSQSAP n (n は各国語 ID) のメンバーです。

PL/I 言語呼び出し可能インターフェースのサンプル・プログラムは、以下の機能を実行します。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

```

DSQABFP: PROCEDURE OPTIONS(MAIN REENTRANT) REORDER;
/*****
/* Sample program: DSQABFP
/* PL/I version of the QMF callable interface
*****/

/*****
/* Include and declare query interface communications area
*****/
%INCLUDE SYSLIB(DSQCOMML);

/*****
/* Built in function
*****/
DCL LENGTH BUILTIN;

/*****
/* Query interface command length and commands
*****/
DCL COMMAND_LENGTH FIXED BIN(31);
DCL START_QUERY_INTERFACE CHAR(5) INIT('START');
DCL SET_GLOBAL_VARIABLES CHAR(10) INIT('SET GLOBAL');
DCL RUN_QUERY CHAR(12) INIT('RUN QUERY Q1');
DCL PRINT_REPORT CHAR(22) INIT('PRINT REPORT (FORM=F1)');
DCL END_QUERY_INTERFACE CHAR(4) INIT('EXIT');

/*****
/* Query command extension, number of parameters and lengths
*****/
DCL NUMBER_OF_PARAMETERS FIXED BIN(31);/* number of variables
DCL KEYWORD_LENGTHS(10) FIXED BIN(31);/* lengths of keyword names*
DCL DATA_LENGTHS(10) FIXED BIN(31);/* lengths of variable data*

/*****
/* Keyword parameter and value for START command
*****/
DCL START_KEYWORDS CHAR(8) INIT('DSQSMODE');
```

```

DCL START_KEYWORD_VALUES CHAR(11) INIT('INTERACTIVE');

/*****
/* Keyword parameter and value for SET command */
*****/
DCL 1 SET_KEYWORDS,
    3 SET_KEYWORDS_1 CHAR(7) INIT('MYVAR01'),
    3 SET_KEYWORDS_2 CHAR(5) INIT('SHORT'),
    3 SET_KEYWORDS_3 CHAR(7) INIT('MYVAR03');

DCL 1 SET_VALUES,
    3 SET_VALUES_1 FIXED BIN(31),
    3 SET_VALUES_2 FIXED BIN(31),
    3 SET_VALUES_3 FIXED BIN(31);

/*****
/* Main program */
*****/
DSQCOMM = '';
DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL;

/*****
/* Start a query interface session */
*****/
NUMBER_OF_PARAMETERS = 1;
COMMAND_LENGTH = LENGTH(START_QUERY_INTERFACE);
KEYWORD_LENGTHS(1) = LENGTH(START_KEYWORDS);
DATA_LENGTHS(1) = LENGTH(START_KEYWORD_VALUES);

CALL DSQCIPX(DSQCOMM,
             COMMAND_LENGTH,
             START_QUERY_INTERFACE,
             NUMBER_OF_PARAMETERS,
             KEYWORD_LENGTHS,
             START_KEYWORDS,
             DATA_LENGTHS,
             START_KEYWORD_VALUES,
             DSQ_VARIABLE_CHAR);

/*****
/* Set numeric values into query using SET command */
*****/
NUMBER_OF_PARAMETERS = 3;
COMMAND_LENGTH = LENGTH(SET_GLOBAL_VARIABLES);
KEYWORD_LENGTHS(1) = LENGTH(SET_KEYWORDS_1);
KEYWORD_LENGTHS(2) = LENGTH(SET_KEYWORDS_2);
KEYWORD_LENGTHS(3) = LENGTH(SET_KEYWORDS_3);
DATA_LENGTHS(1) = 4;
DATA_LENGTHS(2) = 4;
DATA_LENGTHS(3) = 4;
SET_VALUES_1 = 20;
SET_VALUES_2 = 40;
SET_VALUES_3 = 84;

CALL DSQCIPX(DSQCOMM,
             COMMAND_LENGTH,
             SET_GLOBAL_VARIABLES,
             NUMBER_OF_PARAMETERS,
             KEYWORD_LENGTHS,
             SET_KEYWORDS,
             DATA_LENGTHS,
             SET_VALUES,
             DSQ_VARIABLE_FINT);

/*****
/* Run a query */
*****/

```

```

COMMAND_LENGTH = LENGTH(RUN_QUERY);

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             RUN_QUERY);

/*****
/* Print the results of the query */
*****/
COMMAND_LENGTH = LENGTH(PRINT_REPORT);

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             PRINT_REPORT);

/*****
/* End the query interface session */
*****/
COMMAND_LENGTH = LENGTH(END_QUERY_INTERFACE);

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             END_QUERY_INTERFACE);

END      DSQABFP;

```

関連資料:

5 ページの『各国語機能情報に関する規則』

DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語には、各国語機能 (NLF) が提供されます。

PL/I 用の DSQCOMM

PL/I 用のインターフェース連絡域は、DSQCOMMML という名前です。

```

/*****
/* PL/I include for query callable interface */
*****/

/* Structure declare for communications area */
DCL
1 DSQCOMM,
  3 DSQ_RETURN_CODE      FIXED BIN(31), /* Function return code */
  3 DSQ_INSTANCE_ID     FIXED BIN(31), /* Start ID */
  3 DSQ_COMM_LEVEL      CHAR(12),      /* Communications level */
  3 DSQ_PRODUCT         CHAR(2),       /* Query product ID */
  3 DSQ_PRODUCT_RELEASE CHAR(2),       /* Query product release */
  3 DSQ_RESERVE1        CHAR(28),      /* Reserved */
  3 DSQ_MESSAGE_ID     CHAR(8),       /* Completion message ID */
  3 DSQ_Q_MESSAGE_ID   CHAR(8),       /* Query message ID */
  3 DSQ_START_PARM_ERROR CHAR(8),      /* Start parms in error */
  3 DSQ_CANCEL_IND     CHAR(1),       /* Cmd cancel indicator */
                                     /* 1 = cancelled, 0 = not cancelled */
  3 DSQ_RESERVE2        CHAR(23),      /* Reserved */
  3 DSQ_RESERVE3        CHAR(156),     /* Reserved */
  3 DSQ_MESSAGE_TEXT   CHAR(128),     /* QMF command message */
  3 DSQ_Q_MESSAGE_TEXT CHAR(128);     /* QMF query message */

/* Return codes */
DCL
  DSQ_SUCCESS          FIXED BIN(31) INIT(0) STATIC,
  DSQ_WARNING          FIXED BIN(31) INIT(4) STATIC,
  DSQ_FAILURE          FIXED BIN(31) INIT(8) STATIC,
  DSQ_SEVERE           FIXED BIN(31) INIT(16) STATIC;

/* Communications level */

```

```

DCL
    DSQ_CURRENT_COMM_LEVEL  CHAR(12) INIT('DSQL>001002<') STATIC;

/* Query product ID */
DCL
    DSQ_QRW                  CHAR(2) INIT('01') STATIC,
    DSQ_QMF                  CHAR(2) INIT('02') STATIC,
    DSQ_QM4                  CHAR(2) INIT('03') STATIC;

/* Query product release ID */
DCL
    DSQ_QRW_V1R2            CHAR(2) INIT('01') STATIC,
    DSQ_QRW_V1R3            CHAR(2) INIT('02') STATIC,
    DSQ_QMF_V2R4            CHAR(2) INIT('01') STATIC,
    DSQ_QMF_V3R1            CHAR(2) INIT('02') STATIC,
    DSQ_QMF_V3R1M1          CHAR(2) INIT('03') STATIC,
    DSQ_QMF_V3R2            CHAR(2) INIT('04') STATIC,
    DSQ_QMF_V3R3            CHAR(2) INIT('05') STATIC,
    DSQ_QMF_V6R1            CHAR(2) INIT('06') STATIC,
    DSQ_QMF_V7R1            CHAR(2) INIT('07') STATIC,
    DSQ_QM4_V1R1            CHAR(2) INIT('01') STATIC,
    DSQ_QMF_V7R2            CHAR(2) INIT('08') STATIC,
    DSQ_QMF_V8R1            CHAR(2) INIT('09') STATIC,
    DSQ_QMF_V9R1            CHAR(2) INIT('10') STATIC,
    DSQ_QMF_V10R1           CHAR(2) INIT('11') STATIC;

/* Cancelled indicator */
DCL
    DSQ_CANCEL_YES          CHAR(1) INIT('1') STATIC,
    DSQ_CANCEL_NO           CHAR(1) INIT('0') STATIC;

/* Variable types */
DCL
    DSQ_VARIABLE_CHAR       CHAR(4) INIT('CHAR') STATIC,
    DSQ_VARIABLE_FINT        CHAR(4) INIT('FINT') STATIC;

/* Mode */
DCL
    DSQ_INTERACTIVE         CHAR(1) INIT('1') STATIC,
    DSQ_BATCH                CHAR(1) INIT('2') STATIC;

/* Yes or no */
DCL
    DSQ_YES                  CHAR(1) INIT('1') STATIC,
    DSQ_NO                   CHAR(1) INIT('2') STATIC;

/* Query interface entry point */
DCL
    DSQCIPL ENTRY (*, /* Interface block */
                  FIXED BIN(31), /* Length of command */
                  CHAR(*)) /* Command string */
    EXTERNAL OPTIONS(ASSEMBLER);

DCL
    DSQCIPX ENTRY (*, /* Interface block */
                  FIXED BIN(31), /* Length of command */
                  CHAR(*), /* Command string */
                  FIXED BIN(31), /* # of command keywords */
                  *, /* Length of keyword */
                  *, /* Keyword string */
                  *, /* Length of value */
                  *, /* Value of keyword */
                  CHAR(4)) /* Data type of value */
    EXTERNAL OPTIONS(ASSEMBLER);

```


CICS でのプログラムの実行

PL/I でプログラムを作成したら、コンパイルしてから実行する必要があります。

QMF 呼び出し可能インターフェースを使用するプログラムを変換、コンパイル、およびリンク・エディットするときには、以下の条件について考慮してください。

- 連絡域 (DSQCOMML) がコンパイル・ステップで使用可能であるか、それをプログラムにコピーする必要があります。
- プログラムのリンク・エディットのステップで QMF インターフェース・モジュール DSQCIPL および DSQCIPX が使用可能でなければなりません。

次の例では、CICS から提供されるプロシージャー DFHVITVL を使用します。この JCL は、QMF では提供されませんが、コピーして、要件に合うように変更できます。

```
//samPLI JOB
// EXEC PROC=DFHVITVL
//PLI.SYSIN DD *
.
.
.
Your program or copy of QMF sample DSQABFP
.
.
.
/*
/* Provide access to QMF communications macro DSQCOMML
//PLI.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Provide access to QMF interface module
/* Allocation for target library
//LKED.SYSLMOD DD
/* Allocation for QMF load library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCIPL)
INCLUDE QMFLOAD(DSQCIPX)
MODE AMODE(31) RMODE(31)
NAME sampPLI(R)
/*
```

TSO でのプログラムの実行

TSO で PL/I プログラムを実行するには、プログラムをコンパイルし、リンク・エディットしてから、プログラムを実行します。その際、ISPF を使用するかどうかは任意です。

TSO のコンパイルおよびリンク・エディット

PL/I プログラムは、プログラムをコンパイルし、リンク・エディットしてからでないと、TSO で実行できません。

次の JCL では、PL/I コンパイラーを使用して、呼び出し可能インターフェース・アプリケーションをコンパイルしてから、アプリケーションをリンク・エディットします。いくつかのパラメーターは、QMF インストール先によって異なることがあります。

```
//samPLI JOB
//STEP1 EXEC PROC=IBMZCPL
/* Provide Access to QMF Communications Macro DSQCOMML
//PLI.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
```

```

//PLI.SYSIN DD *
.
.
.
Your program or copy of QMF sample DSQABFP
.
.
.
/* Allocation for target library
//LKED.SYSLMOD DD
/* Allocation for QMF load library
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCIPL)
INCLUDE QMFLOAD(DSQCIPX)
ENTRY CEESTART
MODE AMODE(31) RMODE(ANY)
NAME sampPLI(R)
/*

```

ISPF を使用しない TSO での実行

PL/I プログラムのアセンブルが正常に終了したら、ISPF なしでそのプログラムを実行することができます。

ISPF を使用しない TSO でプログラムを実行するには、次に示す CLIST のようなプログラムを作成します。

```

PROC 0
CONTROL ASIS
/*****/
/* Note: QMF, DB2, GDDM and PL/I load libraries must be */
/* allocated before running this CLIST. */
/* Name of QMF load library is "QMF1210.SDSQLOAD". */
/*****/
/* Specify attribute list for dataset allocations */
/*****/
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****/
/* Datasets used by TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGMGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADM SYMBOL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Datasets used by QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/

```

```

/* Start your program using TSO CALL command */
/*****
CALL sampPLI
EXIT CODE(0)

```

ISPF のもとの TSO での実行

PL/I プログラムのアセンブルが正常に終了したら、ISPF のもとでそのプログラムを実行することができます。

ISPF のもとの TSO でプログラムを実行するには、次に示す CLIST のようなプログラムを作成します。

```

PROC 0
CONTROL ASIS
/*****
/* Specify attribute list for dataset allocations */
/*****
ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)
ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)
ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)
ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)
/*****
/* Datasets used by TSO */
/*****
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****
/* Datasets used by ISPF */
/*****
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PLI.PLILINK','PLI.SIBMLINK')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* QMF/GDDM Datasets */
/*****
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****
/* Datasets used by QMF */
/*****
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****
/* Start your program as the initial ISPF dialog */
/*****
ISPSTART PGM(sampPLI) NEWAPPL(DSQE)
EXIT CODE(4)

```

EXIT CODE(4) は ISPF 後処理パネルを抑止します。

REXX 言語インターフェース

REXX は解釈言語なので、コンパイルする必要がありません。

ただし、コンパイルした REXX またはコンパイルした他の言語を使用するプログラムは、解釈 REXX を使用するように作成された同様のプログラムよりパフォーマンスが向上します。REXX コンパイラーは REXX プログラムに使用可能ですが、ロジックを持つプロシージャーには使用できません。

ロジックを持つプロシージャーおよび特定の QMF 書式関数 (計算、定義済みの列、および条件) を使用するときには、TSO 下で REXX 呼び出し可能インターフェースを使用して QMF を開始します。REXX 呼び出し可能インターフェースで REXX サービスを使用するために必要なリソースを削減できます。

例えば、REXX 呼び出し可能インターフェースを使用して QMF が開始された場合は、REPORT パネルで PRINT REPORT コマンドまたは BOTTOM コマンドを発行するために必要となるリソースを削減できます。リソース消費の削減は、かなりの量になり、TSO のもとで QMF を実行する場合に最も顕著になります。

REXX 言語は、コマンドを処理する方法と場所を決定するコマンド環境において常に機能します。QMF コマンドを出す REXX プログラムを作成する場合、ADDRESS QRW コマンドを介して QMF コマンド環境を使用することができます。

制約事項: REXX は、QMF for CICS では使用できないため、REXX 用の QMF 呼び出し可能インターフェースは、CICS の下では機能しません。

関連概念:

31 ページの『第 5 章 ADDRESS QRW および QMF コマンド環境』
QMF を TSO、ISPF、またはネイティブ z/OS で開始すると、QMF は QRW と呼ばれる REXX コマンド環境を作成します。REXX プログラムを実行している場合は、REXX ADDRESS コマンド ADDRESS QRW を発行することで、デフォルトのコマンド環境を QRW に設定できます。ADDRESS QRW を使用した場合、別の ADDRESS コマンドを出すまで、QMF はデフォルトのコマンド環境にとどまります。

REXX 用のインターフェース連絡変数

インターフェース連絡変数は、複数の REXX 変数で構成されます。これらの変数は、各呼び出しの完了後に設定され、呼び出し側プログラムで変更してはいけません。

REXX 変数のインターフェース連絡変数を次の表に示します。

表 49. REXX 用のインターフェース連絡変数

構造名	説明
dsq_return_code	<p>QMF コマンドの実行結果を示す整数</p> <p>値は以下のとおりです。</p> <p>dsq_success 要求が正常に処理された</p> <p>dsq_warning 警告を伴って正常に終了</p> <p>dsq_failure コマンドが正しく処理されなかった</p> <p>dsq_severe QMF セッションを強制終了させる重大エラー QMF への追加の呼び出しを、このインスタンス ID を使用して実行することはできない。 dsq_return_code の値は、REXX 変数 rc にも置かれる。</p>
dsq_instance_id	START コマンドの処理時に QMF によって設定される ID。
dsq_product	<p>使用中の IBM 照会プロダクトを識別する</p> <p>先頭に dsq_qmf が付く変数は、QMF for TSO バージョンを指定します。</p>
dsq_product_release	<p>使用中の QMF のリリース・レベル</p> <p>変数 dsq_qmf_v12r1 は QMF バージョン 12 リリース 1 を指定します。</p>
dsq_message_id	完了メッセージ ID
dsq_q_message_id	照会メッセージ ID
dsq_start_parm_error	パラメーター・エラーのために START が失敗したときの、エラーがあるパラメーター
dsq_cancel_ind	<p>QMF がコマンドを実行している間に、ユーザーがコマンド処理を取り消したかどうかを示すコマンド取り消し標識。指定可能な値は以下のとおり。</p> <p>dsq_cancel_yes ユーザーがコマンドを取り消した。</p> <p>dsq_cancel_no ユーザーがコマンドを取り消さなかった。</p>
dsq_message_text	完了メッセージ・テキスト
dsq_q_message_text	照会メッセージ・テキスト

REXX 用の関数呼び出し

呼び出し可能インターフェースは、通常の REXX 関数呼び出しを使用してアクセスします。QMF には、呼び出し可能インターフェースを介して出されるすべての QMF コマンドを実行するために使用される、DSQCIX と呼ばれる外部サブルーチンが用意されています。

DSQCIX 関数呼び出しの構文は次のとおりです。

```
call DSQCIX cmd parmlist
```

この構文の *cmd* は大文字ストリングで記述した QMF コマンド、*parmlist* はパラメーターと値のペアのリストです。

DSQCIX 関数呼び出しのパラメーター・リストの構文



parmlist を含むコマンド全体を、文字ストリングで記述した単一の REXX 変数として QMF に渡します。このストリングは、引用符 (' ') または (" ") で囲む必要があります。コマンド・ストリングの一部として REXX 変数を使用する場合には、引数を囲まないでください。例えば、次のようにします。

```
CALL DSQCIX "RUN QUERY NAME (&ECN="REXAUG",CONFIRM=YES)"
```

parmname

パラメーターの名前

value

parmname によって指定したパラメーター名に関連させる値

以下は、関数呼び出しの例です。

```
call DSQCIX "RUN QUERY Q1"  
call DSQCIX "PRINT REPORT (FORM=F1"  
call DSQCIX "EXIT"
```

parmlist での以下のエレメントの有無は、結果に影響を与えません。

- パラメーターとパラメーターの間のコンマ (,) (スペースでも同じ結果を得られます)
- 右括弧 (必要ではありません)
- *parmname* と 値 の間の等号 (=) (スペースでも同じ結果を得られます)

以下の各ステートメントでも、同じ結果が得られます。

```
call dsqcix "SET GLOBAL (abc=17, def=26"  
call dsqcix "SET GLOBAL ( abc=17 def=26"  
call dsqcix "SET GLOBAL ( abc=17 , def=26)"  
call dsqcix "SET GLOBAL (abc 17 def=26)"
```

REXX プログラミングの例

QMF には、サンプルの REXX プログラム DSQABFX が用意されています。

サンプル・ソース・コード・リストをここで見ることも、オンラインでアクセスすることもできます。サンプル・プログラムはライブラリー QMF1210.SDSQEXCE のメンバーです。

REXX 言語呼び出し可能インターフェースのサンプル・プログラムは、以下の機能を実行します。

- QMF を開始する。
- 3 つの大域変数を設定する。
- Q1 と呼ばれる照会を実行する。
- 書式 F1 を使用して結果の報告書を印刷する。
- QMF セッションを終了する。

照会 Q1 または書式 F1 は QMF では提供されませんが、サンプル・プログラムは、これらのオブジェクトを使用しています。

```

/*REXX*****
/* Sample program: DSQABFX                               */
/* REXX version of the QMF callable interface             */
/******

/*****
/* Start a query interface session                         */
/*****

call dsqcix "START (DSQSMODE=INTERACTIVE)"
say dsq_message_id dsq_message_text
if dsq_return_code = dsq_severe then exit dsq_return_code

/*****
/* Set numeric values into query using SET command        */
/*****

call dsqcix "SET GLOBAL (MYVAR01=20,SHORT=40,MYVAR03=84)"
say dsq_message_id dsq_message_text
if dsq_return_code = dsq_severe then exit dsq_return_code

/*****
/* Run a query                                             */
/*****

call dsqcix "RUN QUERY Q1"
say dsq_message_id dsq_message_text
if dsq_return_code = dsq_severe then exit dsq_return_code

/*****
/* Print the results of the query                         */
/*****

call dsqcix "PRINT REPORT (FORM=F1)"
say dsq_message_id dsq_message_text
if dsq_return_code = dsq_severe then exit dsq_return_code

/*****
/* End the query interface session                       */
/*****

call dsqcix "EXIT"
say dsq_message_id dsq_message_text
exit dsq_return_code

```

REXX プログラムの実行

REXX でプログラムを作成したら、それを実行する必要があります。

REXX プログラムを TSO で実行するには、以下のようなプログラムを作成します。

```

/*****/
/* Issue TSO allocates for QMF product */
/*****/
Address TSO

"ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)"
"ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)"
"ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)"
"ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)"
"ALLOC FI(SYSPROC) SHR REUSE ",
  "DA('QMF1210.DSQCLSTE',' ",
    "'DSN.DSNCLIST')"
```

```

"ALLOC FI(SYSEXEC) SHR REUSE ",
  "DA('QMF1210.SDSQEXCE')"
```

```

"ALLOC FI(ISPLLIB) SHR REUSE ",
  "DA('QMF1210.SDSQLOAD',' ",
    "'ADM.GDDM.GDDMLOAD',' ",
    "'DSN.DSNLOAD')"
```

```

"ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE"
"ALLOC FI(DSQPRINT) SYSOUT USING(PRINTDCB)"
"ALLOC FI(SYSPRT) SYSOUT(X) LRECL(132) RECFM(FBA) BLKSIZE(132)"
"ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)"
"ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)"
"ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS"
"ALLOC DDNAME(DSQEDIT) UNIT(SYSDA) NEW USING(EDITDCB)"
"ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE"
"ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE"
"ALLOC FI(ADMCFORM) DA('QMF1210.DSQCHART') SHR REUSE"
"ALLOC FI(DSQUCFRM) DA('QMF1210.DSQUCFRM') SHR REUSE"
"ALLOC FI(ADMGDF) DA('GDDM.ADMGDF') SHR REUSE"
"ALLOC FI(ADMSYMBL) DA('ADM.GDDM.GDDMSYM') SHR REUSE"

/* The beginning of your REXX program ..... */
.
.
.
/* The end of your REXX program ..... */
```

INTERACT ループを使用する REXX の例

対話式セッションでの END コマンドの動作を、対話式 QMF での END の動作方法と同様にすることができます。

通常、呼び出し可能インターフェース・プログラムが INTERACT コマンドを出した場合、ユーザーが END コマンドを出すと、QMF はただちに制御をユーザーのプログラムに戻します。しかし、対話式 QMF によって、ユーザーは END コマンドを出して QMF ホーム・パネルに戻ることができます。2 回目の END コマンドを出すと、QMF セッションが終了します。

対話式 QMF の場合と同様に END コマンドを動作させるには、次の例のロジックをプログラムに追加します。

このプログラムは、処理方法を決定するために *dsq_message_id* を使用しています。これらの値は、リリースによって異なる可能性があります。

このプログラムは、QMF と一緒に配布されません。

```

/*REXX*****/
/* Sample program: Using INTERACT loop */
/*****/
/*****/
/* Start an interactive QMF session */
```



```

/*****/
trace error

parms = "START (DSQSMODE=INTERACTIVE"
call dsqcix parms
if dsq_return_code = dsq_severe then exit dsq_return_code
/*****/
/* SET GLOBAL to show panel IDs */
/*****/
call dsqcix "SET GLOBAL (DSQDC_SHOW_PANID=1"
if dsq_return_code = dsq_severe then exit dsq_return_code
/*****/
/* Issue message */
/*****/
call dsqcix "MESSAGE (TEXT='OK, you may enter a command.')"
if dsq_return_code = dsq_severe then exit dsq_return_code
/*****/
/* INTERACT loop */
/*****/
Continue = "yes"
Do while continue = "yes"
  call DSQCIX "INTERACT"
  Select
    When (dsq_return_code = dsq_severe) Then /* Severe error */
      Continue = "no"
    When (dsq_message_id = "DSQ21869") Then /* END from HOME panel */
      Continue = "no"
    When (dsq_message_id = "DSQ90557") Then /* User issued EXIT */
      Continue = "no"
    Otherwise nop /* OK continue session */
  End
End
/*****/
/* End the session */
/*****/
if dsq_message_id <> "DSQ90557" then /* EXIT not issued */
  call dsqcix "EXIT" /* Issue EXIT */
endif

exit dsq_return_code

```


付録 A. プロダクト・インターフェース・マクロ

次の表に、お客様用の汎用プログラミング・インターフェースとして QMF で提供されるマクロをリストします。

重要: ここに記載したマクロ以外の QMF マクロを、プログラミング・インターフェースとして使用しないでください。

表 50. QMF 関数へのインターフェースを提供するマクロ

用途	マクロ名
プロダクト・インターフェース・マクロ	DSQQMF n このプログラム名の n は各国語 ID です。 英語の場合、この ID は E になります。
呼び出し可能インターフェースのマクロ	<ul style="list-style-type: none"> • アセンブラ <ul style="list-style-type: none"> - DSQCIA - DSQCOMMA • COBOL <ul style="list-style-type: none"> - DSQCIB - DSQCOMMB • C/C++ <ul style="list-style-type: none"> - DSQCIC - DSQCICE - DSQCOMMC • Fortran <ul style="list-style-type: none"> - DSQCIF - DSQCIFE - DSQCOMMF • PL/I <ul style="list-style-type: none"> - DSQCIPL - DSQCIPX - DSQCOMML • REXX <ul style="list-style-type: none"> - DSQCIX
コマンド・インターフェースのマクロ	DSQCCI
QMF 管理プログラム出口ルーチン・インターフェース・マクロ	<ul style="list-style-type: none"> • DXEGOVA • DXEXCBA
QMF ユーザー編集出口ルーチン・マクロ	DXE ECS

関連資料:

5 ページの『各国語機能情報に関する規則』
DB2 QMF では、数種類の異なる言語を使用することができます。それぞれの言語
には、各国語機能 (NLF) が提供されます。

付録 B. QMF 大域変数

QMF には、QMF セッションのさまざまな特性や、QMF コマンド、およびパネル表示を制御するために役立つ、多数の大域変数があります。大域変数を使用すると、プロシージャおよびアプリケーションでの QMF 関数の動作を制御することもできます。

QMF 大域変数の命名規則

QMF に用意されているほとんどの大域変数の命名規則は `DSQcc_XXXXXXXXXXXX` です。`cc` は変数のカテゴリーを示し、`XXXXXXXXXXXX` は最大 12 文字の長さの記述名です。下線文字 (`_`) が `cc` の後に含まれています。

`cc` は、以下のいずれかの ID です。

- AP** プロファイル関連状態情報の変数
- AO** その他の (プロファイル関連でない) 状態情報の変数
- CM** 前のコマンドで生じたメッセージに関する情報の変数
- CP** 表エディターに関する情報の変数
- DC** QMF が画面に情報を表示する方法を制御する変数
- EC** QMF がコマンドとプロシージャを実行する方法を制御する変数
- QC** CONVERT QUERY オプションにより値が生成される変数
- QM** RUN QUERY エラー・メッセージ情報を含む変数
- QW** QMF for Workstation に固有の変数

セッション変数

セッション変数は異なる命名規則に従っています。セッション変数は、`DSQEC_SESSGLV_SAV` 大域変数が 1 または 2 に設定されている場合に、ユーザーが一部のパネル上の一部のフィールドに入力する値を保管する大域変数です。セッション変数の命名規則は次のとおりです。

`DXYnpppp_ln_dd`

各部の説明:

- `n` は各国語 ID です。
- `pppp` はパネル ID の最後の 4 文字です。
- `ln` はフィールドに関連付けられている ID です。
- `dd` はフィールドに関連付けられている ID であり、そのフィールドが別のフィールドに依存する場合にのみ使用されます。

大域変数の値の設定と表示

大域変数に割り当てる値が 55 バイト以下である場合は、SET GLOBAL コマンドを使用して値を割り当ててください。変数が 55 バイトを超える場合は、SHOW GLOBALS コマンドを使用します。

このタスクについて

デフォルトでは、大域変数の値は、リセットするまで、または QMF セッションを終了するまで保持されます。ただし、DSQEC_USERGLV_SAV 大域変数は、あるセッションから別のセッションまで大域変数値を保持するように設定できます。

初期化時に大域変数をカスタマイズするには、「TSO/CICS 用 DB2 QMF インストールおよび管理の手引き」で、QMF 開始時の大域変数および QMF セッションの動作の初期化に関する情報を参照してください。

手順

55 バイトを超える値を大域変数に割り当てるには、以下のようになります。

1. SHOW GLOBALS コマンドを使用して GLOBALS パネルを表示します。
2. **Show Field** キーを押して入力フィールド全体を表示します。「大域変数の表示」画面での大域変数の最大長は 32,768 バイトです。
3. 表示されている行に変数の値を入力します。

プロファイル関連でない状態情報の大域変数

DSQAO 大域変数には、状況情報、またはパラメーターやフラグの設定が格納されます。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

表 51. プロファイル関連でない状態情報の大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_APPL_TRACE	DSQATRAC	01	0 レベル A0 の場合 1 レベル A1 の場合 2 レベル A2 の場合
DSQAO_ATTENTION	DSQCATTN	01	ユーザー・アテンション・フラグ
DSQAO_BATCH	DSQABATC	01	バッチ・モードまたは対話モード。値は以下のとおりです。 1 対話式セッションの場合 2 バッチ・モード・セッションの場合

表 51. プロファイル関連でない状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_CONNECT_ID	DSQAAUTH	128	データベースへの接続に使用するユーザー ID。このユーザー ID を使用して作業します。 この変数の値は、以下のコマンドまたはステートメントを発行したときに変更されます。 <ul style="list-style-type: none"> 別の許可 ID でデータベースに再接続する場合は、QMF CONNECT コマンドを発行する DB2 for z/OS データベースの場合は、SET CURRENT SQLID ステートメントを発行する。
DSQAO_CONNECT_LOC	なし	18	ユーザーが現在接続しているデータベースのロケーション名。名前は 16 文字です (必要に応じて、右端を空白で埋めます)。
DSQAO_CURSOR_OPEN	DSQACRSR	01`	データベース・カーソル状況。値は以下のとおりです。 <ol style="list-style-type: none"> カーソルがオープン カーソルがクローズ
DSQAO_DATE_FORMAT	なし	05	SYSIBM.DATE_FORMAT で指定された値が含まれます。 値は ISO、USA、EUR、JIS、LOCAL のいずれかです。
DSQAO_DB_MANAGER	DSQADBMG	01	次のいずれかの値で示されたデータベース・マネージャー。 <ol style="list-style-type: none"> DB2 for VSE and VM DB2 for z/OS DB2 for Linux, UNIX, and Windows DB2 for iSeries
DSQAO_DBCS	DSQADBCS	01	DBCS サポート状況。値は以下のとおりです。 <ol style="list-style-type: none"> DBCS サポートあり DBCS サポートなし
DSQAO_DSQSBSTG	なし	10	DSQSBSTG パラメーターで指定された値、またはこのパラメーターが指定されなかった場合はデフォルトが含まれます。
DSQAO_DSQSFISO	なし	01	DSQSFISO パラメーターで指定された値、またはこのパラメーターが指定されなかった場合はデフォルトが含まれます。 以下の値が使用されます。 <ol style="list-style-type: none"> 0 QMF は、DATE(ISO) や TIME(ISO) を使用してあらかじめコンパイルされることはありません。 1 QMF は DATE(ISO) および TIME(ISO) でプリコンパイルされています。これがデフォルトです。

表 51. プロファイル関連でない状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_DSQSMRFI	なし	01	このフィールドは、QMF の始動時に DSQSMRFI プログラム・パラメーターに指定された値を反映します。 0 DSQSMRFI プログラム・パラメーターに NO が指定されました。これは DB2 単一行フェッチおよび挿入が使用されることを意味します。 1 DSQSMRFI プログラム・パラメーターに YES が指定されました。これは DB2 複数行フェッチおよび挿入が使用されることを意味します。複数行フェッチでは行セット・カーソルを使用します。
DSQAO_DSQSMTHD	なし	01	DSQSMTHD プログラム・パラメーターで指定された値、またはこのパラメーターが指定されなかった場合はデフォルトが入ります。 以下の値が使用されます。 0 NO が指定されました。QMF は 1 つのスレッドで動作します。これがデフォルトです。 1 YES が指定されました。QMF は、コマンド (RUN QUERY、DISPLAY TABLE)、およびオープン・カーソルによるそれ以降の報告書のスクロール (BOTTOM、TOP、FORWARD、BACKWARD、RIGHT、および LEFT) に使用される第 2 スレッドで動作します。
DSQAO_DSQSPILL	なし	01	DSQSPILL パラメーターで指定された値、またはこのパラメーターが指定されなかった場合はデフォルトが含まれます。 以下の値が使用されます。 0 予備記憶域を使用しません。この値は、DSQSPILL パラメーター値の NO に相当します。 1 予備記憶域を使用します。この値は、DSQSPILL パラメーター値の YES に相当します。
DSQAO_DSQSPTYP	なし	5	DSQSPTYP パラメーターで指定された値、またはこのパラメーターが指定されなかった場合はデフォルトが含まれます。 以下の値が使用されます。 FILE データをファイルにスピルします。 64BIT データを拡張仮想記憶域にスピルします。

表 51. プロファイル関連でない状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_DSQSRSTG	なし	8	DSQSRSTG パラメーターで指定された値、またはこのパラメーターが指定されなかった場合はデフォルトが含まれます。
DSQAO_FORM_PANEL	DSQASUBP	02	現在の書式パネル。値は以下のいずれかになります。 1 FORM.MAIN 2 FORM.COLUMNS 3 FORM.PAGE 4 FORM.FINAL 5 FORM.BREAK1 6 FORM.BREAK2 7 FORM.BREAK3 8 FORM.BREAK4 9 FORM.BREAK5 10 FORM.BREAK6 11 FORM.OPTIONS 12 FORM.CALC 13 FORM.DETAIL 14 FORM.CONDITIONS ブランク値は、QMF 一時記憶域に書式がないことを意味します。
DSQAO_INTERACT	DSQAIACT	01	対話フラグの設定。値は以下のとおりです。 0 対話式実行でない場合 1 対話式実行が許可されている場合
DSQAO_LOCAL_DB2	なし	18	ローカル DB2 for z/OS データベースのロケーション名。 この値は、変数 DSQAO_SUBSYS_ID で名指しされたサブシステムのロケーション名です。リモート作業単位環境では、DSQ_LOCAL_DB2 はアプリケーション・リクエストの名前。名前は 16 文字 (必要な場合は、右端をブランクで埋めます)。
DSQAO_LOCATION	DSQAITLO	18	現オブジェクトのロケーション名 (ある場合)。 3 部構成の名前が使用された場合のみ、名前が入ります。名前は 16 文字 (必要な場合は、右端をブランクで埋めます)。
DSQAO_NLF_LANG	DSQALANG	01	ユーザーの国の言語。英語言語環境の場合、この値は 'E' になります。
DSQAO_NUM_FETCHED	DSQAROWS	16	取り出されたデータ行。DATA オブジェクトが空の場合は '0'。

表 51. プロファイル関連でない状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_OBJ_NAME	DSQAITMN	128	現在表示されているパネル上にある表 (報告書に含まれているもの)、照会、プロシージャ、書式の名前。 現パネル上にどのオブジェクトも表示されていないとき、または表示されているオブジェクトに名前がないときは、この変数にはブランクが入っています。
DSQAO_OBJ_OWNER	DSQAITMO	128	現在表示されているパネル上にある表 (報告書に含まれているもの)、照会、プロシージャ、書式の所有者。 現パネル上にどのオブジェクトも表示されていないとき、または表示されているオブジェクトに所有者がないときは、この変数にはブランクが入っています。
DSQAO_OTC_LICENSE	なし	01	これは、DB2 QMF for z/OS OTC 独立型製品がインストールされているかどうかを示します。 0 これは、DB2 QMF for z/OS OTC 独立型製品の製品 ID (5697-QMF) が見つからなかったことを示します。 1 これは、DB2 QMF for z/OS 独立型製品の製品 ID (5697-QMF) が見つかったことを示します。
DSQAO_PANEL_TYPE	DSQAITEM	01	現行パネルのタイプ。値は以下のとおりです。 1 HOME 2 QUERY 3 REPORT 4 FORM 5 PROC 6 PROFILE 7 CHART 8 LIST 9 表エディター A GLOBALS
DSQAO_QMF_RELEASE	DSQAREVN	02	QMF のリリース番号 (数字)。エクスポートされた書式、報告書、および指示照会のヘッダー・レコードに表示されます。QMF バージョン 12 リリース 1 の場合は、この値が「19」になります。
DSQAO_QMF_VER_RLS	DSQAQMF	10	QMF のバージョンおよびリリース。QMF バージョン 12 リリース 1 の場合は、この値が「QMFV12R1.0」になります。

表 51. プロファイル関連でない状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_QMFADM	なし	01	QMF 管理者権限。 0 現行の許可 ID は QMF 管理者権限を持っていません。 1 現行の許可 ID は管理者権限を持っています。
DSQAO_QRY_SUBTYPE	DSQASUBI	01	照会サブタイプ。値は以下のとおりです。 1 SQL サブタイプ 2 QBE サブタイプ 3 PROMPTED サブタイプ ブランクは、現パネルが QUERY でないことを意味します。
DSQAO_QUERY_MODEL	DSQAMODL	01	現行の照会のモデル。指定できる値は「1」(リレーショナル・データ・モデル)のみです。
DSQAO_SAME_CMD	DSQACMDM	01	値は以下のとおりです。 0 2 つのコマンドが同じでない場合 1 2 つのコマンドが同じである場合
DSQAO_STO_PROC_INT	なし	01	QMF for TSO が DB2 for z/OS ストアード・プロシージャーとして開始されたかどうかを示します。値は以下のとおりです。 0 QMF は、ストアード・プロシージャーとして開始されていません。 1 QMF は、ストアード・プロシージャーとして開始されています。
DSQAO_SUBSYS_ID	なし	04	QMF が TSO で稼働している場合は、この値は QMF の接続先のローカル DB2 サブシステムの ID です。 CICS の中で DSQSSUBS プログラム・パラメーターの値を指定すると、この大域変数はその値を含みます。このパラメーターは黙認され、値は処理されません。値は大域変数フィールドに配置され、処理は行われません。この論理によって、同一のプログラムを複数の環境で使用することが可能になります。
DSQAO_SYSTEM_ID	DSQASYST	01	現行のオペレーティング・システム。値は以下のいずれかになります。 2 z/OS 下の TSO 3 TSO またはネイティブ z/OS 5 CICS
DSQAO_TERMINATE	DSQCSESC	01	QMF 終了フラグ。値は以下のとおりです。 0 セッションがマークされていない場合 1 セッションがマークされていた場合

表 51. プロファイル関連でない状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAO_TIME_FORMAT	なし	05	SYSIBM.TIME_FORMAT で指定された値が含まれます。 値は ISO、USA、EUR、JIS、LOCAL のいずれかです。
DSQAO_VARIATION	DSQAVARN	02	書式パネル・バリエーション番号。ブランクは、FORM.DETAIL が現行のパネルではないことを意味します。

プロファイル関連状態情報の大域変数

DSQAP 大域変数は、QMF プロファイル設定に関連する情報を保管します。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

表 52. プロファイル関連状態情報の大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAP_CASE	DSQAPCAS	01	CASE パラメーター。値は以下のとおりです。 <ol style="list-style-type: none"> 1 UPPER の場合 2 MIXED の場合 3 STRING の場合 <p>ご使用のサイトで、TSO 下の大/小文字混合パスワードの RACF サポートを使用している場合は、この値を 2 に設定してください。この設定を行わないと、すべての入力 (パスワードを含む) が大文字に変換されるため、CONNECT コマンドが失敗します。CASE を MIXED に設定する場合、入力はすべて大文字で行うようにしてください。これは、QMF では大文字のコマンドのみが認識されるためです。</p>
DSQAP_CONFIRM	DSQAPRMP	01	CONFIRM パラメーター。値は以下のとおりです。 <ol style="list-style-type: none"> 0 NO の場合 1 YES の場合
DSQAP_DECIMAL	DSQAPDEC	01	DECIMAL パラメーター。値は以下のとおりです。 <ol style="list-style-type: none"> 1 PERIOD の場合 2 COMMA の場合 3 FRENCH の場合
DSQAP_LENGTH	DSQAPLEN	18	LENGTH パラメーター。値はパラメーターの値です ('1' から '999' までか、'CONT')。
DSQAP_PFKKEY_TABLE	DSQAPPFK	31	ファンクション・キー表の名前

表 52. プロファイル関連状態情報の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAP_PRINTER	DSQAPPRT	08	PRINTER パラメーター。値は以下のとおりです。 <ul style="list-style-type: none"> • GDDM プリンターのニックネーム • ブランク (DSQPRINT と関連づけられたプリンター)
DSQAP_QUERY_LANG	DSQAPLNG	01	LANGUAGE パラメーター。値は以下のいずれかになります。 1 SQL 2 QBE 3 PROMPTED の場合
DSQAP_QUERY_MODEL	DSQAMODP	01	MODEL パラメーター (リレーショナルの場合、値は '1')。
DSQAP_RESOURCE_GRP	DSQAPGRP	16	RESOURCE GROUP パラメーター。
DSQAP_SPACE	DSQAPSPC	50	SPACE パラメーター。値はパラメーターの値です。
DSQAP_SYNONYM_TBL	DSQAPSYN	31	現行の QMF セッションに使用される同義語表の名前。ユーザーがコマンド同義語を入力するときには、ここに示す表に同義語定義を保管しておかないと、コマンドが失敗します。
DSQAP_TRACE	DSQAPTRC	18	TRACE パラメーター。値は以下のとおりです。 ALL (最大トレース) NONE (最小トレース) トレース対象のコンポーネントとトレース内容の詳細レベルを示す一連の文字と数字を指定することもできます (例えば、A2L2C1)。
DSQAP_WIDTH	DSQAPWID	18	WIDTH パラメーター。値はパラメーターの値です ('22' から '999')。

CICS 関連の大域変数

DSQAP 大域変数は、CICS 環境に関連付けられています。SET GLOBAL コマンドで変更できるのは、DSQAP_CICS_PQNAME および DSQAP_CICS_PQTYPE のみです。

キュー・タイプが一時データ (TD) であれば、対応するキュー名の最大長は 4 です。例えば、DSQAO_CICS_SQTYPE が TD であれば、DSQAO_CICS_SQNAME の最大長は 4 です。

表 53. CICS 環境関連の大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQAP_CICS_PQNAME	なし	08	QMF 印刷出力を入れる CICS データ・キューの名前。
DSQAP_CICS_PQTYPE			QMF 印刷出力を入れる CICS 記憶域のタイプ。 TS QMF 印刷を、補助記憶装置上の CICS 一時記憶域キューに書き出します。この値がデフォルトです。 TD QMF 印刷を CICS 一時データ・キューに書き出します。
DSQAO_CICS_SQNAME	なし	08	予備ファイルに使用する CICS データ・キューの名前。
DSQAO_CICS_SQTYPE	なし	02	QMF 予備ファイルを入れる CICS 記憶域のタイプ。 TS QMF 予備データを、補助記憶装置上の CICS 一時記憶域キューに書き出します。この値がデフォルトです。 TD QMF 予備データを CICS 一時データ・キューに書き出します。
DSQAO_CICS_TQNAME	なし	08	QMF トレース・データを入れる CICS データ・キューの名前。
DSQAO_CICS_TQTYPE	なし	02	QMF トレース・データを入れる CICS 記憶域のタイプ。 TS QMF トレースを、補助記憶装置上の CICS 一時記憶域キューに書き出します。 TD QMF トレースを CICS 一時データ・キューに書き出します。この値がデフォルトです。

最後に発行されたコマンドで生成されたメッセージに関連する大域変数

DSQCM 大域変数には、最後に発行された QMF コマンドに関する情報が格納されます。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

表 54. 最後に発行されたコマンドに関する情報を取り込む大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQCM_MESSAGE	DSQCM_MESSAGE	80	メッセージ・テキスト
DSQCM_MESSAGE_ALL	DSQCIMSA	360	完全メッセージ・テキスト
DSQCM_MSG_HELP	DSQCIMID	08	メッセージ・ヘルプ・パネルの ID
DSQCM_MSG_NUMBER	DSQCIMNO	08	メッセージ番号。
DSQCM_SUB_TXT_ <i>nn</i>	DSQCIM <i>nn</i>	20	置換値 <i>nn</i>

表エディター関連の大域変数

DSQCP 大域変数は、表エディターの操作に関連付けられています。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

次の表に、表エディターの操作に関連付けられている大域変数を示します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

EDIT TABLE の CONFIRM オプションが NO の場合、表エディターは確認パネルの表示をすべて抑制します。CONFIRM オプションが YES なら、表エディターはこの表に示す大域変数の値を調べ、どのカテゴリーの確認が有効になっているか判断します。

表エディターのデフォルト値は、EDIT TABLE コマンドの SAVE キーワードに応じて次のように異なります。

- SAVE=IMMEDIATE なら、各カテゴリーのデフォルト値が有効です。
- SAVE=END なら、DELETE、MODIFY、および END/CANCEL カテゴリーのデフォルト値が有効です。ADD カテゴリーと CHANGE カテゴリーのデフォルト値は無効です。

表 55. 表エディター関連の大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQCP_RMV_BLANKS	なし	01	<p>VARCHAR 列の末尾ブランクを保持または削除します。この変数は、変更モードの表エディターにのみ作用します。値は以下のとおりです。</p> <p>0 VARCHAR 列の末尾ブランクは削除されません。</p> <p>1 VARCHAR 列の末尾ブランクは削除されます。この値がデフォルトです。</p>
DSQCP_TEADD	なし	01	<p>ADD サブコマンドの後で、確認パネルを表示するかどうか。値は以下のとおりです。</p> <p>0 - パネルを表示しません。</p> <p>1 - パネルを表示します。</p> <p>2 パネルを表示するかどうかは、表エディターのデフォルト値によります。この値がデフォルトです。</p>
DSQCP_TECHG	なし	01	<p>CHANGE サブコマンドの後で、確認パネルを表示するかどうか。値は以下のとおりです。</p> <p>0 - パネルを表示しません。</p> <p>1 - パネルを表示します。</p> <p>2 パネルを表示するかどうかは、表エディターのデフォルト値によります。この値がデフォルトです。</p>

表 55. 表エディター関連の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQCP_TEDEL	なし	01	DELETE サブコマンドの後で、確認パネルを表示するかどうか。値は以下のとおりです。 0 - パネルを表示しません。 1 - パネルを表示します。 2 パネルを表示するかどうかは、表エディターのデフォルト値によります。この値がデフォルトです。
DSQCP_TEDFLT	なし	01	表エディターで列のデフォルト値を示すために使用される予約文字。最初は、正符号 (+) 文字に設定されています。
DSQCP_TEDFLT_DBCS	なし	04	表エディターで漢字ストリング列のデフォルト値を示すのに使用される DBCS の予約文字。 この値は、4 バイトの混合ストリングで、その前にシフトアウト文字を伴い、その後シフトイン文字を伴う 1 文字の DBCS 文字で構成されます。最初は、DBCS の正符号 (+) 文字に設定されています。この大域変数は DBCS 環境でのみ使用されます。
DSQCP_TEEND	なし	01	END サブコマンドまたは CANCEL サブコマンドを発行して表エディター・サブセッションを終了させるときに、確認パネルを表示するかどうかを示します。 パネルは、以下のように複数のバリエーションで表示できます。 <ul style="list-style-type: none"> • END または CANCEL が発行された場合 • データベースが変更された場合 • END または CANCEL の発行時に、画面に変更されたデータが含まれている場合 値は以下のとおりです。 0 - パネルを表示しません。 1 - パネルを表示します。 2 パネルを表示するかどうかは、表エディターのデフォルト値によります。この値がデフォルトです。
DSQCP_TEMOD	なし	01	表示されたデータが変更され、かつ PREVIOUS、CLEAR、SHOW CHANGE、SHOW SEARCH、REFRESH、または NEXT サブコマンドが実行されたときに、確認パネルを表示するかどうかを示します。結果のパネルには、パネル・テキストの一部として、サブコマンドの名前が表示されます。値は以下のとおりです。 0 - パネルを表示しません。 1 - パネルを表示します。 2 パネルを表示するかどうかは、表エディターのデフォルト値によります。

表 55. 表エディター関連の大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQCP_TENULL	なし	01	表エディターで列の NULL 値を示すために使用される予約文字。最初は、ハイフン (-) 文字に設定されています。
DSQCP_TENULL_DBCS	なし	04	表エディターで漢字ストリング列の NULL 値を示すために使用される DBCS の予約文字。この文字は、検索基準のコンテキストで無視する内容を示すためにも使用されます。 この値は、4 バイトの混合ストリングで、その前にシフトアウト文字を伴い、その後シフトイン文字を伴う 1 文字の DBCS 文字で構成されます。最初は DBCS のハイフン (-) 文字に設定されています。この大域変数は DBCS 環境でのみ使用されます。

さまざまな表示を制御する大域変数

DSQDC 大域変数は、特定の種類の情報の表示を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

表 56. 特定の種類の情報の表示を制御する大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_COL_LABELS	なし	01	FORM.MAIN および FORM.COLUMNS に示される列見出しのデフォルト値が、列に割り当てられたデータベースのラベルとなるか、選択元の表の列名となるかを制御します。 0 デフォルトの QMF 書式の列見出しとして列名が使用されます。 1 デフォルトの QMF 書式の列見出しとしてデータベース・ラベルが使用されます。この値はデフォルト値です。
DSQDC_COST_EST	なし	01	データベースのコスト見積の表示を制御します。値は以下のとおりです。 0 コスト見積もりを表示しません。 1 コスト見積もりを表示します。この値がデフォルトです。 2 データベース状況パネルおよびコスト見積パネルを表示しません。

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_CURRENCY	なし	18	<p>DC 編集コードの指定がある場合に使用される通貨記号。</p> <p>この値は、1 から 18 バイトまれの長さのストリングです。英語の場合、デフォルト値はユーロ通貨記号です。デフォルト値は言語によって異なります。DBCS 環境では、この値は SBCS と DBCS 文字の混合ストリングにできます。シフトアウト文字とシフトイン文字を含む混合ストリングの合計の長さは、18 バイトを超えることはできません。</p> <p>キーボード上にない通貨記号が必要な場合でも、その記号を指定できます。ロジックを持つプロシージャに含まれる DSQDC_CURRENCY 変数を、正しい記号に相当する 16 進数値に設定します。例えば、次のプロシージャでは、通貨記号を HEX '9F' に設定します。この値は、英語版 QMF ではユーロ通貨を示します。</p> <pre>/* */ "SET GLOBAL (DSQDC_CURRENCY = " '9F'X</pre> <p>通貨記号に末尾空白が必要な場合は、通貨記号を単一引用符で囲みます。次の例は、フランス語の QMF 用の空白を示しています。</p> <pre>SET GLOBAL (DSQDC_CURRENCY = 'FR '</pre> <p>このコマンドは、コマンド行または線形プロシージャのいずれかから発行できます。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの 変数名	コマンド・インターフェースの 変数名	長さ	説明
DSQDC_DISPLAY_RPT	DSQADPAN	01	<p>RUN QUERY の後で報告書を表示するかどうかを示します。値は以下のとおりです。</p> <p>0 QMF は、RUN QUERY コマンドによる結果の報告書を表示しません。</p> <p>この値は、QMF が DSQQMFn (n は各国語機能 ID) を使用して対話式に開始された場合のデフォルトです。QMF がバッチ・モードで開始された場合も、この値がデフォルトです。QMF がバッチ・モードで起動される場合は、この変数を変更しても、いずれの QMF 画面も表示されません。</p> <p>1 QMF は、報告書を自動的に表示します。</p> <p>QMF が呼び出し可能インターフェースで起動された場合は、この値がデフォルト値です。この値は、START コマンドの DSQADPAN プログラム・パラメータによって指定変更されることがあります。</p> <p>この大域変数を 1 に設定すると、表示されるレポートを確認し、変更をコミットするかロールバックするかを選択できます。これを行うには、変更を確認した後で F3 (END) を押してください。その後、変更をコミットするかロールバックするかを選択するようプロンプトが出されます。変更をコミットするには 1 を、ロールバックするには 2 を選択し、Enter を押してください。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_EC_CHAR	なし	05	<p>文字データ (固定長文字、可変長文字、および非常に長い文字) のユーザー定義デフォルト編集コード。</p> <p>C データの表示を変更しません。これがデフォルトです。</p> <p>CW 列幅の境界でデータを折り返します。</p> <p>CT 列幅の境界でデータを折り返します。その際、空白が含まれていれば、境界より内側の最も近い位置にある空白で改行が行われます。</p> <p>CDx データが 1 行に収まらない場合に、指定した区切り文字 (x) に従って列データを折り返します。区切り文字には、ブランクを含め任意の文字を使用できます。区切り文字自体は、出力には現われません。</p> <p>Uxxxx ユーザー定義の形式設定。xxxx を 0 個から 4 個までの文字 (英字、数字、または特殊文字) に置き換えます。</p> <p>Vxxxx ユーザー定義の形式設定。xxxx を 0 個から 4 個までの文字 (英字、数字、または特殊文字) に置き換えます。</p> <p>B 2 進形式。</p> <p>BW 列幅の境界で列が折り返される 2 進形式。</p> <p>X 16 進形式。</p> <p>XW 列幅の境界で列が折り返される 16 進形式。</p> <p>M 実際のデータではなく、メタデータ (データ・タイプと長さ) を表示します。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_EC_DATE	なし	05	<p>日付データのデフォルト編集コード。値は以下のとおりです。</p> <p>TDYx 年の部分は 4 桁であり、年が最初に記述されます。</p> <p>TDMx 年の部分は 4 桁であり、月が最初に記述されます。</p> <p>TDDx 年の部分は 4 桁であり、日が最初に記述されます。</p> <p>TDYAx 年の部分は下 2 桁であり、年が最初に記述されます。</p> <p>TDYMx 年の部分は下 2 桁であり、月が最初に記述されます。</p> <p>TDDAx 年の部分は下 2 桁であり、日が最初に記述されます。</p> <p>TDL ローカルで定義された日付形式。</p> <p>TD データベース・システムのデフォルト日付形式。これが、この大域変数のデフォルト値です。</p> <p>x は、日付の各部分の区切り文字としてユーザーが指定する文字を表します。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_EC_DEC	なし	05	<p>10 進データのユーザー定義デフォルト編集コード。</p> <p>E または EZ 浮動小数。編集コードの 2 番目の位置にある Z は、ゼロ値を抑止します。</p> <p>D、DC、DZ、DZC、I、IZ、J、JZ、K、KZ、L、LZ、P、PZ 先行ゼロの有無、負数の負符号、3 桁単位の区切り記号、通貨記号、パーセント記号をさまざまに組み合わせた 10 進表記。</p> <p>各コードの後には、小数点の後に許可する桁数を示す値 (0 から 99) を付けることができます。</p> <p>D 編集コードの 2 番目または 3 番目の位置にある C は、標準の通貨記号の代わりに、ユーザー定義の通貨記号を表示します。</p> <p>編集コードの 2 番目の位置にある Z は、ゼロ値を抑止します。</p> <p>デフォルト値は L です。L* が指定された場合、QMF はデータベースの列定義に基づいて 10 進データを形式設定します。この動作には、以前のリリースの QMF との整合性があります。</p> <p>Uxxxx ユーザー定義の形式設定。xxxx を 0 文字から 4 文字 (文字、数字、または特殊文字) に置き換えます。</p> <p>Vxxxx ユーザー定義の形式設定。xxxx を 0 文字から 4 文字 (文字、数字、または特殊文字) に置き換えます。</p> <p>M 実際のデータではなく、メタデータ (データ・タイプと長さ) を表示します。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_EC_NUM	なし	05	<p>数値データ (整数、短精度整数、および 64 ビット整数) のユーザー定義デフォルト編集コード。</p> <p>E または EZ 浮動小数。編集コードの 2 番目の位置にある Z は、ゼロ値を抑止します。</p> <p>D、DC、DZ、DZC、I、IZ、J、JZ、K、KZ、L、LZ、P、PZ 先行ゼロの有無、負数の負符号、3 桁単位の区切り記号、通貨記号、パーセント記号をさまざまに組み合わせた 10 進表記。</p> <p>D 編集コードの 2 番目または 3 番目の位置にある C は、標準の通貨記号の代わりに、ユーザー定義の通貨記号を表示します。</p> <p>編集コードの 2 番目の位置にある Z は、ゼロ値を抑止します。</p> <p>デフォルト値は L です。</p> <p>Uxxxx ユーザー定義の形式設定。xxxx を 0 文字から 4 文字 (文字、数字、または特殊文字) に置き換えます。</p> <p>Vxxxx ユーザー定義の形式設定。xxxx を 0 文字から 4 文字 (文字、数字、または特殊文字) に置き換えます。</p> <p>M 実際のデータではなく、メタデータ (データ・タイプと長さ) を表示します。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_EC_TIME	なし	05	<p>時刻データのデフォルト編集コード。値は以下のとおりです。</p> <p>TTSx 24 時間クロック形式 (秒を含む)。</p> <p>TTCx 12 時間クロック形式 (秒を含む)。</p> <p>TTAx 簡略クロック形式 (秒なし)。</p> <p>TTAN 簡略クロック形式 (秒なし、区切り文字なし)。</p> <p>TTUx USA 形式。</p> <p>TTL ローカルに定義された時刻形式。</p> <p>TT データベース・システムのデフォルト時刻形式。これが、この大域変数のデフォルト値です。</p> <p><i>x</i> は、時刻の各部分の区切り文字としてユーザーが指定する文字を表します。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_LIST_ORDER	なし	02	<p>データベース・オブジェクト・リスト内のオブジェクトに、デフォルトのソート順序を設定します。最初の文字には、以下の値を使用できます。</p> <ul style="list-style-type: none"> 1 リストはデフォルトの順を使用します。 2 リストはオブジェクト所有者によりソートされます。 3 リストはオブジェクト名によりソートされます。 4 リストはオブジェクト・タイプによりソートされます。 5 リストは変更日付によりソートされます。 6 リストは最後に使用された日付によりソートされます。この日付を更新させるコマンドのリストは、DSQEC_LAST_RUN 大域変数によって設定されます。 <p>2 番目の文字の値:</p> <ul style="list-style-type: none"> A リストは昇順にソートされます。 D リストは降順にソートされます。 <p>この変数は、LIST コマンドの結果としてリストされるオブジェクトについてのみ適用されます。「表示プロンプト」パネルなどの他の方法で作られたリストには適用されません。また、表のリストにも適用されません。</p>
DSQDC_POS_SQLCODE	なし	01	<p>データベースから正の SQL コードが戻された場合に QMF が取るアクションを設定します。値は以下のとおりです。</p> <ul style="list-style-type: none"> 0 トレース・データ・ファイル (DSQDEBUG) にメッセージを記録せず、ヘルプ・テキストも提供しません。 1 SQL コードに関連付けられた QMF メッセージをトレース・データ・ファイル (DSQDEBUG) に記録します。 2 QMF メッセージ・ヘルプは、正の SQL コードに使用できます。 <p>この大域変数は、SQL コード +495 および +100 には適用されません。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_SCROLL_AMT	なし	04	<p>QMF パネルのスクロール量を設定します。以下の値を設定できます。</p> <p>Csr スクロール量を「カーソル」に設定します。</p> <p>後方にスクロールすると、QMF は、カーソルがある行または列をスクロール可能域の一番下にスクロールします。同様に、前方にスクロールすると、QMF は一番上にスクロールします。左または右にスクロールすると、左端または右端にスクロールします。</p> <p>Half スクロール量をスクロール可能域の半分に設定します。</p> <p>Page スクロール移動量を全ページに設定します。この値がデフォルトです。</p> <p>n スクロール量を <i>n</i> 行数または列数に設定します。<i>n</i> には 1 から 9999 までの任意の値を指定できます。</p>

表 56. 特定の種類の情報の表示を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQDC_SHORT_EXPT	なし	01	<p>EXPORT コマンドの DATAFORMAT パラメーターに QMF の値を指定してエクスポートされた、データまたは表に適用されます。ヘッダー・レコードに含まれるすべての列名フィールドの長さを制御します。値は以下のとおりです。</p> <p>0 QMF は、ヘッダー・レコードの列フィールドの長さを 30 バイトに設定します。この長さは、以下のバージョンについてはデフォルトの長さです。</p> <ul style="list-style-type: none"> DB2 for z/OS バージョン 8.1.5 以降 DB2 for iSeries バージョン 5.2 以降 DB2 for Linux, UNIX, and Windows バージョン 8.1 以降 <p>1 QMF は、ヘッダー・レコードの列フィールドの長さを 18 バイトに設定します。この長さは、以下のバージョンについてはデフォルトの長さです。</p> <ul style="list-style-type: none"> DB2 for z/OS バージョン 8.1.5 以前 DB2 for iSeries バージョン 5.2 以前 DB2 for Linux, UNIX, and Windows バージョン 8.1 以前 すべての DB2 Server (VSE および VM 版) データベース
DSQDC_SHOW_PANID	DSQCPDSP	01	<p>QMF パネルのパネル ID を表示するかどうか。値は以下のとおりです。</p> <p>0 パネル ID を表示しません。この値がデフォルトです。</p> <p>1 パネル ID を表示します。</p>

関連資料:

240 ページの『コマンドとプロシージャの実行を制御する大域変数』
 DSQEC 大域変数は、コマンドおよびプロシージャの実行方法を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

コマンドとプロシージャの実行を制御する大域変数

DSQEC 大域変数は、コマンドおよびプロシージャの実行方法を制御します。これらの大域変数はいずれも SET GLOBAL コマンドで変更できます。

表 57. コマンドとプロシージャの実行を制御する大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_ALIASES	なし	31	DB2 for z/OS のロケーションから表のリストを要求したときに、表および視点の別名のリストを検索するための視点。現行サーバーが DB2 for z/OS または DB2 for Linux, UNIX, and Windows である場合にも該当します。
DSQEC_BUFFER_SIZE	なし	03	データをデータベースから取り出すために使用されるデータ・バッファの長さを設定します。有効な値は 4 から 256 (整数はそれぞれ 1KB で、例えば 4 は 4K、256 は 256K) です。デフォルト値は 4 (4KB) です。
DSQEC_CC	なし	01	紙送り制御文字が報告書出力形式に出力されないようにします。以下の値を指定できます。 0 紙送り制御文字を 1 桁目に出力しません。 1 紙送り制御文字が有効です。報告書の 1 桁目に紙送り制御文字が出力されます。
DSQEC_COLS_LDB2	なし	31	現行ロケーションが DB2 for z/OS のとき、そこにある表の列情報を検索するための視点。
DSQEC_COLS_RDB2	なし	31	リモート DB2 for z/OS ロケーションが現行ロケーションでないとき、そこにある表の列情報を検索するための視点。
DSQEC_COLS_SQL	なし	31	DB2 for VSE and VM データベース内の表の列情報を検索するための視点。
DSQEC_CON_ACC_RES	なし	01	QMF から DB2 for z/OS へ実行要求される、実行可能な SELECT 照会に適用されます。この変数は、選択するデータが挿入操作、更新操作、または削除操作でロックされているときのデータベースの処理方法を指定するために使用します。この変数を設定すると、QMF は、SELECT 照会の PREPARE ステートメントの concurrent-access-resolution 属性で変数値に関連付けられた文節を指定します。実行可能な SELECT 照会は、QMF 照会 (SQL SELECT 照会、指示照会、QBE P. 照会など) からだけでなく、他の QMF 操作 (DISPLAY TABLE など) からのものも含む可能性があります。 値は以下のとおりです。 0 QMF は、処理中の SQL SELECT ステートメントに関連付けられた PREPARE ステートメントでは、同時アクセス解決のオプションを指定しません。この値がデフォルトです。 1 SKIP LOCKED DATA この値は、DB2 for z/OS バージョン 9 (新機能モード) 以降に対する実行可能 SELECT ステートメントに指定できます。 2 USE CURRENTLY COMMITTED この値は、DB2 for z/OS バージョン 10 (新機能モード) 以降に対する実行可能 SELECT ステートメントに指定できます。 3 WAIT FOR OUTCOME この値は、DB2 for z/OS バージョン 10 (新機能モード) 以降に対する実行可能 SELECT ステートメントに指定できます。
DSQEC_CURR_FOLDER	なし	128	フォルダー処理 (SAVE、LIST、および ERASE) を行うことができる QMF コマンドに使用する現行フォルダーの名前を指定します。デフォルトはブランクです。 フォルダー名がこの大域変数で識別されると、QMF フォルダー・オブジェクトを使用する QMF コマンドの処理時に、そのフォルダーが使用されます。例えば、DSQEC_CURR_FOLDER が設定され、SAVE QUERY AS Q1 コマンドが実行されると、照会は保管され、大域変数で識別されたフォルダーに照会オブジェクトが組み込まれます。 QMF コマンドで FOLDER キーワードを使用してフォルダー名を指定すれば、この大域変数をオーバーライドできます。この場合、FOLDER キーワードで指定したフォルダー名が、DSQEC_CURR_FOLDER 大域変数で指定したフォルダー名をオーバーライドします。この大域変数がブランクであり、FOLDER キーワードを指定しない場合、フォルダー処理は使用されません。 制約事項: この大域変数は、QMF が DB2 Server (VSE および VM 版) に接続している場合はサポートされません。

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_DISABLEADM	なし	01	<p>QMF 管理者権限の抑止。この大域変数の値が変更されると、それが即時に反映されます。指定できる値は次のとおりです。</p> <p>0 QMF 管理者権限が使用可能です (許可 ID が QMF 管理者権限を持つ場合)。</p> <p>1 QMF 管理者権限が抑止されます (許可 ID の権限にかかわらず)。</p> <p>この大域変数の初期のデフォルト値は、DSQUOPTS 初期化出口ルーチンによりオーバーライドできます。</p>
DSQEC_DSALLOC_DIR	なし	03	<p>TSO 内の新規 PDS データ・セットのメンバーをエクスポートするときに使用するディレクトリー・ブロックの数を指定します。PDS データ・セットの場合、値はゼロより大きくなければなりません。</p> <p>サイトのデフォルト・タイプのデータ・セットまたは PDSE データ・セットを使用する場合は、QMF によってこの大域変数の値が無視されます。サイトのデフォルト・タイプのデータ・セットを使用するには、DSQEC_PO を 0 に設定します。PDSE データ・セットを使用するには、DSQEC_PO を 2 に設定します。</p> <p>サイトで順次データ・セットを使用する場合は、この大域変数をゼロに設定します。</p>
DSQEC_DSALLOC_PRI	なし	08	<p>QMF は、データ・セットをトラック単位で割り振ります。この大域変数は、QMF EXPORT コマンドの結果を保管するために使用される TSO データ・セット用のトラックの 1 次数量を指定します。</p> <p>指定できる値は、1 から、記憶域デバイスおよびオペレーティング・システムで許可される最大サイズまでです。デフォルト値は 15 です。値ゼロは指定できません。</p> <p>PS、PDS、および PDSE データ・セットのトラック数の最大値は 16777215 です。</p>
DSQEC_DSALLOC_SEC	なし	08	<p>QMF は、データ・セットをトラック単位で割り振ります。この大域変数は、QMF EXPORT コマンドの結果を保管するために使用される TSO データ・セット用のトラックの 2 次数量を指定します。</p> <p>指定できる値は、ゼロから、記憶域デバイスおよびオペレーティング・システムで許可される最大サイズまでです。デフォルト値は、105 トラックです。</p> <p>PS および PDS データ・セットのトラック数の最大値は 65535 です。PDSE データ・セットのトラック数の最大値は 16777215 です。</p>
DSQEC_DSLRECL1	なし	05	<p>SQL 照会または QMF プロシーチャーが新規データ・セットにエクスポートされるときに使用される論理レコード長 (LRECL) を指定します。有効な値は 79 から 32760 です。</p> <p>デフォルト値は 79 です。</p>
DSQEC_DS_SUPPORT	なし	01	<p>QMF Data Service (QDS) のサポートを提供します。</p> <p>0 QMF Data Service へのアクセスを許可しません。(デフォルト)</p> <p>1 QMF Data Service へのアクセスを許可します。</p> <p>SQL 照会は QMF Data Service コンポーネントによって分析されます。照会で参照されているオブジェクトは QMF Data Service コンポーネントに定義され、照会全体が QDS によって実行されます。照会で参照されているオブジェクトのいずれも、QDS に定義されているオブジェクトにアクセスしない場合、照会は現在の DB2 接続によって実行されます。</p> <p>QDS サービスをロードできなかったか、または使用できない場合、この値は無視され、すべての要求が DB2 に送られます。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_EDITOR	なし	18	<p>EDIT コマンドで EDITOR キーワードを指定しない場合に、EDITOR キーワードに使用する値を指定します。</p> <p>この大域変数に有効な値は以下のとおりです。</p> <p>PDF プロシーチャーまたは照会の編集に ISPF/PDF エディターを使用します。照会またはプロシーチャーの編集に PDF エディターを使用するには、QMF を ISPF ダイアログとして開始します。</p> <p>EE プロシーチャーまたは照会の編集に SQL QUERY 拡張エディターまたは PROC 拡張エディターを使用します。</p> <p>editorname 使用可能なその他のエディターの名前。エディターを開始する CLIST の名前を指定することもできます。使用可能なエディターについて詳しくは、QMF 管理者にお問い合わせください。</p> <p>デフォルト値はブランクです。</p>
DSQEC_EXPL_MODE	なし	07	<p>RUN QUERY コマンドの実行時に DB2 特殊レジスター CURRENT EXPLAIN MODE に使用される設定を指定します。この特殊レジスターは、適格な動的 SQL ステートメントの EXPLAIN 機能の動作を制御します。照会を実行する前に、QMF によって、CURRENT EXPLAIN MODE 特殊レジスターがこの大域変数で指定された値に設定されます。</p> <p>この大域変数に有効な値は以下のとおりです。</p> <p>NO EXPLAIN 機能が無効になり、EXPLAIN 可能な動的ステートメントの実行時に EXPLAIN 情報は取得されません。これがデフォルト値です。</p> <p>YES EXPLAIN 機能が有効になり、適格な動的 SQL ステートメントが準備され実行された後、ステートメントの EXPLAIN 表に EXPLAIN 情報が挿入されます。すべての動的 SQL ステートメントがコンパイルされ、実行されます。</p> <p>EXPLAIN EXPLAIN 機能が有効になり、適格な動的 SQL ステートメントが準備された後、ステートメントの EXPLAIN 表に EXPLAIN 情報が挿入されます。動的ステートメント (SET ステートメントを除く) は実行されません。</p> <p>DB2 for Linux, UNIX, and Windows または DB2 10 for z/OS (新機能モード) 以降以外のサーバーの場合、唯一の有効な値は NO です。</p>
DSQEC_EXTND_STG	なし	31	<p>DSQSPTYP プログラム・パラメーターが 64BIT に設定されている場合に、拡張記憶域管理機能への要求ごとに QMF が獲得する拡張記憶域のメガバイト数を指定します。このプログラム・パラメーターは、TSO 用の QMF のみで使用できます。</p> <p>拡張記憶域を必要とする操作を実行すると、その操作が完了するまで、または拡張記憶域を使い果たすまで、QMF から指定容量が要求されます。</p> <p>この大域変数を設定する場合は、QMF ユーザーが作業する DATA オブジェクトの平均サイズを考慮してください。平均サイズが大きい場合に小さい値を設定すると、DATA オブジェクトを完成させるために、拡張記憶域管理機能への呼び出しが QMF から数多く発行されます。これらの呼び出しが繰り返されると、パフォーマンスに影響する可能性があります。</p> <p>値には 1 から 1000 を設定できます。デフォルト値は 25 で、これは QMF が要求ごとに 25 MB の記憶域を要求することを示します。</p>
DSQEC_FORM_LANG	なし	01	<p>保管、エクスポート、またはインポートされる書式のデフォルト NLF 言語を設定します。値は以下のとおりです。</p> <p>0 書式に主要 NLF 言語を使用します。</p> <p>1 書式に英語を使用します。この値がデフォルトです。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_ISOLATION	なし	01	<p>デフォルトの照会分離レベル。</p> <p>値は以下のとおりです。</p> <p>0 分離レベル UR (非コミット読み取り)</p> <p>非コミット読み取りは分散環境において役立ちます。ただし、非コミット読み取りを使用している場合は、ユーザーが表示するすべてのレポートは、レポートが表示された後にデータベースから削除されたデータを含むことがあります。</p> <p>1 分離レベル CS (カーソル固定)</p> <p>この値がデフォルトです。カーソル固定を使用しているときには、報告書内のデータに影響するすべてのデータベース・コマンドが完了するまで、QMF によって報告書が表示されません。</p>
DSQEC_KEEP_THREAD	なし	01	<p>照会の終了時にスレッドを解放するか保持するかを指定します。</p> <p>この大域変数は、バッチ・モードで実行されるプロシーチャー用に作成されたスレッド、または QMF が CONNECT コマンドによってリモート・データベースに接続されているときに作成されたスレッドには、影響しません。プロシーチャーがバッチ・モードで実行されるときには、プロシーチャーが完了するまで、スレッドが保持されます。QMF がリモート・データベースに接続されているときには、接続が終了するまで、スレッドが保持されません。</p> <p>この大域変数に有効な値は以下のとおりです。</p> <p>0 照会の終了時に、スレッドは解放されます。これがデフォルト値です。</p> <p>この設定を使用する場合、SET DB2 <i>global variable</i> ステートメントは、以下のいずれかの状態で実行されない限り失敗します。</p> <ul style="list-style-type: none"> このステートメントは、バッチ・モードで実行されるプロシーチャーに組み込まれています。プロシーチャーが完了した後、DB2 大域変数は、デフォルト値にリセットされます。 リモート・データベースに接続するために QMF CONNECT コマンドが発行され、リモート・データベースで SET DB2 <i>global variable</i> ステートメントが実行されます。 SET DB2 <i>global variable</i> ステートメントは複数ステートメント照会に組み込まれ、かつ QMF DSQEC_RUN_MQ 大域変数が 1 に設定されています。照会が完了した後、DB2 大域変数はデフォルト値にリセットされます。 <p>1 QMF セッションが終了するまで、または DSQEC_KEEP_THREAD 大域変数が 0 に設定されるまで、スレッドはアクティブのまま保持されます。この設定ではユーザーは、SET DB2 <i>global variable</i> ステートメントを実行して、DB2 大域変数を設定できます。</p> <p>DSQEC_KEEP_THREAD が 1 に設定されているときに DB2 大域変数を設定し、その後 DSQEC_KEEP_THREAD を 0 に変更すると、それらの DB2 大域変数がデフォルト値に戻ります。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_LAST_RUN	なし	01	<p>QMF オブジェクト・リストの LAST_USED フィールドを更新する一連のコマンドを指定します。このフィールドは、Q.OBJECT_DIRECTORY コントロール表の LAST_USED 列に基づきます。LAST_USED 列の値は、発行されたコマンドが正常に実行されたかどうかには無関係に更新されます。ただし、LAST_USED 列は即時に更新されないことがあります。また、QMF が異常終了すると、LAST_USED 列が更新されない場合があります。</p> <p>値は以下のとおりです。</p> <p>0 QMF は、以下のコマンドが発行されるたびに LAST_USED タイム・スタンプを更新します。</p> <ul style="list-style-type: none"> • CONVERT • DISPLAY • EXPORT • IMPORT • LAYOUT • PRINT • RUN • SAVE <p>この値がデフォルトです。</p> <p>1 QMF は、LAST_USED タイム・スタンプの更新を、RUN、SAVE、および IMPORT コマンドのみに制限します。</p> <p>2 QMF によって、LAST_USED タイム・スタンプの更新が RUN コマンドのみに制限されます。</p>
DSQEC_LIST_OWNER	なし	128	<p>LIST コマンドの OWNER パラメーターに対してデフォルト値を提供します。128 文字までの許可 ID を指定してください。この変数は、デフォルトでは空白であり、現在の許可 ID が所有するオブジェクトのリストが表示されます。</p> <p>変数値には選択記号を使用できます。単一文字の代わりに下線 (_) を使用し、ゼロ以上の文字の代わりに % 記号を使用してください。例えば、次のコマンドの後に LIST コマンドを実行すると、QMF に対して、先頭に文字 RO を使用するユーザー ID が所有するオブジェクトのみをリストするように指示します。</p> <p>SET GLOBAL (DSQEC_LIST_OWNER=RO%)</p> <p>以下のコマンドは、デフォルトの所有者を、I で始まり、2 桁目の位置と残りの桁に何らかの文字を持つユーザー ID に設定します。</p> <p>SET GLOBAL (DSQEC_LIST_OWNER=I_%</p> <p>この大域変数で設定する値は、ホーム・パネルを除く QMF パネルの List キーを押したときに表示されるリストには適用されません。</p>
DSQEC_LOB_COLMAX	なし	10	<p>取得される LOB 列の最大データ・サイズ (バイト単位) を指定します。LOB の最大サイズは 2147483637 (つまり 2 GB) です。</p> <p>デフォルトでは、LOB データではなく LOB メタデータが取得されます。ただし、M 以外の編集コードが指定された場合、または DSQEC_LOB_RETRV 大域変数が 3 に設定された場合は、メタデータではなく LOB データが取得されます。この場合は、最大サイズを超える LOB データが含まれた表をユーザーが照会すると、エラーが出され、報告書データは表示されません。ユーザーが、最大サイズを超える LOB データが含まれた表またはデータ・オブジェクトに対して EXPORT TABLE、PRINT TABLE、SAVE DATA、または EXPORT DATA コマンドを発行すると、エラーが出され、コマンドが終了します。</p> <p>デフォルトは 0 で、最大値の指定がないことを示します。</p>
DSQEC_LOB_RETRV	なし	01	<p>LOB データまたはメタデータの取得方法を指定します。有効な値は以下のとおりです。</p> <p>1 結果に LOB メタデータを表示します。実際の LOB データを表示するには、M 編集コードを別の編集コードに変更します。この値が指定されると、QMF によって LOB データへのアクセスに LOB ロケーターが使用されます。これがデフォルト設定です。</p> <p>2 結果に LOB メタデータのみを表示します。M 編集コードは、LOB データに有効な唯一の編集コードです。この値を指定すると、QMF では LOB ロケーターが使用されません。</p> <p>3 実際の LOB データを取得し、結果に表示します。この値が指定されると、QMF によって LOB データへのアクセスに LOB ロケーターは使用されません。</p>

表 57. コマンドとプロシージャの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_LOB_SAVE	なし	01	<p>ユーザーが QMF SAVE DATA コマンドまたは IMPORT TABLE コマンドを使用して LOB データをデータベース内の表に保管できるかどうかを指定します。有効な値は以下のとおりです。</p> <p>0 - LOB 保管を無効にする (Disable LOB Save) 列に LOB データが含まれている場合に、ユーザーが QMF SAVE DATA コマンドまたは IMPORT TABLE コマンドを発行してデータをデータベース内の表に保管できないように指定します。LOB 列が存在する場合は、エラー・メッセージが表示され、データは保管されません。</p> <p>1 - LOB 保管を有効にする (Enable LOB Save) ユーザーが QMF SAVE DATA コマンドまたは IMPORT TABLE コマンドを使用して LOB データをデータベース内の表に保管できるように指定します。これがデフォルト値です。</p>
DSQEC_NLFCMD_LANG	なし	01	<p>コマンド用に予期される NLF 言語を設定します。値は以下のとおりです。</p> <p>0 コマンドには主要 NLF 言語を使用します。この値がデフォルトです。</p> <p>1 コマンドには英語を使用します。</p>
DSQEC_PO	なし	01	<p>QMF オブジェクトを新規 TSO データ・セットにエクスポートするときに作成する、区分 (PO) データ・セットのタイプを指定します。値は以下のとおりです。</p> <p>0 データ・セットのタイプを、使用しているサイトのデフォルトとしてリストされたタイプに割り振ります。このタイプは、SYS1.PARMLIB の IGDSMSxx メンバーの中に指定されています。この値はデフォルト値です。</p> <p>1 エクスポートされたデータに、PDS データ・セットを割り振ります。</p> <p>2 エクスポートされたデータに、PDSE データ・セットを割り振ります。</p>
DSQEC_PRO_ENABLE	なし	01	<p>QMF が QUERY、FORM、PROC、または PROFILE 一時記憶域の内容を上書きまたは破棄する前に、確認パネルを表示するかどうかを制御します。値は以下のとおりです。</p> <p>0 サポートされている一時記憶域の内容が上書きされる前に、確認パネルは表示されません。この値がデフォルトです。</p> <p>1 対象となる一時記憶域に対応する大域変数も 1 に設定されている場合、確認パネルは表示されます。以下の大域変数は、それぞれ対応する一時記憶域の上書きを制御します。</p> <ul style="list-style-type: none"> DSQEC_PRO_FORM は、現在の QMF 報告書の書式指定をストアする FORM 一時記憶域の上書きを制御します。 DSQEC_PRO_PROC は、現在の QMF プロシージャをストアする PROC 一時記憶域の上書きを制御します。 DSQEC_PRO_PROF は、QMF プロファイル設定をストアする PROFILE 一時記憶域の上書きを制御します。 DSQEC_PRO_QUERY は、現在の QMF 照会をストアする QUERY 一時記憶域の上書きを制御します。
DSQEC_PRO_FORM	なし	01	<p>この変数は、QMF によって FORM 一時記憶域の内容が上書きまたは破棄される前に、確認パネルを表示するかどうかを制御します。DSQEC_PRO_ENABLE 大域変数を 1 に設定する必要があります。指定できる値は以下のとおりです。</p> <p>0 一時記憶域の内容が破棄される前に、確認パネルは表示されません。</p> <p>1 確認パネルが表示され、ユーザーは破棄を保留しているコマンドを実行することも取り消すこともできます。その後、一時記憶域の内容は SAVE コマンドで保存できます。</p>
DSQEC_PRO_PROC	なし	01	<p>この変数は、QMF によって PROC 一時記憶域の内容が上書きまたは破棄される前に、確認パネルを表示するかどうかを制御します。DSQEC_PRO_ENABLE 大域変数を 1 に設定する必要があります。指定できる値は以下のとおりです。</p> <p>0 一時記憶域の内容が破棄される前に、確認パネルは表示されません。</p> <p>1 一時記憶域の内容が破棄される前に、確認パネルが表示されます。ユーザーは破棄を保留しているコマンドを実行することも取り消すこともできます。その後、一時記憶域の内容は SAVE コマンドで保存できます。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_PRO_PROF	なし	01	<p>この変数は、QMF によって PROFILE 一時記憶域の内容が上書きまたは破棄される前に、確認パネルを表示するかどうかを制御します。DSQEC_PRO_ENABLE 大域変数を 1 に設定する必要があります。指定できる値は以下のとおりです。</p> <p>0 一時記憶域の内容が破棄される前に、確認パネルは表示されません。</p> <p>1 一時記憶域の内容が破棄される前に、確認パネルが表示されます。ユーザーは破棄を保留しているコマンドを実行することも取り消すこともできます。その後、一時記憶域の内容は SAVE コマンドで保存できます。</p>
DSQEC_PRO_QUERY	なし	01	<p>この変数は、QMF によって QUERY 一時記憶域の内容が上書きまたは破棄される前に、確認パネルを表示するかどうかを制御します。DSQEC_PRO_ENABLE 大域変数を 1 に設定する必要があります。指定できる値は以下のとおりです。</p> <p>0 一時記憶域の内容が破棄される前に、確認パネルは表示されません。</p> <p>1 一時記憶域の内容が破棄される前に、確認パネルが表示されます。ユーザーは破棄を保留しているコマンドを実行することも取り消すこともできます。その後、一時記憶域の内容は SAVE コマンドで保存できます。</p>
DSQEC_RERUN_IPROC	なし	01	<p>END コマンドの実行後に、呼び出しプロシーチャーを再実行します。値は以下のいずれかになります。</p> <p>0 END コマンドの実行後、呼び出しプロシーチャーを再実行しません。</p> <p>1 END コマンドの実行後、呼び出しプロシーチャーを再実行します。この値がデフォルトです。</p> <p>QMF を呼び出しプロシーチャーで起動しておいてから、この変数を '0' に設定すると、プロシーチャー再実行の代わりに QMF が終了します。</p>
DSQEC_RESET_RPT	なし	31	<p>一時記憶域内の不完全な DATA オブジェクトがパフォーマンスに影響する可能性がある場合に、QMF からプロンプトを出すかどうかを決定します。指定できる値は以下のとおりです。</p> <p>0 リセット報告書プロンプト・パネルは表示されず、QMF が実行中の報告書を完成させます。この値はデフォルト値です。</p> <p>1 リセット報告書プロンプト・パネルが表示されます。このパネルは、新しいコマンドを開始する前に現在実行中の報告書を完了させるかリセットするかを尋ねるプロンプトを出します。</p> <p>2 リセット報告書プロンプト・パネルは表示されず、QMF は現在、実行中の報告書を停止させます。</p>
DSQEC_RUN_MQ	なし	01	<p>RUN QUERY コマンドが SQL 照会内の複数のステートメントをサポートするかどうかを指定します。値は以下のとおりです。</p> <p>0 複数の SQL ステートメントはサポートされません。この変数を 0 に設定し、複数のステートメントを含む SQL 照会を実行すると、QMF は、最初のセミコロンが検出された後のすべてのステートメントを無視します。この値がデフォルトです。</p> <p>1 複数の SQL ステートメントがサポートされます。セミコロンは各ステートメント (最後のステートメントを除く) の最後に配置する必要があります。</p> <p>制約事項: SELECT ステートメントは他のステートメントとともに照会に含めることができますが、照会ごとに含めることができる SELECT ステートメントは 1 つだけです。CALL ステートメントおよび CREATE PROCEDURE ステートメントは、SQL 照会内で単独で使用する必要があります。</p>
DSQEC_SAV_ACCELNM	なし	128	<p>SAVE DATA コマンド、IMPORT TABLE コマンド、および RUN QUERY to TABLE コマンドからアクセラレーター専用表を作成する際に使用するデフォルト・アクセラレーターの名前を指定します。この変数は、ACCELERATOR キーワードが指定されていない場合にのみ参照されます。</p> <p>この大域変数は空白に設定できますが、DSQEC_SAV_ALLOWED 大域変数が「4」に設定されている場合は空白に設定しないでください。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_SAV_ALLOWED	なし	01	<p>ユーザーが QMF の SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを使用して、データをデータベース内の新規表に保管するか、アクセラレーター内の新規表に保管するかを制御します。オプション 0 の場合を除き、このフィールドは、置き換えられたデータが入っている、またはデータの追加先である既存の表の場所には影響しません。この変数の設定とは無関係に、既存表のデータはデータベースまたはアクセラレーター内で置換または付加されます。</p> <p>この大域変数の有効な値は以下のとおりです。</p> <p>0 - データ保管を無効にする ユーザーは、QMF の SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを使用してデータをデータベースまたはアクセラレーター内の表に保管できません。エラー・メッセージが表示され、データは保管されません。</p> <p>1 - データベース表へのデータの保管のみを有効にする ユーザーは、QMF の SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを使用してデータをデータベース内の表に保管できます。データをアクセラレーター専用表に保管することはできません。この設定がデフォルトです。</p> <p>2 - アクセラレーター専用表へのデータの保管のみを有効にする ユーザーは、QMF の SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを使用してデータをアクセラレーター専用表に保管できます。データをデータベース表に保管することはできません。DSQEC_SAV_ACCELNM 大域変数にはアクセラレーターのデフォルト名が含まれていますが、ACCELERATOR キーワードによってオーバーライドすることができます。</p> <p>3 - データベース表またはアクセラレーター専用表のいずれかへのデータの保管を有効にする (データベースがデフォルト) ユーザーは、QMF の SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを使用してデータをデータベース内の表またはアクセラレーター専用表のいずれかに保管できます。SPACE や ACCEL などのコマンド・キーワード・オーバーライドがない場合、表はデータベースに保管されます。</p> <p>4 - データベース表またはアクセラレーター専用表のいずれかへのデータの保管を有効にする (アクセラレーターがデフォルト) ユーザーは、QMF の SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを使用してデータをデータベース内の表またはアクセラレーター専用表のいずれかに保管できます。SPACE や ACCELERATOR などのコマンド・キーワード・オーバーライドがない場合、表はアクセラレーターに保管されます。このオプションを選択する場合、DSQEC_SAV_ACCELNM 大域変数にはアクセラレーターの名前が含まれていなければなりません。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_SESSGLV_SAV	なし	01	<p>一部のパネル上の一部のデータ入力フィールドへのユーザー入力を QMF セッション内およびセッション間で保管するかどうかを制御します。ユーザー入力はセッション変数として保管されます。これらのセッション変数は DXY 接頭部が付加された大域変数として Q.GLOBAL_VARS 表に保管されます。DSQEC_SESSGLV_SAV 大域変数は、セッション全体にわたって検査され、QMF の開始時と終了時にも検査されます。有効な値は以下のとおりです。</p> <p>0 QMF の開始時にこの設定を指定すると、Q.GLOBAL_VARS 表からすべてのセッション変数が削除されます。</p> <p>QMF セッション中にこの設定を指定すると、記憶域からすべてのセッション変数が削除されます。この設定が 1 または 2 に変更されない限り、現行セッションではそれ以降、セッション変数は保管されません。</p> <p>QMF の終了時にこの設定を指定すると、Q.GLOBAL_VARS 表からすべてのセッション変数が削除されるため、ユーザー入力は次の QMF セッションには引き継がれません。</p> <p>これがデフォルト値です。</p> <p>1 QMF の開始時にこの設定を指定すると、Q.GLOBAL_VARS 表からユーザーのすべてのセッション変数が復元されます。</p> <p>QMF セッション中にこの設定を指定すると、現行セッションではそれ以降、セッション変数が保管されます。例えば、LIST コマンド・プロンプト・パネルに値を入力し、LIST パネルを終了し、同じセッション内でそのパネルに戻ると、それらのフィールドには、以前入力した値が取り込まれます。</p> <p>QMF の終了時にこの設定を指定すると、現行セッション中にユーザーが作成または変更したすべてのセッション変数が破棄され、Q.GLOBAL_VARS 表に保管されません。現行セッションより前に Q.GLOBAL_VARS 表に存在していたすべてのセッション変数値は、未変更のままになります。このオプションを使用すると、例えば、QMF セッションを開始するたびに、同じセッション変数値を再度初期化できます。</p> <p>次の QMF セッションが開始されると、管理者が設定した初期大域変数によって指定変更されない限り、値が 0 に戻ります。</p> <p>2 QMF の開始時にこの設定を指定すると、Q.GLOBAL_VARS 表からユーザーのすべてのセッション変数が復元されます。</p> <p>QMF セッション中にこの設定を指定すると、この設定が 0 に変更されない限り、現行セッションではそれ以降、セッション変数が保管されます。例えば、LIST コマンド・プロンプト・パネルに値を入力し、LIST パネルを終了し、同じセッション内でそのパネルに戻ると、それらのフィールドには、以前入力した値が取り込まれます。</p> <p>QMF の終了時にこの設定を指定すると、Q.GLOBAL_VARS 表にすべてのセッション変数が保管されるため、そのセッション中に保管されたユーザー入力はすべてその次の QMF セッションにも引き継がれます。</p> <p>このパラメーターは、CONNECT、CONVERT、DISPLAY、DRAW、EDIT、ERASE、EXPORT、IMPORT、LIST、PRINT、RESET、RUN、SAVE、SET、および SHOW コマンドを介してアクセスする、コマンド・プロンプト・パネル上のほとんどのフィールドに適用されます。</p>
DSQEC_SHARE	なし	31	<p>SHARE パラメーターのデフォルト値を指定します。値は以下のいずれかになります。</p> <p>0 データを他のユーザーと共用しない。</p> <p>1 データを他のユーザーと共用する。</p>
DSQEC_SP_RS_NUM	なし	04	<p>報告書を作成するために使用する結果セット (ストアード・プロシーチャーから返される) を指示します。値は以下のとおりです。</p> <p>0 結果セットを無視します。</p> <p>1 最初の結果セットを返します。</p> <p>2 2 番目の結果セットを返します。</p> <p>n n 番目の結果セットを返します。n の最大値は、63 です。</p> <p>ALL すべての結果セットを返します。</p>

表 57. コマンドとプロシージャの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_SPAC_OVRIDE	なし	01	<p>QMF プロファイルで指定されたデフォルト表スペースをユーザーがオーバーライドできるかどうかを指定します。</p> <p>この大域変数の有効な値は以下のとおりです。</p> <p>0 - SPACE キーワード・オプションを無効にする ユーザーは、SPACE キーワード・オプションを指定した SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを発行できません。</p> <p>1 - SPACE キーワード・オプションを有効にする ユーザーは、SPACE キーワード・オプションを指定した SAVE DATA コマンド、RUN QUERY to TABLE コマンド、または IMPORT TABLE コマンドを発行できます。この設定がデフォルトです。</p>
DSQEC_SQLQRYSZ_2M	なし	01	<p>長さが 32,767 バイト (32 KB) を超える SQL 照会を RUN QUERY コマンドがサポートするかどうかを制御します。</p> <p>0 DB2 for z/OS、DB2 for iSeries、および DB2 for Linux, UNIX, and Windows データベースに対する SQL 照会は、32,767 バイト (32 KB) に制限されます。この値がデフォルトです。</p> <p>1 SQL 照会は 32 KB よりも大きくなる場合があります。サポートされる最大照会サイズは、照会の送信先のデータベースのタイプによって異なります。</p> <ul style="list-style-type: none"> DB2 for z/OS に送信される照会の長さは、2 MB まで可能です。 DB2 for iSeries または DB2 for Linux, UNIX, and Windows に送信される照会の長さは、65 KB まで可能です。 <p>これらの最大サイズは、RUN QUERY コマンドの送信先のデータベースのバージョンがこのサイズの照会をサポートしていることを前提としています。DB2 for VSE and VM に対する SQL 照会は、8 KB に制限されます。</p> <p>QMF for CICS から 32 KB より大きな照会を実行する場合は、追加のカスタマイズが必要となる場合があります。</p>
DSQEC_TABS_LDB2	なし	31	<p>現行サーバーで表と視点のリストを検索するための視点。これが DB2 for z/OS または DB2 for Linux, UNIX, and Windows である場合。</p>
DSQEC_TABS_RDB2	なし	31	<p>リモート DB2 サブシステムにある表と視点のリストを取り出すための視点。</p>
DSQEC_TABS_SQL	なし	31	<p>DB2 for VSE and VM データベースの表および視点のリストを取り出すための視点。</p>
DSQEC_TRACE_LIMIT	なし	31	<p>トレース出力量を指定のバイト数に制限します。有効な範囲は 0 から 2147483647 です。</p> <p>この変数を使用すると、QMF トレース出力のサイズを小さくすることができます。</p> <p>この大域変数は通常、IBM ソフトウェア・サポートの指示に従って設定します。</p>
DSQEC_TRACE_MODULE	なし	54	<p>トレース対象の QMF モジュールの名前が入っています。</p> <p>6 つまでのモジュールをコンマで区切って指定できます。</p> <p>モジュール名を大域変数に指定した後、ALL に設定した TRACE キーワードを付けて SET PROFILE コマンドを実行し、トレースを開始してください。例: SET PROFILE (TRACE=ALL)</p> <p>注: コマンド行での SET GLOBAL コマンドによってモジュールを指定する場合、モジュール名は単一引用符で囲む必要があります。</p>

表 57. コマンドとプロシーチャーの実行を制御する大域変数 (続き)

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQEC_TWO_GB_ROW	なし	01	<p>QMF 報告書に返される行の長さを制御します。以下の値のいずれかを使用してください。</p> <p>0 QMF 報告書に XML 列または LOB 列が含まれていない限り、報告書のデータ行の長さを 32 KB に制限します。</p> <p>1 32 KB より大きいデータ行の長さを許可します (最大 2 GB)。</p> <p>重要:</p> <ul style="list-style-type: none"> DSQEC_TWO_GB_ROW 大域変数の設定に関係なく、デフォルトでは、最大 2 GB の XML、CLOB、または BLOB データ、および最大 1 GB の DBCLOB データを表示できます。ただし、LOB 行の最大長は、DSQEC_LOB_COLMAX 大域変数で制限できます。 DSQEC_TWO_GB_ROW 大域変数の設定に関係なく、1 つの表の最大レコード・サイズがそのページ・サイズを超えることはできません。DB2 は、4 KB、8 KB、16 KB、または 32 KB のサイズのページ内にレコードを保管します。そのため、1 つの表からデータを表示または選択するときには、表示できるデータ行の最大長が 32 KB のままになります。2 つ以上の表を結合する視点からデータを表示または選択すると、行の長さは最大 2 GB まで可能です。 <p>これらのページ・サイズが考慮されるため、SAVE DATA コマンドで保存できる QMF 報告書のデータ行の長さも 32 KB に制限されています。LOB データを保管する機能は、DSQEC_LOB_SAVE 大域変数で制御されます。</p>
DSQEC_USERGLV_SAV	なし	01	<p>ユーザーが作成または変更した大域変数 (先頭が「DSQ」である大域変数を含む) が、QMF セッションの終了時に保管されるかどうかを決定します。保管される値は、Q.GLOBAL_VARS 表に保管され、そのセッションのユーザー ID に関連付けられます。値が保管された場合は、そのユーザーによる次の QMF セッションの開始時にそれらの値が復元されます。有効な値は以下のとおりです。</p> <p>0 QMF の終了時に、Q.GLOBAL_VARS 表からすべての大域変数が削除され、現行セッション中の大域変数は Q.GLOBAL_VARS 表に保管されません。これがデフォルト値です。</p> <p>1 QMF の終了時に、現行セッション中にユーザーが作成または変更したすべての大域変数が破棄され、Q.GLOBAL_VARS 表に保管されません。Q.GLOBAL_VARS 表に既に存在していたすべての大域変数値は、現行の QMF セッションより前の状態のまま保持されます。このオプションを使用すると、例えば、QMF セッションを開始するたびに、同じ大域変数値を再度初期化できます。</p> <p>次の QMF セッションが開始されると、管理者が設定した初期大域変数によって指定変更されない限り、値が 0 に戻ります。</p> <p>2 QMF の終了時に、ユーザーが作成または変更したすべての大域変数が Q.GLOBAL_VARS 表に保管されます。ユーザーが QMF を再開すると、そのユーザーの以前のセッションで保管された大域変数が復元されます。変数が読み取り専用として定義された場合を除き、管理者が Q.GLOBAL_VARS 表で定義した値はすべて、ユーザーの値に置き換えられます。</p>

関連資料:

 DB2 の PREPARE ステートメント

PREPARE ステートメントの concurrent-access-resolution 属性に関する情報を参照してください。

CONVERT QUERY の結果を保管する大域変数

DSQQC 大域変数は、CONVERT QUERY コマンドの結果を反映します。これらの大域変数はいずれも、SET GLOBAL コマンドでは変更できません。

表 58. CONVERT QUERY コマンドの結果を反映する大域変数

呼び出し可能インターフェースの変数名	コマンド・インターフェースの変数名	長さ	説明
DSQQC_LENGTH_mnn	DSQCL_mnn	05	変換結果の長さ mnn

表 58. CONVERT QUERY コマンドの結果を反映する大域変数 (続き)

呼び出し可能インターフェースの 変数名	コマンド・インターフェースの 変数名	長さ	説明
DSQQC_QRY_COUNT	DSQCQCNT	03	変換結果に含まれる照会の数。元の照会が QBE I. または QBE U. 照会でない限り、値は常に「1」でなければなりません。
DSQQC_QRY_LANG	DSQCQLNG	01	変換された照会の言語。値は以下のとおりです。 1 SQL 2 QBE 3 Prompted
DSQQC_QRY_TYPE	DSQCQTYP	指定なし	変換結果中の最初の値
DSQQC_RESULT_ <i>nnn</i>	DSQQC <i>nnn</i>	指定なし	<i>nnn</i>

RUN QUERY エラー・メッセージ情報を示す大域変数

DSQQM 大域変数は、RUN QUERY コマンドの結果を保管します。これらの大域変数は、いずれも SET GLOBAL コマンドでは変更できません。

表 59. RUN QUERY コマンドの結果を保管する大域変数

呼び出し可能インターフェースの 変数名	コマンド・インターフェースの 変数名	長さ	説明
DSQQM_MESSAGE	DSQCIQMG	80	照会メッセージのテキスト。
DSQQM_MESSAGE_ALL	DSQCIQMA	360	完全照会メッセージ・テキスト
DSQQM_MSG_HELP	DSQCIQID	08	メッセージ・ヘルプ・パネルの ID
DSQQM_MSG_NUMBER	DSQCIQNO	08	メッセージ番号。
DSQQM_SQL_RC	DSQCISQL	16	最後のコマンドまたは照会からの SQLCODE
DSQQM_SQL_STATE	なし	05	データベース・マネージャーから返されてきたときは、DSQQM_SQL_RC の SQLCODE と関連づけられている SQLSTATE。
DSQQM_SUB_TXT_ <i>nn</i>	DSQCIQ <i>nn</i>	20	置換値 <i>nn</i>
DSQQM_SUBST_VARS	DSQCIQ00	04	置換変数の数

パネル入力値を保管する大域変数

DSQEC_SESSGLV_SAV 大域変数が 1 または 2 に設定されている場合、DXY 大域変数は、ユーザーがデータ入力フィールドに入力した値を保管します。一部のパネル上の一部のデータ入力フィールドへの入力のみが保管されます。

DSQEC_SESSGLV_SAV 大域変数の設定に関係なく、次の表にリストされていないフィールドへのユーザー入力は保管されません。

これらの大域変数はすべて、SET GLOBAL コマンドで変更できます。ただし、これらの変数を変更したり削除したりすると、コマンド・プロンプト・パネルで生成される値が変更されるため、注意が必要です。

表 60. DXY 大域変数とパネル・フィールド名との間のマッピング

大域変数名 (<i>n</i> は各 国語 ID、 <i>ln</i> は複数 行フィールド内の 1 行に関連付けられて いる ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXYnPCO1_ <i>ln</i>	01 - 03	CONNECT	ユーザー
DXYnPCO1_05	-	CONNECT	ロケーション
DXYnPCO3_01	-	CONNECT (CICS)	ロケーション
DXYnPCNV_ <i>ln</i>	02 - 07	CONVERT	名前
DXYnPDSP_ <i>ln</i>	02 - 07	DISPLAY	名前
DXYnPDSP_ <i>ln</i> _01	02 - 07	DISPLAY QUERY	名前
DXYnPDSP_ <i>ln</i> _02	02 - 07	DISPLAY PROC	名前
DXYnPDSP_ <i>ln</i> _03	02 - 07	DISPLAY FORM	名前
DXYnPDSP_ <i>ln</i> _05	02 - 07	DISPLAY REPORT	名前
DXYnPDSP_ <i>ln</i> _07	02 - 07	DISPLAY CHART	名前
DXYnPDSP_ <i>ln</i> _08	02 - 07	DISPLAY TABLE	名前
DXYnPDRS_ <i>ln</i>	01 - 06	DRAW	名前
DXYnPDRS_07	-	DRAW	タイプ
DXYnPDRS_08	-	DRAW	Identifier
DXYnPEDT_01	-	EDIT	タイプ
DXYnPED2_ <i>ln</i>	01 - 06	EDIT TABLE	名前
DXYnPED2_07	-	EDIT TABLE	Mode
DXYnPERA_ <i>ln</i>	02 - 07	ERASE	名前
DXYnPERA_ <i>ln</i> _01	02 - 07	ERASE QUERY	名前
DXYnPERA_ <i>ln</i> _02	02 - 07	ERASE PROC	名前
DXYnPERA_ <i>ln</i> _03	02 - 07	ERASE FORM	名前
DXYnPERA_ <i>ln</i> _08	02 - 07	ERASE TABLE	名前
DXYnPEXM_ <i>ln</i>	02 - 07	EXPORT	名前
DXYnPEXM_ <i>ln</i> _01	02 - 07	EXPORT QUERY	名前
DXYnPEXM_ <i>ln</i> _02	02 - 07	EXPORT PROC	名前
DXYnPEXM_ <i>ln</i> _03	02 - 07	EXPORT FORM	名前
DXYnPEXM_ <i>ln</i> _05	02 - 07	EXPORT REPORT	名前
DXYnPEXM_ <i>ln</i> _06	02 - 07	EXPORT DATA	名前
DXYnPEXM_ <i>ln</i> _07	02 - 07	EXPORT CHART	名前
DXYnPEXM_ <i>ln</i> _08	02 - 07	EXPORT TABLE	名前
DXYnPXM1_ <i>ln</i>	01 - 05	EXPORT の後に Enter (TSO の場合)	To
DXYnPXM1_ <i>ln</i> _01	01 - 05	EXPORT QUERY の 後に Enter (TSO の 場合)	To

表 60. DXY 大域変数とパネル・フィールド名のためのマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PXM1_ <i>ln</i> _02	01 - 05	EXPORT PROC の後に Enter (TSO の場合)	To
DXY n PXM1_06	-	EXPORT の後に Enter (TSO の場合)	Member
DXY n PXM1_06_01	-	EXPORT QUERY の後に Enter (TSO の場合)	Member
DXY n PXM1_06_02	-	EXPORT PROC の後に Enter (TSO の場合)	Member
DXY n PXM2_01_07	-	EXPORT CHART の後に Enter (TSO の場合)	Member
DXY n PXM3_ <i>ln</i> _05	01 - 05	EXPORT REPORT の後に Enter (TSO の場合)	To
DXY n PXM3_06_05	-	EXPORT REPORT の後に Enter (TSO の場合)	Member
DXY n PXM3_08_05	-	EXPORT REPORT の後に Enter (TSO の場合)	データ形式
DXY n PXM4_ <i>ln</i> _06	01 - 05	EXPORT DATA の後に Enter (TSO の場合)	To
DXY n PXM4_06_06	-	EXPORT DATA の後に Enter (TSO の場合)	Member
DXY n PXM4_08_06	-	EXPORT DATA の後に Enter (TSO の場合)	データ形式
DXY n PXM4_09_06	-	EXPORT DATA の後に Enter (TSO の場合)	出力モード
DXY n PXM4_10_06	-	EXPORT DATA の後に Enter (TSO の場合)	ヘッダー
DXY n PXM5_ <i>ln</i> _03	01 - 05	EXPORT FORM の後に Enter (TSO の場合)	To

表 60. DXY 大域変数とパネル・フィールド名とのマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PXM5_06_03	-	EXPORT FORM の後に Enter (TSO の場合)	Member
DXY n PXM5_08_03	-	EXPORT FORM の後に Enter (TSO の場合)	言語
DXY n PXC1_01	-	EXPORT の後に Enter (CICS の場合)	待ち行列名
DXY n PXC1_01_01	-	EXPORT QUERY の後に Enter (CICS の場合)	待ち行列名
DXY n PXC1_01_02	-	EXPORT PROC の後に Enter (CICS の場合)	待ち行列名
DXY n PXC1_02	-	EXPORT の後に Enter (CICS の場合)	待ち行列タイプ
DXY n PXC1_02_01	-	EXPORT QUERY の後に Enter (CICS の場合)	待ち行列タイプ
DXY n PXC1_02_02	-	EXPORT PROC の後に Enter (CICS の場合)	待ち行列タイプ
DXY n PXC1_04	-	EXPORT の後に Enter (CICS の場合)	中断
DXY n PXC1_04_01	-	EXPORT QUERY の後に Enter (CICS の場合)	中断
DXY n PXC1_04_02	-	EXPORT PROC の後に Enter (CICS の場合)	中断
DXY n PXC3_01_05	-	EXPORT REPORT の後に Enter (CICS の場合)	待ち行列名
DXY n PXC3_02_05	-	EXPORT REPORT の後に Enter (CICS の場合)	待ち行列タイプ
DXY n PXC3_04_05	-	EXPORT REPORT の後に Enter (CICS の場合)	中断
DXY n PXC3_05_05	-	EXPORT REPORT の後に Enter (CICS の場合)	データ形式

表 60. DXY 大域変数とパネル・フィールド名との間のマッピング (続き)

大域変数名 (<i>n</i> は各 国語 ID、 <i>ln</i> は複数 行フィールド内の 1 行に関連付けられて いる ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PXC4_01_06	-	EXPORT DATA の 後に Enter (CICS の 場合)	待ち行列名
DXY n PXC4_02_06	-	EXPORT DATA の 後に Enter (CICS の 場合)	待ち行列タイプ
DXY n PXC4_04_06	-	EXPORT DATA の 後に Enter (CICS の 場合)	中断
DXY n PXC4_05_06	-	EXPORT DATA の 後に Enter (CICS の 場合)	データ形式
DXY n PXC4_06_06	-	EXPORT DATA の 後に Enter (CICS の 場合)	出力モード
DXY n PXC4_07_06	-	EXPORT DATA の 後に Enter (CICS の 場合)	ヘッダー
DXY n PXC5_01_03	-	EXPORT FORM の 後に Enter (CICS の 場合)	待ち行列名
DXY n PXC5_02_03	-	EXPORT FORM の 後に Enter (CICS の 場合)	待ち行列タイプ
DXY n PXC5_04_03	-	EXPORT FORM の 後に Enter (CICS の 場合)	中断
DXY n PXC5_05_03	-	EXPORT FORM の 後に Enter (CICS の 場合)	言語
DXY n PIMM_ <i>ln</i>	02 - 07	IMPORT (TSO の場 合)	名前
DXY n PIMM_ <i>ln</i> _01	02 - 07	IMPORT QUERY (TSO の場合)	名前
DXY n PIMM_ <i>ln</i> _02	02 - 07	IMPORT PROC (TSO の場合)	名前
DXY n PIMM_ <i>ln</i> _03	02 - 07	IMPORT FORM (TSO の場合)	名前
DXY n PIMM_ <i>ln</i> _06	02 - 07	IMPORT DATA (TSO の場合)	名前

表 60. DXY 大域変数とパネル・フィールド名との間のマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PIMM_ ln _08	02 - 07	IMPORT TABLE (TSO の場合)	名前
DXY n PIMM_ ln	08 - 13	IMPORT (TSO の場合)	From
DXY n PIMM_ ln _01	08 - 13	IMPORT QUERY (TSO の場合)	From
DXY n PIMM_ ln _02	08 - 13	IMPORT PROC (TSO の場合)	From
DXY n PIMM_ ln _03	08 - 13	IMPORT FORM (TSO の場合)	From
DXY n PIMM_ ln _06	08 - 13	IMPORT DATA (TSO の場合)	From
DXY n PIMM_ ln _08	08 - 13	IMPORT TABLE (TSO の場合)	From
DXY n PIMM_14	-	IMPORT (TSO の場合)	Member
DXY n PIMM_14_01	-	IMPORT QUERY (TSO の場合)	Member
DXY n PIMM_14_02	-	IMPORT PROC (TSO の場合)	Member
DXY n PIMM_14_03	-	IMPORT FORM (TSO の場合)	Member
DXY n PIMM_14_06	-	IMPORT DATA (TSO の場合)	Member
DXY n PIMM_14_08	-	IMPORT TABLE (TSO の場合)	Member
DXY n PIQF_03	-	IMPORT の後に Enter (TSO の場合)	注釈
DXY n PIQF_03_01	-	IMPORT QUERY の後に Enter (TSO の場合)	注釈
DXY n PIQF_03_02	-	IMPORT PROC の後に Enter (TSO の場合)	注釈
DXY n PIQL_03_03	-	IMPORT FORM の後に Enter (TSO の場合)	注釈
DXY n PIQL_04_03	-	IMPORT FORM の後に Enter (TSO の場合)	言語

表 60. DXY 大域変数とパネル・フィールド名のためのマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PITB_01_08	-	IMPORT TABLE の後に Enter (TSO の場合)	注釈
DXY n PIMC_ <i>ln</i>	02 - 07	IMPORT (CICS の場合)	名前
DXY n PIMC_ <i>ln</i> _01	02 - 07	IMPORT QUERY (CICS の場合)	名前
DXY n PIMC_ <i>ln</i> _02	02 - 07	IMPORT PROC (CICS の場合)	名前
DXY n PIMC_ <i>ln</i> _03	02 - 07	IMPORT FORM (CICS の場合)	名前
DXY n PIMC_ <i>ln</i> _06	02 - 07	IMPORT DATA (CICS の場合)	名前
DXY n PIMC_ <i>ln</i> _08	02 - 07	IMPORT TABLE (CICS の場合)	名前
DXY n PIMC_08	-	IMPORT (CICS の場合)	待ち行列名
DXY n PIMC_08_01	-	IMPORT QUERY (CICS の場合)	待ち行列名
DXY n PIMC_08_02	-	IMPORT PROC (CICS の場合)	待ち行列名
DXY n PIMC_08_03	-	IMPORT FORM (CICS の場合)	待ち行列名
DXY n PIMC_08_06	-	IMPORT DATA (CICS の場合)	待ち行列名
DXY n PIMC_08_08	-	IMPORT TABLE (CICS の場合)	待ち行列名
DXY n PIMC_09	-	IMPORT (CICS の場合)	待ち行列タイプ
DXY n PIMC_09_01	-	IMPORT QUERY (CICS の場合)	待ち行列タイプ
DXY n PIMC_09_02	-	IMPORT PROC (CICS の場合)	待ち行列タイプ
DXY n PIMC_09_03	-	IMPORT FORM (CICS の場合)	待ち行列タイプ
DXY n PIMC_09_06	-	IMPORT DATA (CICS の場合)	待ち行列タイプ
DXY n PIMC_09_08	-	IMPORT TABLE (CICS の場合)	待ち行列タイプ

表 60. DXY 大域変数とパネル・フィールド名とのマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PIMC_10	-	IMPORT (CICS の場合)	中断
DXY n PIMC_10_01	-	IMPORT QUERY (CICS の場合)	中断
DXY n PIMC_10_02	-	IMPORT PROC (CICS の場合)	中断
DXY n PIMC_10_03	-	IMPORT FORM (CICS の場合)	中断
DXY n PIMC_10_06	-	IMPORT DATA (CICS の場合)	中断
DXY n PIMC_10_08	-	IMPORT TABLE (CICS の場合)	中断
DXY n PLST_01	-	LIST (QUERIES、PROCS、FORMS、ANALYTICS、QMF、TABLES、または ALL)	タイプ
DXY n PLST_ <i>ln</i>	02 - 04	LIST (QUERIES、PROCS、FORMS、ANALYTICS、QMF、TABLES、または ALL)	所有者
DXY n PLST_ <i>ln</i>	05 - 07	LIST (QUERIES、PROCS、FORMS、ANALYTICS、QMF、TABLES、または ALL)	名前
DXY n PLST_08	-	LIST (QUERIES、PROCS、FORMS、ANALYTICS、QMF、TABLES、または ALL)	ロケーション
DXY n PPRT_ <i>ln</i>	02 - 07	PRINT (TSO の場合)	名前
DXY n PPRT_ <i>ln</i> _01	02 - 07	PRINT QUERY (TSO の場合)	名前
DXY n PPRT_ <i>ln</i> _02	02 - 07	PRINT PROC (TSO の場合)	名前
DXY n PPRT_ <i>ln</i> _03	02 - 07	PRINT FORM (TSO の場合)	名前
DXY n PPRT_ <i>ln</i> _04	02 - 07	PRINT PROFILE (TSO の場合)	名前

表 60. DXY 大域変数とパネル・フィールド名との間のマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PPRT_ ln _05	02 - 07	PRINT REPORT (TSO の場合)	名前
DXY n PPRT_ ln _07	02 - 07	PRINT CHART (TSO の場合)	名前
DXY n PPRT_ ln _08	02 - 07	PRINT TABLE (TSO の場合)	名前
DXY n PPR2_01_07	-	PRINT CHART の後に Enter (TSO の場合)	Printer
DXY n PPR3_01_01	-	PRINT QUERY の後に Enter (TSO の場合)	Printer
DXY n PPR3_01_02	-	PRINT PROC の後に Enter (TSO の場合)	Printer
DXY n PPR3_01_03	-	PRINT FORM の後に Enter (TSO の場合)	Printer
DXY n PPR3_01_04	-	PRINT PROFILE の後に Enter (TSO の場合)	Printer
DXY n PPR3_01_08	-	PRINT TABLE の後に Enter (TSO の場合)	Printer
DXY n PPR4_01_05	-	PRINT REPORT の後に Enter (TSO の場合)	Printer
DXY n PPR5_ ln	02 - 07	PRINT (CICS の場合)	名前
DXY n PPR5_ ln _01	02 - 07	PRINT QUERY (CICS の場合)	名前
DXY n PPR5_ ln _02	02 - 07	PRINT PROC (CICS の場合)	名前
DXY n PPR5_ ln _03	02 - 07	PRINT FORM (CICS の場合)	名前
DXY n PPR5_ ln _04	02 - 07	PRINT PROFILE (CICS の場合)	名前
DXY n PPR5_ ln _05	02 - 07	PRINT REPORT (CICS の場合)	名前
DXY n PPR5_ ln _07	02 - 07	PRINT CHART (CICS の場合)	名前

表 60. DXY 大域変数とパネル・フィールド名とのマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PPR5_ <i>ln</i> _08	02 - 07	PRINT TABLE (CICS の場合)	名前
DXY n PPR5_08	-	PRINT (CICS の場合)	待ち行列名
DXY n PPR5_08_01	-	PRINT QUERY (CICS の場合)	待ち行列名
DXY n PPR5_08_02	-	PRINT PROC (CICS の場合)	待ち行列名
DXY n PPR5_08_03	-	PRINT FORM (CICS の場合)	待ち行列名
DXY n PPR5_08_04	-	PRINT PROFILE (CICS の場合)	待ち行列名
DXY n PPR5_08_05	-	PRINT REPORT (CICS の場合)	待ち行列名
DXY n PPR5_08_07	-	PRINT CHART (CICS の場合)	待ち行列名
DXY n PPR5_08_08	-	PRINT TABLE (CICS の場合)	待ち行列名
DXY n PPR5_09	-	PRINT (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_01	-	PRINT QUERY (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_02	-	PRINT PROC (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_03	-	PRINT FORM (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_04	-	PRINT PROFILE (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_05	-	PRINT REPORT (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_07	-	PRINT CHART (CICS の場合)	待ち行列タイプ
DXY n PPR5_09_08	-	PRINT TABLE (CICS の場合)	待ち行列タイプ
DXY n PPR5_10	-	PRINT (CICS の場合)	中断
DXY n PPR5_10_01	-	PRINT QUERY (CICS の場合)	中断
DXY n PPR5_10_02	-	PRINT PROC (CICS の場合)	中断

表 60. DXY 大域変数とパネル・フィールド名とのマッピング (続き)

大域変数名 (<i>n</i> は各国語 ID、 <i>ln</i> は複数行フィールド内の 1 行に関連付けられている ID)	<i>ln</i> 値の範囲	コマンド	フィールド名
DXY n PPR5_10_03	-	PRINT FORM (CICS の場合)	中断
DXY n PPR5_10_04	-	PRINT PROFILE (CICS の場合)	中断
DXY n PPR5_10_05	-	PRINT REPORT (CICS の場合)	中断
DXY n PPR5_10_07	-	PRINT CHART (CICS の場合)	中断
DXY n PPR5_10_08	-	PRINT TABLE (CICS の場合)	中断
DXY n PRST_01	-	RESET	タイプ
DXY n PRSG_01	-	RESET GLOBAL	ALL を入力...
DXY n PRSG_ <i>ln</i>	02 - 11	RESET GLOBAL	大域変数名
DXY n PRUN_ <i>ln</i>	02 - 07	RUN	名前
DXY n PRUN_ <i>ln</i> _01	02 - 07	RUN QUERY	名前
DXY n PRUN_ <i>ln</i> _02	02 - 07	RUN PROC	名前
DXY n PRU3_ <i>ln</i>	01 - 06	RUN QUERY の後に Enter	書式
DXY n PRU4_01	-	RUN PROC の後に Enter	Arg
DXY n PSAV_01	-	SAVE	タイプ
DXY n PSA2_ <i>ln</i>	01 - 06	SAVE DATA	名前
DXY n PSA2_08	-	SAVE DATA	注釈
DXY n PSA3_ <i>ln</i> _01	01 - 06	SAVE QUERY	名前
DXY n PSA3_ <i>ln</i> _02	01 - 06	SAVE PROC	名前
DXY n PSA3_09_01	-	SAVE QUERY	注釈
DXY n PSA3_09_02	-	SAVE PROC	注釈
DXY n PSA4_ <i>ln</i>	01 - 06	SAVE FORM	名前
DXY n PSA4_09	-	SAVE FORM	注釈
DXY n PSET_01	-	SET	タイプ
DXY n PSGL_ <i>ln</i>	01 - 19 (偶数)	SET GLOBAL	変数
DXY n PSGL_ <i>ln</i>	02 - 20 (奇数)	SET GLOBAL	値
DXY n PSHO_01	-	SHOW	名前を入力...

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

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

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

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

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。

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

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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せんが、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。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

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

© (お客様の会社名) (西暦年).

このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_.

プログラミング・インターフェース情報

本書は、QMF のサービスを利用するためのプログラムを作成するプログラミング・インターフェースを記述します。

商標

IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com)[®] は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

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 の米国およびその他の国における登録商標です。

用語集

異常終了 (abend) (abnormal end of task (abend))

実行中に回復機能が解決できなかったエラー状態のために、タスク、ジョブ、またはサブシステムが終了すること。

アドレス・スペース (address space)

コンピューター・プログラムまたはプロセスで使用可能なアドレスの範囲。アドレス・スペースは、物理記憶域、仮想記憶域、またはその両方を指すことがある。

拡張プログラム間通信 (Advanced Program-to-Program Communication)

APPC を参照。

集約関数 (aggregate function)

列中のデータを総計する関数の任意のグループ。この関数は、形式パネル上で取扱コード

AVERAGE、CALC、COUNT、FIRST、LAST、MAXIMUM、MINIMUM、STDEV、SUM、CSUM、PCT、CPCT、TPCT、TCPCT を使用して要求される。

集約変数 (aggregation variable)

集約関数は、「FORM.BREAK」、
「FORM.CALC」、
「FORM.DETAIL」、または
「FORM.FINAL」パネルを使用して報告書に配置される。その値は、報告書の生成時に、切れ目脚注、明細ブロック・テキスト、または最終テキストの一部として出力される。

別名 (alias)

表、視点、データベース、またはニックネームを識別するために使用される代替名。別名は、同じ DB2 システム/サブシステムにおいて、またはリモートの DB2 システム/サブシステムにおいて、表、視点、またはデータベースを参照する SQL ステートメント内で使用できる。

APAR (プログラム診断依頼書) (APAR (Authorized Program Analysis Report))

サポート対象リリースの IBM 提供プログラムにおける問題点に対する修正要求。

許可プログラム機能 (authorized program facility (APF))

z/OS 環境で、制限付き機能の使用を許可されたプログラムの識別を可能にする機能。

アプリケーション・プログラミング・インターフェース (application programming interface (API)) 高水準言語で記述されたアプリケーション・プログラムが、オペレーティング・システムまたは他のプログラムの特定のデータや機能を使用できるようにするインターフェース。

アプリケーション (application)

QMF サービスを使用して、特定のビジネス・プロセス (複数の場合もあり) を直接サポートする機能を提供する、1 つ以上のコンピューター・プログラムまたはソフトウェア・コンポーネント。

拡張プログラム間通信機能 (Advanced Program-to-Program Communication (APPC))

相互接続した複数のシステムが互いに通信してプログラムの処理を分担できるようにする、SNA LU 6.2 プロトコルのインプリメンテーション。

アプリケーション・プラン (application plan)

バインド処理中に生成される制御構造。QMF バージョン 12.1 アプリケーション・プランのデフォルト名は QMF12 です。

アプリケーション・プログラミング・インターフェース (application programming interface) API を参照。

アプリケーション・リクエスター (application requester)

DRDA に対応したリモート・リレーションアル・データベース管理システム (RDBMS) への要求の送信元。QMF が開

始できる唯一のデータベース・タイプであるため、DB2 for z/OS データベースのみがアプリケーション・リクエスターとして機能できる。

アプリケーション・サーバー (application server)

アプリケーション・リクエスターからの要求のターゲット。アプリケーション・サーバー設置場所のデータベース管理システム (DBMS) が、要求に対応する。QMF for TSO が DB2 for z/OS ストアド・プロシージャとして実行されている場合、リモート・サーバーとの接続はサポートされない。

引数 (argument)

実行時に関数またはプロシージャ間で渡されるか、または返される値。

許可 ID (authorization ID) (authorization identifier (authorization ID))

1 組の特権を指定する文字ストリングで、権限の検証に使用される。許可 ID は、オブジェクト、個々のユーザー、組織グループ、関数、またはデータベース・ロールを表している。QMF は、QMF 初期化中に、Q.PROFILES テーブルの列に対して、データベース許可 ID を認証するか、またはオプションで QMF TSO ログオン ID を認証する。

プログラム診断依頼書 (Authorized Program Analysis Report)

APAR を参照。

許可プログラム機能 (Authorized program facility)

APF を参照。

補助表 (auxiliary table)

定義されている表の外側の列を保管する表。基本表 も参照。

基本製品 (base product)

QMF のインストール時に設定される QMF の英語バージョン。他の言語環境は、その言語に関連付けられた各国語機能 (NLF) をインストールすることにより、インストール後に設定される。

基本表 (base table)

SQL CREATE TABLE ステートメントにより作成され、永続データを保持する表。

バイナリー・ストリング (binary string)

コード化文字セットに関連付けられていないため変換されることのない、バイトの連続。例えば、BLOB データ型はバイナリー・ストリングである。CCSID も参照。

バインド (bind)

DBMS プリコンパイラーからの出力を、アクセス・プラン、アプリケーション・プラン、パッケージなどの、使用可能な制御構造に変換すること。

ビット・データ (bit data)

コード化文字セットに関連付けられていないため変換されることのない、CHAR または VARCHAR データ型のデータ。

バッファ・プール (buffer pool)

データ・ページが読み込まれるメモリー領域であり、処理中にデータ・ページの変更が行われて保存される場所でもある。アドレス・スペースも参照。

組み込み関数 (built-in function)

DB2 データベースに不可欠な、厳密に型指定された高性能関数。組み込み関数は、SQL ステートメント内の、その式が有効なあらゆる場所で参照できる。

呼び出し接続機能 (call attachment facility (CAF))

TSO または z/OS バッチで実行されるアプリケーション・プログラムの DB2 for z/OS 接続機能。CAF は DSN コマンド・プロセッサの代替機能であり、実行環境でのより優れた制御を可能にする。

呼び出し接続機能 (call attachment facility)

CAF を参照。

呼び出し可能インターフェース (callable interface)

QMF オブジェクトとサービスへのアクセスを可能にするプログラミング・インターフェース。

カスケード削除 (cascade delete)

DB2 データベース・マネージャーが、削除された親行の下層行をすべて削除することにより参照制約を強制するプロセス。

カタログ (catalog)

表、視点、索引などのオブジェクトの記述を含む表と視点の集合。QMF オブジェクト・カタログ も参照。

コード化文字セット ID (coded character set identifier (CCSID))

特定のコード化スキーム ID のセット、文字セット ID、コード・ページ ID、およびコード化図形文字表現を一意に識別するその他の情報を含む 16 ビットの数値。QMF では、GDDM が提供する表示サービスを使用するため、GDDM アプリケーション・コード・ページは、データベースで使用中の CCSID と一致している必要がある。バイナリー・ストリング も参照。

文字ストリング (character string)

ビット・データ、1 バイト文字、または 1 バイト文字とマルチバイト文字の組み合わせを表す一連のバイト。

チェック制約 (check constraint)

ユーザー定義の制約で、基本表の特定の列に含むことができる値を指定する。制約も参照。

CICS 顧客情報管理システム (Customer Information Control System (CICS))

オンライン・トランザクション処理サービスおよびビジネス・アプリケーションの管理を提供する IBM ライセンス・プログラム。

節 (clause)

SQL では、SELECT 節や WHERE 節など、言語構造のステートメントの特徴的部分。

CM (互換モード)

QMF バージョン 8.1 および QMF バージョン 9.1 のインストール・モードで、QMF オブジェクト・カタログ内の所有者名およびオブジェクト名をそれぞれ 8 文字と 18 文字に制限する。NFM も参照。

コード・ページ (code page)

GRAPHIC 文字へのコード・ポイントの特定の割り当て。特定のコード・ページ内では、コード・ポイントは 1 つの限定した意味のみを持つことができる。コード・ページは、未定義のコード・ポイントの処理方法も識別する。

コード化文字セット ID (coded character set identifier)

CCSID を参照。

共存 (coexistence)

2 つの QMF リリースが同じ DB2 サブシステムに存在する状態。QMF バージョン 12.1 は、QMF バージョン 9.1 (新機能モード) または QMF バージョン 8.1 (新機能モード) とのみ共存できる。

列 (column)

データベース表の垂直方向のコンポーネント。列は、特定のデータ・タイプ (例えば、文字、10 進数、整数) と名前を持つ。

列関数 (column function)

集約関数 を参照。

列折り返し (column wrapping)

1 つの列内で値が複数の行にまたがる、報告書内の値の書式設定。列折り返しは、XML データの表示を必要とする場合など、列に含まれる値の長さがその列の幅を越えるときに、しばしば使用される。

コマンド・インターフェース (command interface)

QMF コマンドを発行するためのインターフェース。コマンド・インターフェースを使用すれば、QMF のもとで実行中の ISPF ダイアログから QMF コマンドを発行できる。このインターフェースを使用して、QMF は ISPF 変数プールを介してダイアログと通信する。

コマンド同義語 (command synonym)

サイトの定義コマンドの verb または verb / オブジェクトの部分。QMF プロファイルでコマンド同義語を定義してアクティブにすると、ユーザーは、通常の QMF コマンドと同じように、QMF コマンド行で同義語を入力できる。

コマンド同義語表 (command synonym table)

行ごとに 1 つの地域別定義コマンドを保管する表。この表の名前をユーザーのプロファイルに保管することにより、ユーザーにコマンド同義語のセットを割り当てる。

比較演算子 (comparison operator)

SQL の比較式で、2 つの値の関係を指定

するために使用される記号。比較演算子には、= (等しい)、<> (等しくない)、< (より小)、> (より大)、<= (より小か等しい)、および >= (より大か等しい) がある。

互換モード (Compatibility Mode)

互換モード (CM) を参照。

コミット (commit)

作業単位で行われたデータベースの変更を他のプロセスが認識できるように、ロックを解除することにより作業単位を終了すること。この処理は、データ変更を永続的なものにする。

連結 (concatenation)

2 つの文字または文字列を結合して 1 つの文字列にすること。

接続 (connection)

データ通信で、エンティティー間で情報を伝達するために設定されるアソシエーション。SQL 接続も参照。QMF for TSO が DB2 for z/OS ストアード・プロシージャーとして実行されている場合、リモート・サーバーとの接続はサポートされない。

定数 (constant)

変化することのない値を指定する言語エレメント。定数は、ストリング定数または数値定数として分類される。

制約 (constraint)

表での挿入、削除、または更新が可能な値を制限する規則。

制御セクション (control section)

CSECT を参照。

コントロール表 (control tables)

QMF が QMF オブジェクトに関する情報の保管、および QMF 演算を管理するために使用する一連の表。QMF オブジェクト・カタログも参照。

関連参照 (correlated reference)

副照会の外側にある表または視点の列の参照。

関連名 (correlation name)

単一の SQL ステートメント内でオブジェクト (表、視点、表関数の参照、ネストされた表の式、データ変更表の参照など) の

直接的な名前として指定され、使用される識別子。関連名は、SQL ステートメントで、同じ基本表への 2 つの別個の参照を許す場合や、代替名を使用したオブジェクトの表現を許す場合に便利である。

制御セクション (control section (CSECT))

プログラマーによって再配置可能単位として指定されるプログラムの一部。この部分のエレメントは、すべて主記憶域ロケーションにロードされる。

現行ロケーション (current location)

QMF セッションが現在接続されているアプリケーション・サーバー。接続が確立されると、このサーバーはすべての SQL ステートメントを処理する。QMF を初期化する場合、現行ロケーションは DSQSDBNM 開始パラメーターを使用して示すことができる。QMF for TSO が DB2 for z/OS ストアード・プロシージャーとして実行されている場合、リモート・サーバーとの接続はサポートされない。

現行オブジェクト (current object)

一時記憶域に保持されている QMF オブジェクト。そのため、使用のたびにデータベースの検索を必要とせずに簡単にアクセスできる。

QUERY、FORM、PROC、PROFILE、REPORT、DATA、および CHART の 7 つの一時記憶域がある。ユーザーは、SHOW コマンドと DISPLAY コマンドを使用して DATA 領域以外のすべての領域にナビゲートできる。一時記憶域も参照。

カーソル (cursor)

セットのデータ行をポイントして選択するためにアプリケーション・プログラムにより使用される名前付き制御構造。

顧客情報管理システム (Customer Information Control System)

CICS を参照してください。

データ・タイプ (data type)

各種データの 1 つを特定する種別。SQL でのデータ・タイプは、列、リテラル、ホスト変数、特殊レジスター、パラメーター、および関数と式の結果を表す属性。

データベース (database)

1 つ以上のアプリケーションに対して提供するために、まとめて保管されている関連したデータ項目や独立データ項目の集合。

データベース管理者 (database administrator)

データベースの設計、開発、運用、セキュリティ、メンテナンス、および使用を担当する個人。

データベース管理システム (database management system (DBMS))

DBMS を参照。

データベース・マネージャー (database manager)

効率的なアクセス、保全性、回復、並行性制御、プライバシー、およびセキュリティを目的に、集中制御、データ独立性、複雑な物理構造を提供することによりデータを管理するプログラム。

データベース・サーバー (database server)

データベース・マネージャーを使用して他のソフトウェア・プログラムまたはコンピューターにデータベース・サービスを提供するソフトウェア・プログラム。

DBCS (2 バイト文字セット) (DBCS (double-byte character set))

各文字が 2 バイトで表現される文字セット。これらの文字セットは、一般的に日本語や中国語のような 1 バイトで表現できない記号が多い各国語で使用される。SBCS も参照。

データベース管理システム (database management system (DBMS))

データベースの作成と編成、変更、およびそのデータベースに保管されているデータへのアクセスについて、制御を行うソフトウェア・システム。

宛先管理テーブル (destination control table (DCT))

CICS で使用されている、一時データ宛先をそれぞれ記述する表。このテーブルは、区画外、区画内、および間接宛先ごとの項目を含む。

デフォルト形式 (default form)

保存された書式が RUN QUERY コマンドで指定されない場合に作成される QMF 書式。

デフォルト値 (default value)

他に値が指定されていない場合に使用される、事前に定義されている値、属性、またはオプション。DB2 表の列データのデフォルト値は、データを変更する SQL ステートメント (INSERT、UPDATE、MERGE など) に DEFAULT キーワードを指定することにより定義できる。

従属行 (dependent row)

親行の親キーの値と一致する外部キーを含む行。外部キーの値は、従属行から親行への参照を表す。

従属表 (dependent table)

オブジェクトに従属する表。例えば、外部キーを持つ表は、対応する主キーを含む表に従属する。

宛先管理テーブル (destination control table)

DCT を参照。

明細ブロック・テキスト (detail block text)

データ内のある行に対応している、報告書の本文中のテキスト。

明細見出しテキスト (detail heading text)

報告書の見出しにあるテキスト。

詳細バリエーション (detail variation)

「FORM.DETAIL」パネルに指定されるデータ形式設定定義。条件に応じて報告書を全体的にまたは部分的に形式設定するために使用できる。

特殊タイプ (distinct type)

共通の表現を組み込みデータ型と共有する、ユーザー定義のデータ型。

分散データ (distributed data)

複数のシステムに保管され、リモート・ユーザーおよびアプリケーション・プログラムで使用できるデータ。

分散データベース (distributed database)

ユーザーからは論理的に完全な、ローカルからアクセスできるデータベースに見えるが、実際にはデータ通信ネットワークで接

続された複数の場所にあるデータベースで構成されているデータベース。

分散リレーショナル・データベース体系

(Distributed Relational Database Architecture™)

DRDA® を参照。

分散作業単位 (distributed unit of work)

分散リレーショナル・データベース処理の形式。ユーザーやアプリケーション・プログラムが 1 つの作業単位内で、複数の場所でデータの読み取りや更新を行える。1 つの作業単位内では、1 つのシステムで実行されている QMF などのアプリケーションが、複数のリモート・データベース・システムに対して、これらのシステムがサポートしている SQL を使用して SQL 要求を送信できる。要求は、3 部構成の表名または視点名を含む QMF コマンドを介して実行される。3 部構成の名前の QMF コマンドは、DB2 (VM または VSE 版) データベースに送信することはできず、QMF for TSO を DB2 for z/OS ストアード・プロシージャーとして開始した場合には使用することもできない。QMF コマンドの 3 部構成の名前は、ラージ・オブジェクト (LOB) データ・タイプを含む表を参照することもできない。

2 バイト文字セット (double-byte character set)

DBCS を参照。

倍精度浮動小数点数 (double-precision floating-point number)

実数に関する 64 ビットの近似的表現。

DRDA (分散リレーショナル・データベース体系 (Distributed Relational Database Architecture (DRDA)))

リモート・データへの透過的なアクセスを提供する形式とプロトコルを定義する体系。DRDA は、アプリケーション・リクエスト関数とアプリケーション・サーバー関数の 2 種類の関数を定義する。

環境 (environment)

関数のパフォーマンスをサポートするために使用される論理および物理リソースの名前付きコレクション。

出口ルーチン (exit routine)

別のプログラムから制御を受け取って特定の関数を実行するプログラム。

Extensible Markup Language

XML を参照。

拡張構文 (extended syntax)

呼び出し可能インターフェース・アプリケーションで QMF SET GLOBAL コマンド、QMF GET GLOBAL コマンド、および特定の関数呼び出しに使用される構文。拡張構文は、アセンブラー、C、COBOL、Fortran、または PL/I 言語で記述された、QMF 呼び出し可能インターフェース・アプリケーションによって使用されるパラメーターを定義する。

フォールバック (fallback)

現行リリースへの移行を試みたか、または完了した後、ソフトウェア・プログラムの前のリリースに戻る処理。

フェッチ (fetch)

データベースまたはファイルから行を検索して QMF DATA オブジェクトを作成する処理。QMF は、DSQSMRFI パラメーターを介した複数行フェッチをサポートしている。

外部キー (foreign key)

リレーショナル・データベースで、別の表の主キーを参照する表内のキー。

図形データ表示管理プログラム (GDDM)

(Graphical Data Display Manager(GDDM))

ディスプレイ・デバイスまたはプリンターに出力するテキストと図形を定義して表示するグラフィック・ソフトウェア。

大域変数 (global variable)

名前付きエンティティ。その値は、QMF セッションの間はデフォルトで保持される。QMF は、大域変数を使用してセッションとデータベース・アクティビティの両方を管理する。大域変数によっては、SET GLOBAL コマンドで設定できるものもあれば、現在の QMF セッションの状態に関する情報を記録するため、SET GLOBAL コマンドでは設定できないものもある。

グラフィック・ストリング (**graphic string**)
一連の 2 バイト文字セット (DBCS) 文字。

図形データ表示管理プログラム (**Graphical Data Display Manager**)
GDDM を参照。

ホスト (**host**)
データ通信構成内の制御または最上位システム。

ハイパーテキスト・マークアップ言語 (**hypertext markup language (HTML)**)
Standard Generalized Markup Language (SGML) に準拠し、主にハイパーテキスト・リンクを含むテキストおよび図形情報のオンライン表示をサポートするために設計されたマークアップ言語。

ハイパーテキスト・マークアップ言語 (**hypertext markup language**)
HTML を参照。

ICU (対話式図表ユーティリティ) (**ICU (Interactive Chart Utility)**)
プログラマーではなくても図表、グラフ、ダイアグラムを表示、印刷、またはプロット可能にする、IBM の図形データ表示管理プログラム (GDDM) のメニュー方式のコンポーネント。

ID 列 (**identity column**)
表に挿入される行ごとに数値を自動的に生成する方法を DB2 データベース・マネージャーに提供する列。ID 列は、AS IDENTITY 節を使用して定義される。ID 列は、1 つの表に 1 つのみ定義できる。

索引 (**index**)
キーの値によって論理的に順序付けされるポインターのセット。索引は、データに対する迅速なアクセスを提供し、表の行についてキー値の固有性を強制できる。

内部結合 (**inner join**)
結合されている両方の表の一致する行のみを含める結合演算の結果。外部結合 も参照。

インストール検査手順 (**installation verification procedure**)
IVP を参照。

統合交換形式 (**Integrated Exchange Format**)
IXF を参照。

対話式図表ユーティリティ (**Interactive Chart Utility**)
ICU を参照。

対話式システム生産性向上機能 (**Interactive System Productivity Facility**)
ISPF を参照。

ISPF (対話式システム生産性向上機能) (**ISPF (Interactive System Productivity Facility)**)
フルスクリーン・エディターおよびダイアログ・マネージャーとして機能する IBM ライセンス・プログラム。アプリケーション・プログラムを記述するために使用され、標準スクリーン・パネルの生成や、アプリケーション・プログラマーと端末ユーザー間での対話式対話の方法を提供する。

IVP (インストール検査手順) (**IVP (installation verification procedure)**)
製品が正しくインストールされていることを検証するプロシージャまたはプログラム。

IXF (統合交換形式) (**IXF (Integrated Exchange Format)**)
さまざまなソフトウェア・プロダクト間で表データを転送するためのプロトコル。

ジョブ制御言語 (**job control language (JCL)**)
オペレーティング・システムに対してジョブを識別し、ジョブの要求を記述するコマンド言語。

ジョブ制御言語 (**job control language**)
JCL を参照。

結合 (**join**)
一致する列の値に基づいて複数の表からデータを検索できるようにする SQL 関係演算。

キー (**key**)
表、索引、または参照制約の記述の中で識別される、列または列の順序付きコレクション。同じ列を複数のキーの一部とすることができる。

キーワード (keyword)

プログラミング言語、人工言語、アプリケーション、またはコマンドの定義済みワードの 1 つ。

キーワード・パラメーター (keyword parameter)

キーワードと続く 1 つ以上の値で構成されるパラメーター。定位置パラメーターも参照。

ラージ・オブジェクト (large object)

LOB を参照。

リンク・エディット (link-edit)

リンケージ・エディターを使用して、ロード可能なコンピューター・プログラムを作成すること。

リンケージ・エディター (linkage editor)

モジュール間での相互参照を解決することによって、また、必要に応じてアドレスを調整することによって、1 つ以上のオブジェクト・モジュールまたはロード・モジュールからロード・モジュールを作成するコンピューター・プログラム。

リテラル (literal)

値がその文字自体によって与えられる文字ストリング。例えば、数値定数 7 は値 7 を持ち、文字定数 'CHARACTERS' は値 CHARACTERS を持つ。

線形プロシージャー (linear procedure)

QMF コマンドまたはコマンド同義語のシーケンス・セット。複数の演算を一度に実行するために使用できる。ロジックを持つプロシージャー (*procedure with logic*) も参照。

線形構文 (linear syntax)

あるプログラムまたはプロシージャーの 1 ステートメントに入力するか、または QMF コマンド行に入力できる QMF のコマンド構文。

ロード・モジュール (load module)

主記憶域にロードして実行するのに適した形式のプログラム。

LOB ラージ・オブジェクト (large object) (LOB (large object))

サイズが 0 バイトから 2 ギガバイト (マイナス 1 バイト) の範囲のサイズを持つ

一連のバイト。バイナリー・ラージ・オブジェクト (BLOB)、文字ラージ・オブジェクト (CLOB、1 バイト文字のみか、または 1 バイトと 2 バイト文字を混合したものを含めることが可能)、および 2 バイト文字ラージ・オブジェクト (DBCLOB) という 3 種類の LOB データ・タイプがある。QMF は、最大 32 KB の LOB 列サイズをサポートする。

ローカル (local)

QMF が現在実行されているシステムにインストールまたは保管されているデータベース、オブジェクト、またはアプリケーションに関する表現。

場所 (location)

分散リレーショナル・データベース・システム内の特定のリレーショナル・データベース・サーバー。各ロケーションは固有のロケーション名を持つ。

ロケーション名 (location name)

データベース・サーバーの固有の名前。アプリケーションは、ロケーション名を使用して DB2 データベース・サーバーにアクセスする。

ロック (lock)

一連のイベントをシリアライズする、またはデータへのアクセスをシリアライズする方法。

ログ (log)

システムで発生するイベントを順番に記述するレコードの集合。

LUW Linux、UNIX、および Windows の省略形。

各国語機能 (National Language Feature)

NLF を参照。

新機能モード (New Function Mode)

NFM を参照。

NFM (新機能モード)

QMF バージョン 8.1 と QMF バージョン 9.1 のインストール・モードの 1 つで、QMF オブジェクト・カタログ内の所有者名およびオブジェクト名をデータベースで許可されている最大長に制限するモード。QMF バージョン 12.1 では、所有者

名とオブジェクト名もデータベースが許可している長さがにできる。互換モード (CM) も参照。

NLF (各国語機能) (NLF (National Language Feature))

QMF で使用できる複数のオプション・フィーチャーのいずれか。NLF によってユーザーは、特定のネイティブ言語で QMF と対話できる。

オブジェクト (object)

名前付きの記憶域スペースであり、スペース、場合によってはデータを記述する特定のセットから成る。オブジェクトは、記憶域内のスペースを占めるすべてのものを指し、ライブラリーやディレクトリーに配置可能で、機密保護機能を設定可能であり、それに対して定義された操作を実行できる。QMF オブジェクト も参照。

外部結合 (outer join)

結合されている両方の表の一致する行を含み、結合されている表の一致しない行のいくつか、またはすべてを保持する結合演算の結果。内部結合 も参照。

パッケージ (package)

プログラムの準備中に作成される制御構造データベース・オブジェクト。静的 SQL ステートメントまたは XQuery 式の実行可能な書式と、動的 SQL ステートメントの実行可能な書式の配置ホルダーの両方を含むことができる。

パネル (panel)

画面上の情報の定様式表示。項目フィールドを含めることができる。

パラメーター (parameter)

関数、コマンド、またはプログラムに渡される値またはリファレンスで、入力または制御のアクションとして機能する。値はユーザーによって、または別のプログラムやプロセスによって提供される。

区画 (partition)

ページ・セットの一部。各区画は、単一の、独立して拡張可能なデータ・セットに対応している。区画は、区画化されたページ・セットの区画数に応じて、最大 1、2、または 4 ギガバイトまで拡張でき

る。特定のページ・セットの区画は、すべて同じ最大サイズを持つ。

プラン (plan)

アプリケーション・プラン を参照。

定位置パラメーター (positional parameter)

他のパラメーターと相対的な、決まった位置に指定しなければならないパラメーター。キーワード・パラメーター も参照。

精度 (precision)

有効数字の総数を表す数値の属性。

述部 (predicate)

比較演算を記述または暗示する検索条件の要素。

1 次許可 ID (primary authorization ID)

z/OS 用 DB2 に対してアプリケーション・プロセスを識別するために使用される許可 ID。

主キー (primary key)

リレーショナル・データベースで、データベース表の 1 つの行を一意的に識別するキー。

特権 (privilege)

SQL では、GRANT ステートメントを処理することによりユーザーに与えられる機能。

プロシージャ (procedure)

1 つ以上のタスクを実行するために使用されるステートメントまたはコマンドのシーケンス・セット。線形プロシージャ とロジックを持つプロシージャ も参照。

ロジックを持つプロシージャ (procedure with logic)

1 つ以上のタスクを実行するステートメントのセット。REXX コメントで開始し、条件付きロジック (REXX を使用)、計算、ビルド・ストリング、および TSO コマンドまたは CICS コマンドを許可するロジックを持つプロシージャ。線形手順 も参照。

プロファイル (profile)

ユーザーのセッションの特性に関する情報が含まれているオブジェクト。

プログラムの一時修正 (program temporary fix)

PTF を参照。

指示照会 (prompted query)

ユーザー提供のパラメーターによって制御されるメニュー方式の照会。

PTF (program temporary fix)

System i[®]、System p、および System z[®] の各製品について、IBM によってテストされた修正。すべてのお客様が利用できる。

QBE (例示照会) (QBE (Query-by-Example))

照会を図形によって作成できるようにする QMF のコンポーネント。

QMF 管理者権限 (QMF administrator authority)

ユーザーに Q.PROFILES 制御表での行の挿入や削除を許可する権限。この権限を持つユーザーは、他のユーザーが所有する QMF 照会、書式、プロシージャーで、SAVE、ERASE、IMPORT、EXPORT、および DISPLAY コマンドを実行できる。その所有者にこれらのオブジェクトをすべてのユーザーと共有するように強制する必要がない。QMF は、初期化中に各ユーザー ID の管理者権限をチェックする。このチェックは DSQUOPTS 出口ルーチンか、または別の任意のプログラムに DSQEC_DISABLEADM 変数を設定することで無効にできる。

QMF 管理者 (QMF administrator)

QMF 管理者権限を持つユーザー。

例示照会 (Query-by-Example)

QBE を参照。

QMF オブジェクト (QMF object)

QMF ユーザーがデータを照会、形式化、および表示するために、または QMF とデータベースとの間の対話を管理するために使用するオブジェクト。QMF オブジェクトは、照会および照会結果データ、書式、プロシージャー、報告書、図表、および QMF プロファイルを含む。各 QMF オブジェクトは、オブジェクトを表示するために使用される名前付き一時記憶域を持つ。報告書と図表を除くすべてのオブジェクトをデータベースに保存できる。報告書と図表は、ユーザー要求時に、特定の QMF 書式仕様をデータベースから返され

た結果データに適用することにより、動的に作成される。一時記憶域 も参照。

QMF オブジェクト・カタログ (QMF object catalog)

QMF 照会、プロシージャー、書式、フォルダー、および分析オブジェクトに関する情報を保管するコントロール表の集合。これらの管理表は、Q.OBJECT_DIRECTORY、Q.OBJECT_DATA、および Q.OBJECT_REMARKS を含む。

修飾子 (qualifier)

QMF オブジェクトについて用いられるときは、オブジェクトの所有者または場所を識別する名前の部分。TSO データ・セットについて用いられるときは、それ以外の名前の部分とピリオドで区切られている名前の部分。例えば、'TCK'、'XYZ'、および 'QUERY' は、すべてデータ・セット名 'TCK.XYZ.QUERY' の修飾子である。

照会 (query)

特定の条件を基準に、データベースの情報を要求すること。例えば、顧客表の残高が 1000 ドルを超える全顧客のリストの要求など。QMF では、照会は Prompted Query、QBE、または SQL 照会パネルから実行された SQL ステートメントも、これらのステートメントが情報を要求しない (SELECT ステートメント) の場合でも参照する。

リソース管理テーブル (resource control table (RCT))

CICS トランザクションと DB2 リソースとの間の関係を定義する DB2 管理テーブル。

リレーショナル・データベース管理システム (relational database management system (RDBMS))

リレーショナル・データベースを編成し、リレーショナル・データベースへのアクセスを提供するハードウェアとソフトウェアの集合。

オンライン・リソース定義 (resource definition online (RDO))

CICS で、CICS を実行しながらユーザーが特定の CICS リソースを対話式に定義

できるようにする機能。具体的には、RDO によってユーザーは端末、プログラム、およびトランザクションを対話式に定義できる。

レコード (record)

行またはその他のデータの記憶表現。

レコード長 (record length)

行やその他のデータを表す記憶の長さ。

再入可能 (reentrant)

すべてのデータベース・スレッドに対して 1 つの共有コピーとして記憶域に常駐できる実行可能コード。再入可能コードは、自己書き換えコードではなく、スレッドごとに別々の記憶域を提供する。

参照制約 (referential constraint)

指定された外部キーの NULL 以外の値は、親表の主キーの値としても表現される場合のみ有効であるという要件。参照制約は、常に従属ファイルの視点から定義される。

リレーショナル・データベース (relational database)

表のセットとして見なすことができ、データのリレーショナル・モデルに従って取り扱われるデータベース。各データベースは、データの論理構造と物理構造を記述するシステム・カタログ表のセット、データベースに割り振られているパラメーター値を含む構成ファイル、およびに実行中のトランザクションとアーカイブ可能なトランザクションによるリカバリー・ログを含む。

リレーショナル・データベース管理システム (relational database management)

RDBMS を参照。

リモート (remote)

QMF が現在実行しているシステム以外のシステムにインストールまたは保管されているデータベース、オブジェクト、またはアプリケーションに関する表現。QMF CONNECT コマンドを使用して、リモート・サーバーでオブジェクト (QMF 照会、書式、プロシージャ、フォルダー、および分析オブジェクト) にアクセスできる。リモート・ロケーションから、表または視点のみにアクセスする場合は、3 部構

成の表名、または視点名で QMF コマンドを使用することもできる。リモート・アクセスは、QMF for TSO が DB2 for z/OS ストアード・プロシージャとして実行されている場合は許されない。

リモート作業単位 (remote unit of work)

QMF などのアプリケーション・プログラムが、1 つの作業単位内で 1 つのリモート・データベースのデータへアクセスが行える分散リレーショナル・データベース処理の形式。接続は、QMF CONNECT コマンドによって確立される。CONNECT コマンドは、QMF for TSO が DB2 for z/OS ストアード・プロシージャとして実行されている場合には使用できない。

リクエスター (requester)

アプリケーション・リクエスターを参照。

リソース (resource)

ロックまたはクレームのオブジェクト。表スペース、索引スペース、データ・パーティション、索引パーティション、またはロジカル・パーティションが挙げられる。

リソース・コントロール表 (resource control table) RCT を参照。

オンライン・リソース定義 (resource definition online)

RDO を参照。

再構造化拡張実行プログラム (Restructured Extended Executor)

REXX を参照。

再構造化拡張実行プログラム (REXX)

(Restructured Extended Executor(REXX))

汎用高水準プログラミング言語。特にパーソナル・コンピューティングのための EXEC プロシージャまたはプログラムに適している。

ロールバック (roll back)

SQL ステートメントによって変更されるデータを、その最後のコミット・ポイントの状態にリストアすること。複数のステートメントを含み、COMMIT ステートメントは含まない照会でエラーが発生した場合、QMF セッション (SET など) に影響するものを除いて、すべてのステートメン

トがロールバックされる。1 つ以上の COMMIT ステートメントを含む照会で障害が発生した場合、正常に実行された前回の COMMIT ステートメントより後で行われた更新はすべてロールバックされる。いずれの場合も、障害発生後に照会は終了する。

ルーチン (routine)

プログラム、またはプログラムによって呼び出される一連の命令。通常、ルーチンは用途が広く、頻繁に使用される。

行 (row)

表の水平方向のコンポーネントで、表の列ごとに 1 つ、一連の値で構成されている。

実行時変数 (runtime variable)

プロシージャまたは照会に含まれる変数。その値は、プロシージャまたは照会の実行時にユーザーによって指定される。実行時変数の値は、現行のプロシージャまたは照会でのみ使用できる。大域変数も参照。

1 バイト文字セット (single-byte character set (SBCS))

各文字が 1 バイトコードで表現されるコード化文字セット。最大 256 文字の表現を許容する 1 バイト・コード・ポイント。2 バイト文字セットも参照。

スカラー関数 (scalar function)

オプションで引数を受け付け、呼び出されるたびに単一のスカラー値を返す SQL 関数。スカラー関数は、SQL ステートメントの中で式が有効なあらゆる場所で参照できる。

スクラッチパッド域 (scratchpad area)

プログラムの実行中に、アプリケーション・プログラムからの情報を保持するため、会話型処理で使用される作業領域。

検索条件 (search condition)

表から行を選択するための基準。検索条件は、1 つ以上の述部で構成される。

2 次許可 ID (secondary authorization ID)

z/OS 用の DB2 で、許可出口ルーチンにより 1 次許可 ID に関連付けられる許可 ID。1 次許可 ID も参照。

セグメント表スペース (segmented table space)

セグメントと呼ばれる同じサイズのページ・グループに分割される表スペース。セグメントは、別の表の行が同じセグメントに保管されないように表に割り当てられる。表スペースも参照。

サーバー (server)

アプリケーション・サーバーを参照。

セッション (session)

ユーザーが QMF を呼び出した時点から、EXIT コマンドを出すまでの、ユーザーと QMF 間で行うすべての対話。

シフトイン文字 (shift-in character)

EBCDIC システムで使用される制御文字 (X'0F')。後続のバイトが、SBCS 文字を表すことを示すため使用される。シフトアウト文字も参照。

シフトアウト文字 (shift-out character)

EBCDIC システムで使用される制御文字 (X'0E') で、後続のバイトが、次のシフトイン制御文字まで、DBCS 文字を表すことを示す。シフトイン文字も参照。

1 バイト文字セット (single-byte character set) SBCS を参照。

単精度浮動小数点数 (single-precision floating-point number)

実数に関する 32 ビットの近似的表現。

SQL (構造化照会言語) (SQL (Structured Query Language))

リレーショナル・データベースにあるデータの定義および操作のための標準化言語。

SQL 許可 ID (SQL authorization ID)

SQLID を参照。

SQL 接続 (SQL connection)

アプリケーション・プロセスとローカルまたはリモート・サーバー、またはデータベース・サーバー間のアソシエーション。リモート作業単位、分散作業単位も参照。

SQL 関数 (SQL function)

SQL ステートメントと SQL PL ステートメントのサブセットを使用することにより完全に実装される関数。

SQL 許可 ID (SQL authorization ID) (SQL ID (SQL authorization ID))

z/OS 用の DB2 で、いくつかの状況で動的 SQL ステートメントの許可のチェックに使用される ID。

SQL 戻りコード (SQL return code)

SQL ステートメントが前回 1 つ以上の警告、またはエラーなしで正常に完了したかどうかを示す SQLSTATE または SQLCODE。

構造化照会言語通信域 (Structured Query Language Communication Area (SQLCA))

アプリケーション・プログラムに、その SQL ステートメントの実行に関する情報、またはデータベース・マネージャーからの要求を提供する一連の変数。エラーが SQL コードに関連する場合、QMF メッセージ・ヘルプ (ヘルプ・キーを押して選択可能)が SQLCA の内容を表示する。

ストアド・プロシージャ (stored procedure)

ホスト言語ステートメントと SQL ステートメントの両方を含む演算を実行するため、SQL CALL ステートメントを使用して呼び出すことができるルーチン。

ストアド・プロシージャ・インターフェース (stored procedure interface)

QMF for TSO へのインターフェース。QMF を DB2 for z/OS ストアド・プロシージャとして開始し、QMF 照会または必要な作業を実行するプロシージャの名前を渡し、トレース出力の結果セットを含む最大 21 個の結果セットを受け取ることができる。QMF for TSO は、DB2 for z/OS ストアド・プロシージャを実行可能なあらゆる製品からこの方法で開始できる。

構造化照会言語 (Structured Query Language)

SQL を参照。

構造化照会言語通信域 (Structured Query Language Communication Area)

SQLCA を参照。

副照会 (subquery)

別の照会の WHERE 節または HAVING 節に表現される SQL 照会一式。

置換変数 (substitution variable)

(1) 値が大域変数または実行時変数のいずれかによって指定される、プロシージャまたは照会内の変数。(2) 値が大域変数によって指定される、QMF 書式の変数。

サブストリング (substring)

文字ストリングの一部。

サブシステム (subsystem)

z/OS 用の DB2 では、リレーショナル・データベース管理システム (RDBMS) の個別インスタンス。

表 (table)

リレーショナル・データベースで、固定数の列で構成されるデータベース・オブジェクト。順序付けられていない行セットの保管に使用される。基本表 も参照。

表スペース (table space)

データベース内の記憶域の論理単位。z/OS 用の DB2 では、表スペースは、ページ・セットであり、1 つ以上の表を含むことができる。DB2 for Linux, UNIX, and Windows では、表スペースはコンテナの集合であり、表のデータ、索引、長フィールド、および LOB の部分は、同じ表スペースに保管することも、別々の表スペースに保管することもできる。

一時記憶域 (temporary storage)

ユーザーが作業中に QMF オブジェクトを一時的に保管するために使用する記憶域。データベースをそれ以上検索せずに、簡単にオブジェクトにアクセスできる。QUERY、DATA、FORM、PROC、REPORT、CHART、または PROFILE の 7 つの一時記憶域がある。照会結果データ (DATA オブジェクト) を除き、これらの記憶域にある QMF オブジェクトは、SHOW コマンドの後に一時記憶域の名前を指定して実行することにより表示できる。DATA 領域の内容は直接表示できないが、ユーザーは、SHOW REPORT または SHOW CHART コマンドを発行して、FORM 領域の現行書式の仕様で書式化された照会データを表示できる。『QMF オブジェクト』、『現行オブジェクト』も参照。

一時記憶域キュー (temporary storage queue)

CICS の、あらゆる順序で読み取りおよび再読み取りが可能なデータ項目のキュー。キューは、タスクにより作成され、同じタスクまたは別のタスクによって削除されるまで存続する。一時データ・キュー も参照。

スレッド (thread)

DB2 の構造であり、アプリケーションの接続を記述し、その進行を追跡し、リソース関数を処理し、さらに DB2 リソースとサービスへのアクセス可能性の範囲を決定する。ほとんどの DB2 機能は、スレッド構造のもとで実行される。

3 部構成の名前 (three-part name)

表、視点、または別名のフルネーム。場所名、許可 ID、およびオブジェクト名で構成され、それぞれピリオドで区切られている。3 部構成の名前を含む QMF コマンドは、DB2 for z/OS データベースからのみ開始することができ、DB2 (VM/VSE 版) を除くすべてのデータベースに送信できる。QMF for TSO が DB2 for z/OS ストアード・プロシージャとして開始されている場合、3 部構成の名前を持つ QMF コマンドはサポートされない。

タイム・シェアリング・オプション (Time Sharing Option)

TSOを参照してください。

トレース (trace)

コンピューター・プログラムまたはトランザクションの処理の記録。トレースから収集された情報は、問題とパフォーマンスの評価に使用される。

トランザクション (transaction)

1 つ以上のアプリケーション・プログラムで構成される処理単位。1 つ以上のオブジェクトに影響し、単一の要求によって開始される。

一時データ・キュー (transient data queue)

CICS 記憶域。オブジェクトは、後続の内部処理または外部処理のために、ここに保管される。一時データ・キュー も参照。

トリガー (trigger)

単一の基本表または視点に関連付けられたデータベース・オブジェクト。規則を定義

する。規則は一連の SQL ステートメントで構成され、関連付けられた基本表または視点で挿入、更新、または削除のデータベース操作が発生すると実行される。

TSO (タイム・シェアリング・オプション) (TSO (Time Sharing Option))

ユーザーがシステムと対話式に作業できるようにする、z/OS オペレーティング・システムの基本エレメント。

2 フェーズ・コミット (two-phase commit)

外部サブシステム内のリカバリー可能リソースをコミットする 2 段階の処理。最初の段階では、データベース・マネージャー・サブシステムがポーリングされ、コミットの準備ができていることが確認される。サブシステムすべてがポジティブに応答した場合、データベース・マネージャーはこれらのサブシステムにコミットするよう指示する。

ユーザー定義関数(user-defined function (UDF))

CREATE FUNCTION ステートメントを使用して DB2 データベース・システムに定義される関数。定義後は、SQL ステートメントで参照できる。UDF は SQL 関数でも外部関数でもよい。

ユニコード (Unicode)

文字エンコード規格。世界の共通語で記述されたテキストに加え、一部の古典的および伝統的テキストの交換、処理、および表示をサポートする。ユニコード規格には、ISO 10646 で規定された 16 ビットの文字セットがある。

リカバリー単位 (UR) (unit of recovery (UR))

整合点と整合点の間にある作業単位内の一連の操作。

作業単位 (UOW) (unit of work (UOW))

アプリケーション・プロセス内のリカバリー可能な一連の操作。アプリケーション・プロセスは任意の時点で単一の作業単位であるが、その活動中には、コミット操作またはロールバック操作の結果として多くの作業単位が関わる場合がある。マルチサイト更新操作では、単一の作業単位が複数の回復単位を含む場合がある。複数のステートメントを含み、COMMIT ステートメントは含まない QMF SQL 照会において

は、すべてのステートメントが単一の作業単位を構成するため、セッションに影響するステートメント (SET など) を除くすべてのステートメントが障害時にロールバックされる。複数のステートメントを含み、1 つ以上の COMMIT ステートメントを含む QMF SQL 照会においては、作業単位は COMMIT ステートメントと、照会の先頭または最後の COMMIT ステートメントまでさかのぼるすべてのステートメントで構成される。障害が発生した場合、正常に実行された前回の COMMIT ステートメントより後で行われた更新はすべてロールバックされる。

ユーザー定義関数 (user-defined function)

UDF を参照。

視点 (view)

基礎となる一連の表に保管されているデータに基づく論理表。視点によって返されるデータは、基礎表で実行されている SELECT ステートメントによって決定される。

XML (Extensible Markup Language)

Standard Generalized Markup Language(SGML) に基づくマークアップ言語を定義するための標準メタ言語。

z/OS 64 ビットの実記憶を使用する IBM メインフレーム・オペレーティング・システム。

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]

アセンブラー

- 機能呼び出し 154
- 言語インターフェース 153
- 高水準アセンブラー (HLASM) 153
- サンプル・プログラム 156
- マクロ 215
- 連絡域 163
- CICS 153
 - サンプル・プログラム 156
 - z/OS 164
- TSO サンプル・プログラム 160, 165

アプリケーション

- インプリメンテーション・メソッド 3
- 開始 2
- 開発 1
- コマンド 1
 - 概要 45
 - 処理 17
 - INTERACT 48
- コマンド同義語 2
- 制御 1
- タイプ 1
- データ・レコード 109
- デバッグ 149
- プロシージャ 7
- プロシージャ、ロジックを持つ 3
- 2 か国語使用の 41
- CICS 環境 4
- ISPF 要件 33
- アプリケーションのアテンション・フラグ 218
- アプリケーションのデバッグ
 - ISPF の使用 37
 - PDF ダイアログ・テスト 37
- アプリケーション・プラン
 - デフォルト ID、V12.1 57
- アプリケーション・プログラミング・インターフェース
 - コマンド・インターフェース
 - 参照: コマンド、インターフェース
 - ストアード・プロシージャ (TSO のみ)
 - 参照: ストアード・プロシージャ・インターフェース
 - 呼び出し可能インターフェース
 - 参照: 呼び出し可能インターフェース
- 移行情報 103
- 一時記憶域 23

一時記憶域 (続き)

- 上書きの確認 240
- キュー 146
- 制限 3
- トレース用の大域変数 225
- 変更 23
- CICS
 - 印刷関連の大域 225
 - 予備ファイルに関連する大域 225
- 一時記憶域の上書き、回避 240
- 一時データ
 - 一時記憶域キューとの対比 146
 - 印刷関連の大域 225
 - トレース用の大域変数 225
 - 予備ファイルに関連する大域 225
- インポート 73, 103
 - オブジェクト・レベル情報 103
 - 指示照会オブジェクト 140
 - 書式オブジェクト 113
 - QMF の外で作成された表 75
- エクスポート
 - オブジェクトのタイプ 73
 - 記憶域についての考慮事項 146
 - 指示照会 103
 - 書式 103
 - 書式オブジェクト 73
 - 図表 143
 - 対保管 74
 - データおよび表 75
 - データ・タイプの幅計算 75
 - データ・タイプ用 16 進コード 75
 - ファイルの最大サイズ 143
 - プロシージャ 142
 - 編集コードに使用されるキーワード 113
 - 報告書 103
 - CSV データ 101
 - LOB データ 99
 - QBE 照会 143
 - SQL 照会 142
- エクスポート後の XML ファイル内のスキーマ定義 99
- エクスポート後の XML ファイルの検証 99
- エクスポート後の XML ファイルのスタイル・シート 99
- エクスポート後のオブジェクトでの制御域 105
 - 書式ファイルのレコード 105
 - 報告書ファイルのレコード 105
 - T レコード 107
- エクスポート後のデータ・タイプの 16 進コード 113
- エクスポートされた XML 内の CDATA タグ 99
- エクスポートされた XML 内のセル・タグ 99
- エクスポートされたファイルの表レコード 81
- エクスポート時の 2 次スペース割り振り 240

- エクスポート書式に表示される BINARY キーワード 113
- エクスポート書式に表示される CHAR キーワード 113
- エクスポート書式に表示される DATE キーワード 113
- エクスポート書式に表示される GRAPHIC キーワード 113
- エクスポート書式に表示される NUMERIC キーワード 113
- エクスポート書式に表示される TIME キーワード 113
- エクスポート書式に表示される TIMEST キーワード 113
- エクスポート書式に表示される TSTMP TZ キーワード 113
- エクスポート書式に表示される UNKNOWN キーワード 113
- エラー時のシグナル命令 13
- オブジェクト
 - 共用 240
 - 最終使用日 240
- オブジェクト (object)
 - 現行に関連する大域変数 218
 - 最終使用日 229
 - タイプ
 - リストの順序付け 229
- オブジェクトの終わりレコード (E) 109
- オブジェクトの最終使用日
 - リストのソート 229
 - RUN、SAVE、IMPORT に限定 240
- オペレーティング・システム、大域変数 218
- オンライン・ヘルプ
 - QMF メッセージ・ヘルプ
 - 正の SQL コードの表示 229

[カ行]

- カーソル
 - 固定、使用可能化 240
- カーソル (cursor)
 - 状況 218
- 拡張記憶域、予備データに使用
 - 必須のプログラム・パラメーター 57
- 拡張記憶域、予備データ用に使用
 - 設定量 240
 - 必須のプログラム・パラメーター 57
- 拡張形式 10 進浮動小数点データ
 - 参照： DECFLOAT データ・タイプ
- 拡張構文
 - SET GLOBAL コマンド 54
- 確認パネル
 - 一時記憶域の上書き 240
 - リセット報告書 240
- 仮想記憶域
 - 参照： 記憶域
- 各国語フィーチャー (NLF)
 - 参照： 多文化サポート
- 貨幣価値、通貨記号の変更 229
- 紙送り制御文字、抑止 240
- 環境大域変数 218
- 管理者権限、大域変数 218, 240
- キーワード 57
 - 編集コード用、エクスポート・ファイル内 113
 - START コマンド 57

- 記憶域
 - エクスポートされるファイルの最大サイズ 143
 - エクスポートでの指定 240
 - エクスポートについての考慮事項 146
 - XML データ 102
 - 拡張、予備データ (TSO) 57
 - 予備データ
 - 拡張された仮想 240
- 機能呼び出し
 - C 169
 - DSQCIC 169
 - DSQCICE 169
- キュー
 - 印刷用の大域変数 225
 - 予備データ用の大域変数 225
- 行
 - 最大長 240
- 強調表示規則 viii
- 強調表示の規則 viii
- 切れ目パネル 113
- 区切り文字
 - SQL 照会内のステートメント間 240
- 形式 124
 - エンコード
 - 定義 73
 - 指示照会オブジェクト 132
 - 書式オブジェクト 113
 - データのエクスポート 75
 - 表 75
 - ヘッダー・レコード 75
 - 報告書オブジェクト 124
 - 制御情報なし 129
 - 列データ 81
 - IXF 73, 81
- 結果セット
 - エクスポートされた XML 内のレコード 99
 - 報告書用を指定 240
- 現行ロケーション 45
- コード、SQL
 - 参照： SQL コード
- コード化文字セット ID、XML エクスポート 99
- ご意見、IBM への送付 x
- 降順リスト 229
- 構文解析サービスおよび XML のエクスポート 99
- 構文図、読み方 viii
- 互換モードおよび複数行フェッチ 57
- コマンド
 - アプリケーション 45
 - インターフェース 2
 - サンプル・プログラム 25
 - 説明 25
 - プログラムからの呼び出し 26
 - 戻りコード 27
 - 要件 4
 - SELECT サービス 26
 - 各国語、設定 240

コマンド (続き)

- 環境 31
- 言語変数 41
- サポートする大域変数 218
- システム固有 9
- 長さ 17
- メッセージ出力を保管する大域 226
- 戻りコード 13
- 2 か国語使用のアプリケーション 43
- INTERACT 48
- RUN 10
- SET GLOBAL 54

コマンド同義語

- 作成 71
- 定義 224
- 例 2
- NLF 表 39

[サ行]

サービス情報 viii

サービス・トレース 63

- PL/I の例 64

最終使用日、オブジェクト・リスト 229, 240

最大長

- SQL 照会 240

サブシステム ID、大域変数 218

サポートされている言語

翻訳

- 参照：多文化サポート

呼び出し可能インターフェースのプログラミング 4

サポート情報 viii

指示照会

- ヘッダー・レコード 132

指示照会 (prompted query)

- インポート / エクスポート・ファイル仕様 143
- エクスポート 103, 132
 - ファイルの最大サイズ 143
- エクスポート形式 132
- データ・レコード 132
- 表番号 132
- フィールド番号 132

視点

- LIST コマンド、関連する大域 240

終了フラグ変数 218

順序

- LIST コマンド項目、変更 229

照会

実行

- 参照：RUN QUERY コマンド

実行後の報告書

- 抑制 229

大域に保管されるサブタイプ 218

分離レベル 240

変換

- 参照：CONVERT QUERY コマンド

照会 (続き)

- 見積もられたコスト、使用不可にする 229

モデル大域変数 218

CALL ステートメント

- 報告書の結果セットを指定 240

- 参照：CALL ステートメント

QUERY パネルの上書きを回避 240

SQL

- 参照：SQL 照会

照会のコスト見積、使用不可にする 229

照会の分離レベル 240

昇順リスト 229

初期プロシージャ

- 作成 7

- 名前の指定 7

- 保管 7

2 か国語使用のアプリケーション 43

CONNECT コマンド 7

書式

- インポート 122

- エクスポート 103

- ファイルの最大サイズ 143

- エクスポート時のデータ・タイプ・キーワード 113

- エクスポート時のデータ・レコード 113

- エクスポート時の見出しレコード 113

パネル

- 関連する大域 218

- 表番号 113

- フィールド番号 113

- 変換 113

FORM.COLUMNS

- 列名とラベル 229

FORM.MAIN

- 列名とラベル 229

LAYOUT コマンド 113

SAVE、EXPORT、IMPORT の多文化サポート 240

所有者名

- 関連する大域変数 218

- リストの順序付け 229

- LIST コマンドのデフォルト 240

新機能モードおよび複数行フェッチ 57

数値

データ

- 通貨記号、変更 229

スクロール量、設定 229

図形データ形式 (GDF) 143

ストアード・プロシージャ

QMF for TSO の開始

- 大域変数のサポート 218

ストアード・プロシージャ (stored procedure)

- 報告書の結果セットを指定 240

ストアード・プロシージャ・インターフェース

- リモート・データ・アクセスの制限 9

- DSQSDBLG パラメーター 57

図表オブジェクト 143

制御情報

- 報告書から削除 129
- 正の SQL コード、メッセージ・サポート 229
- 正の SQL コードの通知 229
- セッション、状態を記録する変数 218
- セッション大域変数 251
- 線形プロシージャ
 - 抑制 51
 - STOPPROC オプション 51
- 前提となる DB2 (z/OS 版) の知識 vii
- 操作モード
 - 表示する大域変数 218
- 相対コスト見積パネル、使用不可にする 229

[タ行]

大域変数

- アプリケーション・トレース・レベル 218
- 一時記憶域の上書き 240
- 印刷出力内の紙送り制御文字 240
- オブジェクトの最終使用日 240
- 管理者権限 218, 240
- 規則 55
- 起動プロシージャ、再実行 240
- クラス
 - ストアード・プロシージャ・インターフェース 218
 - DSQAO 218
 - DSQAP 224, 225
 - DSQCM 226
 - DSQCP 227
 - DSQEC 240
 - DSQQC 250
 - DSQQM 251
 - DXY 251
- 現行オブジェクト 218
- 現行パネル名 218
- 現在の書式パネル 218
- コマンド同義語の定義 224
- コマンド・サポート 218
- 作成 54
- サブシステム ID 218
- 照会サブタイプ 218
- 照会の分離レベル 240
- 初期化時の設定 218, 240
- 所有者名 218
- スクロール量 229
- ストアード・プロシージャの結果セット 240
- ストアード・プロシージャ・インターフェース 218
- 正の SQL コードの通知 229
- 設定 54
 - SET GLOBAL コマンド 54
- 設定/表示 218
- 相対コスト見積パネル 229
- 多文化サポート 218, 240
- 通貨記号 229
- データベース・カーソル状況 218

大域変数 (続き)

- データベース・マネージャー 218
 - 同時アクセスの解決 240
 - 取り出された行、番号 218
 - のリスト 217
 - バージョン/リリース 218
 - バッチと対話式操作 218
 - パネル ID、表示 229
 - 複数ステートメント SQL 照会 240
 - プロファイルの CASE パラメーター 224
 - プロファイルの CONFIRM パラメーター 224
 - プロファイルの LANGUAGE パラメーター 224
 - プロファイルの LENGTH パラメーター 224
 - プロファイルの MODEL パラメーター 224
 - プロファイルの PRINTER パラメーター 224
 - プロファイルの RESOURCE GROUP パラメーター 224
 - プロファイルの SPACE パラメーター 224
 - プロファイルの TRACE パラメーター 224
 - プロファイルの WIDTH パラメーター 224
 - 変数の作成 54
 - メッセージ出力 226
 - モデル照会 218
 - ユーザー・アテンション・フラグ 218
 - 予備データの拡張記憶域 240
 - リスト順序 229
 - リセット報告書パネルの表示 240
 - リモート・ロケーション名 218
 - 列ラベルと名前 229
 - ローカル・データベース名 218
 - 32 KB を超えた SQL 照会 240
 - CICS
 - 印刷 225
 - トレース 225
 - 予備データ 225
 - CONNECT ID 218
 - DBCS サポート 218
 - EXPORT コマンドの記憶域 (TSO) 240
 - EXPORT の列名の長さ 75, 229
 - LIST コマンド
 - サポートする視点 240
 - OWNER のデフォルト 240
 - QMF 報告書の行の長さ 240
 - RUN QUERY 後の報告書表示 229
 - RUN QUERY メッセージ 251
 - RUW を介して使用する QMF 224
 - SAVE コマンドの SHARE パラメーター 240
- ### 対話式 QMF
- 操作モードの大域変数 218
- ### 対話の切り替え (DSQAO_INTERACT) 50
- ### 対話モード
- 初期プロシージャ 7
 - GDDM ICU 48
 - QMF 48
- ### 大/小文字混合パスワード 224
- ### 多文化サポート
- 関連する大域変数 218, 240

多文化サポート (続き)
書式 (SAVE/EXPORT/IMPORT) 240
置換変数 10
値の割り当て 10
構文 10
大域変数の設定 10
REXX プログラム呼び出し 15
注意
特記事項 263
注釈
アプリケーション・データ・レコード 109, 122
エクスポート・フォーマット 122
長形式 10 進浮動小数点データ
参照: DECFLOAT データ・タイプ
追跡可能性 63
通貨記号、変更 229
データ
エクスポート 75
オブジェクト (object)
形式 75
IXF のエクスポート後の形式 81
タイプの幅 75
レコードのエクスポート 75, 99
D レコード 81
データおよび表をエクスポートする場合の QMF 形式 75, 143
データのロック
エスカレーションの回避 240
同時アクセスの解決のオプション 240
データベース
非コミット読み取りとカーソル固定 240
データベース (database)
サブシステム ID、大域変数 218
SQL コード
参照: SQL コード
データベース・マネージャー、タイプを保管する大域 218
データベース・リモート接続 9
データ・セット、エクスポートでの定義 240
データ・タイプ、エクスポートについての考慮事項 75, 113
データ・レコードの形式
XML データ・タイプ 99
ディレクトリー・ブロック、エクスポート時の指定 240
デバッグ用の A オプション 150
デバッグ用の L オプション 149
同義語、コマンド 71
同時アクセスの解決 240
トレース
アプリケーション・トレース・レベル 218
オフにする 150
正の SQL コード 229
設定 149
大域変数 225
トレース定義の作成 37
ファイルの割り振り 151
プロファイル・パラメーター 224
例 151
A オプション 150

トレース (続き)
ISPF コマンド 37
L オプション 149

[ナ行]

名前
リストの順序付け 229
列用、データベース・ラベルへの変更 229
LIST コマンドをサポートする視点、大域 240

[ハ行]

バージョン番号
エクスポート時の IXF バージョン 81
QMF、保管する大域 218
パスワード、大/小文字混合 9
バッチ QMF
操作モードの大域変数 218
バッチ QMF セッション、大域 218
パネル 48
確認
一時記憶域の上書き、回避 240
現行の 48
相対コスト見積、使用不可にする 229
対話式 48
名前
関連する大域変数 218
ID
表示 229
パフォーマンス
同時アクセスの解決のオプション 240
バリエーション、FORM.DETAIL
番号を保管する大域変数 218
バリエーション・パネル 113
引数 12
非コミット読み取り、使用可能化 240
表
エクスポート 75
オブジェクト (object)
インポート 75
インポート / エクスポートの規則 102
インポート / エクスポート・ファイル仕様 143
処理 75
記述レコード (T) 81, 107
行レコード (R) 108
作成、QMF の外での 81
指示照会の番号 132
書式番号 113
報告書番号 124
LIST コマンド
関連する大域変数 240
表エディター内の + 符号、変更 227
ファンクション・キー
定義が保管される場所 224

- フィードバック、IBM への送付 x
- フェッチ、複数行 57
- 不完全なデータ・オブジェクト
 - プロンプト・パネル 102
 - リセット報告書パネルの使用可能化 240
- 複数行取り出し 57
 - 3 部構成の名前の障害 57
- 複数言語環境 44
- 複数ステートメント照会 240
- プログラミング・インターフェース情報 264
- プログラム呼び出し 17
- プロシージャ
 - エクスポート
 - ファイルの最大サイズ 143
 - 起動、再実行 240
 - 初期化、変数の設定 218
 - ストアード・プロシージャ
 - 参照：ストアード・プロシージャ
 - PROC パネルの上書きを回避 240
- プロダクト・インターフェース・マクロ 215
- プロファイル
 - 関連する大域変数 224
 - 保管されていない値の上書きを回避 240
- ヘッダー・レコード
 - オブジェクト・レベル 103
 - 形式 75
 - 書式オブジェクト 113
 - フィールド 103
 - IXF 81
 - XML エクスポート 99
- 別名
 - LIST の別名を検索する視点 240
- 変換可能なアプリケーション 44
- 変更日、リストの順序付け 229
- 編集コード、エクスポートで表示されるキーワード 113
- 変数
 - エラー処理 23
 - 規則 55
 - グローバル 10, 217
 - 置換 10
 - 設定 10
 - 置換 10
 - 入力プロンプト 11
 - プール 17
 - rc 13
- ホーム・パネル 7
- 報告書
 - 印刷
 - 紙送り制御文字 240
 - エクスポート 75
 - ファイルの最大サイズ 143
 - エクスポート形式 124
 - エクスポートの使用 124
 - エクスポート例 124
 - エクスポート・レコード 124
 - オブジェクト (object) 2

- 報告書 (続き)
 - エクスポート形式 124
 - 表番号 124
 - フィールド番号 124
 - 横方向 124
- 行 (L) レコード 110
- 行データ 108
- 行の長さ、設定 240
- サンプル・ヘッダー 124
- 照会実行後の抑止 229
- ストアード・プロシージャの実行
 - 参照：ストアード・プロシージャ (stored procedure)
- 制御情報なしでのエクスポート 129
- テキストの表示 51
- パネル 2
- 表データ 107
- 表番号 124
- フィールド番号 124
- ミニ・セッション 68
- 横方向 124
- 横方向エクスポート 130
- レコード 103
- HTML 128

報告書内のドル記号、変更 229

[マ行]

- マクロ、プロダクト・インターフェース 215
- 見出し、列
 - 参照：ヘッダー
- 見積もられた照会コスト、使用不可にする 229
- ミニ・セッション
 - 報告書 68
 - 無効なコマンド 68
 - 有効なコマンド 68
- メッセージ
 - 関連する大域変数
 - 正の SQL コードのメッセージ・サポート 229
 - 前のコマンドからのメッセージ 226
 - RUN QUERY メッセージ 251
- 戻りコード
 - コマンド・インターフェース 27
 - ゼロでない 13
 - メッセージ 13
 - 呼び出し可能インターフェース 20

[ヤ行]

- ユーザー ID、データベース接続 218
- ユーザー定義編集コード
 - エクスポート書式に表示されるキーワード 113
- ユーザー・アテンション・フラグ 218
- ユーロ通貨記号、使用可能化 229
- 呼び出し可能インターフェース
 - アプリケーションの実行 22

呼び出し可能インターフェース (続き)

- アプリケーションのデバッグ 149
- 言語 4, 17, 153
- コマンド 21
- コマンド処理情報 17
- サンプル・プログラム 3
 - アセンブラー 156
 - C 170
 - COBOL 181
- 説明 17
- プログラム 3
- マクロ 215
- 戻りコード 20
- 連絡域 3
 - エラー処理 23
 - 定義 18
 - フィールドの設定 18
 - C 173
 - COBOL 182
- ロジックを持つプロシージャからの呼び出し 15
- CICS のもとでの実行 23
- COBOL 178
- FORTRAN 188
- GET GLOBAL コマンド 47
- ISPF 3
- PL/I 198
- REXX
 - 使用 3
 - 説明 208
 - で呼び出す 7
 - 連絡変数 208
- START コマンド 3
 - 構文 56
 - QMF の開始 22
- 予備ファイル
 - サポートする大域変数 225
 - TSO での拡張記憶域の使用 240

[ラ行]

- リセット報告書パネル、使用可能化 240
- リソース競合、削減 240
- リモート作業単位
 - コマンドの動作 45
- リモート・サーバーとの接続
 - ストアド・プロシージャ・インターフェースの制限 45
- リモート・データ・アクセス
 - CONNECT のユーザー ID 218
- レコード
 - アプリケーション・データ (*) 109
 - 可変長形式 105
 - 形式 75
 - 固定形式 103
 - データ (D) 81
 - データ値 (V) 106
 - データ継続 (C) 113

レコード (続き)

- 表の記述 (T) 81
- ヘッダー 103
- 列 (C) 81
- 列
 - エクスポート時の幅 75
 - データ形式 81
 - 表エディター内のデフォルトのインディケータ 227
 - ヘッダー
 - ラベルと名前 229
 - C レコード 81
 - DESCRIBE コマンド
 - 参照: DESCRIBE コマンド
 - EXPORT の名前の長さ 75, 229
 - 列見出し用のラベルと名前 229
- レポート用の HTML 形式
 - エクスポート 128
 - ファイルの最大サイズ 143
- 連絡域
 - 定義 18
 - COBOL 178, 182
 - FORTRAN 194
 - PL/I 203
- ログ、トレース 229
- ロケーション名
 - 保管する大域変数 218
 - 参照: リモート・データ・アクセス

[数字]

- 1 次スペース割り振り、エクスポートでの 240
- 10 進浮動小数点データ・タイプ
 - 参照: DECFLOAT データ・タイプ
- 2 か国語使用のオブジェクト 41
- 64BIT オプション、DSQSPTYP パラメーター 57

A

- ADD コマンド
 - 表エディターの確認 227
- ADDRESS コマンド 15, 31
- ARG ステートメント 12

B

- BIGINT データ・タイプ
 - エクスポート 75
 - 編集コード・キーワード、エクスポートされた書式 113
 - 列データ形式、IXF 81
 - エクスポート時の列幅 75
- BINARY データ・タイプ
 - エクスポート 75
 - 編集コード・キーワード、エクスポートされた書式 113
 - 列データ形式、IXF 81
 - エクスポート時の SQLTYPE コード (QMF 形式) 75

BINARY データ・タイプ (続き)
エクスポート時の列幅 75

C

C 言語

インターフェースの要件 169
機能呼び出し 169
サンプル・プログラム 170
呼び出し可能インターフェース 167
連絡域 167
マッピング 167
DSQCOMM 173
CICS 175
ISPF 176
TSO 176

CALL ステートメント

報告書の結果セットを指定 240

CALL 命令 15

CANCEL コマンド

表エディターの確認 227

CHANGE コマンド

表エディターの確認 227

CHAR データ・タイプ

エクスポート
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

CICS 環境

アセンブラー 4
z/OS 要件 164
関連する大域変数 225
データ・キュー 4
一時データ・キュー 146
IXF 形式 81
QMF オブジェクトを転送するための使用 74
プログラム開始パラメーターの指定変更 23
呼び出し可能インターフェース 4
領域 23

COBOL プログラム 185

CONNECT コマンド 9

DB2 対話 23

CLEAR コマンド

表エディターの確認 227

COBOL

機能呼び出し 179
区切り文字 184
サンプル・プログラム 181
実行の要件 184
マクロ 215
呼び出し可能インターフェース 178
連絡域 178
CICS 185
DSQCOMM 182
ISPF 186

COBOL (続き)

TSO 186

CONNECT コマンド

初期プロシージャ 7
説明 45
大/小文字混合パスワード 9, 224
プロシージャ 9
例 45
CONNECT ID の大域変数 218
DB2 (VM または VSE 版) 9

CONVERT QUERY コマンド

最終使用日の更新を制限 240
大域変数 250

CSV エクスポート形式

ファイルの最大サイズ 143

CSV データ

エクスポート 101

D

DATA オブジェクト

関連する大域変数 218
不完全、リセット報告書パネルの使用可能化 240

DATAFORMAT パラメーター値、EXPORT コマンド

QMF 75, 229

DATE データ・タイプ

エクスポート
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の ISO 形式 75
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

DB2 (VM または VSE 版)

リモート接続 9
CONNECT コマンド 9

DBCS サポート

関連する大域変数 218
デフォルトのインディケーターの変更、表エディター 227
NULL インディケーターの変更、表エディター 227

DECFLOAT データ・タイプ

エクスポート
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

DECIMAL データ・タイプ

エクスポート
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

DELETE コマンド

表エディターの確認 227

DESCRIBE コマンド

サポートする視点 240

DISPLAY コマンド
最終使用日の更新を制限 240
DSQ1SCHEM スキーマ・ファイル 99
DSQ1STSH スタイル・シート・ファイル 99
DSQABFA 160
DSQABFAC 156
DSQAO 大域変数 218
DSQAP 大域変数 224, 225
DSQCIA 154
DSQCIX サブルーチン 210
DSQCM 大域変数 226
DSQCOMM
アセンブラー 163
エラー処理 23
定義 18
フィールドの設定 18
メッセージ・テキスト 153
C 167, 173
COBOL 178
DSQCOMMMA 163
DSQCOMMCC 173
DSQCP 大域変数 227
DSQDEBUG トレース・ログ
正の SQL コードのログギング 229
参照: トレース
DSQEC 大域変数 240
DSQEC_NLFCMD_LANG 変数 41, 240
DSQQC 大域変数 250
DSQQM 大域変数 251
DSQSDBLG パラメーター 57
DSQUOPTS 初期化ルーチン 240
DXY 大域変数 251
D, DC, DZ, DZC 編集コード
通貨記号、変更 229

E

EDIT コマンド 36
デフォルト・エディター 240
END コマンド
コマンド・インターフェース 27
初期プロシーチャーの再実行 7
セッション・タイプ 46
説明 46
対話式セッション 212
表エディターの確認 227
EXIT コマンド 47
EXPORT コマンド
最終使用日の更新を制限 240
使用された XML の仕様 99
書式
編集コード・キーワード 113
書式、使用される各国語 240
データ・オブジェクト 75
表オブジェクト 75
列名の長さ 75, 229

EXPORT コマンド (続き)
CICS の使用 146
DATA 74
IXF オプション 81
TSO
記憶域の指定 240
Extensible Markup Language (XML) データ・タイプ
参照: XML データ・タイプ

F

FILE オプション、DSQSPTYP パラメーター 57
FLOAT データ・タイプ
エクスポート
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75
FORTRAN
機能呼び出し 189
サンプル・プログラム 191
マクロ 215
呼び出し可能インターフェース 188
連絡域 188
DSQABFF 191
DSQCOMM 194
ISPF 196
TSO 195

G

GDDM (図形データ表示管理プログラム) 48
GET GLOBAL コマンド 21, 47
GRAPHIC データ・タイプ
エクスポート
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

I

ICU (対話式図表ユーティリティ) 48
IMPORT コマンド
アクセラレーター表 240
最終使用日の更新を制限 240
実行中のエラーと警告 113
使用された XML の仕様 99
使用される各国語、IMPORT FORM 240
定義 73
CICS の使用 146
DATA オプション 102
INTEGER データ・タイプ
エクスポート
編集コード・キーワード、エクスポートされた書式 113

INTEGER データ・タイプ (続き)
 エクスポート (続き)
 列データ形式、IXF 81
 エクスポート時の SQLTYPE コード (QMF 形式) 75
 エクスポート時の列幅 75

INTERACT コマンド
 コマンド形式 50
 セッション
 終了 47, 48
 書式 48
 説明 48

ISPF (対話式システム生産性向上機能)
 呼び出し可能インターフェース 35

IXF エクスポート形式
 説明 81
 バージョン番号 81
 ファイルの最大サイズ 143
 文字 81
 例 81
 ロング・ネーム・サポート 81
 2 進 81

L

LANGUAGE パラメーター、QMF プロファイル 224

LAYOUT コマンド 113
 最終使用日の更新を制限 240

LIST コマンド
 基本視点
 視点名を保管する大域 240
 項目の順序、変更 229
 OWNER パラメーターのデフォルト 240

LIST コマンドのソート順序 229

LOB データ
 エクスポート 99

LONG VARCHAR データ・タイプ
 エクスポート
 列データ形式、IXF 81

LONG VARGRAPHIC データ・タイプ
 エクスポート
 列データ形式、IXF 81

M

MESSAGE コマンド
 オプション 51
 説明 51
 線形手順実行の抑止 51
 テキストの表示 51
 トレース 151
 例 51
 ISPF パネル 51
 QMF ヘルプ・パネル 51

MODEL パラメーター、QMF プロファイル 224

MR パラメーター
 3 部構成の名前の障害 57

N

NEXT コマンド
 表エディターの確認 227

NLF (各国語機能)
 言語 39
 パネル要件 39

NLF (各国語機能) (NLF (National Language Feature))
 言語 ID 33
 セッション環境 39
 定義済み 39
 複数言語環境 44
 参照： 多文化サポート

NULL
 values
 デフォルトの文字、表エディター 227

P

PARSE ARG ステートメント 12

PDF 36, 37

PDS および PDSE データ・セット
 エクスポート記憶域の定義 240
 QMF へのタイプの定義 240

PL/I
 機能呼び出し 199
 サンプル・プログラム 201
 マクロ 215
 呼び出し可能インターフェース 198
 連絡域 198

CICS 205

DSQABFP 201

DSQCOMM 203

ISPF 206

TSO 205

z/OS 205

PREPARE ステートメント、同時アクセスの解決 240

PREVIOUS コマンド
 表エディターの確認 227

PRINT コマンド
 大域変数
 紙送り制御文字の抑止 240
 最終使用日の制限 240

CICS
 キュー名/タイプ 225

PRINTER パラメーター
 QMF プロファイル 224

PS データ・セット、エクスポートのための定義 240

Q

- QBE (例示照会)
 - インポート / エクスポート・ファイル仕様 143
 - エクスポート時のファイルの最大サイズ 143
 - エクスポート・フォーマット 143
- QMF VUE のご使用条件大域変数 218
- QMF 管理者権限
 - 参照： 管理者権限、大域変数
- QMF コマンド内の 3 部構成の名前
 - MR=YES のときの障害 57
- QMF セッションの状態、変数 218
- QMF で選択可能な翻訳
 - 参照： 多文化サポート
- QMF の対話式実行、大域変数 218
- QMF のリリース番号、大域変数 218
- QMF プロファイルの CASE パラメーター 224
- QMF プロファイルの CONFIRM パラメーター 224
- QMF プロファイルの DECIMAL パラメーター 224
- QMF プロファイルの LENGTH パラメーター 224
- QMF へのインターフェース
 - コマンド・インターフェース
 - 参照： コマンド、インターフェース
 - ストアード・プロシージャ・インターフェース
 - 参照： ストアード・プロシージャ・インターフェース
 - 呼び出し可能インターフェース
 - 参照： 呼び出し可能インターフェース
- qmf_data.xsd スキーマ・ファイル 99
- Q.SYSTEM_INI プロシージャ 218

R

- RACF と大/小文字混合パスワード 224
- REFRESH コマンド
 - 表エディターの確認 227
- RESOURCE GROUP パラメーター、QMF プロファイル 224
- RUN QUERY コマンド
 - アクセラレーター表 240
 - 複数ステートメント照会 240
 - メッセージ用の大域変数 251
 - 32 KB を超えた SQL 照会 240
- RUN コマンド
 - 組み込み置換変数 15
 - 最終使用日の更新を制限 240
 - 置換変数 10
 - プロンプト・パネル 10
- RUW (リモート作業単位) 9

S

- SAVE DATA コマンド 74
 - アクセラレーター表 240
- SAVE オプション
 - EDIT TABLE コマンド
 - 関連する大域 227

- SAVE コマンド
 - 最終使用日の更新を制限 240
 - 書式、使用される各国語 240
 - SHARE パラメーター、設定する大域 240
- SAVE コマンドの SHARE パラメーター 240
- SELECT PGM サービス、ISPF 35
- SELECT ステートメント
 - 同時アクセスの解決のオプション 240
- SELECT ステートメントの SKIP LOCKED DATA オプション 240
- SELECT ステートメントの USE CURRENTLY COMMITTED オプション 240
- SELECT ステートメントの WAIT FOR OUTCOME オプション 240
- SET GLOBAL コマンド 218
 - 拡張構文 54
 - 構文 54
 - 変数値の入力プロンプト 11
 - 呼び出し可能インターフェース 21, 54
- SHOW コマンド
 - SHOW CHANGE、表エディターの確認 227
 - SHOW GLOBALS 218
 - SHOW SEARCH、表エディターの確認 227
- SMALLINT データ・タイプ
 - エクスポート
 - 編集コード・キーワード、エクスポートされた書式 113
 - 列データ形式、IXF 81
 - エクスポート時の SQLTYPE コード (QMF 形式) 75
 - エクスポート時の列幅 75
- SPACE パラメーター、QMF プロファイル 224
- SQL キーワード
 - 複数ステートメント照会
 - 参照： 複数ステートメント照会
 - SELECT
 - 同時アクセスの解決のオプション 240
- SQL コード
 - 最後のコマンドからの表示 251
 - 正、メッセージ・サポートの使用可能化 229
- SQL 照会 142
 - エクスポート
 - ファイルの最大サイズ 143
 - 32 KB を超えた長さ 240
- SQLID 特殊レジスター 218
- SQLSTATE 情報、表示 251
- START コマンド
 - インターフェース連絡域 18
 - エラーのデバッグ 153
 - キーワード
 - リスト 57
 - DSQADPAN 57
 - DSQALANG 57
 - DSQSBSTG 57
 - DSQSCMD 57
 - DSQSDBCS 57
 - DSQSDBNM 57
 - DSQSDBQN 57

START コマンド (続き)

キーワード (続き)

DSQSDBQT 57
DSQSDEBUG 57
DSQSIROW 57
DSQSMODE 57
DSQSMRFI 57
DSQSPILL 57
DSQSPRID 57
DSQSPTYP 57
DSQSRSTG 57
DSQSRUN 57
DSQSSPQN 57
DSQSSUBS 57

構文 56

QMF の始動 22, 56

START コマンドの DSQ パラメーター

DSQADPAN 57
DSQALANG 57
DSQSBSTG 57
DSQSCMD 57
DSQSDBCS 57
DSQSDBNM 57
DSQSDBQN 57
DSQSDBQT 57
DSQSDEBUG 57
DSQSIROW 57
DSQSMODE 57
DSQSMRFI 57
DSQSPILL 57
DSQSPLAN 57
DSQSPRID 57
DSQSPTYP 57
DSQSRSTG 57
DSQSRUN 57
DSQSSPQN 57
DSQSSUBS 57

T

TIME データ・タイプ

エクスポート

編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の ISO 形式 75
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

TIMESTAMP WITH TIME ZONE データ・タイプ

エクスポート

エクスポート用の IXF バージョン番号 81
編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
列のレコード形式 81
エクスポート時の ISO 形式 75
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

TIMESTAMP データ・タイプ

エクスポート

編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の ISO 形式 75
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

TRACE コマンド 63

PL/I の例 64

TSO

アセンブラ呼び出し可能インターフェース・プログラム 165
アセンブラ・プログラム 165
予備データ用拡張記憶域 57
C プログラム 176
C 呼び出し可能インターフェース・プログラム 176
RACF での大/小文字混合パスワード 9
REXX プログラム 211
REXX 呼び出し可能インターフェース・プログラム 211

V

VARBINARY データ・タイプ

エクスポート

編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

VARCHAR データ・タイプ

エクスポート

列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

VARGRAPHIC データ・タイプ

エクスポート

編集コード・キーワード、エクスポートされた書式 113
列データ形式、IXF 81
エクスポート時の SQLTYPE コード (QMF 形式) 75
エクスポート時の列幅 75

VBS 形式、XML データ・エクスポート 99

VM プラットフォーム

3 部構成の名前の制限 57

VSE プラットフォーム

3 部構成の名前の制限 57

VUE ご使用条件大域変数 218

VUE ご使用条件の条項 (大域変数) 218

W

WIDTH パラメーター、QMF プロファイル 224

X

XML エクスポートに使用する CCSID 99

XML エクスポート/インポートの仕様 99

XML データ・タイプ

 エクスポート 99

 記憶域についての考慮事項 102

 データ・レコードの形式 99

 ファイルの最大サイズ 143



プログラム番号: 5615-DB2
5697-QM2

Printed in Japan

SC43-3486-00



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21