

DB2 照会報告書作成プログラム



DB2 QMF アプリケーション開発の手引き

バージョン 8 リリース 1

DB2 照会報告書作成プログラム



DB2 QMF アプリケーション開発の手引き

バージョン 8 リリース 1

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、277 ページの『付録 F. 特記事項』に記載されている情報をお読みください。

本書は、IBM DB2 ユニバーサル・データベース・サーバー (z/OS 版) (DB2 UDB for z/OS) バージョン 8 リリース 1 (プログラム番号 5625-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に適用されます。また改訂版やテクニカル・ニュースレターなどで特に断りのない限り、これ以降のすべてのリリースにも適用されます。

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

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

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

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

(URL は、変更になる場合があります)

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：	SC18-7651-00 DB2 Query Management Facility Developing DB2 QMF Applications Version 8 Release 1
発行：	日本アイ・ビー・エム株式会社
担当：	ナショナル・ランゲージ・サポート

第1刷 2003.12

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

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

© Copyright International Business Machines Corporation 1982, 2003. All rights reserved.

© Copyright IBM Japan 2003

目次

本書について	vii	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	30
本書の使用法	vii	コマンド・インターフェースの呼び出し	31
本書を読むための前提条件	vii	END コマンド	31
第 1 章 QMF アプリケーション開発の概要	1	コマンド・インターフェースでの変数の使用	32
QMF でのアプリケーション開発とは	1	コマンド・インターフェースの戻りコード	33
エンド・ユーザーがアプリケーションを使用する方法	2	第 5 章 ADDRESS QRW: QMF コマンド環境の使用	37
使用可能なアプリケーション開発ツール	3	第 6 章 ISPF を使用する QMF アプリケーションの作成	39
第 2 章 アプリケーションとしてのプロシージャの使用	7	ISPF アプリケーションからの QMF の開始および実行	39
プロシージャを使用できない場合	7	変数を含む照会の実行	40
初期プロシージャ	8	ISPF のもとでの QMF ロジックを持つプロシージャからのプログラムの呼び出し	41
プロシージャ内での QMF CONNECT の使用	9	ロジックを持つプロシージャからの ISPF コマンドの使用	41
プロシージャ内の置換変数	10	呼び出し可能インターフェースの考慮事項	42
ロジックを持つプロシージャ内での REXX 変数の使用	12	ISPF で EDIT コマンドを使用する方法	42
ロジックを持つプロシージャへの引き数の受け渡し	12	アプリケーションをデバッグするための ISPF の使用	43
ロジックを持つプロシージャでの REXX エラー処理ステートメントの使用	13	第 7 章 2 か国語使用のアプリケーションの作成	45
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	15	アプリケーション用の 2 か国語使用のオブジェクトの作成	45
第 3 章 呼び出し可能インターフェース	19	コマンド言語変数の使用	46
呼び出し可能インターフェースとは	19	2 か国語使用のアプリケーションでの初期プロシージャの使用	47
インターフェース連絡域 (DSQCOMM) の定義	21	英語のコマンドの使用	47
戻りコード	23	複数言語環境	48
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	23	QMF セッション環境	48
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	25	変換可能なアプリケーションの作成	50
QMF 内からの呼び出し可能インターフェースの使用	25	第 8 章 アプリケーションでの QMF コマンド	51
エラー処理	26	CONNECT	51
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	27	END	54
第 4 章 アプリケーション用コマンド・インターフェースの使用	29	EXIT	56

GET GLOBAL	57
INTERACT	58
MESSAGE.	61
SET GLOBAL	64
START.	67
コマンド同義語の使用.	74

第 9 章 QMF オブジェクトのインポートと エクスポート 79

エクスポートしたファイル、データ・セッ ト、または CICS データ・キューを使用して 何ができるか.	80
データのエクスポートと保管	81
データ・オブジェクトと表オブジェクト.	81
プロシージャおよび SQL 照会	86
図表オブジェクト	87
エンコード・オブジェクト	88
指示照会オブジェクト	102
書式オブジェクト.	105
報告書オブジェクト	113
QBE 照会	122
外部化 QMF オブジェクトの仕様	122
CICS キューを使用する場合の規則と考慮事 項	123

第 10 章 QMF アプリケーションのデバッ グ 125

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.	125
START コマンドおよび他の QMF コマンド 上のエラーのデバッグ	129

付録 A. 呼び出し可能インターフェース言語 のサンプル・コード 131

アセンブラ言語インターフェース.	131
C 言語インターフェース	154
COBOL 言語インターフェース	172
FORTRAN 言語インターフェース	189
PL/I 言語インターフェース	205
REXX 言語インターフェース.	222

付録 B. エクスポート / インポート・フォ ーマット 231

データの QMF フォーマット.	231
指示照会オブジェクトの表番号とフィールド 番号	233
書式オブジェクトの表番号とフィールド番号	236

報告書オブジェクトの表番号とフィールド番 号	241
QMF 報告書で使用される HTML タグ.	243

付録 C. 統合交換フォーマット (IXF) . . . 245

ヘッダー・レコード (H)	246
表レコード (T).	246
列レコード (C).	247
データ・レコード (D)	249
列データ・フォーマット	249
IXF の例.	255

付録 D. プロダクト・インターフェース・マ クロ 259

付録 E. QMF グローバル変数表 261

プロファイル関連状態情報の DSQ グローバ ル変数	261
プロファイル関連でない状態情報の DSQ グ ローバル変数	263
CICS に関連付けられている DSQ のグロー バル変数.	266
先行コマンドで作成されたメッセージに関連 する DSQ グローバル変数.	267
表編集機能に関連付けられている DSQ のグ ローバル変数	267
画面への情報表示方法を制御する DSQ のグ ローバル変数	270
コマンドとプロシージャの実行を制御する DSQ グローバル変数.	272
CONVERT QUERY の結果を示す DSQ グロ ーバル変数	275
RUN QUERY エラー・メッセージ情報を示 す DSQ のグローバル変数.	275

付録 F. 特記事項. 277

商標	279
参考文献	281
CICS の資料	281
COBOL の資料	281
DB2 ユニバーサル・データベース (z/OS 版) の資料	282
Document Composition Facility (DCF) の資料	282
Distributed Relational Database Architecture (DRDA) の資料	283

Graphical Data Display Manager (GDDM) の資料	283	REXX の資料	284
High Level Assembler (HLASM) の資料	283	VM/ESA の資料	284
Interactive System Productivity Facility (ISPF) の資料	283	VSE/ESA の資料	284
OS/390 の資料	283	用語集	285
OS PL/I の資料	284	索引	305

本書について

本書は、アプリケーション・プログラマーが DB2 照会報告書作成プログラム (QMF™) を使用するアプリケーションを作成する際の指針として書かれたものです。

本書の使用方法

本書には、QMF アプリケーションを作成する前に決めておく必要のある設計上の決定事項の概説、様々なプログラミング技法の説明、および QMF を使用するアプリケーション・プログラムの作成のポイントを示す例が記載してあります。付録には、アプリケーション開発に役立つ参照情報があります。

本書は、z/OS®、VM®、および VSE™ のユーザーを対象にしています。各システム間、または CICS®、CMS、TSO およびネイティブ z/OS バッチ間の相違点は、必要に応じて説明してあります。それ以外の点については、QMF はそれぞれのシステムで同じように機能すると考えてください。

本書は、DB2 QMF for TSO/CICS バージョン 8 リリース 1 用に更新されました。QMF 製品の VM 版と VSE 版については、バージョン 8.1 への更新は行われていません。DB2 QMF 製品の VM 版および VSE 版に特定した情報は、バージョン 7.2 以前に関するものです。

本書を読むための前提条件

QMF アプリケーションを使用すれば、QMF がサポートする言語のいずれかで作成さ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ができます。本書では、読者が照会とプロシージャの作成方法、報告書のフォーマット設定方法、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。

QMF コマンドまたは呼び出し可能インターフェースを使用して QMF アプリケーションを作成するには、以下のいずれかのプログラム言語の知識が必要です。

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

アセンブラー、PL/I、C、REXX、COBOL、FORTRAN

ISPF のもとで稼働する任意の言語

作成するアプリケーションのタイプによっては、パネル表示アプリケーションも必要になります。

第 1 章 QMF アプリケーション開発の概要

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。たとえば、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ーの実行
- QMF オブジェクト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データベース内のデータの変更

さらに、リモート・ロケーションで QMF 報告書を印刷するユーザー定義コマンド、または週ごとの売上結果の図表を自動的に生成するファンクション・キーのような便利な機能を、QMF のエンド・ユーザーに提供するアプリケーションも作成することができます。

本章では、QMF アプリケーションの 2 つの主要なタイプについて、また、アプリケーションを作成する場合に役立つ、QMF によって提供されるアプリケーション開発ツールについて説明します。

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 において、アプリケーションとは、プロシージャー、プログラム、または、QMF コマンドを実行したり、Export および Import QMF コマンドを実行して QMF オブジェクトを変更するための exec です。

アプリケーション開発とは、アプリケーションを作成する過程を意味します。これには、以下の作業が含まれます。

- アプリケーションが解決する問題の理解
- 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

上記の定義を終えたら、エンド・ユーザーがアプリケーションを使用する方法、またアプリケーションが QMF と対話できるようにするために使用する QMF アプリケーション・ツールを決定する設計を開始することができます。

エンド・ユーザーがアプリケーションを使用する方法

エンド・ユーザーに主としてアプリケーションと対話させるようにすることもできるし、またアプリケーションを QMF のカスタマイズされた機能として使用させることもできます。

- アプリケーションが QMF に慣れていないエンド・ユーザーを対象にしている場合、エンド・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。実際、QMF がアクティブになっていることをエンド・ユーザーが知らないで済むようにしたい場合があります。このような場合、アプリケーションは QMF サービスを使用しますが、QMF の外側に存在します。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。
- エンド・ユーザーが QMF を熟知している場合、エンド・ユーザーがアプリケーションを QMF の拡張部分または QMF をカスタマイズしたものと見なすようにすることが望ましいでしょう。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。

主としてアプリケーションと対話するエンド・ユーザー

QMF サービスを使用するアプリケーションを作成するものとします。このアプリケーションは、エンド・ユーザーに 図 1 に示すようなメニュー方式のインターフェースを提供します。

```
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

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 プロシーチャーを実行します。

この例では、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が付きません。

QMF セッション内でアプリケーションを開始するエンド・ユーザー

あるユーザーから別のユーザーに QMF 報告書を送るアプリケーションを作成するものとします。

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。ユーザーがコマンド同義語 (SEND_TO) をコマンド行から発行するか、その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ができます。

ユーザーは報告書を生成した後、図 2 に示すように QMF コマンド行に SEND_TO SMITH と入力すれば、Smith 氏にこの報告書を送ることができます。

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

使用可能なアプリケーション開発ツール

エンド・ユーザーがアプリケーションをどのように見ているかに関係なく、次の任意のアプリケーション開発ツールを使用してアプリケーションを作成することができます。

- QMF プロシージャ
- QMF 呼び出し可能インターフェース
- QMF コマンド・インターフェース
- QMF 外部フォーマット

QMF アプリケーション開発の概要

- QMF コマンド同義語
- QMF にブリッジする他の IBM プロダクト

QMF プロシージャー

QMF プロシージャーは、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 プロシージャーは、インストール先システム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、線形プロシージャーとロジックを持つプロシージャーという 2 つのタイプのプロシージャーがあります。

- 線形プロシージャー 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、線形プロシージャー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ー 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。ロジックを持つプロシージャーは、QMF がサポートするすべての環境 (CICS を除く) で使用できます。ロジックを持つプロシージャーには、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。

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 の作成に関する一般情報については、「DB2 QMF 使用の手引き」を参照してください。QMF プロシージャーを使用してアプリケーションを作成する方法については、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の使用』を参照してください。

バージョン 3.3 以降、QMF はシステム初期化プロシージャーを提供します。これを使用すると、ユーザーが QMF ホーム・パネルを表示する前に、コマンドを実行してグローバル変数を設定することができます。

QMF 呼び出し可能インターフェースおよびコマンド・インターフェース

QMF プロシージャーの使用を選択しない場合、プログラムが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どちらを介して QMF と通信するかを決定する必要があります。

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は、QMF の外側から呼び出され、QMF セッションを開始し、コマンドを実行のために QMF に送るアプリケーションの作成に使用されます。

呼び出し可能インターフェースは、QMF がサポートするすべての環境で使用することができます。これは、z/OS、VM、および VSE 環境での照会用の共通プログラミング・インターフェースであり、5 ページの表 1 に示すように様々な言語で使用することができます。

表 1. 呼び出し可能インターフェースのサポート

	z/OS の もとの CICS	VSE の もとの CICS	CMS	TSO	APPC	SRPI	ネイティブ z/OS バッチ
アセンブラー	×	×	×	×	×	×	×
C	×	×	×	×	×	×	×
COBOL	×	×	×	×	×	×	×
FORTRAN			×	×	×	×	×
PL/I	×	×	×	×	×	×	×
REXX			×	×	×	×	×

呼び出し可能インターフェースの詳細については、19 ページの『第 3 章 呼び出し可能インターフェース』を参照してください。

コマンド・インターフェース

QMF コマンド・インターフェースによって、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。

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。

QMF コマンド・インターフェースの詳細については、29 ページの『第 4 章 アプリケーション用コマンド・インターフェースの使用』を参照してください。

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比較

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相違点は、次のとおりです。

呼び出し可能インターフェース

- QMF がサポートするすべての環境で使用可能
- ISPF が不要
- アプリケーションを実行する前に QMF を開始することが不要

コマンド・インターフェース

- QMF および ISPF がサポートするすべての環境で使用可能
- ISPF が存在していて、アクティブであることが必要
- アプリケーションを実行する前に QMF を開始することが必要
- ISPF アプリケーションと QMF の間の通信用の変数の提供

- ISPF がサポートするプログラム言語が必要

QMF オブジェクトの外部フォーマット

アプリケーションは、QMF プロダクトの外側にあるファイルに QMF オブジェクトをエクスポートすることができます。たとえば、書式を CMS 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することができます。各オブジェクトには特定のフォーマットがあります。アプリケーションは、このフォーマットを編集し、他のシステムに転送したり、QMF にインポートしたりすることができます。

QMF オブジェクトの外部フォーマットの詳細については、79 ページの『第 9 章 QMF オブジェクトのインポートとエクスポート』を参照してください。

コマンド同義語

QMF を使用すれば、コーディングするプログラムまたはプロシージャーに関するコマンド同義語を指定することができます。これらのコマンド同義語を使用すれば、エンド・ユーザーは、コーディングされたプログラムまたはプロシージャーを QMF コマンドのいずれかを使用しているかのように使用することができます。

コマンド同義語の詳細については、74 ページの『コマンド同義語の使用』を参照してください。

QMF にブリッジする他の IBM プロダクト

QMF の機能を拡張するために、次の IBM プロダクトを QMF と共に使用することができます。

アプリケーション・システム (AS)

AS は QMF コマンドを発行し、QMF 照会を定義することができます。次に、AS は QMF 照会の結果を AS プロセスへの入力として使用することができます。

ECF 拡張接続機能 (ECF) を使用すれば、ワークステーション・ユーザーはホストのリレーショナル・データにアクセスすることができます。ワークステーション・ユーザーは ECF 機能を使用して要求をホストに送信し、保管されている QMF 照会またはプロシージャーを実行し、取り出したデータをワークステーションにダウンロードします。

GDDM 対話式図表ユーティリティ (ICU) は、図表を表示するために QMF で使用しますが、実際は図形データ表示管理プログラム (GDDM[®]) の機能です。

ISPF 対話式システム生産性向上機能を使用すれば、ユーザーは、コマンド・インターフェースを介して QMF と対話できるパネルを生成することができます。

Lotus 1-2-3/M

Lotus[®] 1-2-3[®] のホスト版は、QMF にアクセスし、照会結果にスプレッドシート分析を行うことができます。

第 2 章 アプリケーションとしてのプロシージャの使用

アプリケーションをすべてプロシージャとして作成することができます。プロシージャは、開発システム上で作成し、個人で使用するために保持することも、あるいは、実動システムに移して共通のプロシージャとして使用することもできます。

CICS 環境で QMF を使用する場合には、QMF 線形プロシージャを使用することができます。CMS、TSO、またはネイティブ z/OS バッチ環境で QMF を使用する場合には、QMF プロシージャ内で REXX のステートメントおよび関数も使用することができます。REXX 関数およびロジックを持つプロシージャは、CICS 環境では使用できません。

本章では、QMF プロシージャを使用してアプリケーションをインプリメントするために必要な情報について説明します。

プロシージャを作成し、構築し、実行する方法については、「DB2 QMF 使用の手引き」を参照してください。

QMF プロシージャで ISPF サービスを使用するには、このほかにいくつかのステップが必要です。ISPF のもとで実行中の QMF ロジックを持つプロシージャから ISPF コマンドを実行する方法については、41 ページの『ロジックを持つプロシージャからの ISPF コマンドの使用』を参照してください。

プロシージャを使用できない場合

QMF 一時記憶域内のプロシージャ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として作成することはできません。プロシージャを実行すると、それが QMF 一時記憶域内の現在のプロシージャになるからです。

たとえば、アプリケーションをプロシージャとして作成し、そのアプリケーションが QMF の一時記憶域内の現在のプロシージャを保管するようにコーディングした場合、アプリケーションが実行中はそのアプリケーションが QMF 一時記憶域内の現在のプロシージャであるため、アプリケーションは自分自身を保管することになります。

初期プロシージャ

初期プロシージャは、QMF セッションの開始直後に実行されるプロシージャです。DSQSRUN パラメータを使用してこのプロシージャの名前を指定してください。以下のコマンドで DSQSRUN を使用することができます。

- QMF を対話式で開始した場合、DSQQMFE コマンド
- 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

使用中の QMF セッションのタイプによって、QMF が初期プロシージャを実行する方法が異なります。QMF が初期プロシージャを使用する方法の詳細については、54 ページの『初期プロシージャ (DSQSRUN) を伴う対話式セッション』を参照してください。

TSO 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 exec を使用して QMF START コマンドの DSQSCMD パラメータの記述に従って、プログラム・パラメータを設定することもできます。QMF CICS は REXX をサポートしないので、START コマンドで DSQSMODE=INTERACTIVE を使用してすべてのプログラム・パラメータを指定する必要があります。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 BATCH です。

初期プロシージャ作成に関する考慮事項

- デフォルトでは、DSQQMFE によって開始された QMF の対話式セッションでユーザーが END コマンドを発行した時は、必ず QMF が初期プロシージャを再実行します。DSQEC_RERUN_IPROC グローバル変数が、初期プロシージャ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を再実行することを示しています。0 は、初期プロシージャを再実行しません。

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は再実行されないので、このグローバル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を作成する場合、このプロシージャの最後の現行パネルがホーム・パネルになるようにしてはなりません。この場合、QMF はプロシージャ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、ユーザーと対話しないで初期プロシージャを再実行します。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を回避するには、以下のいずれかのタスクを実行してください。

- プロシージャ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。

- 初期プロシージャ内で置換変数の値を指定する場合に、変数名の前に使用する必要があるアンパーサンド (&) の数は、ユーザー環境によって異なります。たとえば、DSQSRUN を次のように指定することができます。

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

VAR1 と共に指定する必要があるアンパーサンドの数は、QMF を CICS、CMS、TSO、またはネイティブ z/OS パッチのもとで実行しているかどうか、ISPF が存在しているかどうか、および QMF を開始したプログラムが REXX で作成されているかどうかによって異なります。

初期プロシージャおよびリモート作業単位

初期プロシージャは 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。

QMF CONNECT コマンドを初期プロシージャから、または初期プロシージャによってセットアップした対話式セッション中にコマンド行から使用する場合、END コマンドを出して初期プロシージャを再び呼び出す前に元のロケーションに再接続する必要があります。

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。

プロシージャ内での QMF CONNECT の使用

QMF CONNECT コマンドを使用すれば、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ことができます。このコマンドは、線形プロシージャまたはロジックを持つプロシージャ内で使用できます。

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を作成する場合は、以下の点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、そのプロシージャおよびそのプロシージャ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 (たとえば、照会、プロシージャ、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。

QMF CONNECT コマンドの使用およびリモート作業単位サポートの詳細については、「DB2 QMF 解説書」を参照してください。

- 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。

アプリケーションとしてのプロシージャの使用

- プロシージャにシステム固有のコマンド (CICS、CMS、または TSO) が入っている場合、これらのコマンドは、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。
プロシージャに、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入っている場合、そのプロシージャは正常に実行されません。
- システム固有のコマンドが使用するすべてのファイルまたはデータ・セットは、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。

プロシージャ内の置換変数

QMF 置換変数は、線形プロシージャおよびロジックを持つプロシージャ内で使用できます。

置換変数は、QMF コマンド内で使用できる任意の変数です。QMF は、これらの変数を管理します。置換変数の前には常にアンパーサンド (&) が付きます。置換変数に値を割り当てるには、グローバル変数を設定するか、RUN コマンドで値を指定するか、RUN コマンド・プロンプト・パネルで値を指定します。グローバル変数の設定については、64 ページの『SET GLOBAL』を参照してください。

QMF で置換変数にアンパーサンドを使用する方法の詳細については、「DB2 QMF 使用の手引き」を参照してください。

RUN コマンドでの値の指定

以下のように RUN コマンドを使用して、置換変数に値を割り当てることができます。

- 線形プロシージャの場合

```
RUN PROC SCHEDULE (&&TYPE='VACATION'
```
- ロジックを持つプロシージャの場合

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

PROC または QUERY パネル内から QMF RUN 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。RUN コマンドは、各パネルより呼び出された際に、これらの値を想定します。

&TYPE の値は、SCHEDULE と呼ばれるプロシージャ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。

置換変数に関する上記の値は、その置換変数を定義しているプロシージャ内でのみ有効です。この値は、置換変数を定義しているプロシージャが呼び出したプロシージャまたはモジュールでは有効ではありません。

RUN コマンド・プロンプト・パネルでの値の指定

置換変数が含まれている照会またはプロシージャを実行する場合、この変数にグローバル変数によって、または RUN コマンド上で値が割り当てられなければ、QMF は RUN コマンド・プロンプト・パネルを表示します。ユーザーは、このパネルで変数の値を指定することができます。

置換変数に関する上記の値は、その置換変数を定義しているプロシージャ内でのみ有効です。この値は、置換変数を定義しているプロシージャが呼び出したプロシージャまたはモジュールでは有効ではありません。

線形手順内の変数値の入カプロンプト

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カプロンプト

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。たとえば、QMF が値の入力をプロンプト指示する必要のある変数を含んでいるステートメントがロジックを持つプロシージャに 3 つある場合、QMF は、ステートメントごとに 1 回ずつ、合計 3 回の入力をプロンプト指示します。

ロジックを持つプロシージャに、線形プロシージャと同様に、1 度に必要なすべての変数値の入力のプロンプトを出したい場合は、ダミー・プロシージャを使用します。図 3 に示すように、ロジックを持つプロシージャで 2 つの異なる行に指定されている置換変数 LASTNAME と DEPT_NUM の置換変数に対して 1 回プロンプトを出してもらいたいとします。

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

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

図 3. 変数値を含むロジックを持つプロシージャ

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次の行を追加します。

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

アプリケーションとしてのプロシージャの使用

ここで、PROMPT_ME は、図 4 に示すように、コメント行のみで命令が含まれていないロジックを持つプロシージャです。

完成後のロジックを持つプロシージャは次のようになります。

```
/* This proc is a dummy proc that provides prompting. */
/* This procedure runs two queries, displaying the report after each */
/* procedure 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、次のように SET GLOBAL 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での REXX 変数の使用

ロジックを持つプロシージャで REXX 変数を使用することができます。これらの変数の値は、これらの変数を定義したプロシージャ内でのみ認識されます。以下の操作が可能です。

- 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グローバル変数を 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。

REXX 変数の詳細については、使用しているシステムの REXX 解説書を参照してください。GET GLOBAL および SET GLOBAL コマンドの詳細については、「DB2 QMF 解説書」を参照してください。

ロジックを持つプロシージャへの引き数の受け渡し

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。このオプションによって、引き数または値をロジックを持つプロシージャに渡すことができます。

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に、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

```

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)
```

ARG オプションを使用すれば、プロシージャ間で値を渡すこともできます。

ロジックを持つプロシージャでの REXX エラー処理ステートメントの使用

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 エラー時のシグナル 命令は、非ゼロの戻りコードが検出された場合に、REXX に、現在行からエラー とマークされているラベルに分岐するように指示します。このステートメントには以下の 2 つの部分が必要です。

- エラー時のシグナル

各コマンドの後で、REXX は rc と呼ばれる変数にコマンドの戻りコードを置きます。

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。

エラー時のシグナル は、REXX 呼び出し可能インターフェースからではなく、QMF REXX プロシージャ (ADDRESS QRW) コマンド環境からエラーを戻します。

- エラー ・ ラベル

エラー時のシグナル 命令には、プロシージャが非ゼロの戻りコードを検出した場合に分岐するラベルに関する、ユーザーによる指定が必要です。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 ロジックを持つプロシージャ */
error:
  exit rc

```


REXX EXIT ステートメントによるメッセージの使用

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 命令を使用した場合に表示されるメッセージは、以下の要因によって異なります。

- 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

表 2 は、一定の条件のもとで表示されるメッセージを示しています。

表 2. QMF から戻されるメッセージ

最後の QMF コマンドからの 非ゼロの戻りコード	プロシージャの 戻りコード	プロシージャの完了時のメッセージ
なし	0	OK, your procedure was run (プロシージャは実行されました)
なし	非ゼロ	プロシージャからの戻りコードは 8
あり	0	QMF によって提供されるエラー・メッセージ
あり	非ゼロ	QMF によって提供されるエラー・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、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。

最後のコマンドからのエラー・メッセージを表示し、かつ QMF 戻りコードと共に終了したい場合には、次の例に示すように MESSAGE コマンドと EXIT DSQ_RETURN_CODE を使用してください。

```

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

```

図 5. エラー・メッセージと戻りコードの表示

変数 dsq_message_text と dsq_return_code は QMF が提供する REXX 変数です。(これらの変数の完全なリストが、222 ページの『REXX 言語インターフェース』にあります。) 15 ページの図 6 に示すように、MESSAGE コマンドと dsq_message_text 変数を使用してメッセージを保管してから、他の処理の実行後にメッセージを表示することができます。


```

/* 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;

```

図 6. プロシージャ内でのメッセージの保管および取り出し

MESSAGE コマンドの詳細については、61 ページの『MESSAGE』を参照してください。

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

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場合、アプリケーション内の置換変数に指定するアンパーサンド (&) の数に注意してください。呼び出されるプログラムに、RUN QUERY WEEKLY_Q (&&DEPT=58 のように置換変数が指定されている RUN コマンドがある場合には、特に注意してください。

置換変数がない REXX プログラムの呼び出し

REXX プログラムに、置換変数が組み込まれている RUN コマンドがない場合には、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てください。

- ADDRESS 命令

この命令は、コマンド環境を確立します (コマンド環境の詳細については、37 ページの『第 5 章 ADDRESS QRW: QMF コマンド環境の使用』を参照してください)。プログラムの名前が PANDA で、CMS 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 CMS "PANDA"
```

- CALL 命令

この命令はプログラムを呼び出します。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()
```

アプリケーションとしてのプロシージャの使用

上記のコマンドの詳細については、使用中のシステムに関する REXX 解説書を参照してください。

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。このような場合、QMF がユーザーに変数値を入力するようプロンプトを出します。

置換変数がある REXX プログラムの呼び出し

REXX アプリケーションに、置換変数が組み込まれた QMF RUN コマンドがある場合には、そのアプリケーションを CMS program_name または TSO program_name を使用して呼び出す必要があります。

実行中のプログラムがロジックを持つプロシージャであるか、またはロジックを持つプロシージャによって呼び出された呼び出し可能インターフェース・プログラムであるかに関係なく、コマンドは同様に QMF に入ります。したがって、呼び出し可能インターフェース・プログラムは、プロシージャ自身の論理的な拡張になります。

次のコマンドについて考えてみます。

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

ロジックを持つプロシージャでは、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身のために解決するので、その変数を照会に渡しません。

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し、かつ、そのアプリケーションにコマンド RUN QUERY WEEKLY_Q (&DEPT=58 がある場合、QMF はその変数を、呼び出しプロシージャの変数であるかのように解決します。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。

- CMS または 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 1 つ追加して、ロジックを持つプロシージャが解決しないようにします。

- グローバル変数を使用する。

アプリケーションの開始時にグローバル変数を定義して、QMF セッションで使用します。

第 3 章 呼び出し可能インターフェース

本章では、QMF 呼び出し可能インターフェースの概要を述べます。特定言語の QMF 呼び出し可能インターフェースについては、言語ごとの呼び出し可能インターフェースを説明している、『付録 A. 呼び出し可能インターフェース言語のサンプル・コード』の節を参照してください。

アセンブラー

131 ページの『アセンブラー言語インターフェース』

C 言語 154 ページの『C 言語インターフェース』

COBOL

172 ページの『COBOL 言語インターフェース』

FORTRAN

189 ページの『FORTRAN 言語インターフェース』

PL/I 205 ページの『PL/I 言語インターフェース』

REXX 222 ページの『REXX 言語インターフェース』

呼び出し可能インターフェースとは

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。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、最初に、プログラムと QMF の間の通信を開始するための、QMF 提供のルーチンへの呼び出しを行わなければなりません。この呼び出しは、QMF 提供のインターフェース・ルーチンに対して行います。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、最初の開始呼び出しの後で 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、QMF が提供する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

呼び出し可能インターフェース

び出し可能インターフェースが呼び出しアプリケーション・プログラムに制御を戻したとき、アプリケーションはこれらの変数を参照できますが、変更してはなりません。

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 提供の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

- QMF に対する呼び出しの後、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、主要言語として指定した NLF 言語で作成し、かつ、英大文字で作成する (または英大文字に変換する) 必要があります。
- 渡すコマンドの最大長は 256 バイトです。

図 7 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。

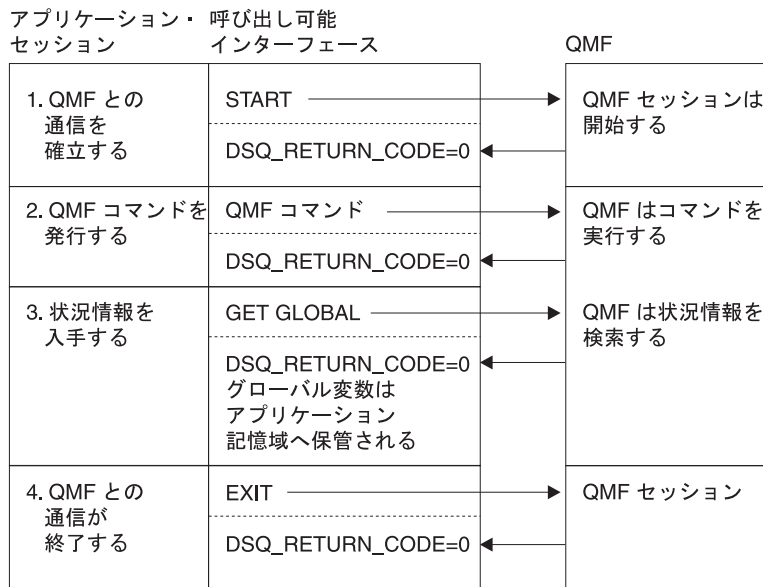


図 7. QMF との通信に QMF 呼び出し可能インターフェースを使用するアプリケーション

呼び出し可能インターフェースを介してのコマンド発行の結果は、一般的に、コマンドを対話式で出した場合と同様です。

インターフェース連絡域 (DSQCOMM) の定義

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通信マクロが用意されています。このマクロには、以下の情報が入っています。

- インターフェース連絡域 (DSQCOMM) または連絡変数
- 戻りコードと理由コードの定義
- QMF に対する関数呼び出しの定義

このマクロは、上記のリストに示した変数が入っている記憶域を定義します。この記憶域は呼び出し可能インターフェースの連絡域であり、この域内に保管された変数は、QMF と呼び出し可能インターフェース・アプリケーションの両方からアクセス可能です。ただし、値を変更するのは QMF だけにすべきです。アプリケーション・プログラムは、これらの変数を読み取り専用と見なすべきです。

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 CMS 仮想計算機内
- 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を更新します。

インターフェース連絡域は、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。これは、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、最

呼び出し可能インターフェース

初の呼び出しで `DSQ_INSTANCE_ID` をゼロ (0) に設定して、`QMF` が変数を最初の `START` コマンドによって設定された値にリセットするようにならなければなりません。

`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。

サポートされる言語にはそれぞれ、インターフェース連絡域について記述する固有の連絡マクロがあります。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または連絡変数には、呼び出し側プログラムが変更してはならない、表 3 に示した情報が入ります。

表 3. 変更してはならない `DSQCOMM` フィールド

フィールド	説明
戻りコード	<code>QMF</code> がコマンドを処理した後の、 <code>QMF</code> 処理の状況を示す。
インスタンス ID	<code>START</code> コマンドによって開始された <code>QMF</code> のインスタンスを示す。
完了メッセージ ID	コマンドがユーザーの端末から出された場合に、 <code>QMF</code> がユーザーの端末に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 <code>QMF</code> 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 <code>QMF</code> が表示するメッセージが入る。
照会メッセージ ID	<code>RUN QUERY</code> コマンドの結果として出される <code>QMF</code> メッセージのメッセージ ID が入る。これは、ユーザーの照会で表示されるメッセージのメッセージ ID である。 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 <code>QMF</code> が表示するメッセージが入る。
エラーがある <code>START</code> 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 <code>START</code> コマンドが失敗した場合に、エラーがあるパラメーターが入る。
取り消し標識	<code>QMF</code> がコマンドを実行中に、ユーザーがコマンド処理を取り消したかどうかを示す。
完了メッセージ	ユーザーの端末に <code>QMF</code> が表示する完了メッセージが入る。

表 3. 変更してはならない *DSQCOMM* フィールド (続き)

フィールド	説明
照会メッセージ	<p><code>RUN QUERY</code> コマンドの結果の照会メッセージ・テキストが入る。これは、ユーザーの照会で <code>QMF</code> が表示するテキストである。</p> <p>たとえば、エラーがある照会オブジェクトを実行すると、<code>QMF</code> 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。照会メッセージには、このエラー・メッセージ・テキストが入る。</p>

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` と共に出荷される連絡マクロによって記述されます。

アプリケーションをシステム間で移植可能にしたい場合、戻りコード値は、システムによって異なる可能性があるため、変数名で参照する必要があります。

呼び出し可能インターフェースからの戻りコードは、以下の状態を示します。

- `QMF` が正常に要求を処理した。
- 警告状態にもかかわらず、`QMF` が要求を処理した。
- `QMF` がコマンドを正しく処理しなかった。
- 重大エラーのため、`QMF` のこのインスタンスが終了した。

各戻りコードの定義については、本書の該当するプログラム言語の節を参照してください。

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

プロシージャ内で任意の `QMF` コマンドを使用するために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。ただし、以下の 3 つのコマンドの場合、特殊な呼び出し可能インターフェースの構文になります。

- `START`
- `GET GLOBAL`、拡張構文
- `SET GLOBAL`、拡張構文

`START` 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` を使用する場合は、拡張構文を使用してください。SET GLOBAL コマンドの拡張構文によって、長さが 32 768 文字までの値を入れることができるグローバル変数を設定できます。アプリケーションでの `GET GLOBAL` コマンドと `SET`

GLOBAL コマンドの使用方法の詳細については、57 ページの『GET GLOBAL』および 65 ページの『SET GLOBAL: 拡張構文』を参照してください。

呼び出し可能インターフェース・アプリケーションで使用できる上記のコマンドおよび他のコマンドについては、51 ページの『第 8 章 アプリケーションでの QMF コマンド』を参照してください。各言語の START コマンドおよび SET GLOBAL コマンドの例については、各言語のサンプル・プログラムを参照してください。

アセンブラー

134 ページの『アセンブラー プログラミングの例』

C 言語 158 ページの『C 言語プログラミングの例』

COBOL

175 ページの『COBOL プログラミングの例』

FORTRAN

192 ページの『FORTRAN プログラミングの例』

PL/I 208 ページの『PL/I プログラミングの例』

REXX 225 ページの『REXX プログラミングの例』

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、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 がすでに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、まだ開始されていなければ、開始されます。QMF がすでに開始されていれば、非ゼロの戻りコードが戻され、次のメッセージ番号とメッセージが表示されます。

```
DSQ50719 QMF already active. Secondary session not permitted.
```

REXX 呼び出し可能インターフェースを使用している場合、次のプログラムを実行することもできます。

```
/* test to see if QMF is active */  
"SUBCOM QRW"  
if rc = 0  
    then say "QMF is active"  
else say "QMF is not active"
```

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、QMF はエラーがあるまま開始されます。この場合、EXIT コマンドを出して QMF を停止す

ることができます。 START コマンドをもう一度出す必要がある場合があります。エラーが続くようであれば、インターフェース連絡域の内容または QMF トレース・データ出力でエラーの原因を調べてください。

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。

START コマンドで使用する構文およびキーワードの詳細については、67 ページの『START』を参照してください。

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、対話式 QMF を実行する場合と同様に実行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、131 ページの『付録 A. 呼び出し可能インターフェース言語のサンプル・コード』にある各言語の該当するコーディング・サンプルを参照してください。

QMF 内からの呼び出し可能インターフェースの使用

CICS ユーザーへの注

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。

CICS を除く、サポートされているすべての環境で、QMF 内から呼び出し可能インターフェースを使用して、QMF の一時記憶域を変更するアプリケーションを実行することができます。たとえば、QMF セッションの最中でも、呼び出し可能インターフェースを介してファイルのエクスポートまたはインポートを実行する必要がある場合があります。

CMS または 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によって、これを実行することができます。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。QMF はすでにアクティブなので、START コマンドを出す必要はありません。

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

CICS 以外の環境で QMF を実行している場合には、アプリケーション内でのエラー処理のために、QMF が提供する REXX 変数または DSQCOMM 連絡域内の同様の値を使用することができます。

たとえば、REXX 変数 `dsq_message_text` または DSQCOMM のメッセージ・テキスト・フィールドには、QMF メッセージが入ります。

REXX では、QMF は各 QMF コマンドの完了後に、変数 `dsq_return_code` に以下のいずれかの値を割り当てます。

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

REXX 以外の言語の場合、QMF は同じ値を DSQCOMM の戻りコード・フィールド `DSQ_RETURN_CODE` に入れます。

このような戻りコードと値をアプリケーションで使用することができます。以下の例は、REXX 呼び出し可能インターフェース・アプリケーションでエラー処理変数を使用する方法を示しています。

```
⋮  
call dsqcix "CONVERT QUERY MYQUERY"  
if dsq_return_code ^= dsq_success then ...  
⋮  
call dsqcix "PRINT REPORT"  
if dsq_return_code=dsq_severe | dsq_return_code=dsq_failure then ...
```

QMF は、メッセージ番号とメッセージ・テキストが入る変数も提供します。

各 DSQCOMM 内の変数またはフィールドの完全なリストについては、131 ページの『付録 A. 呼び出し可能インターフェース言語のサンプル・コード』の各言語に該当する節を参照してください。

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 プログラムをインストールする通常の方法を使用してそのプログラムを CICS にインストールします。CICS におけるアプリケーションの詳細については、「*CICS (VSE/ESA 版) 適用業務プログラミングの手引き*」を参照してください。QMF アプリケーション・プログラムのインストールについての詳細は、「*CICS (VSE/ESA 版) システム定義の手引き*」を参照してください。

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、非常に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言語プログラムです。QMF は、他のアセンブラ言語プログラム、またはいずれかの高水準言語 (VS COBOL II、PL/I、または C/370) で作成されたプログラムとリンクすることができます。高水準言語を使用して QMF を呼び出す場合、高水準言語プログラムを最初にリンクし、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。高水準言語プログラミングの手引き、および *CICS アプリケーション・プログラミングの手引き* の言語に関する考慮事項の節を参照してください。

CICS において、デフォルトの QMF 開始パラメーターのいずれかを指定変更するには、START コマンドでこれらのキーワードを指定する必要があります。たとえば、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 BATCH です。対話式 QMF セッションを実行する場合には、DSQSMODE=INTERACTIVE を使用して、START コマンドを出す必要があります。

- CICS の領域 (z/OS) または区画 (VSE) 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

- **DB2 (VSE または VM 版)** 呼び出し可能インターフェースを介して QMF を呼び出すと、CICS トランザクションは、インストール済みのデータベース・パッケージを使用して QMF を実行するので、それ以上のアクションは不要です。

- **DB2 (z/OS 版)** ユーザーのプログラムを呼び出す CICS トランザクションも、リソース管理テーブル (RCT) 項目によって DB2 に記述する必要があります。RCT 項目に関する詳細については、「*DB2 for z/OS Administration Guide*」と「*CICS/ESA システム定義の手引き*」を参照してください。

呼び出し可能インターフェース

RCT PLAN 名は、呼び出し可能インターフェース・プログラムと QMF プロダクトの両方で同じでなければなりません。

第 4 章 アプリケーション用コマンド・インターフェースの使用

QMF には、ISPF ダイアログから QMF サービスを使用するためのアプリケーション・インターフェースが用意されています。このインターフェースは、コマンド・インターフェースです。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 コマンドを出すことができます。このインターフェースを使用して、QMF は、図 8 に示すように、ISPF 変数プールを介してダイアログと通信することができます。

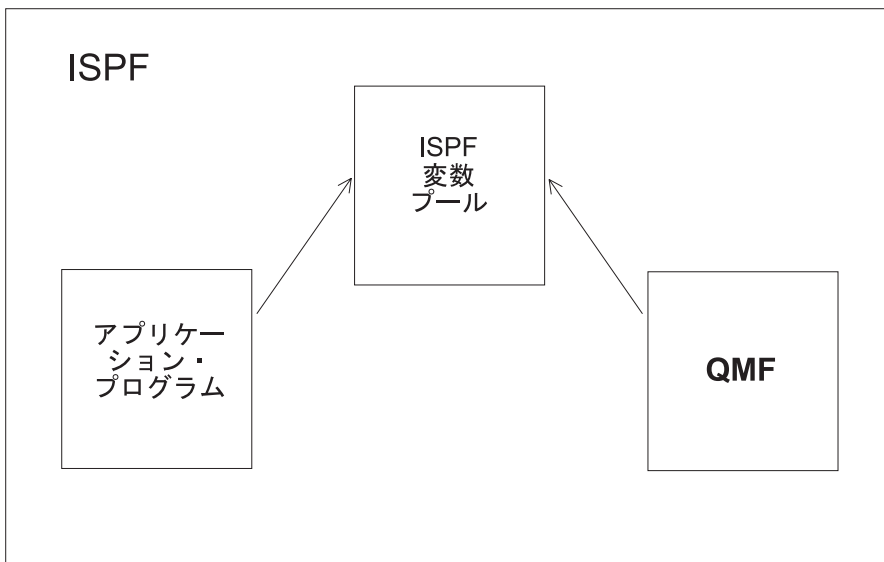


図 8. QMF と対話する QMF コマンド・インターフェース・アプリケーション

CICS ユーザーへの注

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。

アプリケーション用コマンド・インターフェースの使用

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が必要です。ISPF の使用上の詳細については、「*ISPF: Dialog Management Guide and Reference*」を参照してください。

コマンド・インターフェース (DSQCCI) を使用するには、QMF セッションが実行中ではなければなりません。コマンド・インターフェースを使用して QMF セッションを開始することはできません。次のように、ISPSTART コマンドを使用すれば、ISPF のもとで QMF セッションを開始することができます。

```
ISPSTART PGM(DSQMF) NEWAPPL(DSQE) PARM(...)
```

ISPF SELECT PGM サービスを使用して、QMF コマンド・インターフェース・プログラム DSQCCI を呼び出すことによって、ISPF ダイアログから QMF コマンドを使用します。SELECT PGM サービスの PARM オプションを使用して QMF コマンドを渡します。コマンド・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、以下のステップを実行する必要があります。

1. ISPF を開始する
2. QMF を開始する
3. CMS または TSO コマンドを使用してプログラムを実行する

重要: 上記のいずれかのステップを省略すると、プログラムは失敗します。

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

ユーザーに照会名の指定を入力し、照会を実行し、報告書を表示する ISPF パネルを表示するために、コマンド・インターフェースを使用する必要があるものとします。

このシナリオでは、以下を行います。

1. コマンド・インターフェース REXX プログラムを作成する。プログラムは以下の処理を行う。
 - a. DISPLAY サービスを使用して、ISPF パネル QRYNAME を表示する。

```
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。ここで、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. 次のコマンドを使用して、照会の結果をユーザーに表示する。

```
ADDRESS ISPEXEC "SELECT PGM(DSQCCI) PARM(INTERACT)"
```
2. ISPF を開始する。
3. QMF を開始する。

4. QMF コマンド行から CMS または TSO コマンドを使用して作成したプログラムを呼び出す。たとえば、プログラムの名前が GETINFO である場合、コマンドは使用中のシステムにより以下のいずれかになります。

```
CMS GETINFO
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。

ISPF SELECT サービスを介してコマンド・インターフェースを呼び出す場合、実行する QMF コマンドを大文字で PARM オペランドに入れて渡します。次のコマンドを実行してください。

```
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 は、自動的にコマンドを小文字から大文字に変換しません。QMF コマンドを小文字で指定すると、QMF はそのコマンドを認識しません。QMF がコマンドを実行しているときにプロンプト指示を出したい場合は、QMF コマンドの直前に INTERACT コマンドをコーディングしてください。

呼び出し時に、NEWPOOL または NEWAPPL オプションを指定してはなりません。NEWPOOL または NEWAPPL オプションを省略すれば、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。

SELECT サービスを使用するには、RUN QUERY コマンドで 2 つのアンバーサンドを使用する必要があります。これによって、ISPF はその変数を自分自身のものと解釈しなくなります。

END コマンド

コマンド・インターフェース (DSQCCI) の実行中に、END コマンドを実行すると、DSQCCI 呼び出しが終了し、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。コマンド・インターフェースの呼び出し中は、アバンド (異常終了) だけが QMF セッションを終了します。

QMF は、コマンド・インターフェースの呼び出し中に EXIT コマンドまたは重大エラーに遭遇すると、DSQCSESC を設定して、そのセッションを終了とマークします。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の使用

STATE コマンドは、QMF で提供された変数ごとに現行値を提供します。このコマンドは、コマンド・インターフェースでしか使用できません。QMF 変数は、このコマンドの実行時に、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。表 4 は、QMF が ISPF 変数プールに置く使用可能な QMF 変数のサブセットを示しています。

表 4. ISPF 変数プールの QMF 変数

変数のタイプ	変数名	説明	
STATE コマンド	DSQAAUTH	DSQAPLEN	アプリケーションが STATE コマンドを出すと、QMF はこれらの変数を更新する。
	DSQABATC	DSQAPLNG	
	DSQACMDM	DSQAPPFK	
	DSQACRSR	DSQAPPRT	
	DSQADBCS	DSQAPRMP	
	DSQADBMG	DSQAPSPC	
	DSQAIACT	DSQAPSYN	
	DSQAITEM	DSQAPTRC	
	DSQAITLO	DSQAPWID	
	DSQAITMN	DSQAQMF	
	DSQAITMO	DSQAREVN	
	DSQALANG	DSQAROWS	
	DSQAMODL	DSQASUBI	
	DSQAMODP	DSQASUBP	
	DSQAOGRP	DSQATRAC	
	DSQAPCAS	DSQAVARN	
DSQAPDEC			

表 4. ISPF 変数プールの QMF 変数 (続き)

変数のタイプ	変数名	説明
CONVERT コ マンド	DSQCL nnn	CONVERT コマンドの処理中に、 QMF はこれらの変数を更新する。
	DSQCQ nnn	
	DSQCQCNT	
	DSQCQLNG	
	DSQCQTYP	
コマンド・メ ッセージ	DSQCATTN	コマンド・インターフェースによっ て出されたコマンドを処理するた びに、QMF はこれらの変数を更新す る。
	DSQCIM00	
	DSQCIM nn	
	DSQCIMID	
	DSQCIMNO	
	DSQCIMSG	
	DSQCSESC	
照会メッセー ジ	DSQCIQ00	RUN QUERY がエラー・メッセー ジを戻すと、QMF はこれらの変数 を更新する。
	DSQCIQ nn	
	DSQCIQID	
	DSQCIQMG	
	DSQCIQNO	
	DSQCISQL	

ISPF 変数プールで QMF 変数を使用する場合には、変数に 8 文字の名前を使用してください。上記の変数の値および拡張名の説明については、261 ページの『付録 E. QMF グローバル変数表』を参照してください。

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、アプリケーションの言語に関係なく同じです。戻りコードは正の値またはゼロです。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。

アプリケーション用コマンド・インターフェースの使用

戻りコードは、ユーザーの EXEC または CLIST 内の変数に入れます。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 に入れます。

次の例は、戻りコードを調べる EXEC を示しています。

例

アプリケーションには、次のコードが含まれています。

```
ADDRESS ISPEXEC SELECT PGM(DSQCCI) PARM(RUN QUERYA (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
```

上記のコードは、照会を実行してから、REXX RC を使用してエラーをテストします。

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 ~ 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・インターフェースがこれらのいずれかのコードを戻す場合、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 共用プールに戻します。各コードを表 5 に示します。

表 5. 戻りコード 0 ~ 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ってのみ戻されます。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ユーザーに戻します。

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。失敗によって、アプリケーションの共用プールへの変数のコピーが不可能になっています。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ことが言えます (変数は、アプリケーションの共用プールにコピーされていれば、設定されています)。

これらの戻りコードは、通常、0 ~ 16 の戻りコードより重大なエラーを示しています。そのうちのいくつかについては、IBM 担当員に問い合わせる必要があるかもしれません。

次の表は、20 以上の値を持つ戻りコードの説明を示しています。共用変数は、QMF 変数 (および、現行コマンドが STATE コマンドの場合は、STATE 変数) を参照します。

コードによっては、コマンドが実行されても共用変数が設定されないことがあります。このことは、QMF が STATE コマンドを正しく実行したことを意味します。QMF は、コマンド・インターフェースが更新済みの共用 QMF 変数と STATE 変数を設定すると想定しました。しかし、コマンド・インターフェースは、エラー・コードの説明に示されている理由で、設定に失敗しました。各コードを表 6 に示します。

表 6. 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 VCOPIY 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。
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。たとえば、ユーザーが ISPF SELECT PGM コマンドを使用しないで DSQCCI を呼び出した。
100	アンカーが見つからない。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: QMF コマンド環境の使用

CICS ユーザーへの注

REXX は、QMF CICS ではサポートされていないので、ADDRESS QRW は、CICS 環境では機能しません。

REXX 言語は、コマンドのデフォルトの解決を決定するコマンド環境において機能します。デフォルトのコマンド環境は、オペレーティング・システムによって、CMS または TSO です。

QMF を開始するとき、REXX ADDRESS コマンドを介して、QMF をデフォルトのコマンド環境として設定することができます。このコマンドを単独で、または QMF コマンドの前で使用することができます。

```
ADDRESS QRW
ADDRESS QRW command
```

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。ADDRESS QRW コマンドを使用した場合、QMF は、そのコマンドについてだけのコマンド環境になります。

QMF ロジックを持つプロシージャーを使用している場合、QRW がデフォルトのコマンド環境になります。呼び出し可能インターフェースまたは REXX コマンド環境 ADDRESS QRW のいずれを使用しても QMF は同様に機能します。

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
⋮
```

図9. QMF コマンド環境の使用例

第 6 章 ISPF 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。こうしたアプリケーションを作成する方法の 1 つに、ISPF を使用する方法があります。

CICS ユーザーへの注

ISPF は CICS 環境で実行できないので、ISPF サービスを CICS のもつで使用することはできません。

ISPF を使用して、メインフレーム・システムでエンド・ユーザー・インターフェースを提供することができます。ISPF は、QMF 呼び出し可能インターフェースまたはコマンド・インターフェースと共に使用することができます。

本章では、ISPF と共に呼び出し可能インターフェースを使用する方法に関する考慮事項について概説します。呼び出し可能インターフェースの使用法の一般情報については、19 ページの『第 3 章 呼び出し可能インターフェース』を参照してください。コマンド・インターフェースの使用法については、29 ページの『第 4 章 アプリケーション用コマンド・インターフェースの使用』を参照してください。

ISPF アプリケーションからの QMF の開始および実行

呼び出し可能インターフェースに関する考慮事項は、以下のとおりです。

呼び出し可能インターフェースは **ISPF** ダイアログの言語と一致しなければならない

ISPF ダイアログが PL/I プログラムの場合、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。

正しい言語 **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 がファンクション・キーの設定やラベルなどの ISPF 環境を指定変更しないようにすることができます。QMF を開始するアプリケーションを開始するには、次の ISPF ステートメントを使用してください。

```
SELECT PGM(MYPROG) NEWAPPL(DSQn)
```

ISPF を使用する QMF アプリケーションの作成

ここで、 n は NLF ID です。次に、PL/I プログラム MYPROG が、呼び出し可能インターフェース START コマンドを使用して QMF を開始します。

ID DSQn は、QMF の開始後も、ISPF 環境が変更されないように保証します。

NLF ID のリストについては、70 ページの表 7 を参照してください。

STATE コマンドの代わりに GET GLOBAL または SET GLOBAL を使用する

GET GLOBAL および SET GLOBAL コマンドは、すべての QMF グローバル変数に対して機能しますが、STATE コマンドは、状態情報が入っている変数に対してしか機能しません。261 ページの『付録 E. QMF グローバル変数表』にあるこれらの変数に関する表を参照してください。

変数を含む照会の実行

アプリケーションは、変数を含む照会を実行することができます。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現れます。これらの変数に適切な値を与えると、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 VDEFINE 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。ISPF 対話管理の手引き および解説書 を参照してください。

- 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。PDF マクロは、REXX プログラムを使用して作成することができます。

- ISPF ダイアログを使用して照会を作成する。

SQL 照会が入っているファイルを作成するために、プログラムは 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することができます。次に、このファイルを QMF にインポートして実行することができます。

ISPF のもとでの QMF ロジックを持つプロシージャからのプログラムの呼び出し

ISPF のもとで QMF を実行している場合、ロジックを持つプロシージャから呼び出し可能インターフェース・プログラムまたは REXX プログラムを呼び出すために、ISPF SELECT サービス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・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、PGM キーワードを使用してください。このコマンドの構文は次のとおりです。

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

REXX プログラムの場合は、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)"
```

cmdname は呼び出し可能インターフェース・プログラムまたは REXX プログラムの名前です。

ロジックを持つプロシージャからの ISPF コマンドの使用

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。QMF プロシージャから、ISPF SELECT CMD を発行してください。

正しい ISPF 環境を設定し、ISPF コマンドが入っている REXX 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 コマンドを使用してください。

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

userprogram は、ISPF コマンドが入っている REXX プログラムです。

たとえば、ISPF コマンドが入っている REXX 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)"
```

ISPF の詳細については、*ISPF 対話管理の手引きおよび解説書* を参照してください。

ISPF コマンドが入っている REXX プログラムを実行するために、**CMS DIALOG** または **TSO DIALOG** のように、QMF の CMS コマンドまたは TSO コマンドを使用することもできます。QMF が ISPF SELECT CMD ステートメントを出してくれます。

ISPF を使用する QMF アプリケーションの作成

ISPF のもとで QMF を実行中に、ロジックを持つプロシージャが ISPF サービスを必要とするプログラムを開始する場合、このプロシージャは、上の例で示した ISPF SELECT CMD 環境を使用してこのプログラムを開始する必要があります。たとえば、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が DB2 の DSN コマンドを開始するものとします。DSN コマンドは ISPF サービスを使用するので、以下のいずれかのコマンドを使用して DSN コマンドを出す必要があります。

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

または

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

ここで、DSNEXEC には ADDRESS TSO DSN ステートメントが入っています。

呼び出し可能インターフェースの考慮事項

QMF のアプリケーション内で、QMF バージョン 7 より前のバージョンでリンク・エディットされており、かつ、呼び出し可能インターフェースを使用する LIBDEF 関数を使用したい場合は、QMF バージョン 7 インターフェース・モジュールを使用してユーザーのアプリケーションをリンク・エディットし直す必要があります。

ISPF で EDIT コマンドを使用する方法

ISPF のもとで QMF アプリケーションを実行する場合、QMF SQL の照会またはプロシージャを次のコマンドを使用して編集することができます。

```
EDIT QUERY  
EDIT PROC
```

PROC または QUERY パネル内から 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。ここで、 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。

このデフォルトを変更するには、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
```

name には、以下のいずれかを指定できます。

- 使用可能なエディター

- DSQE 以外のアプリケーション ID を指定している REXX プログラムの名前。
QMF が提供するファンクション・キーとは異なるファンクション・キーを使用する場合 QMF アプリケーション ID とは異なるアプリケーション ID を使用することができます。

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。たとえば、PDF EDIT RECOVERY オプションが DSQ n EDRT PROFILE データ・セット・メンバーを必要としている場合 (ここで、 n は NLF 文字)、このメンバーは EDIT コマンドを使用する前に存在していなければなりません。

QMF EDIT コマンドの詳細については、オンライン・ヘルプ、および「DB2 QMF 解説書」を参照してください。

アプリケーションをデバッグするための ISPF の使用

QMF トレース機能は、QMF のメッセージとコマンドしかトレースしません。アプリケーションの ISPF コマンドをトレースするには、メッセージを ISPF ログ・ファイルまたはデータ・セットに書き込みます。この ISPF サービスは、125 ページの『第 10 章 QMF アプリケーションのデバッグ』に示す QMF トレース機能を補います。

ISPF ログ・サービスの使用

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。たとえば、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)
```

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 です。

PDF ダイアログ・テストの使用

インストール先に PDF がある場合、ダイアログ・テスト・サービス (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することができます。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。デバッグは対話式に行うことが可能です。アプリケーションの全部または一部を実行し、結果を調査し、変更し、再実行することができます。さらに、ダイアログ・テストを使用して、次の処理を実行できます。

- 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

ISPF を使用する QMF アプリケーションの作成

- 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

ダイアログ・テスト・サービスのトレース・オプション (TRACES) によって、トレース定義を作成、変更、および削除することができます。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ことができます。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・テストの LOG オプションを使用して、ISPF ログをブラウズすることができます。あるいは、ISPF を終了するときに印刷出力を調べることができます。

一般的な ISPF サービスおよび特殊なダイアログ・テストの詳細については、「*ISPF Dialog Management Guide and Reference*」を参照してください。

第 7 章 2 か国語使用のアプリケーションの作成

多くの対話式ビジネス・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。

英語以外の言語の QMF 環境は、各国語機能 (NLF) です。NLF は、ドイツ語や日本語などの特定の言語用に調整した QMF セッションをユーザーに提供します。

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、また変換可能なアプリケーションを作成することができます。本章では、複数の言語環境、または英語以外の言語環境で QMF を使用する方法について説明します。

アプリケーション用の 2 か国語使用の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 か国語使用の方法は、特定のオブジェクトによって異なります。

照会 指示照会および QBE 照会は自国語で作成することができ、SQL 照会は英語で作成することができます。

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。

グローバル変数 DSQEC_FORM_LANG は、SAVE コマンドに使用す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、書式を主要セッション言語で保管することを指定します。

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。

SAVE コマンドを使用することによって、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。たとえばフランス語では、SEMAINE_F という名前の書式を WEEKLY_F という名前で英語で保管するコマンドは、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。

コマンド言語変数の使用

NLF セッションで英語コマンドが使用されるようにするには、主要言語変数 DSQEC_NLFCMD_LANG を英語に設定します。この変数によって、英語と NLF セッションの主要言語との間での切り替えを行うことができます。

以下の例では、アプリケーションは WEEKLY_P という名前のプロシージャーです。

```
"GET GLOBAL (CURR_LANG=DSQEC_NLFCMD_LANG"  
"SET GLOBAL (DSQEC_NLFCMD_LANG='1')"  
"RUN PROC WEEKLY_P"  
"SET GLOBAL (DSQEC_NLFCMD_LANG=CURR_LANG"
```

上記のコマンドは、初期プロシージャー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。これらのコマンドは、以下の処理を行います。

主要言語値を保管する

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。値を保管したら、DSQEC_NLFCMD_LANG を英語を示す値 1 にリセットします。

アプリケーションを実行する

QMF セッションを英語に設定すると、英語のアプリケーションを実行することができます。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せん。

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」パネルの場合、デフォルトのファイル・タイプも主要言語で表示されます。

NLF がプロファイル内で大文字オプションを指定している場合、ユーザーが英語のコマンドを実行しても、QMF はユーザーの主要言語オプションに従います。

主要言語に戻る

アプリケーションが終了した後、コマンド言語変数を元の値にリセットする必要があります。

2 国語使用のアプリケーションでの初期プロシージャーの使用

アプリケーションが QMF を開始し、初期プロシージャーを実行すると、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ーを実行します。このプロシージャーがエラーを検出すると、QMF は終了します。たとえば、ユーザーが英語で実行中に主要言語で END コマンドを出すと、QMF はそのコマンドをエラーとして解釈し、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 国語使用のアプリケーションを処理するように、初期プロシージャーを変更する。
- 2 国語使用の初期プロシージャーには、図 10 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"
```

図 10. 2 国語使用のアプリケーション内の初期プロシージャー

- END コマンドの後に、初期プロシージャー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ーを実行しないようにすることができます。

英語のコマンドの使用

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。しかし、プロンプト・パネルまたはメッセージを主要言語で表示するためには、主要言語変数が英語に設定されていない場合でも、いくつかの英語のコマンドは任意の NLF で実行できなければなりません。

たとえば、英語で作成し、しかもある NLF で実行させたい対話式アプリケーションがある場合、ユーザー・カスタマイズ・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、次の例のように、フランス語の NLF セッションで実行できるメッセージの表示には INTERACT コマンドが必要です。

```
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 */
```

2 か国語使用のアプリケーションの作成

以下のコマンドは、どの NLF でも機能します。

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

複数言語環境

QMF のインストール先に 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には、それぞれの QMF セッションごとに、主要言語を選択することができます。たとえば、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。

QMF セッション環境

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。以下の節では、NLF と英語環境の類似点と相違点をリストしています。

環境の類似点

多くの点で、QMF セッション環境は、どの NLF が作動中であるかは関係なく同じです。最も重要な類似点は次のとおりです。

能力

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。すべての一時記憶域オブジェクトを作成して保管し、報告書をフォーマット設定して印刷し、SQL コマンドを出すことができます。また、指示照会、SQL 照会、および QBE 照会を実行し、QMF プロシーチャーを実行することができます。英語環境と NLF 環境の間の相違点は、何ができるかではなく、端末で何を入力する必要があるかということと端末画面にどの言語が表示されるかです。

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

これらは同じものなので、変換されません。

システム・コマンド

CMS、TSO、または CICS および ISPF コマンドは、QMF の CMS、TSO、または CICS コマンドを介して QMF から出すことができます。このコマンドは、変換による影響を受けません。CMS、TSO、または CICS の後ろ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重要な相違点をいくつか示します。

QMF 言語

各 NLF には、QMF 言語用の動詞とキーワードの完全なセットがあります。NLF 言語環境で操作中の場合には、これらの動詞とキーワードが QMF コマンドに現れていなければなりません。特定の NLF では、これらの語は変換される場合があります。

たとえば、ドイツ語の NLF では、動詞 DISPLAY とキーワード PROC は、それぞれ ANZEIGEN および PROZEDUR に変換されています。ドイツ語の NLF セッションでは、QMF はコマンド ANZEIGEN PROZEDUR を理解しますが、DISPLAY PROC を理解しません。

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。QMF のメッセージとパネルは、QMF コマンドの動詞とキーワードと同様に、変換され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。照会名のような可変情報は変換されません。

許可されるパネル入力

ユーザー入力を必要とするプロンプト・パネルや書式パネルのような多くの QMF パネルでは、入力の範囲がキーワードの小さなセットに制限されているものがあります。許可される値の大部分が変換されます。たとえば、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。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。しか

2 か国語使用のアプリケーションの作成

し、QMF が提供するキーワードの一部として、その名前は変換することができます。パラメーターによっては、想定できる値も変換することができます。

たとえば、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、可能な値は、GROSS、KETTE、および GEMISCHT になります。

エクスポート後、および保管後の書式オブジェクト

SAVE、EXPORT、および IMPORT コマンドによって、書式オブジェクトを保管するための言語を指定することができます。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。これらのコマンドの詳細については、「DB2 QMF 解説書」を参照してください。

サンプル表およびサンプル照会

IBM は、英語のサンプル表およびサンプル照会の変換バージョンを、いくつかの NLF に用意しています。たとえば、日本語のユーザー用の英語の表から変換されたサンプル表があります。

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。このような変数には、以下のものがあります。

- QMF コマンドの動詞、オブジェクト名、およびオプション ID
- インストール先で定義するパネル名
アプリケーション用の独自のパネルを作成する場合、アプリケーションを実行される言語ごとに変換したパネルのセットが必要です。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。
- インストール先定義のメッセージ ID
メッセージもパネルと同様に、適切な NLF 言語に変換する必要があります。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。

変数を使用すれば、複数の NLF で同じプログラム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。さらに、QMF には、アプリケーション用として設計されたコマンドが用意されています。

本章では、ユーザーが一般的にプログラムで使用する QMF コマンド、およびアプリケーション開発でのこれらのコマンドの使用方法について説明します。コマンドとその構文の詳細については、「DB2 QMF 解説書」を参照してください。

CONNECT

QMF CONNECT コマンドを使用して QMF セッション中に、リモート作業単位を分散ネットワーク内の別のシステムに関連付けることができます。また、QMF CONNECT コマンドを使用して、QMF によってサポートされるリモート・データベースにアクセスすることもできます。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場合、このコマンドを以下のインターフェースやプロシージャから出すことができます。

- 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（線形プロシージャまたはロジックを持つプロシージャ）

QMF CONNECT コマンドを使用してリモート作業単位を開始すると、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 を開始し QMF コマンドを発行するすべての呼び出し可能インターフェース・プログラムおよびコマンド・インターフェース・プログラムは、ユーザーと同じシステム（ローカル・システム）に常駐していなければなりません。プログラムは、ローカル・システムで QMF を開始すれば、QMF CONNECT を発行してリモート・データベースに接続することができます。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、現行ロケーション（リモート・データベース）で実行されます。
- QMF によって開始されるすべてのプログラムは、QMF を実行中のオペレーティング・システム（ローカル・システム）の規則に従っていなければなりません。
- 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を使用する場合、システム固有のすべてのコマンドおよび大

アプリケーションでの QMF コマンド

部分の QMF コマンドは、QMF を実行中のシステム (通常は、ローカル・システム) で実行されることに注意してください。ただし、QMF コマンドが、

- 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

には、これらのコマンドは、現行ロケーションのデータベースに影響を与えます。

注: DB2 QMF バージョン 8.1 では、z/OS の CONNECT コマンドが変更され、128 バイトの許可 ID がサポートされています。許可 ID は、QMF 線形プロシージャ内で複数の行にわたって継続する場合は、二重引用符で区切る必要があります。すべての継続行の 1 桁目には、正符号 (+) がなければなりません。詳しくは、「DB2 QMF 解説書」を参照してください。DB2 QMF バージョン 8.1 は、接続に使用されている現行の許可 ID よりも最大長が短い許可 ID を持つ QMF オブジェクト表を含むデータベースとの接続をサポートしていません。例えば、現行の許可 ID が "A23456789" で、最大長が 8 の許可 ID を持つ QMF オブジェクト表を含むデータベースに接続しようとすると、エラーが戻され、そのデータベースとの接続は確立されません。

新規のロケーションへの接続が長い名前に与える影響

QMF の始動時に、新しいロケーションまたは初期ロケーションに接続する場合、長い名前のサポートは、以下に示すように、接続しようとしているデータベースに有効なデータベース制限と QMF オブジェクト表によって異なります。

- 接続に使用される許可 ID の長さは、データベースでサポートされている許可 ID、または QMF 制御表でサポートされている許可 ID のいずれかよりも長くなってはいけません。
- 表名の最大長は、接続先のデータベースでサポートされている最大長によって異なります。
- 表の列名の最大長は、接続先のデータベースでサポートされている最大長によって異なります。
- QMF オブジェクト名の最大長は、QMF 制御表でサポートされている最大長によって異なります (QMF バージョン 7.2 以前の場合は 18 バイト、長い名前の QMF オブジェクト表にマイグレーションした後の DB2 QMF バージョン 8.1 の制御表の場合は 128 バイト)。

例

CMS を実行中のローカル VM システム (SANJOSE) にログオンしているものとし、以下に挙げることを行う REXX 呼び出し可能インターフェース・プログラムを作成します。

1. QMF セッションを開始する。

```
CALL DSQCIX "START"
```

2. リモート DB2 データベース (DALLAS) に接続する。

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

3. リモート・データベースにデータを照会し、データをフォーマット設定し、報告書を印刷するロジックを持つプロシージャーを実行する。

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

プロシージャー EARNINGS には次のロジックが含まれています。

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

このプロシージャーには、別の CONNECT コマンドは含まれていません。

4. QMF セッションを終了する。

```
CALL DSQCIX "EXIT"
```

このプログラムを作成する場合、以下のことに注意してください。

- アプリケーション・プログラムはローカル (SANJOSE) VM システムに常駐していなければなりません。
- QMF セッションはローカル (SANJOSE) VM システムで開始します。
- プロシージャーはリモート・データベース (DALLAS) に常駐していなければなりません。3 のステップでアプリケーションがプロシージャーを実行すると、DALLAS が現行ロケーションになります。
- 2 のステップで CONNECT コマンドの後にアプリケーションまたはプロシージャーで使用するすべての QMF オブジェクト (この場合は、照会および書式) は、リモート・データベース (DALLAS) に常駐していなければなりません。
- 3 のステップでプロシージャーで実行する SQL 照会 EARNQ は、DALLAS での DB2 データベースに対して実行されます。
- プロシージャー EARNINGS の PRINT コマンドは、現行ロケーション (DALLAS) のプロファイルによって指定されているプリンターで報告書を印刷します。この例では、現行ロケーション (DALLAS) のプロファイルによって、プリンターがローカル VM システム (SANJOSE) にあると定義されていると想定しています。

QMF CONNECT コマンドを使用してのリモート・ロケーションへの接続の詳細については、オンライン・ヘルプを参照してください。

END

QMF セッションを終了するために、アプリケーション中に END コマンドを組み込むことができます。エンド・ユーザーが対話式 QMF セッションを終了して、制御をアプリケーションに戻す場合は、終了ファンクション・キーを押すか、コマンド行に END コマンドを入力しなければならないように、アプリケーションを設計することもできます。

END コマンドを制御する規則は、それを実行するセッションのタイプによって異なります。この節では、以下の各タイプの QMF セッションでの END コマンドの機能について説明します。

- 呼び出し可能インターフェースによって開始されたセッション
- 初期プロシージャーを伴う、ISPF を使用する対話式セッション
- 初期プロシージャーを伴わない、ISPF を使用する対話式セッション
- INTERACT コマンドによって開始された対話式セッション
- バッチ・モード・セッション

呼び出し可能インターフェースによって開始されたセッション

呼び出し可能インターフェースによって開始された対話式セッションで、ユーザーによって End コマンドが実行されると、対話式セッションが終了し、制御が呼び出し側のアプリケーションに戻ります。QMF がアクティブ・セッションを終了する前に、「ホーム」パネルが現行パネルになります。QMF はアクティブのままです。QMF は、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開始された後は、EXIT コマンドまたは重大エラーによってのみ終了します。

初期プロシージャー (DSQSRUN) を伴う対話式セッション

次のキーワードを使用して QMF を開始すると、QMF は初期プロシージャーを実行する対話式セッションを開始します。

```
DSQSRUN=xxxxx,DSQSMODE=I
```

ここで、xxxxx (DSQSRUN キーワードの値) は、QMF 初期プロシージャーの名前です。このキーワードの説明は 68 ページの『START コマンドのキーワード』にあります。

QMF を開始すると、QMF は初期プロシージャーを実行します。このプロシージャーが終了すると、現行パネルが「ホーム」パネルでない限り、ユーザーは対話式セッションに入っています。現行パネルがホーム・パネルの場合、QMF は対話式セッションを開始しません。その代わりに、以下の両方の条件が真の場合、QMF はただちに初期プロシージャーを再始動します。

- 重大エラーが発生していない。
- DSQEC_RERUN_IPROC グローバル変数が 1 に設定されている。

プロシージャの最後の現行パネルが「ホーム」パネルになるような初期プロシージャは作成しないようにしてください。初期プロシージャの最後の現行パネルがホーム・パネルであると、割り込み不能ループが発生し、QMF は、開始されていないか、または初期プロシージャを実行中であるように見えます。プロシージャに QMF EXIT または INTERACT コマンドのいずれかが含まれているようにしても、この問題を回避することができます。

エンド・ユーザーが対話式セッションで END コマンドおよび DSQEC_RERUN_IPROC= 1 を出すと、QMF は単純に初期プロシージャを再始動します。セッションを終了するには、EXIT コマンドを使用してください。

QMF が呼び出し可能インターフェースによって開始されていない場合には、DSQ_RERUN_IPROC を使用して、QMF が初期プロシージャを再実行するかどうかを制御することができます。DSQEC_RERUN_IPROC = 0 を設定すると、END コマンドの実行時に初期プロシージャが再実行されないで QMF が終了します。この変数は、呼び出し可能インターフェース・アプリケーションには影響を及ぼしません。

初期プロシージャを伴わない対話式セッション

この場合、QMF の開始時に DSQSRUN パラメーターは指定しません。これによって、ユーザーが制御を受ける前にどのプロシージャも実行されなくなります。

エンド・ユーザーがこのような対話式セッション内から END コマンドを実行すると、QMF は以下のいずれかのタスクを実行します。

- 現行パネルがホーム・パネルでない場合、現行パネルをホーム・パネルにします
- 現行パネルが「ホーム」パネルであれば、セッションを終了とマークします。END コマンドがオンラインで実行されると、セッションは即時に終了します。このコマンドがアプリケーションで実行されれば、アプリケーションが終了すると必ずセッションも終了します。

INTERACT コマンドによって開始された対話式セッション

アプリケーションは、58 ページの『INTERACT』で説明する INTERACT コマンドを使用すれば、現在の対話式 QMF セッション内で新しい対話式 QMF セッションを開始することができます。古いセッションは、初期プロシージャを伴うかどうかに関係なく、1 次セッションであっても、別のアプリケーションによって開始されたセッションであってもかまいません。

新しいセッションでの END コマンドは、ユーザーがオンラインで入力することもアプリケーションで発行することもできます。いずれの方法で END コマンドを実行しても、現行パネルが何であれ、対話式セッションは終了とマークされます。

END コマンドをオンラインで出せば、セッションはただちに終了します。アプリケーションで出すと、セッションはアプリケーションの終了時に終了します。セッションが終了すると、制御はそれを開始したアプリケーションに戻ります。

バッチ・モード・セッション

QMF バッチ・モード・セッションは、QMF でサポートされるすべての環境の非対話式セッションで実行されます。デフォルトであるバッチ・モード (DSQSMODE=BATCH) を指定することによって、QMF を画面表示なしで開始することができます。

DSQQMFE を使用する場合、DSQSRUN を使用して初期プロシージャーを指定する必要があります。ただし、呼び出し可能インターフェースを使用する場合、初期プロシージャーを指定する必要はありません。

バッチ QMF セッション中に、初期プロシージャーは END コマンドを出すことも、END コマンドを出すアプリケーションを開始することもできます。結果は、初期プロシージャーを伴わない対話式セッションの場合と似ています。END コマンドによって、以下の処理が行われます。

- 現行パネルがホーム・パネルでない場合、現行パネルをホーム・パネルにする
- 現行パネルがホーム・パネルであれば、セッションを終了とマークする

END コマンドは、初期プロシージャーによって出された場合、セッションをただちに終了します。アプリケーションで出された場合、セッションはアプリケーションが QMF EXIT コマンドを出すと、そのたびに終了します。

セッション中は、対話は許可されません。したがって、セッションは新規セッションを開始できません。

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。バッチ・モードには、1 つのセッションしかありません。対話式セッションには、1 次セッションと、INTERACT コマンドで開始されたすべてのセッションがあります。

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。

QMF を使用してアプリケーションを実行する場合、呼び出し可能インターフェース・プログラムに QMF EXIT ステートメントを組み込むことが大切です。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

アプリケーションだけが、このグローバル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。

GET GLOBAL

GET GLOBAL コマンドを使用すると、アプリケーションの QMF グローバル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 コマンドに指定するパラメーターは、アプリケーション・プログラムが変数名および GET GLOBAL コマンドによって戻される値を保管するために使用する記憶域を定義します。

number of varnames (変数の数)

要求する変数の数

varname lengths (変数名の長さ)

指定する各変数名の長さのリスト

変数名の長さは、記憶域内のグローバル変数名の実際の長さに等しくなければなりません。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 (変数名)

QMF 変数の名前前のリスト

グローバル変数名には、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 (値の長さ)

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、QMF は右側を切り捨て、右側が切り捨てられた値を戻します。
- 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、QMF は後書きブランクで埋めた値を戻します。
- 整数の長さは、常に 4 バイトでなければなりません。

values (値)

変数値のリスト

value type (値のタイプ)

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 コマンドはユーザーを対話式 QMF セッションまたは GDDM ICU セッションに置きます。ユーザーは、これらのセッションにいる間、これらのプロダクトの通常の対話式セッションにいる場合と同様にコマンドを入力することができます。

INTERACT 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。

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

通常 QMF パネルを表示する QMF コマンドの後に INTERACT コマンドを出します。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

通常 QMF に GDDM ICU を開始させ、「ICU」パネルを表示させるコマンドの後に INTERACT コマンドを実行します。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

次の例では、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ジャが実行されます。

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

図 11. INTERACT コマンドがない単純なプロシージャ

QMF は、フォーマット設定済みデータおよび "OK,your procedure was run (プロシージャが実行されました)" というメッセージが入っている REPORT パネルを表示します。

複数のステップからなるプロシージャを作成することもできます。プロシージャの中間結果を見たい場合、INTERACT コマンドを使用する必要があります。複数の照会を

実行するプロシージャーの中間結果を見るには、最初の RUN 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 12. プロシージャーでの INTERACT の使用

「ホーム」パネルからこのプロシージャーを実行すると、QMF はフォーマット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。このプロシージャーは、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。

INTERACT コマンドを、呼び出し可能インターフェースを介して出しても、同じ結果が得られます。ただし、REXX では同じコマンドが次のようになります。

```
⋮
call dsqcix "RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
call dsqcix "INTERACT"
call dsqcix "RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"
⋮
```

図 13. REXX アプリケーションでの INTERACT の使用

Call dsqcix "INTERACT" の行は、呼び出し可能インターフェースを介して INTERACT コマンドを出すための REXX 構文です。呼び出し可能インターフェースを介して INTERACT コマンドを実行する場合には、使用するプログラム言語に適した構文を使用しなければなりません。

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 グローバル変数をゼロ (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、このグローバル変数を設定することができます。

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

INTERACT 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。たとえば、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した場合、この新しいセッションが終了するとき、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ています。

ユーザーが QMF オブジェクト・パネルから END コマンドを出した後、(対話式 QMF のように) アプリケーションに QMF ホーム・パネルを表示させたい場合、228 ページの『INTERACT ループを使用する REXX の例』のロジックを追加してください。

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 コマンドのキーワード DSQSMODE を設定することです。DSQSMODE キーワードの詳細については、70 ページの表 7 を参照してください。

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

`INTERACT command`

ここで、*command* は指定するコマンドです。QMF は、QMF とユーザーの間でコマンドの実行に関するダイアログが実際に行われると、このコマンドを対話式に実行します。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。

たとえば、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 ?`

QMF バッチ・セッションのように対話式実行が許可されていない場合は、コマンド・フォームの INTERACT はこの後のコマンドに影響を与えません。

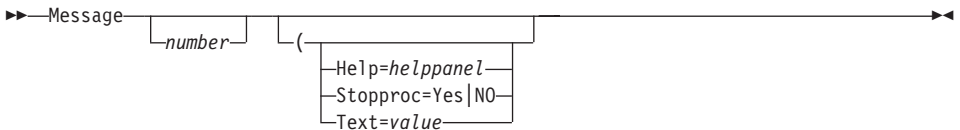
DSQAO_INTERACT という名前の変数を調べれば、現行セッションで対話式実行が許可されるかどうかを検査することができます。値 1 は INTERACT が許可されること

を意味します。バッチ・アプリケーションでは、対話式実行が許可されません。DSQAO_INTERACT の詳細については、263 ページの『付録 E. QMF グローバル変数表』を参照してください。

MESSAGE

アプリケーションを作成する場合、ユーザーに表示する情報、またはユーザーが次に実行すべき機能に関する特定のメッセージをそれらのユーザーに送信しなければならない場合があります。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 が表示するように指定することもできます。

MESSAGE コマンドの構文:



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 では、独自のパネルを作成して表示したい場合は、そのパネルの定義を ISPF パネル・ライブラリーに入れる必要があります。このライブラリーは、ISPLIB ファイルまたはデータ・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 において、*number* (番号) を指定した場合、*helppanel* (ヘルプ・パネル) はデフォルトで、*number* (番号) 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・パネル標識を定義していない場合、MESSAGE コマンドはメッセージ・ヘルプを提供しません。その代わりに、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

Stopproc を使用すると、プロシージャ終了スイッチを設定することによって線形プロシージャの実行を抑止できます。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だけに影響を与えません。

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。

変数 DSQCM_MESSAGE を調べれば、プロシージャ終了スイッチがオンかどうかを検査することができます。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せん。

TEXT オプション

TEXT= 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。ブランク文字を含む値は、区切り文字で囲む必要があります。メッセージ値に使用することが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。区切り文字に二重引用符を使用した場合、二重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。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 はテキストを大文字変換しませんが、MESSAGE が DSQCCI (コマンド・インターフェース) を介して出された場合、ISPF はテキストを大文字変換する場合があります。

メッセージに引用符が入っている場合、**TEXT=** の指定では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に残ります。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。

メッセージ生成のための MESSAGE コマンドの使用例

プロシージャーを使用して、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 が最初の報告書を表示した後、2 番目の報告書に進む準備ができたときに対話式セッションを終了するように、ユーザーに指示するメッセージを表示する必要があるものとします。図 14 に示すような線形プロシージャーを作成することができます。このプロシージャーには、REPORT パネル上に表示される、MESSAGE コマンドによって定義したメッセージが含まれています。メッセージを REPORT パネルに表示するには、MESSAGE コマンドを INTERACT コマンドの直前に置いてください。

```

:
:
:   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 14. MESSAGE コマンドの使用例

ロジックを持つプロシージャーを組み込む場合、図 15 に示すように、テキスト・ストリングの代わりに REXX 変数を使用することができます。REXX 変数を使用する場合、*mesagetext* 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 15. プロシージャーにおける MESSAGE コマンドでの REXX 変数の使用

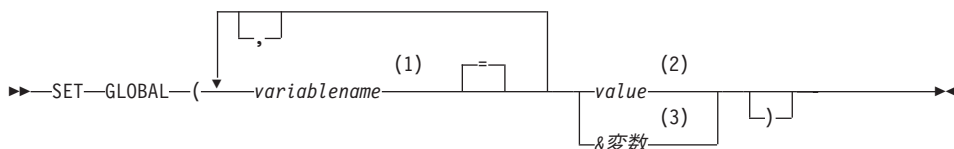
QMF REXX プロシージャーからのメッセージの発行例

```

/* QMF REXX PROCEDURE */
MSGTEXT="ZAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"||,
"AAAAAAAAAX"MESSAGE(TEXT=("MSGTEXT"))/*MAXTEXT=360PARANS*/EXIT

```

QMF 線形プロシージャーからの QMF MESSAGE コマンドの実行例



注:

- 1 値を割り当てられたグローバル変数の識別
- 2 グローバル変数を構成する文字ストリング。空白文字を含む値は、区切り文字で囲む必要があります。グローバル値に使用することが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。区切り文字として二重引用符を使用する場合、二重引用符もグローバル変数の一部に組み込まれます。SET GLOBAL コマンドが線形プロシージャから入力され、変数が複数行に渡る場合、値は引用符で囲み、継続文字 (+) を各行の先頭に使用する必要があります。区切り文字として二重引用符を使用する場合、二重引用符もグローバル変数値の一部に組み込まれます。複数行に渡る場合、括弧を区切り文字として使用することはできません。
- 3 グローバル変数の内容を含むグローバル変数名

varname=value (変数名 = 値)

変数名に値を割り当てます。

たとえば、DEPT という名前のグローバル変数を設定するには、次のコマンドを出します。

- 線形プロシージャの場合
SET GLOBAL (DEPT=38
- ロジックを持つプロシージャの場合
"SET GLOBAL (DEPT=38"

SET GLOBAL コマンドの詳細については、「DB2 QMF 解説書」を参照してください。

グローバル変数は、定義後、変数をリセットするか、QMF セッションを終了するまで定義済みのまま残ります。RESET GLOBAL コマンドの使用方法については、「DB2 QMF 解説書」を参照してください。

SET GLOBAL: 拡張構文

REXX 以外の言語 (アセンブラ、C、COBOL、FORTRAN、または PL/I) で作成したアプリケーション内でグローバル変数の値を変更するには、拡張構文で SET GLOBAL コマンドを使用する必要があります。このコマンドの例は、131 ページの『付録 A. 呼び出し可能インターフェース言語のサンプル・コード』の該当する言語のサンプル・プログラムを参照してください。

アプリケーションでの QMF コマンド

SET GLOBAL 拡張構文コマンドで使用する変数名の最大長は 17 文字です。変数値の最大長は 32 768 文字です。

▶▶SET GLOBAL—(—| 変数定義 |—————▶▶

変数定義:

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

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

number of varnames (変数の数)

要求する変数の数。

varname lengths (変数名の長さ)

指定する各変数名の長さのリスト。

変数名の長さは、記憶域内のグローバル変数名の実際の長さに等しくなければなりません。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 (変数名)

QMF 変数の名前前のリスト。

value lengths (値の長さ)

変数値の長さのリスト。

次の規則が変数値に適用されます。

- 指定した値の長さが、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。
- 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長い場合、QMF に保管されるときに値の中に認識できない文字が組み込まれる可能性があります。
- 整数の長さは、常に 4 バイトでなければなりません。

QMF は、記憶域内の、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 (値)

変数値のリスト。

value type (値のタイプ)

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。

REXX 呼び出し可能インターフェースで SET GLOBAL を使用している場合は、64 ページの『SET GLOBAL』に示しているように、SET GLOBAL コマンドの線形構文しか使用できません。この線形構文の場合、グローバル変数名の最大長は 17 文字で、変数値の最大長は 2,000 文字です。

グローバル変数の使用規則

- 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バーサンドが付きません。
- QMF 書式は、変数名または総計変数名を形成するために設定されたグローバル変数を認識しません。
- QMF 書式は、名前に疑問符があるグローバル変数を認識しません。

グローバル変数名の定義規則

- グローバル変数名は、コマンド行で入力する場合は 17 文字に、呼び出し可能インターフェースを介して入力する場合は 18 文字に制限されています。ただし、SET GLOBAL コマンドの制限のため、使用する名前は 17 文字にしてください。
- グローバル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。
- グローバル変数は DSQ で開始してはなりません。これらの文字は、QMF の事前定義グローバル変数として QMF によって予約されています。
- グローバル変数名の最初の文字は、英字 (A から Z) か、以下に示すいずれかの特殊文字でなければなりません。

¢ ! \$ ~ { } ? @ # % ¥

- グローバル変数名には、空白や、以下の文字が含まれてはなりません。

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

- グローバル変数名に後書き空白があっても、それは認識されません。

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。1 時点では 1 つの QMF セッションしかアクティブにできません。QMF がすでに開始されているかどうかをアプリケーションにテストさせたい場合には、24 ページの『アプリケーションからの QMF の開始』を参照してください。

この節には、START コマンドの構文とキーワードに関する情報、およびキーワードの説明の表があります。

▶—START—(— | キーワード定義 | —————▶

キーワード定義:

|—number of keywords—, —keyword lengths—, —キーワード—, —————|

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

アプリケーションでの QMF コマンド

アセンブラー、C、COBOL、FORTRAN、および PL/I は、START コマンドに関する以下の仕様を使用します。

number of keywords (キーワードの数)

START コマンドで使用する開始コマンド・キーワードの数

keyword lengths (キーワードの長さ)

指定する各開始コマンド・キーワードの長さ

keywords (キーワード)

開始コマンド・キーワードの名前

『START コマンドのキーワード』を参照してください。

value lengths (値の長さ)

各開始コマンド・キーワードの値の長さが入っているリスト

values (値)

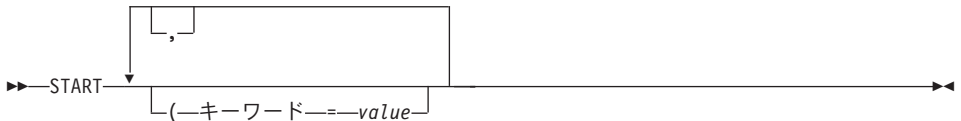
このコマンドに指定する開始コマンド・キーワードの値のリスト

value type (値のタイプ)

値が入る記憶域のデータ・タイプ。値のタイプは、START コマンドの場合は、文字でなければなりません。

REXX 呼び出し可能インターフェースの場合の START コマンドの構文

REXX 呼び出し可能インターフェースの場合、START コマンドの構文は次のとおりです。



START コマンドのキーワード

START コマンドには、以下のいずれかのキーワードを指定します。

DSQADPAN

DSQSIROW

DSQALANG

DSQSMODE

DSQSBSTG

DSQSPILL

DSQSCMD (CMS および TSO のみ)

DSQSPLAN (TSO のみ)

DSQSDBCS

DSQSPRID (TSO のみ)

DSQSDBNM

DSQSRSTG (CMS および TSO のみ)

DSQSDBQN (CICS のみ)

DSQSRUN

DSQSDBQT (CICS のみ)

DSQSSPQN (CICS のみ)

DSQSDBG

DSQSSUBS (TSO のみ)

DSQSDCSS (CMS のみ)

DSQSUSER (CICS/VSE のみ)

これらのキーワードの説明は、70 ページの表 7 にあります。

QMF では、以下の規則に従って START コマンド・キーワードを指定することができます。

- START コマンドで任意の開始コマンド・キーワードを指定することができます。QMF でサポートされるすべての環境 (CICS を除く) では、REXX プログラム中で、DSQSCMD パラメーターによって指定した DSQSCMD 以外の任意のキーワードを指定することができます。QMF 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。
- 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。
- アプリケーションまたは初期プロシージャで、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。これにより、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することができます。

これらのキーワードおよび環境依存性による影響の詳細については、ご使用のプラットフォームの「インストールと管理の手引き」を参照してください。

アプリケーションでの QMF コマンド

表 7. 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	DSQDC_DISPLAY_RPT グローバル変数を設定する。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ことを指定するには、値を 0 に設定する。	呼び出し可能インターフェース: 1 バッチ・モードの場合、または DSQQMFE を使用して QMF を対話式で開始した場合: 0																												
DSQALANG	<p>開始するセッションの主要言語を指定する。このパラメーターの値は、1 文字の言語 ID である。このキーワードで指定した主要言語で QMF コマンドを入力または指定する。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 (45 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』参照)。次の表は、この変数に有効な言語 ID の完全なリストである。</p> <table border="0"> <tr> <td>ID</td> <td>言語</td> </tr> <tr> <td>D</td> <td>ドイツ語</td> </tr> <tr> <td>E</td> <td>英語</td> </tr> <tr> <td>F</td> <td>フランス語</td> </tr> <tr> <td>H</td> <td>ハングル語 (韓国)</td> </tr> <tr> <td>I</td> <td>イタリア語</td> </tr> <tr> <td>K</td> <td>漢字 (日本)</td> </tr> <tr> <td>P</td> <td>ブラジル・ポルトガル語</td> </tr> <tr> <td>Q</td> <td>デンマーク語 (QMF VSE では使用不可)</td> </tr> <tr> <td>S</td> <td>スペイン語</td> </tr> <tr> <td>U</td> <td>大文字英語</td> </tr> <tr> <td>V</td> <td>スウェーデン語 (QMF VSE では使用不可)</td> </tr> <tr> <td>Y</td> <td>スイス・フランス語</td> </tr> <tr> <td>Z</td> <td>スイス・ドイツ語</td> </tr> </table>	ID	言語	D	ドイツ語	E	英語	F	フランス語	H	ハングル語 (韓国)	I	イタリア語	K	漢字 (日本)	P	ブラジル・ポルトガル語	Q	デンマーク語 (QMF VSE では使用不可)	S	スペイン語	U	大文字英語	V	スウェーデン語 (QMF VSE では使用不可)	Y	スイス・フランス語	Z	スイス・ドイツ語	E (英語)
ID	言語																													
D	ドイツ語																													
E	英語																													
F	フランス語																													
H	ハングル語 (韓国)																													
I	イタリア語																													
K	漢字 (日本)																													
P	ブラジル・ポルトガル語																													
Q	デンマーク語 (QMF VSE では使用不可)																													
S	スペイン語																													
U	大文字英語																													
V	スウェーデン語 (QMF VSE では使用不可)																													
Y	スイス・フランス語																													
Z	スイス・ドイツ語																													

表 7. 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>報告書作成に使用する記憶域のバイト数を QMF に伝える。これにより、CICS 内の同じアドレス・スペースに複数のユーザーが存在する場合に、記憶域の量を制限することができ。TSO および CMS でも、この変数を同様の目的で使用できる。</p> <p>TSO および CMS では、このキーワードと DSQSRSTG キーワードの両方を指定した場合、このキーワードが優先する。このキーワードを指定しない場合、DSQSRSTG キーワードが使用される。</p> <p>CICS ユーザーへの注: CICS の場合、DSQSBSTG が常に使用され、DSQSRSTG は使用されない。</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サイズを自動的に割り振る。この最小サイズは、環境によって異なる。大きな報告書には最小サイズより多くの記憶域が必要な場合がある。</p>	<p>CICS の場合: 500 000 バイト</p> <p>CMS または TSO の場合: 0 バイト</p>
DSQSCMD (CMS および TSO の み)	<p>QMF プログラム・パラメーターを設定する REXX プログラムを指定する。</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、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す。この REXX プログラムは、START コマンドに指定されていないキーワードのデフォルトとして、QMF が使用する QMF プログラム・パラメーターの値を提供する。</p> <p>START (DSQSCMD=<i>yourprogram</i>)</p> <p>CICS ユーザーへの注: QMF CICS は、REXX をサポートしないので、CICS のもとでは、DSQSCMD はサポートされない。CICS のもとで呼び出し可能インターフェースを使用して QMF を開始した場合に、QMF プログラム・パラメーターを設定したければ、START コマンドでキーワードを設定する必要がある。</p>	DSQSCMDE

アプリケーションでの QMF コマンド

表 7. 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
DSQSDBCS	<p>端末が 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 端末から 2 バイト文字セット (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、値を YES に設定する必要がある。他の場合、値を NO に設定する必要がある。</p>	NO
DSQSDBNM	<p>QMF セッションの開始時に、接続するロケーションを指定する。ヌル値は、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> <p>インストール先で通信データベースをセットアップしていない場合、ヌル以外の値またはデフォルト・データベースの名前の指定を試みると、エラーが発生する。</p>	NULL
DSQSDBQN (CICS のみ)	<p>CICS 記憶域を QMF トレース・データ用に使用することを指定する。名前は、DSQSDBQT によって選択される 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。CICS 名前指定の詳細については、CICS アプリケーション・プログラミングの手引きを参照。</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 一時データを使用する。</p> <p>TS CICS 補助一時記憶域を使用する。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
DSQSDEBUG	<p>QMF 初期化時にプロダクト・トレースを行う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 最小限の QMF トレースを指定する。</p>	NONE

表 7. 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
DSQSDCSS (CMS のみ)	QMF 実行モジュールが入っている DCSS (非連続共用セグメント) の名前を指定する。	QMF720n、ここで n は各国語 ID である。 英語の場合、デフォルトは QMF720E である。
DSQSIROW	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。	100
DSQSMODE	作業するモードを QMF に指示する。 I 対話モードを指定する。 B バッチ・モードを指定する。 DSQSMODE の値が B の場合、QMF がバックグラウンド・ジョブで実行できるように、パネル表示は禁止される。	B (バッチ)
DSQSPILL	QMF が予備ファイルまたはデータ・セットを使用するかどうかを指定する。可能な値は YES または NO である。	CICS の場合: NO CMS および TSO の場合: YES
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。	QMF810
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
DSQSRSTG (CMS および TSO のみ)	ユーザーのアプリケーションおよびユーザーのアプリケーションが呼び出す他のアプリケーションのために予約する、仮想記憶域のバイト数を指定する。QMF 環境からアプリケーションを実行する予定の場合、このパラメーターを使用する。ユーザーのアプリケーションのために記憶域を予約しない場合、QMF は使用可能なすべての仮想記憶域を使用して、大きな報告書を作成する可能性がある。	ゼロ (0)

アプリケーションでの QMF コマンド

表 7. 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
DSQSRUN	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。 このプロシージャに、グローバル変数およびプロファイル変数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のコマンドを組み込むことができる。このプロシージャは、アプリケーションが QMF を対話式に実行する場合にユーザーを QMF 内に置いたり、バッチ操作用の QMF セッションを準備したりすることができる。	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。	DSQSid、ここで、id は CICS 端末 ID。
DSQSSUBS(TSO のみ)	QMF がインストールされている DB2 サブシステムの ID を指定する。	DSN
DSQSUSER (CICS/VSE のみ)	CONNECT コマンドに DB2 Server (VSE 版) 許可 ID とパスワードを指定する。DSQSUSER キーワードを指定するには、次のように入力する。 DSQSUSER=SQL ID/password ここで SQL ID は、QMF を開始したユーザーの DB2 Server (VSE 版) 許可 ID。	システム・カタログに定義されている 3 バイトの VSE オペレーター ID およびパスワード。DSQSMODE=B を指定して QMF を開始する場合は、指定する必要がある。

コマンド同義語の使用

QMF では、QMF コマンドに似たコマンドであるコマンド同義語を作成できます。コマンド同義語を使用すれば、コマンドの柔軟性が増すので、エンド・ユーザーにとって非常に有用です。たとえば、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。QMF ユーザーがコマンド同義語にアクセスできるようにするには、コマンド同義語を 1 つまたは複数のコマンド同義語表に入力する必要があります。ユーザーがコマンド同義語を出すと、QMF はユーザーのアプリケーションを開始する TSO、RUN、CICS、または CMS コマンドを実行します。

コマンド同義語の作成

コマンド同義語を作成するためには、以下のことを行います。

1. 記述コマンドを作成する。

QMF コマンドは、動詞 - 目的語 フォーマットに従います。すべてのコマンドは動詞 (作用語) であり、多くのコマンドには、その後に目的語 (記述名詞) があります。たとえば、END は動詞のみのコマンドであり、CONVERT QUERY は動詞 - 目的語コマンドです。

既存の QMF コマンドと同じ動詞を使用してコマンド同義語を作成することができます。この場合、元の QMF コマンドも、その前にコマンド QMF を指定すれば、使用することができます。コマンド QMF については、「DB2 QMF 解説書」を参照してください。

たとえば、週の売上高が入力されたかどうかを調べるために、プロシージャーが報告書を実行するとします。今週のデータがない場合、プロシージャーは、表エディターを呼び出して、最新の情報を表に追加します。プロシージャーの名前に関係なく、コマンド同義語はユーザーにわかりやすくする必要があります。UPDATE SALES のような動詞 - 目的語のペアを選択することができます。

作成するコマンド同義語でパラメーターまたはオプションが必要である場合、置換変数 &ALL を使用できます。

2. 該当するコマンド同義語表を、新しいコマンド同義語を使用して更新する。

使用するコマンド同義語表の名前を知っていなければなりません。変数 DSQAP_SYNONYM_TBL に、各ユーザーのコマンド同義語表の名前が入っています。

データベース管理者は、コマンド同義語表へアクセスすることができます。個人用のコマンド同義語を作成したい場合、ユーザーがコマンド同義語を追加できるように定義した視点を作成することができます。

たとえば、コマンド同義語表には、コマンド同義語 UPDATE SALES に関する次のような情報が含まれています。

```

ADD                Q.COMMAND_SYNONYMS

VERB. . . . . ( UPDATE                )
OBJECT. . . . . ( SALES                )
SYNONYM_DEFINITION. . ( RUN PROC WEEKLY_SALES
REMARKS . . . . . ( procedure that checks to see if the weekly sales fi>
1 to 4 of 4

```

追加ファンクション・キーを押すと、QMF は、このコマンド同義語をユーザーの表に追加します。ただし、このコマンドを使用する前に、QMF に再接続する必要があります。

3. 必要に応じて、プロファイルを更新する。

このコマンド同義語を新しい表または視点に追加する場合、新しい表または視点の名前をユーザーのプロファイルに追加してください。

4. QMF セッションを終了する。

新しい QMF セッションを開始するまで、QMF は、ユーザーによるコマンド同義語表およびプロファイル表への変更を認識しません。

RUN QUERY の報告書ミニ・セッション

QMF 報告書を生成するアプリケーションを作成する場合、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することができます。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・セッションに有効なコマンドと無効なコマンドは、77 ページの表 8 と 77 ページの表 9 にリストされています。

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することができません。

有効な QMF グローバル変数 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。なぜなら、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からです (1 に設定されている場合は表示し、0 に設定されている場合は表示を抑止します)。

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。

- グローバル変数 DSQDC_DISPLAY_RPT のデフォルト値は 1 です。(DSQQMFE を使用して QMF を開始した場合、それが対話式であるかバッチ・モードであるかに関係なく、このグローバル変数のデフォルト値は 0 です)。
- 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。このミニ・セッション内で、QMF は照会結果の報告書を表示します。
- 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 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 に報告書ミニ・セッションを開始させたくない場合、以下のいずれかを実行してください。

- 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。

グローバル変数の詳細については、64 ページの『SET GLOBAL』を参照してください。

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 (制約事項は括弧内に示してあります)。

表 8. ミニ・セッションで有効なコマンド

• BACKWARD	• FORWARD	• RETRIEVE
• BOTTOM	• GET GLOBAL	• RIGHT
• CANCEL (ポップアップ・ウィンドウがアクティブの場合)	• HELP	• SAVE (データ)
• CICS	• INTERACT	• SET (プロファイル、グローバル変数)
• CMS	• ISPF	• SHOW (報告書、図表)
• DISPLAY (報告書、図表)	• LEFT	• SWITCH (ヘルプがアクティブの場合)
• END	• MESSAGE	• TOP
• ENTER	• PRINT (報告書、図表)	• TSO
	• QMF	

表 9 は、ミニ・セッションで無効なコマンドのリストです。

表 9. ミニ・セッションで無効なコマンド

• ADD	• ERASE	• REDUCE
• CANCEL	• EXIT	• REFRESH
• CHANGE	• EXPORT	• RESET GLOBAL
• CHECK	• EXTRACT	• RESET (照会、プロシージャ、書式)
• CLEAR	• GETQMF	• RUN
• CONNECT	• IMPORT	• SAVE
• CONVERT	• INSERT	• SEARCH
• DELETE	• INTERACT	• SHOW
• DESCRIBE	• IRM	• SORT
• DISPLAY (照会、プロシージャ、プロファイル、書式)	• LIST	• SPECIFY
• DRAW	• NEXT	• START
• EDIT	• PREVIOUS	• SWITCH
• ENLARGE	• PRINT (照会、プロシージャ、プロファイル、書式)	

アプリケーションでの QMF コマンド

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

第 9 章 QMF オブジェクトのインポートとエクスポート

QMF 環境の外で QMF オブジェクトを使用するアプリケーションを作成することができます。QMF EXPORT コマンドおよび IMPORT コマンドを使用して、QMF 環境の外に QMF オブジェクトを配置してください。

以下のオブジェクトをエクスポートすることができます。

図表 データ
書式 プロシージャ
照会 報告書
表

オブジェクトをエクスポートすると、オブジェクトは QMF によって外部フォーマットに変換され、ファイル、データ・セット、または CICS データ・キューに置かれます。QMF オブジェクトの外部フォーマットは、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、外部フォーマットをファイル、データ・セット、または CICS データ・キューから読み取り、(コマンドの発行方法に従って) 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。

データ・オブジェクトおよび表オブジェクトは、QMF フォーマットまたは IXF フォーマットでエクスポートすることができます。書式オブジェクト、指示照会オブジェクト、および報告書オブジェクトのフォーマットは、エンコード・フォーマットと呼ばれる複雑なフォーマットです。図表は、図形データ・フォーマット (GDF)、GDDM フォーマットでエクスポートされます。

本章では、すべての QMF エクスポート・フォーマットについて説明し、アプリケーションでの使用方法を示します。231 ページの『付録 B. エクスポート / インポート・フォーマット』で、データの QMF フォーマットについて説明し、エンコード・フォーマット・オブジェクトに関する表番号とフィールド番号を定義しています。IXF フォーマットの詳細については、245 ページの『付録 C. 統合交換フォーマット (IXF)』を参照してください。

CICS ユーザー: IMPORT コマンドまたは EXPORT コマンドを使用するアプリケーションを作成する場合には、123 ページの『CICS キューを使用する場合の規則と考慮事項』を参照してください。

IMPORT コマンドおよび EXPORT コマンドの構文については、「DB2 QMF 解説書」を参照してください。QMF オブジェクトのインポートおよびエクスポートについては、オンライン・ヘルプを参照してください。

エクスポートしたファイル、データ・セット、または CICS データ・キューを使用して何ができるか

インポート/エクスポート機能によって以下の操作を実行できます。

- 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フォーマットを使用して、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。オブジェクトが入っているファイル、データ・セット、または CICS データ・キューを QMF にインポートすると、新しい QMF オブジェクトが作成されます。

報告書と図表は、QMF にインポートできません。

- データベースに非 QMF オブジェクトを保管する。

オブジェクトをプロシージャ・オブジェクトまたは照会オブジェクトとしてインポートする場合、QMF はオブジェクトをそのまま QMF 環境に持ち込み、追加のレコードまたはフィールドをインポート・ファイルに挿入しません。レコード長が 79 バイト以下の任意のプログラムまたはファイルをインポートすることができます。

- 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。

注意:

システムまたは環境間で転送された、エクスポート・オブジェクトには、そのエクスポート・オブジェクトを変更または破棄してしまう可能性のある変換が行われるリスクがあります。エクスポート・オブジェクトを、たとえば、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稼働している環境間や、異なる NLF 環境間で転送しないでください。

CONVERT QUERY コマンドを使用すれば、指示照会または QBE 照会を SQL 照会に変換し、それをエクスポートして他のプロダクトで使用することができます。

CONVERT コマンドの詳細については、「DB2 QMF 解説書」を参照してください。

QMF オブジェクトを以下のように転送することができます。

- VM 内の CMS セッション間
- TSO またはネイティブ z/OS バッチ環境の QMF と、CICS 区画外一時データ・キューを使用する CICS 環境の QMF との間
- SENDFILE によるネットワーク上
- 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。
たとえば、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 と SAVE DATA 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。これによって結果の使用方法が異なります。

- データ・オブジェクトをエクスポートすると、ファイル、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。保管済みのデータを使用して行うアクションは、データベースを介して行う必要があります。

データ・オブジェクトと表オブジェクト

照会を実行すると、QMF は結果を報告書に表示します。報告書は、データ・オブジェクトまたは報告書オブジェクトとしてエクスポートすることができます。報告書オブジェクトをエクスポートする場合、このオブジェクトは、書式オブジェクトに指定されているデータのフォーマットを保持します。報告書オブジェクトを HTML 報告書としてエクスポートすると、適切な HTML コーディングでパッケージされます。この報告書をインターネット上で表示するために Web サーバーに置くことができます。QMF のデータ・オブジェクトと表オブジェクトは、生データの形でエクスポートされます。報告書オブジェクトの詳細については、113 ページの『報告書オブジェクト』を参照してください。

表の表示用の生データは、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りレシヨナル表および視点は、表オブジェクトと呼ばれます。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・フォーマットは同じです。データとしてエクスポートしたオブジェクトを表としてインポートできます。また、逆も可能です。

データ・オブジェクトと表オブジェクトは、QMF フォーマットまたは統合交換フォーマット (IXF) でエクスポートすることができます。

EXPORT コマンドに、DATAFORMAT=QMF または DATAFORMAT=IXF を指定します。QMF フォーマットはデフォルトです。詳しくは、231 ページの『データの QMF フォーマット』を参照してください。

QMF オブジェクトのインポートとエクスポート

IXF には、2 進フォーマットと文字フォーマットという 2 つのフォーマットがあります。この説明は、85 ページの『2 進数と文字』にあります。IXF フォーマットの説明は、245 ページの『付録 C. 統合交換フォーマット (IXF)』にあります。

&ファイルにユーザー固有の表を作成するには、QMF フォーマットまたは IXF フォーマットを指定して必要なデータが入っているファイル、データ・セット、または CICS データ・キュー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。次に、このファイル、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 にインポートしてください。ファイル、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データベースにインポートするコマンドの例を次に示します。

```
IMPORT TABLE MYTABLE FROM MYDATA
```

EXPORT コマンドおよび IMPORT コマンドの詳細については、「DB2 QMF 解説書」を参照してください。CICS/VSE ユーザーは、123 ページの『CICS キューを使用する場合の規則と考慮事項』を参照してください。

QMF フォーマットのデータ・オブジェクトの解釈: 例

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。この例では、各データ・レコードの長さは 23 バイトです。231 ページの『データの QMF フォーマット』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説明しています。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。

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

Q.STAFF から次のデータをエクスポートする場合は、

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

各列の幅を計算するために、次のような表を使用します。

表 10. 列の幅の計算

列名	データ・タイプ	データ・タイプの幅 (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

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ブランクが埋め込まれます。

図 16 は、報告書からのヘッダーおよびその 16 進数表示を示しています。反転表示の番号は、図の下の注を示しています。

```

      R E L      1 . 0                I D
1   D9 C5 D3 40 F1 4B F0 40 0004 0003 C9 C4 40 40 40 40 40 40 40 40
   1                                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 40 01C0 0009 E8 00 C3 D6 D4 D4 40 40 40 40
                                  Y
4   40 40 40 40 40 40 40 40 40 01E4 07 02 E8 00 40 40 40 40 40 40

```

図 16. QMF フォーマット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・ブランク文字を表します。

図 17 は、報告書からのデータおよびその 16 進数表示を示しています。各バイト位置の意味の詳細については、231 ページの『データの QMF フォーマット』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
   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 17. QMF フォーマット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 1.0

オブジェクト・フォーマット・レベル: 1.0

オブジェクト・フォーマット・レベルは、このオブジェクトが使用するオブジェクト・フォーマットのバージョンを QMF に指示します。QMF オブジェクト・フォーマットを変更するたびに、レベル番号が変更されます。オブジェクト・フォーマットは、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'

ヌルの使用: 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: 列にはヌル値が入る

データ・オブジェクトまたは表オブジェクトをエクスポートしたときに生成されるファイル、データ・セット、または CICS データ・キューの詳細については、122 ページの『外部化 QMF オブジェクトの仕様』を参照してください。

IXF フォーマットの例については、245 ページの『付録 C. 統合交換フォーマット (IXF)』を参照してください。

データ・オブジェクトと表オブジェクトのエクスポート / インポートに関する規則と情報

データ・オブジェクトまたは表オブジェクトのインポートおよびエクスポートに関する一般的な考慮事項を以下に示します。

ファイル、データ・セット、または CICS データ・キューは割り振られたままになる

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。ファイル、データ・セット、または CICS データ・キューはオープンの状態であり、QMF に割り振られています。ユーザーがファイル、データ・セット、または CICS データ・キュー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。この時点で、ファイル、データ・セット、または CICS データ・キューはクローズされ、QMF に割り振られているとは見なされなくなります。これは、アプリケーションが、IMPORT DATA コマンドによって QMF に割り振られたファイル、データ・セット、または CICS データ・キューを、削除または変更してはならないことを意味します。アプリケーションは、読み取っていた

データ・セットを変更または削除する前に、別のデータ・ソースを使用して開始するか、QMF 一時データ記憶域を空にする (RESET DATA) 必要があります。

IMPORT コマンドの実行中、QMF は、ファイル、データ・セット、または CICS データ・キュー を読み取る時にロックしません。すなわち、ファイル、データ・セット、または CICS データ・キュー を読み取る時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 CICS データ・キュー がなんらかの方法で消去または変更されると、結果は予測不可能になり、システム・エラーの原因になります。

QMF がオブジェクトを完成する必要がある場合で、そのデータ・オブジェクト用に十分な記憶域がない場合、不完全データ・プロンプトが表示されます。QMF がデータ・オブジェクトを完成する必要があるのは、たとえば、同じファイル、データ・セット、または CICS データ・キュー へのオブジェクトのエクスポートが要求されたときです。この状態は、現在 EXPORT コマンドで指定した同じファイル、データ・セット、または CICS データ・キュー から、以前に IMPORT DATA コマンドを実行していることを意味します。

2 進数と文字

QMF フォーマット、または IXF フォーマットの 2 進数フォーマット

(OUTPUTMODE=BINARY) を使用して、データ・オブジェクトまたは表オブジェクトをエクスポートすると、データは生の 2 進数フォーマットになります。しかし、IXF の文字フォーマット (OUTPUTMODE=CHARACTER) を使用すると、エクスポート後のデータは EBCDIC 形式になります。書式、報告書、プロシージャ、および SQL 照会オブジェクトのエクスポート後のデータも EBCDIC 形式になります。

PL/I、COBOL、および アセンブラーなどの言語で作成したアプリケーション・プログラムは、通常、2 進数データを文字データより速く、効率的に読み取り、処理することができます。1 つの IBM プロダクトから別のプロダクトへデータを交換する場合には、2 進数の方がより効率的に実行されます。しかし、アプリケーション・プログラムを REXX で作成する場合、またはデータをエディターで処理する場合は、EBCDIC (文字) データの方がより効率的です。

エラー

QMF は、ファイル、データ・セット、または CICS データ・キュー からデータをインポートした後、報告書パネルおよび確認メッセージを表示します。ファイル、データ・セット、または CICS データ・キュー にフォーマット・エラーが含まれていると、QMF は報告書パネルを表示しないで、代わりに、IMPORT コマンドを処理する前に現行パネルであったオブジェクト・パネルに、エラー・メッセージを表示します。ただし、現行のオブジェクト・パネルが報告書パネルであった場合、QMF は、インポート後のデータにエラーを検出すると、ホーム・パネルとエラー・メッセージを表示します。

書式オブジェクトと異なり、データ・オブジェクトまたは表オブジェクトをインポートする場合の入力ファイル、データ・セット、または CICS データ・キュー のフォーマット

トは、同じオブジェクトを EXPORT DATA または EXPORT TABLE コマンドを使用してエクスポートしたときに生成されるはずの出力ファイル、データ・セット、または CICS データ・キューのフォーマットと正確に一致していなければなりません。

ロング・ネーム用の書式更新

18 文字を超える列名がある場合は、DATA、TABLE、または IXF フォーマットの列名フィールド・サイズを 30 文字まで増やす必要があります。

DATA または TABLE フォーマット

テーブル名および許可 ID は、DATA または TABLE フォーマットに含まれません。

18 文字を超える列名がある場合、列名フィールド・サイズを 18 から 30 文字まで増やす必要があります。リリース番号は、QMF DATA オブジェクトでのより長い列名の使用を示すために、REL 3.0 に変更されました。この新しいリリース番号を使用すると、オブジェクトをバージョン 8.1 以前の QMF へインポートする場合、エラーが生じます。

IXF フォーマット

IXF (T) テーブル・レコードには、QMF EXPORT TABLE コマンドを使用してエクスポートされているテーブルの所有者および名前が含まれます。しかし、RUN QUERY コマンドの結果である QMF データ・オブジェクトをエクスポートするとき、(T) テーブル・レコードにはブランクの所有者と名前が含まれます。ロング・ネーム・サポートの場合、テーブル名は 18 文字で、所有者名は 8 文字で切り捨てられます。

18 文字を超える列名がある場合は、(C) 列レコード内の列名フィールドを 18 から 30 文字まで増やす必要があります。長い列名が存在する場合、(H) ヘッダー・レコード内の IXF バージョン番号は 0001 に設定されます。18 文字を超える列名がない場合は、以前のリリースの QMF と互換性のある古い IXF フォーマットが使用されます。これらの場合、(H) ヘッダー・レコード内の IXF バージョン番号は 0000 になります。

プロシージャおよび SQL 照会

これらのオブジェクトを示すファイル、データ・セット、または CICS データ・キューのフォーマットは、すべてのオブジェクトのなかで最も単純です。ファイル、データ・セット、または CICS 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。

次に SQL 照会を示します。

```
SQL query                                MODIFIED LINE    1
SELECT *
FROM Q.STAFF
```


次に、対応する外部フォーマットの照会を示します。

```
*** Top of File ***  
SELECT *  
FROM Q.STAFF
```

```
*** End of File ***
```

レコード・フォーマットが単純なので、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非常に単純です。SQL 照会またはプロシーチャーは、79 バイトの照会またはプロシーチャーのレコードが入っている固定長のファイル、データ・セット、または CICS データ・キューから構成されます。結果のファイル、データ・セット、または CICS データ・キューをインポートすれば、照会またはプロシーチャーは QMF 一時記憶域に置かれるので、実行が可能になります。

図表オブジェクト

図表オブジェクトは、エクスポートして、QMF 環境の外で処理することができます。図表は、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り出したりすることができません。図表は、QMF にインポートできません。

QMF が図表オブジェクトをエクスポートする場合、データを報告書フォーマットから図形データ・フォーマット (GDF) に変換します。GDDM フォーマットの 1 つである GDF は、データ交換用の既存の標準です。たとえば、エクスポート後の図表データを GDDM ユーティリティを使用して印刷したり、文書スクリプト・ファイルに組み込んだりすることができます。GDF フォーマットの詳細については、*GDDM アプリケーション・プログラミングの手引き* を参照してください。

エクスポート後の図表オブジェクトを、GDF フォーマットに設定されたファイルまたはデータ・セットと同様に使用することができます。以下に例を示します。

- アプリケーションは、文書構成プログラム (DCF) を使用して QMF 報告書 (印刷済みの報告書またはエクスポート後の報告書) と QMF 図表 (エクスポート後の図表) を結合し、フォーマット設定済みの情報をプリンターに送信することができます。
- アプリケーションは、GGXA などのグラフィックス・エディターを使用して、エクスポート後の QMF 図表を変更したり微調整したりすることができます。

エンコード・オブジェクト

書式オブジェクトおよび指示照会オブジェクトは、エンコード・フォーマットでエクスポートとインポートが行われます。エンコード・フォーマットは、オブジェクトを表構造に変換するフォーマットです。エンコード・フォーマットによって、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。報告書オブジェクトもエンコード・フォーマットでエクスポートされますが、報告書をインポートすることはできません。

書式、報告書、または指示照会 (リレーショナルまたはエンティティ・リレーションシップ) のエンコード・フォーマットは、以下のレコードから構成されます。

- 固定書式レコード: ヘッダー・レコード (H) (88 ページを参照)
- 可変長フォーマット・レコード
 - データ値レコード (V) (92 ページを参照)
 - データ表記述レコード (T) (94 ページを参照)
 - 表行レコード (R) (98 ページを参照)
 - オブジェクト・レコードの終わり (E) (99 ページを参照)

アスタリスク (*) によって示されているアプリケーション・データ・レコードは、エクスポート後のファイル内のオブジェクトに関連する情報とコメントを保管するために、アプリケーション・プログラムで使用することができます。詳細については、100 ページの『アプリケーション・データ・レコード (*)』を参照してください。

上記のレコードの他に、エクスポート後の報告書に以下のレコードを入れることができます。

- 報告書行レコード (L) (116 ページを参照)
- データ継続レコード (C) (118 ページを参照)

エクスポートされたファイル、データ・セット、または CICS データ・キューの仕様については、122 ページの『外部化 QMF オブジェクトの仕様』を参照してください。

固定フォーマット・レコード

大部分のレコードは、可変長フォーマットです。しかし、ヘッダー・レコードは、レコードが入っているファイルまたはデータ・セットが可変長フォーマットの場合でも固定フォーマットです。

ヘッダー・レコード (H)

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・ファイルの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・レコードには、表 11 に示す情報が入っています (アスタリスクは、そのフィールドがインポートに必要であることを示しています)。

表 11. 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)
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、または指示照会がエクスポートされた QMF のリリース・レベル: 11 (QMF バージョン 8 を示す)
09	ブランク
10*	オブジェクトのタイプ: F (書式)、 R (報告書)、 T (リレーショナル指示照会)、 E (ER 指示照会)
11	ブランク
12-13*	QMF オブジェクト・レベル: 01 (報告書)、 04 (書式)、 01 (リレーショナル指示照会または ER 指示照会)
14	ブランク
15*	オブジェクトのフォーマット: E (書式、報告書、およびリレーショナル指示照会または ER 指示照会の各オブジェクトのエクスポートに使用されたフォーマット)
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: W (全オブジェクト)
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国語: E (英語)
22	ブランク
23*	インポート時に、一時記憶域内のオブジェクトに行うアクション: R (オブジェクトの置換)
24	ブランク
25-26	後続の各レコードの先頭にある制御域の長さ: 01 (書式)、 02 (報告書)、 01 (リレーショナル指示照会または ER 指示照会)
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	日付スタンプ: yy/mm/dd
39	ブランク
40-44	タイム・スタンプ: hh:mm
45	ブランク
51	ブランク

H レコード内のオブジェクト・レベルが、オブジェクトのフォーマットの変更を示します。すべてのオブジェクト・フォーマットは、オブジェクト・レベル 01 から開始されます。QMF の新しいリリースでオブジェクト・フォーマットが変更されると、オブジ

QMF オブジェクトのインポートとエクスポート

エクト・レベルは 1 ずつ増加します。オブジェクト・レベルは、フォーマットの変更によって、アプリケーションにエラーが生じる可能性がある場合にのみ増加します。報告書、書式、および指示照会の各オブジェクト・タイプのレベルの変更を検査してください。このエンコード・フォーマットについては、83 ページの図 17 を参照してください。

インポートするオブジェクトによってアプリケーションに問題が生じないことを確認するために、オブジェクト・レベル値を検査してください。

指示照会用の H レコードの例

H QMF 11 T 01 E V W E R 01 03 98/11/20 17:12

例からの値	説明
H QMF 11 T	バージョン 8 QMF リレーショナル指示照会のヘッダー・レコード
01	指示照会の構造のオブジェクト・レベルは 1
E	フォーマット・タイプは、書式、報告書、および指示照会用
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	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
98/11/20	日付スタンプ
17:12	タイム・スタンプ

指示照会エンコード・フォーマットの完全な例については、104 ページの図 19 を参照してください。

書式用の H レコードの例

H QMF 11 F 04 E V W E R 01 03 98/12/16 22:08

例からの値	説明
H QMF 11 F	バージョン 8 QMF フォーム・ヘッダー・レコード
04	書式の構造のオブジェクト・レベルは 4
E	フォーマット・タイプは、書式、報告書、および指示照会用
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

例からの値	説明
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
98/12/16	日付スタンプ
22:08	タイム・スタンプ

書式エンコード・フォーマットの完全な例については、107 ページの図 20 を参照してください。

報告書用の H レコードの例

H QMF 11 R 01 E V W E R 02 03 98/10/14 16:20

例からの値	説明
H QMF 11 R	バージョン 8 QMF レポート・ヘッダー・レコード
01	報告書の構造のオブジェクト・レベルは 1
E	フォーマット・タイプは、書式、報告書、および指示照会用
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	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	無視される
02	制御域の長さは 2 バイト
03	整数長さフィールドの長さは 3 バイト
98/10/14	日付スタンプ
16:20	タイム・スタンプ

報告書エンコード・フォーマットの完全な例については、115 ページの図 22 を参照してください。

可変長フォーマット・レコード

固定フォーマット・レコードである H レコードを除いて、すべてのレコードは可変長フォーマット・レコードです。

標識 レコード・タイプ

V データ値 (92 ページの『データ値レコード (V)』を参照)

T データ表記述 (94 ページの『データ表記述レコード (T)』を参照)

R 表行 (98 ページの『表行レコード (R)』を参照)

QMF オブジェクトのインポートとエクスポート

- E** オブジェクトの終わり (99 ページの『オブジェクトの終わりレコード (E)』を参照)
- *** アプリケーション・データ (100 ページの『アプリケーション・データ・レコード (*)』を参照)
- L** 報告書行 (116 ページの『報告書行レコード (L)』を参照)
- C** データの継続 (118 ページの『データ継続レコード (C)』を参照)

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。これは、レコードを含んでいるファイル、データ・セット、または CICS データ・キューではなく、レコード自体のことです。可変長フォーマット・レコードの一般的なフォーマットは次のとおりです。

制御域	レコード・データ域
-----	-----------

制御域は次のとおりです。

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。特定のタイプの変長フォーマット・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。この区域の各フィールドは、区切り文字 (本書の場合はブランク文字) によって分離されません。

データ値レコード (V)

値レコードは、オブジェクトの単一フィールドの値 (たとえば、書式でのヘッダーの前のブランク行) 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

231 ページの『付録 B. エクスポート / インポート・フォーマット』は、指示照会、書式、および報告書オブジェクトに含まれているフィールドに割り当てられているフィールド番号をリストしています。

V レコードの内容は次のとおりです。

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

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

注:

1. 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。
2. 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ブランクのみ) は、フィールドにヌル値が入っていることを示します。
3. 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。
4. 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。

V レコードの例

書式 V 1511 * NONE

(フィールド番号の全リストについては、236 ページを参照してください。)

フィールド

折り返し報告書行の幅

値 'NONE'

報告書 V 1001 006 PERIOD

(フィールド番号の全リストについては、241 ページを参照してください。)

フィールド

プロファイル DECIMAL オプション

長さ 6

値 'PERIOD'

指示照会 V 1501 001 K

(フィールド番号の全リストについては、233 ページを参照してください。)

フィールド

重複行

長さ 1

値 'K' (保持)

データ表記レコード (T)

エンコード・フォーマットでは、オブジェクトの大部分のデータが表に現れます。これらはデータベースのリレーショナル表ではなく、エンコードされたフォーマット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 (たとえば、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。エンコード表および各列のフィールド番号については、231 ページの『付録 B. エクスポート / インポート・フォーマット』を参照してください。

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。

アプリケーション・プログラムがエクスポート後の書式、報告書、または指示照会の表の内容を参照する場合は、T レコード内のエンコードを使用して、R レコード内の値を正しく見つけるようにしてください。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 レコードの内容は次のとおりです。

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** 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)

T レコードのレコード・データ域

以下のリストに示したバイト位置は、制御域の終わりからのオフセットです。制御域の長さはヘッダー・レコードに示されています。

バイト位置

説明

- 01** ブランク
- 02-05** 表番号 (1001-9999)
- 06** ブランク
- 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)

QMF オブジェクトのインポートとエクスポート

バイト 11-13 (列数) は、後続のフィールド番号 / データ値の長さのペアの個数を示します。これは、バイト 15-22 の情報が列ごとに反復されることを意味します。

T レコードの例 (書式)

T 1110 * 002 1112 007 1113 018

(フィールド番号の全リストについては、236 ページを参照してください。)

フィールド

列ヘッダー表

行 すべて

列 2

列フィールド

列データ・タイプ

長さ 7

列フィールド

列ヘッダー

長さ 18

T レコードの例 (指示照会)

T 1110 008 002 1112 001 1113 027

(フィールド番号の全リストについては、233 ページを参照してください。)

フィールド

表定義の表

行 8

列 2

列フィールド

表 ID

長さ 1

列フィールド

表名

長さ 27

T レコードの例 (報告書)

T 1010 005 003 1012 008 1013 003 1014 006

(フィールド番号の全リストについては、241 ページを参照してください。)

フィールド

フォーマット設定済み報告書表

行 5

列 3

列フィールド

BREAKn

長さ 8

列フィールド

データ用の編集コード

長さ 3

列フィールド

データが入っているフィールドの開始位置

長さ 6

規則および注:

1. 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。
2. 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。
3. 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。
4. 長さがゼロの列 (またはこの表に含まれていない列) 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。指示照会の場合は、常にそうとは限りません。可能な場合はデフォルトが提供されますが、他の場合はエラーになります。
5. 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)

R レコードは、エンコード表内の単一行に関する値のセットを提供します。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 レコードの内容は次のとおりです。

R レコードの制御域

バイト位置

説明

- 01 行レコード ID (R)
- 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)

R レコードのレコード・データ域

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。フォーマットは次のとおりです。

`_val..._val..._val..`

ここで、`val...` はこの行と列のデータ値であり、`_` は区切り文字です。

R レコードの例

この例で、列値の長さは、常に、その列の T レコードに示されています。

書式 R 2 SALARY

(フィールド番号の全リストについては、236 ページを参照してください。)

列値 ' 2'

列値 'SALARY'

報告書 R GROUP L2

(フィールド番号の全リストについては、241 ページを参照してください。)

列値 'GROUP '

列値 'L2'

指示照会 R C A.DEPT

(フィールド番号の全リストについては、233 ページを参照してください。)

列値 'C'

列値 'A.DEPT'

規則および注:

1. 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。
2. データ値の数 (v..v) 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。
3. 関連する 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、すなわち、デフォルト値に設定されることを示します。ただし、T レコード内にフィールドが存在する場合、R レコードにこのフィールド用の余分なブランクが入っていないなければなりません (長さがゼロの値によって、R レコード内では 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)

E レコードは、エクスポートし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中のファイルに組み込まれていない場合、QMF 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します。

アプリケーション・データ・レコード (*)

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、外部ファイルの特定のオブジェクトに関連するコメントなどの固有のデータを組み込むことができます。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基本的に無視され、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、作成後のファイル、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。これらのレコードは、コメント・レコードとして役に立ちます。アプリケーション・データ・レコードの内容は次のとおりです。

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

アプリケーション・データ・レコードの例

*This is the form that groups by DEPT.

このようなコメントが、エクスポートされた書式内に組み込まれます。

エンコード・フォーマット・オブジェクトのエクスポート

エンコード・フォーマットを使用してオブジェクトをエクスポートすると、以下の処理が行われます。

- 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。

エンコード・フォーマット・オブジェクトのインポート

書式、報告書、または指示照会をインポートする場合、以下のこと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。122ページの『外部化 QMF オブジェクトの仕様』および 231ページの『付録 B. エクスポート / インポート・フォーマット』を参照してください。
- 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。
- 各データ・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・フィールドの場合、レコード終わりが区切り文字のように機能するので、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、IMPORT 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。たとえば、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。

QMF オブジェクトのインポートとエクスポート

- 表番号、フィールド番号、および数値の長さ、先行ゼロまたは先行ブランクを組み込むことができます。しかし、後書きブランク（ブランク区切り文字を除く）は許可されません。フィールドは右寄せしなければなりません。
- 長さまたはカウントの代わりに * を使用した場合、左寄せされ、後書きブランクで埋められます。
- 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、後書きブランクで埋められます。長い場合は、切り捨てられます。
- レコードが固定フォーマット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ブランクと想定されます。

指示照会オブジェクト

この節では、リレーショナル指示照会の外部フォーマットを示します。

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。エクスポート後の指示照会ファイル、データ・セット、または CICS データ・キューは、QMF 一時記憶域にインポートしたり、データベースに直接入れることができます。指示照会をインポートする場合、QMF はインポートする照会が、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します。たとえば、インポートする指示照会の表 XYZ に列 A、B、および C がある場合、QMF は、列 A、B、および C がある表 XYZ がデータベース内に存在するかどうかを検査します。

指示照会オブジェクトに関連する表番号とフィールド番号のリストについては、233 ページの『指示照会オブジェクトの表番号とフィールド番号』を参照してください。

指示照会オブジェクトのエクスポート

この節では、エクスポート後の指示照会の例を示します。103 ページの図 18 は、エクスポート前の指示照会の指示照会基本パネル確認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
```

図 18. エクスポート前のサンプル指示照会

104 ページの図 19 は、エクスポート後の指示照会のフォーマットを示しています。

QMF オブジェクトのインポートとエクスポート

```
H QMF 11 T 01 E V W E R 01 03 98/11/20 17:12
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
```

図 19. エクスポートしたファイル、データ・セット、または CICS データ・キュー

指示照会のインポート

アプリケーションが編集または生成した指示照会オブジェクトをインポートする場合は、以下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、インポートするレコードは、ヘッダー (H) レコードの後に決まった順に並んでいなければなりません。並び順は次のとおりでなければなりません。
 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)
- 5. 行条件レコード (フィールド番号 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、すなわち、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。

6. 残りのレコードの並び順は任意です。
- 表に関する表は、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・フォーマットは、使用する言語に関係なく同じです。フォーマット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。指示照会のエクスポート時に使用されるコードについては、233 ページの『指示照会オブジェクトの表番号とフィールド番号』を参照してください。

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。これらは、変換されない SQL 記号です。

書式オブジェクト

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、変更した書式パネルが QMF によってエンコード・フォーマットに変換されます。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・フォーマットになります。

- FORM.BREAK n 、ここで $n = 3 \sim 6$
- 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

外部フォーマット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。

デフォルト書式の作成: 例

LAYOUT コマンドによって、ユーザーは QMF 一時記憶域にデータがない状態でも (QMF 一時記憶域またはデータベース内の) 書式に基づいたサンプル報告書を表示することができます。LAYOUT は、サンプル・データを生成し、QMF にインポートし、それに書式を適用して報告書を作成します。

CICS ユーザーへの注

LAYOUT コマンドには ISPF が必要ですが、CICS では利用できません。

ユーザーは、書式を作成し、エクスポートし、初期化の一部として QMF にインポートすることによって、照会を実行せずに書式を表示することができます。初期プロシージャー中に書式をインポートすれば、ユーザーは SHOW FORM を入力することによって書式にアクセスすることができます。

インポートできる最小の書式はヘッダー・レコードと終了レコードです。しかし、FORM.COLUMNS を使用するためには、少なくとも 1 列の情報が必要です。

空の報告書を作成する照会を実行することによって、デフォルト書式を作成することができます。

SQL query

```
SELECT JOB  
FROM Q.STAFF  
WHERE NAME='empty_set'
```

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。DEFAULT と名付けられたファイル、データ・セット、または CICS データ・キューには、107 ページの図 20 に示されている情報が入ります。

```
H QMF 11 F 04 E V W E R 01 03 98/12/16 22:08

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
```

図 20. エクスポートされた書式のサンプル・フォーマット (1/2)

```
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
```

図 20. エクスポートされた書式のサンプル・フォーマット (2/2)

初期プロシージャ内でコマンド `IMPORT FORM FROM DEFAULT` (CICS では `(QUEUETYPE=xx` パラメーターを含めて) を出すことによって ユーザーがログオンするたびに、デフォルトのファイル、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。

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

アプリケーションで QMF 書式を使用する場合、以下の点に注意してください。

- **QMF の外側での書式ファイル、データ・セット、または CICS データ・キューの作成**

QMF の外で書式ファイルを作成する場合 (`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` レコードと `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ファイル、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ファイル、データ・セット、または CICS データ・キューと正確に同じである必要はありません。たとえば、書式をエクスポートするとき、QMF によって値 (V) レコード内のすべてのデータ値の前に長さが置かれますが、書式をインポートするとき、データ値をレコード終わりによって区切ることを指定するための (*) 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すで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 QMF によって保持されます。

- **ヘッダー・レコード内のオブジェクト・レベルの検査**

書式ファイル、データ・セット、または CICS 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、書式の生成時のフォーマット構造のレベルを示します (オブジェクト・レベルは、88 ページで説明したようにヘッダー・レコードのバイト 12 および 13 に示されます)。アプリケーションが書式ファイル、データ・セット、または CICS データ・キューの内容を適切に解釈しているかを確認するには、オブジェクト・レベルが、アプリケーションがベースとしているフォーマットを示しているかどうかを検査します。

- **アプリケーション・データ・レコードの使用**

100 ページの『アプリケーション・データ・レコード (*)』で説明したアプリケーション・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。このレコードによって、書式オブジェクトに関するファイル、データ・セット、または CICS データ・キュー内に、ユーザー固有のコメントを組み込むことができます。このレコードは、ファイル、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 * 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。

- **日付 / 時刻情報のインポートとエクスポート**

インストール先が日付 / 時刻データ・タイプをサポートしている場合に、日付 / 時刻情報がある書式をエクスポートした後、日付 / 時刻データ・タイプをサポートしない QMF のインストール先で、この書式をインポートすることはできません。インポートを試みると、IMPORT コマンド処理が停止し、QMF によってエラー・メッセージが出されます。

- **CICS で書式を使用する場合の制約**

QMF CICS のもとでは、REXX は使えないので、REXX に基づいている QMF 書式上の区域は、CICS 環境でその書式を実行する場合、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。したがって、REXX 計算、条件付き行フォーマット設定、および列定義は、QMF CICS ユーザーは使えません。

入出力用の書式ファイル、データ・セット、または CICS データ・キューを制御についてのその他の情報および規則については、101 ページの『エンコード・フォーマット・オブジェクトのインポート』を参照してください。

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

QMF オブジェクトのインポートとエクスポート

- 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)
 - オブジェクトのフォーマット (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 (NUMERIC、CHAR、GRAPHIC、UNKNOWN)。インストール先が日付 / 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、および TIMEST 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は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。

ヘッダー・レコードの直後に、COLUMNS 表の T レコード (フィールド番号 1110) が続いていなければなりません。このレコードには、表内の行数の数値カウントが入っていないければなりません (* 行カウントは許可されません)。

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、特定のバリエーション・パネルの他のすべての V、T、および R レコードの前に来なければなりません。

特定のバリエーション用の値が、エンコード・フォーマットに複数回現れた場合、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。

変換後の書式

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、H レコード内の各国語 ID が E であると、QMF は自動的に書式内の予約語を現行セッションの言語に変換します。

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。

表 12 には、列データ・タイプ・フィールドの値に関する情報が示してあります。

表 12. 列データ・タイプ・フィールドの値

データ・タイプ値 (QMF 書式)	データ・タイプ番号 (データベース書式)	文字ストリング (データベース書式)	意味
DATE	384	DATE	日付
TIME	388	TIME	時刻
TIMEST	392	TIMESTAMP	タイム・スタンプ
NUMERIC	496	INTEGER	整数
	500	SMALLINT	小整数
	484	DECIMAL	10 進数
	480	FLOAT	浮動小数点
CHAR	448	VARCHAR	可変文字
	452	CHAR	固定文字
	456	LONG VARCHAR	長可変 文字
	904	ROWID	行 ID
GRAPHIC	464	VARGRAPHIC	可変図形
	468	GRAPHIC	固定図形
	472	LONG VARGRAPHIC	長可変 図形

上記のデータ・タイプ値の他に、U、V、または無効な編集コードに対して QMF が使用する UNKNOWN データ・タイプがあります。

インポート中のエラーの検出

QMF がインポート中に書式ファイルのフォーマットでエラーを検出すると、インポート機能はエラーおよびファイル内のエラーの場所について示すメッセージを出して終了します。

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すで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。

QMF オブジェクトのインポートとエクスポート

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。次に例を示します。

- V レコード
 - ゼロの長さフィールド
 - 指定された長さフィールドが、実際に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 2 つのうちのいずれかの方法で、エラーと警告に応答することができます。

- 1 度に 1 つずつ問題を修正する
- SET PROFILE (TRACE=L2) を指定して IMPORT FORM を実行する

SET PROFILE (TRACE=L2) を指定して IMPORT FORM を実行すれば、IMPORT コマンドに関連するすべてのメッセージ番号のリストを入手できます。次のコマンド

```
HELP message_number
```

を実行すると、当該メッセージ番号のエラー・メッセージが表示されます。

書式オブジェクトのエクスポート

236 ページの『書式オブジェクトの表番号とフィールド番号』は、書式オブジェクトの様々な部分へのフィールド番号の割り当てをリストしています。また、エクスポート後のファイル内で、書式のどの部分が表で、どの部分がそれぞれの値であるかも示しています。列データ・タイプ (フィールド 1112) は、書式パネルに表示されませんが、外部フォーマット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、QMF 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、109 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 が提供するデータ・タイプ・キーワード (フィールド番号 1112) は、指定された編集コードに基づいています。U、V、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。113 ページの表 13 は、QMF が、指定された編集コードについて生成するデータ・タイプ・キーワードを示しています。この表では、x は表示される 10 進数の桁数を表し、0 から 99 の整数になります。

表 13. 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
B、BW、C、CW、CT、CDx、X、XW	CHAR
G、GW	GRAPHIC
E、D、I、J、K、L、P、EZ、DZ、IZ、JZ、KZ、LZ、PZ、DZC、Dx、Ix、Jx、Kx、Lx、Px	NUMERIC
TDXx	DATE
TTXx	TIME
TSI	TIMEST
U、V	UNKNOWN
上記以外 (無効)	UNKNOWN

バリエーション・パネル

書式をエクスポートする場合、QMF はデフォルト値から変更された値を持つバリエーション・パネルだけをエクスポートします。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。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。

変換後の書式

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。このため、H レコード内の各国語 ID が、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。

報告書オブジェクト

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。しかし、エクスポート後の報告書を使用して、以下のことを行えます。

- 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。

報告書のエクスポート例

この例は、レベル 1 の切れ目がある報告書を示しています。114 ページの図 21 は、エクスポートされる報告書を示しています。横方向の報告書の例については、119 ページを参照してください。

QMF オブジェクトのインポートとエクスポート

フィールド番号のリストについては、241 ページの『報告書オブジェクトの表番号とフィールド番号』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. QMF の表報告書

115 ページの図 22 は、図 21 に示した報告書のエクスポート後のフォーマットを示しています。

```

H QMF 11 R 01 E V W E R 02 03 98/10/14 11:24
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 J & H SUPPLY COMPANY
L 110 10100000 AVERAGE SALARIES (DEPTS 10, 15, 20)
L 110 10100000 REPORT 17
L 110 10000000
L 110 10000000
L 170 10000000 AVERAGE
L 170 10000000 DEPT JOB SALARY
L 170 10010000 -----
L 181 11000000 10 MGR 20865.86
L 151 10010000 -----
L 151 11100000 * 20865.86
L 151 10000000
L 181 11000000 15 CLERK 12383.35
L 181 11000000 MGR 20659.80
L 181 11000000 SALES 16502.83
L 151 10010000 -----
L 151 11100000 * 15482.33
L 151 10000000
L 181 11000000 20 CLERK 13878.67
L 181 11000000 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
    
```

図 22. エクスポート後のサンプル報告書のフォーマット

QMF は、報告書をエクスポートするとき、フォーマット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。

- 報告書行レコード (L)
- データ継続レコード (C)

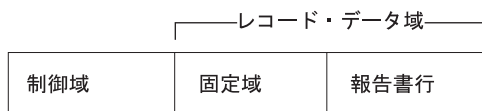
QMF オブジェクトのインポートとエクスポート

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フォーマット設定済みデータのみを使用したい場合、QMF に印刷出力をファイル、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。このファイル、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、フォーマット設定済みデータのみが入ります。

報告書行レコード (L)

報告書内のフォーマット設定済みの各行は、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。他の可変長フォーマット・レコード (V、T、R) と同様に、L レコードは制御域に続くレコード・データ域から構成されます。制御域のフォーマットは、他のレコードに似ています。レコード・データ域は、固定域に続くフォーマット設定済みの報告書行自体からなります。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 レコードの内容は次のとおりです。

L レコードの制御域:

バイト位置

説明

- 01 値レコード ID (L)
- 02 継続標識。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 (118 ページの『データ継続レコード (C)』参照)。
 - C は継続を示す
 - 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
 - ブランクは継続しないことを示す(以下の説明の後にある注 1 および 2(117 ページ) 参照。)

L レコードのレコード・データ域 (固定域)

バイト位置

説明

- 01 ブランク
- 02-04 報告書構成部分標識。 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

05 ブランク

06-13 行タイプ属性。バイト 06 は常に 1。バイト 7-13 の各バイトは、フォーマット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す (1 = 属性がある、0 = 属性がない)。

バイト位置

説明

- 06 1
- 07 データ
- 08 テキスト
- 09 区切り文字
- 10 列の折り返し。注 3 を参照。
- 11 行の折り返し。注 3 を参照。
- 12 2 番目のデータ行 (横方向報告書のみ)。注 4 を参照。
- 13 予約済み
- 14 ブランク

L レコードのレコード・データ域 (報告書行):

バイト位置

説明

01- 終わり

実際のフォーマット設定済み報告書行

L レコードの例

```
L 151 11100000      DEPARTMENT TOTALS      93,659.45
```

(切れ目 1 後書き行にはテキストとデータが入る)

注:

1. 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。
2. 制御域内の継続標識として D が指定されている場合、現行レコードが長すぎて単一の物理レコードに収まらないこと、および、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持するために、現行レコードと次のレコードに、S0 (シフトアウト) と SI (シフトイン) 文字が追加されたことを意味します。

3. 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。特定の L フォーマット・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 フォーマット・レコードに継続されることを意味します。
4. 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 (さらに、切れ目および最終) 合計ごとに 2 つのデータ行を入れることができます。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。

データ継続レコード (C)

C レコードは、値または値のセットを複数のレコードに継続するために使用します。このレコードは、継続するレコードの直後に続けます。C レコードのフォーマットは、継続する元のレコードのフォーマットに対応します。QMF は、L レコードを継続するためにだけ C レコードを使用します。C レコードには以下の区域があります。

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 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。

- C は継続を示す
- 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
- ブランクは継続しないことを示す

(1 (117 ページ) および 2 (117 ページ) の注を参照してください。)

C レコードのレコード・データ域

以下のリストに示したバイト位置は、制御域の終わりからのオフセットです。制御域の長さはヘッダー・レコードに示されています。

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

C レコードの例

単一の報告書行の値がテキストの途中で分割されるが、別の継続レコードには継続されない報告書行の継続

C ARS ----> <---- TOTAL ---->

単一の報告書行の値がテキストの途中で分割され、かつ、レコードが別の C レコードに継続される報告書行の継続

CC ERK ----> <---- MGR ----> <---- SAL

横方向報告書のエクスポート例

図 23 は、エクスポート後の横方向スタイルの報告書を示しています。

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
-----	-----	-----	-----
10		20865.86	20865.86
15	12383.35	20659.80	15482.33
20	13878.68	18357.50	16071.53
38	12482.25	17506.75	15457.11
	=====	=====	=====
	12914.76	19998.21	16880.26
COMPANY NAME REPORT 18 PAGE 1			

図 23. サンプル横方向報告書：この報告書は QMF 横方向報告書機能を使用しています。

120 ページの図 24 は、図 23 の結果のエンコード・フォーマットを示しています。

```

H QMF 11 R 01 E V W E R 02 03 98/10/14 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 10010000
L 181 11000000
L 181 11000000
L 181 11000000
L 181 11000000
L 190 10010000
L 190 11000000
L 120 10000000
L 120 10000000
L 120 10100000
L 120 10100000
L 120 10100000
E

                                J & H SUPPLY COMPANY
                                DEPT AVERAGE SALARIES
                                REPORT 18 (ACROSS REPORT)

                                <----- JOB ----->
                                <- CLERK --> <-- MGR ---> <- SALES --> <- TOTAL -->
                                AVERAGE AVERAGE AVERAGE AVERAGE
                                SALARY SALARY SALARY SALARY
                                DEPT
                                -----
                                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
    
```

図 24. エクスポート後の横方向スタイル報告書のフォーマット

HTML 報告書

HTML 用の報告書をエクスポートすると、QMF はその本文の前後に必要な HTML タグを置きます。これによって、その報告書を Web サーバー上に置いて、HTML 準拠の Web ブラウザーで表示することができます。121 ページの図 25 は、QMF が報告書の前後に置く HTML コーディングを示しています。

```
<HTML>
<HEAD>
<TITLE>
Report
</TITLE>
</HEAD>
<BODY>
<PRE>

                J & H SUPPLY COMPANY
      AVERAGE SALARY (DEPTS 10, 15, 20)
                REPORT 17

      DEPT  JOB          AVERAGE
      ----  -
      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>
```

図 25. HTML 報告書のコーディング・サンプル

QBE 照会

QBE 照会オブジェクトは、QMF の内部フォーマットを使用してエクスポートされません。このフォーマットは、どのような方法でも変更できません。

外部化 QMF オブジェクトの仕様

表 14 は、TSO ファイル、CMS IMPORT ファイル、および CMS EXPORT ファイルの仕様をリストしています。

CICS の場合、レコード・サイズは表 14 に示すサイズと同じですが、強制されるわけではありません。たとえば、一時記憶域キューから、レコード・サイズが 32k 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・フォーマットは、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、レコードをそのフォーマット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・フォーマットは無視されます。

キュー名は、ユーザーが生成し、デフォルトの接頭部または接尾部はありません。CICS TS キュー名は 8 バイトです。TD キュー名は 4 バイトです。

表 14. 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・フォーマット (CMS/TSO)
データまたは表 (QMF フォーマット)	最大サイズ: 7,000 バイト	レコードは固定長でなければならない。
データまたは表 (IXF フォーマット)	最大サイズ: 32,756 (注 2 参照) インポート時に IXF ファイル、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	レコードは可変長でなければならない。
指示照会	最大: 7,290 バイト 最小: EXPORT では 266 バイト、IMPORT では 41 バイト	レコードは EXPORT では可変長でなければならないが、IMPORT では固定長または可変長のいずれでもよい。
SQL 照会	EXPORT では 79 バイト、IMPORT では 255 バイト以下ならばよいが、79 バイトに切り捨てられる。	レコードは EXPORT では固定長でなければならないが、IMPORT では固定長または可変長のいずれでもよい。
QBE 照会	1,024 バイトでなければならない (注 3 参照)。	レコードは可変長でなければならない。
書式	最大: 7,290 バイト 最小: EXPORT では 161 バイト、IMPORT では 23 バイト	レコードは EXPORT では可変長でなければならないが、IMPORT では固定長または可変長のいずれでもよい。

表 14. 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・フォーマット (CMS/TSO)
プロシージャー	EXPORT では 79 バイト、IMPORT ではどのようなサイズでもよいが、79 バイトに切り捨てられる。	レコードは EXPORT では固定長でなければならないが、IMPORT では固定長または可変長のいずれでもよい。
報告書	最大: 7,290 バイト 最小: 65 バイト	レコードは可変長でなければならない。
HTML 報告書	最大: 32,000 バイト	レコードは可変長でなければならない。

注:

1. EXPORT または IMPORT コマンドでは、ファイル、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。名前の詳細については、「DB2 QMF 解説書」を参照してください。
2. 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。書式に列定義情報が入っていない場合、CMS の最小 LRECL は 113 バイトです。
3. 空の QBE 照会は 828 バイトです。
4. 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 (ヌル標識および DBCS 区切り文字のスペースを含む)、プラス 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) です。行の長さから引き出されたレコード・サイズが、最長の IXF ヘッダー・レコードの長さ (81 バイト) より短い場合、レコード・サイズは 81 バイトに設定されます。

CICS キューを使用する場合の規則と考慮事項

規則:

1. CICS では、IMPORT と EXPORT のいずれの場合も、QUEUE TYPE 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。
2. 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。キューは、QMF がその内容を取り出すと、空になります。たとえば、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ーの場合に、「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、(訂正したオブジェクト・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 1 度正常に出すことはできません。
3. 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い限りなりません。

- 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。CICS 一時データ・キューに関する情報については、「CICS (VSE/ESA 版) 適用業務プログラミングの手引き」を参照してください。

考慮事項:

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、オブジェクトを画面に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。さらに、表示するオブジェクトが大きい場合、QMF が画面にオブジェクトを表示する前に、遅れが生じる可能性があります。CICS の区画内一時データ・キューは、最高 32K のデータ行まで保持できます。区画外一時データ・キューは、オブジェクトを保持するために必要な大きさになります。
- 一時記憶域キュー:** QMF は、ユーザーに表示する前に、約 100 行の一時記憶域を読み取ります。一時記憶域キューは、最高 32K のデータ行まで保持することができます。
- キューへの QMF オブジェクトの追加:** QMF は、IMPORT および EXPORT コマンドで SUSPEND パラメーターを使用して、CICS にコマンドの実行時期を制御させます。

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。これは、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 に指示して、QMF トランザクションがキューによって処理中の他のジョブを妨害しないようにします。

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。QMF が自動 ENQ を発行した場合、EXPORT および IMPORT コマンドの SUSPEND オプションに反映されます。

第 10 章 QMF アプリケーションのデバッグ

エラー処理コマンドとアプリケーション・サポート・コマンドに加えて、QMF にはユーザーのプログラムのためのデバッグ機能も用意されています。本章で説明する技法は、呼び出し可能インターフェース・アプリケーションに適用されます。

ISPF のデバッグ技法については、39 ページの『第 6 章 ISPF を使用する QMF アプリケーションの作成』を参照してください。また、TRACE ステートメントを介して REXX トレース機能を使用することができます。このステートメントの詳細については、「REXX 解説書」を参照してください。

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、およびいくつかの異なるレベルのトレース方法が用意されています。

トレース用の L オプションの使用

L オプションによって、QMF セッションを開始する前に割り振った外部 QMF トレース・データ出力にメッセージとコマンドを書き込むように QMF に指示することができます。選択できる L オプションは 2 つあります。

- L1 すべての QMF メッセージを QMF トレース・データ出力に書き込む。
- L2 すべての 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。たとえば、L2 を使用して、Q.SYSTEM_INI システム初期化プロシージャ・コマンドおよびメッセージをトレースして、デバッグすることができます。

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。

1. DISPLAY PROFILE コマンドを出し、PROFILE オブジェクトが表示されたら、TRACE オプションを L1 または L2 に変更する。
2. 次のコマンドを出します。

```
SET PROFILE (TRACE=x
```

ここで x は L1 または L2。

トレース・データ出力を自分で割り振った場合には、端末でトレース情報を印刷したり、後で表示したりすることができます。いずれの場合も、QMF セッションの後でデータを調べることができます。割り振りの詳細については、127 ページの『QMF トレース・データ出力の割り振り』を参照するか、インストール先の情報センターにお問い合わせください。

トレース用の A オプションの使用

A オプションによって、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 オプションの設定値は、A0、A1、または A2 です。A0 はデフォルト値で、A トレースを実行しないことを示すシグナルとして解釈されます。A1 および A2 により、この順に詳細さのレベルを上げて結果を要求することができます。これは、他の QMF トレース・オプションにも使用されるパターンです。

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 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。たとえば、デバッグするアプリケーションを呼び出す直前に、次のように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。この値は、0、1、または 2 で、それぞれ設定値 A0、A1、または A2 を示します。DSQAO_APPL_TRACE の値を使用すれば、図 26 に示すようにアプリケーションに必要なトレースの種類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
end
```

図 26. トレース設定のサンプル REXX プログラム

図 26 に示すようなネスト DO グループは、アプリケーションのどこにでも置くことができます。このような場合、アプリケーションのデバッグに役立つように特定のデータ域のスナップショット・ダンプが取られるか、特定の重要な変数の値が印刷されるか、

デバッグ・モジュールがロードされるか他の診断プロシージャーが実行されます。何が実行されるかは、アプリケーションの実行時に A オプションとして有効である設定値によって異なります。

A オプション・コードは、大きなアプリケーションで使用するのに最適です。デバッグを終了した後、アプリケーションにこのコードを残すことを考えてください。この場合、A0 を設定してアプリケーションを実行すれば、A トレース出力は作成されません。アプリケーションを変更してバグが入り込んだ可能性がある場合、このコードを再び実行することができます。

トレースをオフにする

アプリケーションをテストした後、トレースをオフにするには、次のコマンドを出します。

```
SET PROFILE (TRACE=NONE
```

これによって、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 を呼び出す前に、QMF トレース・データ出力を割り振る必要があります。この出力を、始動プロシージャーを介して自動的に割り振ることができます。この場合でも、元の割り振りが要件に合わない場合には、出力の再割り振りが必要なことがあります。

CMS または TSO で QMF トレース・データ出力を割り振る方法の例については、該当する言語について説明している章にある、アセンブルまたはコンパイルおよび実行のコーディング例を参照してください。

アセンブラー

アセンブラー言語インターフェース (131 ページ以降)

C 言語 C 言語インターフェース (154 ページ以降)

COBOL

COBOL 言語インターフェース (172 ページ以降)

FORTRAN

FORTRAN 言語インターフェース (189 ページ以降)

PL/I PL/I 言語インターフェース (205 ページ以降)

REXX REXX 言語インターフェース (222 ページ以降)

各例のコマンドは、順次トレース・データ出力を割り振っています。この出力は QMF セッションの終了後に端末で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。トレース情報は、1 行につき 80 文字にフォーマット設定されます。端末画面で、出力の行全体を表示することができます。

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。CICS 一時記憶域は、メッセージまたは少量のアプリケーション・トレース・データのみを使用することをお勧めします。

QMF MESSAGE コマンドによるトレースの使用

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。そのためには、TRACE に関する L 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 (125 ページの『トレース用の L オプションの使用』を参照)。これにより、MESSAGE コマンドを介して処理されたすべてのメッセージが、他の QMF メッセージ (および L2 を使用した場合には、コマンド) と共に、QMF トレース・データ出力に記録されます。

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、端末または印刷出力で調べることができます。QMF トレース・データ出力の詳細については、127 ページの『QMF トレース・データ出力の割り振り』を参照してください。

例

アプリケーションが、次の例に示すようなコマンドを出したとします。

```
call dsqcix "SET PROFILE (TRACE=L2"  
:  
call dsqcix "MESSAGE (TEXT='QUERYA COMPLETED SUCCESSFULLY'"  
:  
:  
call dsqcix "MESSAGE (TEXT='EXECB ENTERED WITH VALUE OF 7'"  
:  
:
```

メッセージ 'QUERYA COMPLETED SUCCESSFULLY' および 'EXECB ENTERED WITH 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すべきではありません。

START コマンドおよび他の QMF コマンド上のエラーのデバッグ

DSQCOMM のレベルによっては、DSQCOMM 内にメッセージ・テキストが入ることがあります。START コマンド (または他の QMF コマンド) が失敗した場合、このメッセージ・テキストはデバッグにとって非常に価値があります。QMF のすべてのエラー・メッセージについては、「DB2 QMF メッセージおよびコード」を参照してください。

付録 A. 呼び出し可能インターフェース言語のサンプル・コード

この付録には、以下の QMF 呼び出し可能インターフェース言語のサンプル・コードがあります。

アセンブラー

『アセンブラー言語インターフェース』

C 言語 154 ページの『C 言語インターフェース』

COBOL

172 ページの『COBOL 言語インターフェース』

FORTRAN

189 ページの『FORTRAN 言語インターフェース』

PL/I 205 ページの『PL/I 言語インターフェース』

REXX 222 ページの『REXX 言語インターフェース』

この付録には、QMF によってサポートされる各言語のサンプル・プログラムがあります。各サンプル・プログラムは、以下のことを行います。

- QMF を開始する。
- 3 つのグローバル変数を設定する。
- 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するように書かれています。

この付録では、呼び出し可能インターフェースを使用して、プログラムをアセンブルまたはコンパイルし、リンク・エディットし、実行する方法も示しています。QMF には、以下の例にある REXX EXEC、JCL、または CLIST は含まれていませんが、これらをコピーしてインストール先に合うように変更することができます。

アセンブラー言語インターフェース

アセンブラー言語を使用する場合、呼び出し可能インターフェースでアセンブラー H または高水準アセンブラー (HLASM) を使用する必要があります。QMF には、アセンブラー言語用に 1 つの関数呼び出し DSQCIA が用意されています。

CICS/VSE の場合、31 ビット・アドレッシングを構成するために HLASM を使用する必要があります。

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 はアセンブラー用の DSQCOMM マッピングであり、プロダクトと共に出荷されます。表 15 は、DSQCOMMA の各値を示しています。

表 15. DSQCOMMA 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	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
DSQ_COMM_LEVEL	DS CL12	DSQCOMM のレベルを識別する。QMF START コマンドを出す前に、DSQ_CURRENT_COMM_LEVEL の値に設定する必要がある
DSQ_PRODUCT	DS CL2	使用中の IBM 照会プロダクトを識別する
DSQ_PRODUCT_RELEASE	DS CL2	使用中の照会プロダクトのリリース・レベルを識別する
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 が失敗したときの、エラーがあるパラメーター
DSQ_CANCEL_IND	DS C	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	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

表 15. DSQCOMMA 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_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

アセンブラー言語の関数呼び出しには、DSQCIA と DSQCIA 拡張構文という 2 つのフォーマットがあります。

DSQCIA

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

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

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

拡張構文フォーマットの DSQCIA 関数は、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 3 つの QMF コマンド (START および拡張フォーマットの GET GLOBAL と SET GLOBAL) 用です。

```
CALL DSQCIA, (DSQCOMM, CMDLTH, CMDSTR,
              PNUM, KLTH, KWORD, VLTH, VALUE, VTYPE), 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・キーワード数。FULLWORD パラメーター

呼び出し可能インターフェースのサンプル

KLTH

指定する各キーワードの長さ。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 (単数または複数)。 KLTH によって指定した長さと同じの、文字または文字の構造

VLTH

キーワードに関連する各値の長さ。 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する値。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、文字の構造、FULLWORD パラメーター、または FULLWORD パラメーターの配列のいずれかです。

VTYPE

値ストリング VALUE の QMF データ・タイプ。このタイプの値は、通信マクロ DSQCOMMA に提供されている以下の 2 つの値のいずれかです。

- 文字値を示す DSQ_VARIABLE_CHAR。VTYPE 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。VTYPE 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 VTYPE 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

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。

- z/OS の場合、サンプル・プログラムはライブラリー QMF810.SDSQSAPE のメンバーです。
- VM の場合、サンプル・プログラムはプロダクション・ディスクにあります。
- VSE の場合、サンプル・プログラムは QMF サブライブラリーにあり、名前は DSQABFAC.Z です。

アセンブラー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 つのグローバル変数を設定する。
- 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。

この節では、呼び出し可能インターフェースを使用するアセンブラー・プログラムをアセンブルし、リンク・エディットし、実行する方法も示します。この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

CICS/MVS および CICS/VSE 用のサンプル・アセンブラー・プログラム
プログラム DSQABFAC は、CICS 用の QMF と共に出荷されます。

```

TITLE 'Sample HLASM Query Callable Interface'                00001000
***** 00002000
* 00003000
* Sample Program: DSQABFAC * 00004000
* Assembler Version of the QMF Callable Interface * 00005000
* 00006000
***** 00007000
DSQABFAC DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11) 00008000
SPACE 1 00009000
***** 00010000
* Start a query interface session * 00011000
***** 00012000
LA R4,CICOMM ESTABLISH ACCESS TO DSQCOMM 00013000
USING DSQCOMM,R4 00014000
SPACE 1 00015000
MVC DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL 00016000
ST R4,QMFP1 Address of DSQCOMM 00017000
LA R1,STARTQIL Address of START command length 00018000
ST R1,QMFP2 00019000
LA R1,STARTQI Address of START command 00020000
ST R1,QMFP3 00021000
LA R1,1 One Start command parameter 00022000
ST R1,NUMPARMS 00023000
LA R1,NUMPARMS Address of number of parameters 00024000
ST R1,QMFP4 00025000
LA R1,STARTKYL Address of keyword lengths 00026000
ST R1,QMFP5 00027000
LA R1,STARTKY Address of keywords 00028000
ST R1,QMFP6 00029000
LA R1,STARTVL Address of value lengths 00030000
ST R1,QMFP7 00031000
LA R1,STARTV Address of values 00032000
ST R1,QMFP8 00033000
LA R1,DSQ_VARIABLE_CHAR Address of value data type 00034000
ST R1,QMFP9 00035000
OI QMFP9,X'80' Set end of parameter list 00036000
LA R1,QMFLIST Address of parameter list 00037000
CALL DSQCIA 00038000
SPACE 1 00039000

```

図 27. DSQABFAC、CICS/MVS および CICS/VSE 用のサンプル HLASM プログラム (1/5)

```

***** 00040000
* Set numeric values into query using SET command * 00041000
***** 00042000
SPACE 1 00043000
LA R1,20 Set values for SET GLOBAL command 00044000
ST R1,VVAL1 00045000
LA R1,40 00046000
ST R1,VVAL2 00047000
LA R1,84 00048000
ST R1,VVAL3 00049000
LA R1,SETGL Addr of SET GLOBAL command length 00050000
ST R1,QMFP2 00051000
LA R1,SETG Address of SET GLOBAL command 00052000
ST R1,QMFP3 00053000
LA R1,3 Three SET GLOBAL variables 00054000
ST R1,NUMPARMS 00055000
LA R1,NUMPARMS Address of number of parameters 00056000
ST R1,QMFP4 00057000
LA R1,VNAME1L Address of variable name lengths 00058000
ST R1,QMFP5 00059000
LA R1,VNAME1 Address of variable names 00060000
ST R1,QMFP6 00061000
LA R1,VVAL1L Address of value lengths 00062000
ST R1,QMFP7 00063000
LA R1,VVAL1 Address of values 00064000
ST R1,QMFP8 00065000
LA R1,DSQ_VARIABLE_FINT Address of value data type 00066000
ST R1,QMFP9 00067000
OI QMFP9,X'80' Set end of parameter list 00068000
LA R1,QMFP9L Address of parameter list 00069000
CALL DSQCIA 00070000
SPACE 1 00071000
***** 00072000
* Run a query * 00073000
***** 00074000

```

図 27. DSQABFAC、CICS/MVS および CICS/VSE 用のサンプル HLASM プログラム (2/5)

```

LA R1,QUERYL          Addr of RUN QUERY command length  00075000
ST R1,QMFP2           00076000
LA R1,QUERY          Address of RUN QUERY command  00077000
ST R1,QMFP3           00078000
OI QMFP3,X'80'        Set end of parameter list    00079000
LA R1,QMFPLIST        Address of parameter list    00080000
CALL DSQCIA           00081000
SPACE 1               00082000
*****
* Print the result of the query * 00084000
***** 00085000
LA R1,REPTL          Addr of PRINT Report command lth  00086000
ST R1,QMFP2           00087000
LA R1,REPT           Address of PRINT Report command  00088000
ST R1,QMFP3           00089000
OI QMFP3,X'80'        Set end of parameter list    00090000
LA R1,QMFPLIST        Address of parameter list    00091000
CALL DSQCIA           00092000
SPACE 1               00093000
***** 00094000
* End the query interface session * 00095000
***** 00096000
LA R1,ENDQIL          Address of EXIT command length  00097000
ST R1,QMFP2           00098000
LA R1,ENDQI           Address of EXIT command  00099000
ST R1,QMFP3           00100000
OI QMFP3,X'80'        Set end of parameter list    00101000
LA R1,QMFPLIST        Address of parameter list    00102000
CALL DSQCIA           00103000
SPACE 1               00104000
***** 00105000
* Return * 00106000
***** 00107000
SPACE 1               00108000

```

図 27. DSQABFAC、CICS/MVS および CICS/VSE 用のサンプル HLASM プログラム (3/5)

```

XR      R15,R15          ZERO RETURN CODE          00109000
DFHEIRET RCREG=15      00110000
*****
* Data Areas *
*****
SPACE 1          00114000
* Query Interface commands 00115000
SPACE 1          00116000
STARTQI DC C'START'          START FUNCTION      00117000
SETG    DC C'SET GLOBAL'    SET GLOBAL FUNCTION 00118000
QUERY   DC C'RUN QUERY Q1'  RUN QUERY          00119000
REPT    DC C'PRINT REPORT (FORM=F1)' PRINT REPORT 00120000
ENDQI   DC C'EXIT'          END INTERFACE      00121000
SPACE 1          00122000
DS      0F          00123000
STARTQIL DC AL4(L'STARTQI)  LENGTH OF START FUNCTION 00124000
SETGL   DC AL4(L'SETG)     LENGTH OF SET GLOBAL FUNCTION 00125000
QUERYL  DC AL4(L'QUERY)    LENGTH OF RUN QUERY COMMAND 00126000
REPTL   DC AL4(L'REPT)     LENGTH OF PRINT REPORT COMMAND 00127000
ENDQIL  DC AL4(L'ENDQI)    LENGTH OF END INTERFACE COMMAND 00128000
SPACE 1          00129000
* START command keyword 00130000
SPACE 1          00131000
STARTKY DC C'DSQSMODE'     00132000
STARTV  DC C'INTERACTIVE'  00133000
DS      0F          00134000
STARTKYL DC AL4(L'STARTKY) 00135000
STARTVL DC AL4(L'STARTV)   00136000
SPACE 1          00137000
* SET GLOBAL command variable names 00138000
SPACE 1          00139000
VNAME1  DC C'MYVAR01'     00140000
VNAME2  DC C'SHORT'       00141000
VNAME3  DC C'MYVAR03'     00142000
DS      0F          00143000
VNAME1L DC AL4(L'VNAME1)  00144000
VNAME2L DC AL4(L'VNAME2)  00145000
VNAME3L DC AL4(L'VNAME3)  00146000
SPACE 1          00147000
* SET GLOBAL command values 00148000
SPACE 1          00149000
VVAL1L  DC AL4(L'VVAL1)   00150000
VVAL2L  DC AL4(L'VVAL2)   00151000
VVAL3L  DC AL4(L'VVAL3)   00152000
* Callable interface communications definition 00153000
DSQCOMMA 00154000

```

図 27. DSQABFAC、CICS/MVS および CICS/VSE 用のサンプル HLASM プログラム (4/5)

```

* Equates for registers 0-15                                00155000
R0      EQU  00                                           00156000
R1      EQU  01                                           00157000
R2      EQU  02                                           00158000
R3      EQU  03                                           00159000
R4      EQU  04                                           00160000
R5      EQU  05                                           00161000
R6      EQU  06                                           00162000
R7      EQU  07                                           00163000
R8      EQU  08                                           00164000
R9      EQU  09                                           00165000
R10     EQU  10                                           00166000
R11     EQU  11                                           00167000
R12     EQU  12                                           00168000
R13     EQU  13                                           00169000
R14     EQU  14                                           00170000
R15     EQU  15                                           00171000
* Local variables located in CICS working storage          00172000
DFHEISTG DSECT                                           00173000
          ORG  DFHEIUSR                                   00174000
NUMPARMS DS  F          NUMBER OF KEYWORDS              00175000
* QMF SET GLOBAL command values                          00176000
VVAL1   DS  F                                           00177000
VVAL2   DS  F                                           00178000
VVAL3   DS  F                                           00179000
* QMF Callable interface parameter list                  00180000
QMFP1   DS  0D                                           00181000
QMFP1   DS  F                                           00182000
QMFP2   DS  F                                           00183000
QMFP3   DS  F                                           00184000
QMFP4   DS  F                                           00185000
QMFP5   DS  F                                           00186000
QMFP6   DS  F                                           00187000
QMFP7   DS  F                                           00188000
QMFP8   DS  F                                           00189000
QMFP9   DS  F                                           00190000
* Callable interface communications area                  00191000
CICOMM  DS  CL(DSQCOMM_LEN)                               00192000
          CSECT                                         00193000
          SPACE 1                                       00194000
          END  DSQABFAC                                  00195000

```

図 27. DSQABFAC、CICS/MVS および CICS/VSE 用のサンプル HLASM プログラム (5/5)

TSO および CMS 用のサンプル・アセンブラー・プログラム

TSO および CMS の場合、QMF はプロダクトと共に次のプログラムを出荷します。プログラム名は DSQABFA です。

```

DSQABFA  TITLE 'SAMPLE CALLABLE INTERFACE'
DSQABFA  CSECT
*****
*
* 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,              X
    (CICOMM,             QI COMMON AREA          X
    STARTQIL,            START COMMAND LENGTH    X
    STARTQI,             START COMMAND           X
    NUMPARMS,            NUMBER OF KEYWORDS      X
    STARTKYL,            KEYWORD LENGTHS        X
    STARTKY,             KEYWORDS                X
    STARTVL,             VALUE LENGTHS          X
    STARTV,              VALUES                X
    DSQ_VARIABLE_CHAR),VL VALUES ARE CHARACTERS
SPACE 1

```

図 28. DSQABFA、TSO および CMS 用のサンプル・アセンブラー・プログラム (1/4)

```

*****
* 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, X
(CICOMM, X
SETGL, SET GLOBAL COMMAND LENGTH X
SETG, SET GLOBAL COMMAND X
NUMPARMS, NUM OF VARIABLES TO BE SET X
VNAME1L, VARIABLE NAME LENGTHS X
VNAME1, VARIABLE NAMES X
VVAL1L, VALUE LENGTHS X
VVAL1, VALUES X
DSQ_VARIABLE_FINT),VL VALUES ARE INTEGERS
SPACE 1
*****
* Run a query *
*****
SPACE 1
CALL DSQCIA, X
(CICOMM, X
QUERY, QUERY COMMAND LENGTH X
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

```

図 28. DSQABFA、TSO および CMS 用のサンプル・アセンブラー・プログラム (2/4)


```

*****
* 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

```

図 28. DSQABFA、TSO および CMS 用のサンプル・アセンブラー・プログラム (3/4)

```

* 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 DSQBFA

```

図 28. DSQBFA、TSO および CMS 用のサンプル・アセンブラー・プログラム (4/4)

アセンブラー用の DSQCOMM

この連絡域は、バージョン 2.4 とバージョン 3.2 の間で変更されました。 QMF はこのファイルを DSQCOMMA として出荷します。

```

MACRO                                00001000
DSQCOMMA                              00002000
*****                              00003000
* Callable Interface - variable constants * 00004000
*****                              00005000
*                                     00006000
* Communications Level ID            00007000
*                                     00008000
DSQ_CURRENT_COMM_LEVEL DC CL12'DSQL>001002<' 00009000
*                                     00010000
* Query Product IDs                  00011000
*                                     00012000
DSQ_QRW DC C'01'                      00013000
DSQ_QMF DC C'02'                      00014000
DSQ_QM4 DC C'03'                      00015000
*                                     00016000
* Query Product Release IDs          00017000
*                                     00018000
DSQ_QRW_V1R2 DC C'01'                 00019000
DSQ_QRW_V1R3 DC C'02'                 00020000
DSQ_QMF_V2R4 DC C'01'                 00021000
DSQ_QMF_V3R1 DC C'02'                 00022000
DSQ_QMF_V3R1M1 DC C'03'              00023000
DSQ_QMF_V3R2 DC C'04'                 00024000
DSQ_QMF_V3R3 DC C'05'                 00025000
DSQ_QMF_V6R1 DC C'06'                 00026000
DSQ_QM4_V1R1 DC C'01'                 00027000
*                                     00028000
* Extended parameter data types      00029000
*                                     00030000
DSQ_VARIABLE_CHAR DC C'CHAR'          00031000
DSQ_VARIABLE_FINT DC C'FINT'          00032000
*                                     00033000
* Return codes                       00034000
*                                     00035000
DSQ_SUCCESS EQU 0                      00036000
DSQ_WARNING EQU 4                      00037000
DSQ_FAILURE EQU 8                      00038000
DSQ_SEVERE EQU 16                      00039000
*                                     00040000
* Instance ID values                 00041000
*                                     00042000
DSQ_CONTINUE EQU 0                     00043000
*                                     00044000

```

図 29. DSQCOMMA、アセンブラー連絡域 (1/2)

* Cancel indicator				00045000
*				00046000
DSQ_CANCEL_YES	EQU	C'1'		00047000
DSQ_CANCEL_NO	EQU	C'0'		00048000
*				00049000
*				00050000
DSQ_INTERACTIVE	EQU	C'1'		00051000
DSQ_BATCH	EQU	C'2'		00052000
*				00053000
DSQ_YES	EQU	C'1'		00054000
DSQ_NO	EQU	C'2'		00055000
*				00056000
*****				00057000
* Callable Interface Communications Area			*	00058000
*****				00059000
DSQCOMM	DSECT			00060000
DSQ_RETURN_CODE	DS	F	FUNCTION RETURN CODE	00061000
DSQ_INSTANCE_ID	DS	F	ID ESTABLISHED IN START CMD	00062000
DSQ_COMM_LEVEL	DS	CL12	COMMUNICATIONS LEVEL ID	00063000
DSQ_PRODUCT	DS	CL2	QUERY PRODUCT ID	00064000
DSQ_PRODUCT_RELEASE	DS	CL2	QUERY PRODUCT RELEASE ID	00065000
DSQ_RESERVE1	DS	CL28	RESERVED	00066000
DSQ_MESSAGE_ID	DS	CL8	COMPLETION MESSAGE ID	00067000
DSQ_Q_MESSAGE_ID	DS	CL8	QUERY MESSAGE ID	00068000
DSQ_START_PARM_ERROR	DS	CL8	START PARAMETER IN ERROR	00069000
DSQ_CANCEL_IND	DS	C	CMD CANCEL INDICATOR	00070000
DSQ_RESERVE2	DS	CL23	RESERVED	00071000
DSQ_RESERVE3	DS	CL156	RESERVED	00072000
DSQ_MESSAGE_TEXT	DS	CL128	COMPLETION MESSAGE	00073000
DSQ_Q_MESSAGE_TEXT	DS	CL128	QUERY MESSAGE	00074000
SPACE 1				00075000
DSQCOMM_LEN	EQU	*-DSQCOMM	LENGTH OF DSQCOMM AREA	00076000
MEND				00077000

図 29. DSQCOMM、アセンブラー連絡域 (2/2)

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。この節にリストした例は、そのために必要なステップを示しています。この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

z/OS での CICS 用の変換、アセンブル、およびリンク・エディット

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し、アセンブルし、リンク・エディットする場合、以下のことに注意してください。

- 連絡域マクロ DSQCOMMA 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・フェーズで 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。

次の JCL は、CICS 提供のプロシージャ DFHEBTAL の例を示しています。このプロシージャの使用方法については、「CICS (VSE/ESA 版) システム定義の手引き」を参照してください。

```
//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=QMF810.SDSQSAPE,DISP=SHR
/* Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE CICSLOAD(DFHEAI)
          INCLUDE CICSLOAD(DFHEAI0)
          INCLUDE QMFLOAD(DSQCIA)
          ORDER DFHEAI,DFHEAI0
          ENTRY sampasm
          MODE AMODE(31) RMODE(ANY)
          NAME sampasm(R)
/*
```

図 30. CICS 変換プログラム、アセンブラー、リンケージ・エディターの実行用 JCL

VSE での CICS 用の変換、アセンブル、およびリンク・エディット

以下の VSE ジョブ制御は、VSE/ESA で実行中の CICS に HLASM プログラムをインストールする例です。この例は QMF に含まれており、DSQ3CIAC.Z. という名前で QMF サブライブラリーに置かれています。詳細については、「CICS (VSE/ESA 版) システム定義の手引き」を参照してください。

インストール先で HLASM を使用している場合、システム管理者が E-Deck のマクロ処理を扱う VSE ライブラリー出口を設定していることを確認してください。この出口は DSQCOMMA を読み取ります。この出口のセットアップ方法の詳細な説明については、「VSE Guide to System Functions」および「IBM High-Level Assembler Programmer's Guide for OS/390, VM and VSE」を参照してください。

呼び出し可能インターフェースのサンプル

次の HLASM コンパイラー・オプションを使用して、プログラムをアSEMBルしてください。

```
'LIBMAC,USING(NOLIMIT,NOWARN),EXIT(LIBEXIT(EDECKXIT(ORDER=EA)))'
```

LIBEXIT パラメーターには、CICS 変換プロセスによって作成された CICS マクロ定義が入っています。

```
// JOB DSQ3CIAC
* -----
* Install QMF Callable Interface (HLASM)
* -----
// SETPARM VOLID=valid      *-- update valid for syspch
// SETPARM START=rtrk      *-- update start track/block (syspch)
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks (syspch)
* -----
// DLBL IJSYSPH,'ASM.TRANSLATION',0
// EXTENT SYSPCH,,1,0,&START,&SIZE.
ASSGN SYSPCH,DISK,VOL=&VOLID.,SHR
* Library search chain must contain the QMF, CICS and HLASM sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate Callable Interface program
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC DFHEAP1$
* $$ SLI MEM=DSQABFAC.Z,S=PRD2.PROD
/*
```

図 31. VSE でアSEMBラーおよびエディターを実行するためのジョブ制御 (1/2)

```

* -----
* Step 2: Assemble Callable Interface program
* -----
CLOSE SYSPCH,00D
// DLBL IJSYSIN,'ASM.TRANSLATION',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID.,SHR
// OPTION CATAL,DECK,SYM,ERRS
    PHASE DSQABFAC,*
        INCLUDE DFHEAI
        INCLUDE DFHEAI0
        INCLUDE DSQCIA
        INCLUDE DSQCLOD2
        INCLUDE DSQCMCVP
// EXEC ASMA90,SIZE=(ASMA90,50K),
        PARM='LIBMAC,USING(NOLIMIT,NOWARN),EXIT(LIBEXIT(EDECKXITC
        (ORDER=EA)))'
CLOSE SYSIPT,SYSRDR
/*
* -----
* Step 3: Link-edit Callable Interface program
* -----
// EXEC LNKEDT,PARM='AMODE=31,RMODE=ANY'
/*
/&
// JOB RESET
ASSGN SYSIPT,SYSRDR IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,00D IF 1A93D, CLOSE SYSPCH,00D
/&

```

図 31. VSE でアセンブラーおよびエディターを実行するためのジョブ制御 (2/2)

VM の CMS のもとでのプログラムのアセンブルおよび実行

次のサンプル・プログラムは、アセンブラー H 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アセンブルして実行します。この例の REXX EXEC は QMF では出荷されませんが、ここからコピーして、インストール先に合うように変更することができます。

```

/*****/
/* Assemble your program and execute it. */
/*****/
TRACE off
ADDRESS CMS

/*****/
/* Assemble the program */
/*****/
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMMA"
Maclist = "TEMPP DMSSP CMSLIB OSMACRO"
"GLOBAL MACLIB" Maclist
"HASM yourname"

/*****/
/* Access DB2 Server for VM and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"

/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"

/*****/
/* Issue Filedefs for QMF product */
/*****/
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A (PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 (PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * (PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQLDLIB LOADLIB *"

```

図 32. プログラムをアSEMBルし、実行する REXX プログラム (1/2)

```
/* Provide access to QMF and GDDM program libraries */
GLOBAL LOADLIB DSQDLIB "
GLOBAL TXTLIB ADMRLIB ADMPLIB ADMGLIB"

Say "Starting to execute 'ASSEMBLER' program"
ADDRESS CMS "RUN yourname"

Exit 0
```

図 32. プログラムをアセンブルし、実行する REXX プログラム (2/2)

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。以下の節には、プログラムをアセンブルしリンク・エディットするサンプル・ジョブ、および ISPF を使用する場合と使用しない場合に、TSO 内でアセンブル済みプログラムを実行するためのサンプル・プログラムがあります。

TSO でのアセンブルとリンク・エディット

次のサンプル・ジョブは、アセンブラー H を使用してプログラムをアセンブルし、リンク・エディットします。いくつかのパラメーターは、インストール先によって異なることがあります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```
//sampasm JOB
//STEP1 EXEC PROC=ASMHCL
/* Provide Access to QMF Communications Macro DSQCOMM
//C.SYSLIB DD DSN=QMF810.SAMPLIB,DISP=SHR
//C.SYSIN DD *
        .
        Your program or copy of QMF sample DSQABFA
        .
/*
/* Provide Access to QMF Interface Module
//L.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//L.SYSIN DD *
        INCLUDE QMFLOAD(DSQCIA)
        ENTRY sampasm
        MODE AMODE(31) RMODE(ANY)
        NAME sampasm(R)
/*
```

図 33. TSO でのアセンブラーおよびリンケージ・エディターの実行のための JCL

プログラムは、正常にアセンブルした後、実行することができます。

ISPF のもとの TSO での実行

ユーザーのプログラムは、正常にアセンブルした後、次のようなプログラムを作成することによって、ISPF のもとで実行することができます。

```

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('QMF810.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****
/* Datasets used by ISPF */
/*****
ALLOC FI(ISPLLIB) SHR REUSE +

        DA('QMF810.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD')
ALLOC FI(ISPLIB) SHR REUSE +
        DA('QMF810.SDSQMLBE','ISR.ISRMLIB','ISP.ISPLIB')
ALLOC FI(ISPLLIB) SHR REUSE +
        DA('QMF810.SDSQPLBE','ISR.ISRPLIB','ISP.ISPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
        DA('QMF810.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
        DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* QMF/GDDM Datasets */
/*****
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****
/* Start your program as the initial ISPF dialog */
/*****
ISPSTART PGM(sampasm) NEWAPPL(DSQE)
EXIT CODE(4)

```

図 34. ISPF のもとの TSO でのプログラム実行のための CLIST

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

ユーザーのプログラムは、正常にアセンブルした後、次のようなプログラムを作成することによって、ISPF を使用しない TSO で実行することができます。

```

PROC 0
CONTROL ASIS
/*****
/* Note: QMF, DB2 and GDDM load libraries must be allocated */
/* before executing this CLIST. */
/* Name of QMF load library is "QMF810.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('QMF810.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****
/* QMF/GDDM Datasets */
/*****
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****
/* Start your program using TSO CALL command */
/*****
CALL sampasm
EXIT CODE(0)

```

図 35. ISPF を使用しない TSO でのプログラム実行用の CLIST

C 言語インターフェース

C 言語のために、QMF には DSQCOMMC 通信マクロ、および DSQCICE と DSQCIC という 2 つの関数呼び出しが用意されています。

注: C++ アプリケーションから QMF にアクセスするには、C で作成されたインターフェースが必要です。

C 言語用のインターフェース連絡域マッピング (DSQCOMM)

DSQCOMM は C 言語用の DSQCOMM マッピングであり、プロダクトと共に出荷されます。表 16 は、DSQCOMM の各値を示しています。

表 16. 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。QMF START コマンドを出す前に、DSQ_CURRENT_COMM_LEVEL の値に設定する必要がある
DSQ_PRODUCT	文字、長さ 2	使用中の IBM 照会プロダクトを識別する
DSQ_PRODUCT_RELEASE	文字、長さ 2	使用中の照会プロダクトのリリース・レベルを識別する
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。 <ul style="list-style-type: none"> • 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。

```
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。

DSQCICE

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 3 つの QMF コマンド (START および拡張フォーマットの GET GLOBAL と SET GLOBAL) 用の拡張構文があります。

```
DSQCICE (&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。これは、無符号文字タイプの配列です。QMF コマンドは大文字でなければなりません。

PNUM

コマンド・キーワード数。これは、長整数パラメーターです。

KLTH

指定する各キーワード &KEYWORD の長さ。これは、長整数パラメーターまたは長整数パラメーターの配列です。

KEYWORD

QMF キーワード (単数または複数)。各キーワードは、無符号文字配列です。

VLTH

キーワードに関連する各値の長さ。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する値。この値のタイプは、**VTTYPE** パラメーターに指定します。この値は、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です。

VTTYPE

値ストリング **VALUE** のデータ・タイプ。このタイプの値は、通信マクロ **DSQCOMM** に提供されている以下の 2 つの値のいずれかです。

- 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、**VTTYPE** に指定する必要があります。

C 言語インターフェースは、他の言語のインターフェースと似ています。しかし、パラメーターに関する以下の考慮事項があります。

- コマンド・ストリング、**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。これらのパラメーターとして、C では、ヌル値で終了する記憶域を渡す必要があります。ヌル値はパラメーターの長さを含めなければなりません。**QMF** インターフェースに渡すパラメーター長を入手するために、コンパイル時長さ関数を使用する必要があります。
- ストリングが、ストリングの終わりに到達する前にヌル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。ヌル値 (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** コマンドによって入手した値を含む) の場合、**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、ストリングの終わりにヌル標識が設定されます。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。ヌル標識は、常にデータ・ストリングの終わりに置かれます。

移行情報

DSQCOMM はバージョン 2.4 とバージョン 3.2 の間で変更されました。

- 古い **DSQCOMM** を使用し続ける場合、プログラムを再コンパイルする必要はありません。
- バージョン 3.2 の **DSQCICX** を使用する場合、バージョン 2.4 のプログラムを再リンク・エディットする必要があります。

ただし、バージョン 3.2 の **DSQCOMM** には、**START** コマンドにエラーがある場合に特に有用なメッセージ・テキストが用意されています。新しい **DSQCOMM** を使用する

C 言語インターフェース

場合、プログラムを再コンパイルして、DSQ_COMM_LEVEL (DSQCOMM の) を DSQ_CURRENT_COMM_LEVEL に初期化する必要があります。この値を設定しない場合、QMF は DSQCOMM をバージョン 2.4 のレベルとして扱います。

z/OS での CICS ユーザーへの注

DSQCICX はバージョン 3 リリース 1 モディフィケーション 1 とバージョン 3 リリース 2 の間で変更されました。QMF が提供する関数呼び出しとメイン QMF プログラム間のインターフェースが、CALL インターフェースから EXEC CICS LINK インターフェースに変更されました。新しいインターフェースは、ユーザー・プログラムと QMF プロダクトからの分離性が高くなっています。インターフェースが変更されたので、呼び出し可能インターフェースを使用したプログラムを再びリンク・エディットする必要があります。

C 言語プログラミングの例

この例は、IBM C 言語用の 呼び出し可能インターフェースを示しています。

以下のプログラム DSQABFC は QMF プロダクトと共に出荷されます。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。

- z/OS の場合、サンプル・プログラムはライブラリー QMF810.SDSQSAPE のメンバーです。
- VM の場合、サンプル・プログラムはプロダクション・ディスクにあります。
- VSE の場合、サンプル・プログラムは QMF サブライブラリーにあり、名前は DSQABFC.Z です。

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 つのグローバル変数を設定する。
- 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。

この節では、呼び出し可能インターフェースを使用する C 言語プログラムをコンパイルし、リンク・エディットし、実行する方法も示します。この例の REXX

EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

```

/*****/
/* 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.H>

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 */

```

図 36. DSQABFC、サンプル C プログラム (1/3)

```
/* ***** */
/* 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]);
```

図 36. DSQABFC、サンプル C プログラム (2/3)

```

/*****
/* 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);
    dsqic(&communication_area, &command_length,;
          &run_query[0]);

/*****
/* Print the results of the query                         */
/*****
    command_length = sizeof(print_report);
    dsqic(&communication_area, &command_length,;
          &print_report[0]);

/*****
/* End the query interface session                       */
/*****
    command_length = sizeof(end_query_interface);
    dsqic(&communication_area, &command_length,;
          &end_query_interface[0]);
    exit(0);
}

```

図 36. DSQABFC、サンプル C プログラム (3/3)

C 用の DSQCOMM

この組み込みファイル DSQCOMM は QMF プロダクトと共に出荷されます。

```

/*****
/* C Include for Query Callable Interface (MVS/VM) */
/*****

/* 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_QM4_V1R1 "01"

```

図 37. DSQCOMM、C 連絡域 (1/2)

```

/* 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) \
    dsqcicx( parm1, parm2, parm3)

/* Calling format for call with CMD_EXT area 9 parameters */
#define dsqcice(parm1, parm2, parm3, \
    parm4, parm5, parm6, parm7, parm8, parm9) \
    dsqcicx( parm1, parm2, parm3, \
    parm4, parm5, parm6, \
    parm7, parm8, parm9 )

/* DECLARE OS LINKAGE FORMAT                           */
#pragma linkage(dsqcicx, OS)

```

図 37. DSQCOMMC、C 連絡域 (2/2)

CICS のもとでのプログラムの実行

プログラムを作成した後、実行する前に、変換し、コンパイルし、リンク・エディットする必要があります。この節の例は、必要なステップを示しています。この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

CICS のもとで 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し、コンパイルし、リンク・エディットする場合、以下のことについて考慮してください。

- 連絡域マクロ DSQCOMMC 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。

C 言語インターフェース

- リンク・エディット・フェーズで 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。

z/OS での CICS の場合の変換、コンパイル、およびリンク・エディット

以下の例は CICS 提供のプロシージャ DFHEBTDL を使用しています。

```
//sampleC JOB
//          EXEC PROC=DFHEBTDL
//TRN.SYSIN DD *
#pragma XOPTS(CICS translator options .....
             *
             Your program or copy of QMF sample DSQABFC
             *
/*
/* Provide Access to QMF Communications Macro DSQCOMM
//ASM.SYSLIB DD DSN=QMF810.SDSQSAPE,DISP=SHR
/* Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
             INCLUDE CICSLOAD(DFHELII)
             INCLUDE QMFLOAD(DSQCICX)
             ORDER DFHELII
             ENTRY sampleC
             MODE AMODE(31) RMODE(ANY)
             NAME sampleC(R)
/*
```

図 38. CICS 変換プログラム、C コンパイラ、リンケージ・エディターの実行用 JCL

C/370 言語プログラムは AMODE=31 を指定してリンク・エディットする必要があります。

VSE での CICS の場合の変換、コンパイル、およびリンク・エディット

C/370 プリリンク・ステップ中に、サブライブラリー PRD2.PROD (QMF デフォルト・インストール・サブライブラリー) にある IBM 提供のインターフェース・オブジェクト (DSQCICX.OBJ、DSQCLOD2.OBJ、および DSQCMCVP.OBJ) が、LIBDEF * 検索チェーン内で使用可能でなければなりません。

リンク・エディットの段階では、165 ページの図 39 に示すように、CICS アセンブラー・インターフェース DFHEAIO が LIBDEF * 検索チェーン内に存在していなければなりません。

このサンプル・ジョブ制御は、DSQ3CIC.Z. として PRD2.PROD に保管されています。

```

// JOB DSQ3CIC   Sample job to Install QMF Callable Interface (C/370)
* -----
* Install QMF Callable Interface Example (C/370)
* -----
// SETPARM VOLID=volid      *-- update volid for syspch
// SETPARM START=rtrk      *-- update start track/block
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks
// SETPARM VOLID2=volid2   *-- update volid for work area
// SETPARM START2=rtrk    *-- update start track/block
// SETPARM SIZE2=ntrks    *-- update number of tracks/blocks
* -----
// DLBL IJSYSPH,'c.translation',0
// EXTENT SYSPCH,,1,0,&START,&SIZE
ASSGN SYSPCH,DISK,VOL=&VOLID,SHR
* Library search chain must contain the QMF, CICS and C/370 sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate callable interface program (C/370)
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC DFHEDP1$,SIZE=256K
..* $$ SLI MEM=DSQABFC.Z,S=PRD2.QMFD
/*
CLOSE SYSPCH,00D
* -----
* Step 2: Compile callable interface program (C/370)
* -----
// DLBL IJSYSIN,'c.translation',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID,SHR
// DLBL IJSYSPH,'compiler.output',0
// EXTENT SYSPCH,,1,0,&START2,&SIZE2
ASSGN SYSPCH,DISK,VOL=&VOLID2,SHR
// OPTION DECK
// EXEC EDCCOMP,SIZE=EDCCOMP,PARM='RENT'
CLOSE SYSIPT,SYSRDR
CLOSE SYSPCH,00D
* -----
* Step 3: Pre-link callable interface program (C/370)
* -----
// DLBL IJSYSIN,'compiler.output',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID2,SHR
// OPTION CATAL,NODECK
  PHASE DSQABFC,*
    INCLUDE DFHELII
    INCLUDE DFHEA10
// EXEC EDCPRLK,SIZE=EDCPRLK
CLOSE SYSIPT,SYSRDR
/*

```

図 39. CICS/VSE 変換プログラム、C コンパイラ、およびリンケージ・エディターの実行用のジョブ制御 (1/2)

```
* -----  
* Step 4: Link-edit callable interface program (C/370)  
* -----  
// EXEC LNKEDT,PARM='AMODE=24,RMODE=24'  
/*  
/ &  
// JOB RESET  
ASSGN SYSIPT,SYSRDR IF 1A93D, CLOSE SYSIPT,SYSRDR  
ASSGN SYSPCH,00D IF 1A93D, CLOSE SYSPCH,00D  
/ &
```

図 39. CICS/VSE 変換プログラム、C コンパイラー、およびリンケージ・エディターの実行用のジョブ制御 (2/2)

VM の CMS のもとでのプログラムのコンパイルと実行

次のプログラムは、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実行します。

この例の REXX EXEC は QMF では出荷されませんが、ここからコピーして、インストール先に合うように変更することができます。


```

/*****
/* Compile your program and run it. */
/*****
TRACE off
ADDRESS CMS
/*****
/* Access C product disk using an exec, PRODUCT, that you write. */
/*****
EXEC PRODUCT ADC370
/*****
/* Compile the program */
/*****
"GLOBAL TXTLIB IBMLIB SCEELKED"
"GLOBAL LOADLIB EDCLINK SCREERUN"
"CC" PNAME "(SOURCE SHOWINC"
/*****
/* Create an executable "C" module file
/*****
"GLOBAL LOADLIB DSQDLIB SCREERUN"
"GLOBAL TXTLIB EDCBASE ADMRLIB ADMPLIB ADMGLIB"
"MOD yourname DSQCICX DSQCLOD2 DSQCMCVP"
/*****
/* Access DB2 Server for VM and initialize database */
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"
/*****
/* Access GDDM product disk */
/*****
"EXEC PRODUCT GDDM"
/*****
/* Issue Filedefs for QMF product
/*****
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A (PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 (PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * (PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"
/*****
/* Provide access to QMF and C program libraries
/*****
"GLOBAL LOADLIB DSQDLIB SCREERUN"
"GLOBAL TXTLIB EDCBASE ADMRLIB ADMPLIB ADMGLIB"
Say "Starting to run 'C' program"
"yourname"

Exit 0

```

図 40. ユーザーのプログラムをコンパイルし実行する REXX プログラム

このプログラムは、インストール先に合うように変更する必要がある場合があります。

TSO での C プログラムの実行

次の節には、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・エディットするサンプル・ジョブ、および ISPF を使用する場合と使用しない場合に、コンパイル済みプログラムを実行するためのサンプル・プログラムがあります。

TSO でのコンパイルとリンク・エディット

以下のジョブは、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、インストール先によって異なることがあります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```
//sampleC JOB
//STEP1 EXEC PROC=EDCCL,LPARM='MAP'
/** Provide Access to QMF Communications Macro DSQCOMM
//COMPILE.SYSLIB DD DSN=QMF810.SAMPLIB,DISP=SHR
//COMPILE.SYSIN DD DATA,DLM='<>'

      .
      Your program or copy of QMF sample DSQABFC
      .
<>
/** Provide Access to QMF Interface Module DSQCICX
//LKED.SYSLIB DD DSN=QMF810.SDSQLOAD,DISP=SHR
/**
```

図 41. TSO での C コンパイラーとリンケージ・エディターの実行のための JCL

ISPF を使用しない TSO でのプログラムの実行

ユーザーのプログラムを正常にコンパイルした後、実行するために、次のようなプログラムを作成することができます。

```
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 "QMF810.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('QMF810.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
```

図 42. ISPF を使用しない TSO でのプログラム実行のための CLIST (1/2)

```
/* **** */
/* * QMF/GDDM Datasets * */
/* **** */
ALLOC FI(ADMGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMDF) 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('QMF810.DSQPNLE') SHR REUSE
/* **** */
/* * Start your program using TSO CALL command * */
/* **** */
CALL sampleC
EXIT CODE(0)
```

図 42. ISPF を使用しない TSO でのプログラム実行のための CLIST (2/2)

ISPF のもとの TSO でのプログラムの実行

ユーザーのプログラムを正常にコンパイルした後、実行するために、次のようなプログラムを作成することができます。

```
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('QMF810.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
```

図 43. ISPF のもとでの TSO での DSQABFC の実行のための CLIST (1/2)

```

/*****
/* Datasets used by ISPF */
/*****
ALLOC FI(ISPLLIB) SHR REUSE +

        DA('QMF810.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
          'EDC.SEDCLINK','PLI.SIBMLINK')
ALLOC FI(ISPMLIB) SHR REUSE +
        DA('QMF810.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
        DA('QMF810.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
        DA('QMF810.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
        DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* QMF/GDDM Datasets */
/*****
ALLOC FI(ADMGMGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****
/* Start your program as the initial ISPF dialog */
/*****
ISPSTART PGM(sampleC) NEWAPPL(DSQE)
EXIT CODE(4)

```

図 43. ISPF のもとの TSO での DSQABFC の実行のための CLIST (2/2)

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

COBOL で呼び出し可能インターフェース・プログラムを作成する場合、VS COBOL II、COBOL/370、z/OS および VM 用 IBM COBOL、または VSE 用 IBM COBOL を使用する必要があります。¹。

COBOL 用のインターフェース連絡域マッピング (DSQCOMMB)

DSQCOMMB は COBOL 用の DSQCOMM マッピングであり、プロダクトと共に出荷されます。表 17 は、DSQCOMMB の各値を示しています。

表 17. 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 警告を伴って通常終了した。 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 セッションが終了した。
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。QMF START コマンドを出す前に、DSQ_CURRENT_COMM_LEVEL の値に設定する必要がある
DSQ-PRODUCT	PIC X(2)	使用中の IBM 照会プロダクトを識別する
DSQ-PRODUCT-RELEASE	PIC X(2)	使用中の照会プロダクトのリリース・レベルを識別する
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

1. COBOL/370 は、CICS/VSE ではサポートされません

表 17. 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 1 つの関数呼び出し DSQCIB が用意されています。これは、通信マクロ DSQCOMMB に記述されています。この関数呼び出しには、DSQCIB と DSQCIB 拡張フォーマットという 2 つのフォーマットがあります。

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。これは、整数パラメーターです。

CMDSTR

実行する QMF コマンド。これは、CMDLTH によって指定した長さの大文字のストリングです。

DSQCIB、拡張フォーマット

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 3 つの QMF コマンド (START および拡張フォーマットの 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。これは、整数パラメーターです。

CMDSTR

実行する QMF コマンド。これは、CMDLTH によって指定した長さの大文字のストリングです。

PNUM

コマンド・キーワード数。これは、整数パラメーターです。

KLTH

指定する各キーワードの長さ。これは、整数パラメーターまたは整数パラメーターの配列です。

KWORD

QMF キーワード (単数または複数)。各キーワードは、KLTH によって指定した長さの文字または文字の構造です。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

キーワードに関連する各値の長さ。これは、整数パラメーターまたは整数パラメーターの配列です。

VALUE

各キーワードに関連する値。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、文字の構造、整数パラメーター、または整数パラメーターの配列です。

VTYPE

値ストリング VALUE の QMF データ・タイプ。このタイプの値は、通信マクロ DSQCOMM 提供されている以下の 2 つの値のいずれかです。

- 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 との ISPF LIBDEF サービスの使用

QMF インターフェース DSQCIB への動的呼び出しを使用していて、ユーザーの QMF アプリケーションにおいて LIBDEF 関数を使用したい場合は、動的呼び出しを静的呼び出しに変更してください。たとえば、ID 呼び出しステートメント

```
CALL DSQCIB USING ...
```

を次のリテラル呼び出しの形に変更します。

CALL "DSQCIB" USING ...

移行情報

DSQCOMM はバージョン 2.4 とバージョン 3.2 の間で変更されました。

- 古い DSQCOMM を使用し続ける場合、プログラムを再コンパイルする必要はありません。
- バージョン 3.2 の DSQCIB を使用する場合、バージョン 2.4 のプログラムを再びリンク・エディットする必要があります。

新しい DSQCOMM には、START コマンドにエラーがある場合に特に有用なメッセージ・テキストが用意されています。新しい DSQCOMM を使用する場合、プログラムを再コンパイルして、DSQ_COMM_LEVEL (DSQCOMM の) を DSQ_CURRENT_COMM_LEVEL に初期化する必要があります。この値を設定しない場合、QMF は DSQCOMM をバージョン 2.4 のレベルとして扱います。

z/OS での CICS ユーザーへの注

DSQCIB はバージョン 3 リリース 1 モディフィケーション・レベル 1 とバージョン 3 リリース 2 の間で変更されました。QMF が提供する関数呼び出しとメイン QMF プログラム間のインターフェースが、CALL インターフェースから EXEC CICS LINK インターフェースに変更されました。新しいインターフェースは、ユーザー・プログラムと QMF プロダクトからの分離性が高くなっています。インターフェースが変更されたので、呼び出し可能インターフェースを使用したプログラムを再びリンク・エディットする必要があります。

COBOL プログラミングの例

次のプログラム DSQABFCO は QMF プロダクトと共に出荷されます。この例は、VS COBOL II を使用しています。

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。

- z/OS の場合、サンプル・プログラムはライブラリー QMF810.SDSQSAPE のメンバーです。
- VM の場合、サンプル・プログラムはプロダクション・ディスクにあります。
- VSE の場合、サンプル・プログラムは QMF サブライブラリーにあり、名前は DSQABFCO.Z 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。

COBOL 言語インターフェース

- QMF を開始する。
- 3 つのグローバル変数を設定する。
- 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。

この節では、呼び出し可能インターフェースを使用する COBOL プログラムをコンパイルし、リンク・エディットし、実行する方法も示します。この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

```

*****
*   The following is a VS COBOL II 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.

```

図44. DSQABFCO、サンプル COBOL プログラム (1/2)

```

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, S NAMES,
                  QIVLTHS, SVALUES, DSQ-VARIABLE-CHAR.
*
* 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, V NAMES,
                  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.

```

図44. DSQABFCO、サンプル COBOL プログラム (2/2)

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。

COBOL 用の DSQCOMM

この組み込みファイルは DSQCOMMB と呼ばれ、QMF プロダクトと共に出荷されま
す。

```

*****
* COBOL INCLUDE FOR QUERY CALLABLE INTERFACE (MVS/VM)
*****
* STRUCTURE DECLARE FOR COMMUNICATIONS AREA
01 DSQCOMM.
    03 DSQ-RETURN-CODE PIC 9(8) USAGE IS COMP.
    * 03 DSQ-INSTANCE-ID PIC 9(8) USAGE IS COMP.
    * 03 DSQ-COMM-LEVEL PIC X(12).
    * 03 DSQ-PRODUCT PIC X(2).
    * 03 DSQ-PRODUCT-RELEASE PIC X(2).
    * 03 DSQ-RESERVE1 PIC X(28).
    * 03 DSQ-MESSAGE-ID PIC X(8).
    * 03 DSQ-Q-MESSAGE-ID PIC X(8).
    * 03 DSQ-START-PARM-ERROR PIC X(8).
    * 03 DSQ-CANCEL-IND PIC X(1).
    * 03 DSQ-RESERVE2 PIC X(23).
    * 03 DSQ-RESERVE3 PIC X(156).
    * 03 DSQ-MESSAGE-TEXT PIC X(128).
    * 03 DSQ-Q-MESSAGE-TEXT PIC X(128).
    *
    * 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.

```

図 45. DSQCOMM、COBOL 連絡域 (1/2)

* VALUES FOR DSQ-COMM-LEVEL		00052000
		00053000
01 DSQ-CURRENT-COMM-LEVEL	PIC X(12) VALUE "DSQL>001002<".	00054000
		00055000
* VALUES FOR DSQ-PRODUCT		00056000
		00057000
01 DSQ-QRW	PIC X(2) VALUE "01".	00058000
01 DSQ-QMF	PIC X(2) VALUE "02".	00059000
01 DSQ-QM4	PIC X(2) VALUE "03".	00060000
		00061000
* VALUES FOR DSQ-PRODUCT-RELEASE		00062000
		00063000
01 DSQ-QRW-V1R2	PIC X(2) VALUE "01".	00064000
01 DSQ-QRW-V1R3	PIC X(2) VALUE "02".	00065000
01 DSQ-QMF-V2R4	PIC X(2) VALUE "01".	00066000
01 DSQ-QMF-V3R1	PIC X(2) VALUE "02".	00067000
01 DSQ-QMF-V3R1M1	PIC X(2) VALUE "03".	00068000
01 DSQ-QMF-V3R2	PIC X(2) VALUE "04".	00069000
01 DSQ-QMF-V3R3	PIC X(2) VALUE "05".	00070000
01 DSQ-QMF-V6R1	PIC X(2) VALUE "06".	00071000
01 DSQ-QM4-V1R1	PIC X(2) VALUE "01".	00072000
		00073000
* VALUES FOR DSQ-CANCEL-INDE		00074000
		00075000
01 DSQ-CANCEL-YES	PIC X(1) VALUE "1".	00076000
01 DSQ-CANCEL-NO	PIC X(1) VALUE "0".	00077000
		00078000
* VALUES FOR MODE		00079000
		00080000
01 DSQ-INTERACTIVE	PIC X(1) VALUE "1".	00081000
01 DSQ-BATCH	PIC X(1) VALUE "2".	00082000
		00083000
* VALUES YES AND NO		00084000
		00085000
01 DSQ-YES	PIC X(1) VALUE "1".	00086000
01 DSQ-NO	PIC X(1) VALUE "2".	00087000
		00088000
* CALLABLE INTERFACE PROGRAM NAME		00089000
		00090000
01 DSQCIB	PIC X(6) VALUE "DSQCIB".	00091000
		00092000
* VALUES FOR VARIABLE TYPE ON CALL PARAMETER		00093000
		00094000
01 DSQ-VARIABLE-CHAR	PIC X(4) VALUE "CHAR".	00095000
01 DSQ-VARIABLE-FINT	PIC X(4) VALUE "FINT".	00096000

図 45. DSQCOMMB、COBOL 連絡域 (2/2)

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プログラムを変換し、コンパイルし、リンク・エディットする場合は、以下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・サブプログラムとして実行されます。COBOL プログラムは、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 CALL する必要があります。

- 引用符またアポストロフィ

COBOL プログラム内では、リテラルを区切るために引用符 (") 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 によって配布される連絡域 DSQCOMMB およびサンプル COBOL プログラム DSQABFCO は、リテラルの区切り文字として引用符を使用しています。インストール先またはユーザーの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、QMF によって配布された 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。

- 通信マクロ DSQCOMMB

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

プログラムのリンク・エディット・フェーズで 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

プログラムを作成した後、実行する前に、必要に応じて、変換し、コンパイルし、リンク・エディットしなければなりません。この節でリストされているプログラムは、必要なステップを示しています。

この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

z/OS での CICS の場合の変換、コンパイル、およびリンク・エディット

次の例は、COBOL をサポートする、CICS 提供のプロシージャ DFHEBTVL を示しています。CICS 内で使用するためにプログラムを変換する方法については、CICS ライブラリーを参照してください。

```

//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=QMF810.SDSQSAPE,DISP=SHR
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE CICSLOAD(DFHECI)
          INCLUDE QMFLOAD(DSQCIB)
          ORDER DFHECI
          ENTRY samCOBOL
          MODE AMODE(31) RMODE(ANY)
          NAME samCOBOL(R)
/*

```

図 46. CICS 変換プログラム、COBOL コンパイラー、リンケージ・エディターの実行用 JCL

VSE での CICS の場合の変換、コンパイル、およびリンク・エディット

183 ページの図 47 の VSE ジョブ制御は、COBOL プログラムを VSE で実行中の CICS にインストールする例です。COBOL プログラムを変換およびコンパイルする方法については、CICS ライブラリーを参照してください。

この例は QMF に含まれており、DSQ3CICO.Z という名前で QMF サプライブラリーに置かれています。


```

* $$ JOB JNM=DSQ3CICO,DISP=D,CLASS=0
// JOB DSQ3CICO   Sample job to Install QMF Callable Interface (COBOL)
* -----
*   Install QMF Callable Interface Example (COBOL)
* -----
// SETPARM VOLID=valid      *-- update valid for syspch
// SETPARM START=rtrk      *-- update start track/block (syspch)
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks (syspch)
* -----
// DLBL   IJSYSPH,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSPCH,,1,0,&START,&SIZE
ASSGN SYSPCH,DISK,VOL=&VOLID,SHR
* Library search chain must contain the QMF, CICS and COBOL sub library
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate callable interface program (COBOL)
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC   DFHECP1$,SIZE=256K,PARM='XOPTS(CICS,QUOTE)'
* $$ SLI MEM=DSQABFCO.Z,S=PRD2.PROD
/*
* -----
* Step 2: Compile callable interface program (COBOL)
* -----
CLOSE SYSPCH,00D
// DLBL   IJSYSIN,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID,SHR
// OPTION NODECK,CATAL
    PHASE DSQABFCO,*
        INCLUDE DFHECI
// EXEC   IGYCRCTL,PARM='SZ(MAX),OBJECT,MAP,RES,NODYNAM,QUOTE,LIB,RENT'
CLOSE SYSIPT,SYSRDR
/*
* -----
* Step 3: Link-edit callable interface program (COBOL)
* -----
// EXEC   LNKEDT,PARM='AMODE=31,RMODE=ANY'
/*
/&
// JOB   RESET
ASSGN SYSIPT,SYSRDR   IF 1A93D, CLOSE SYSIPT,SYSRDR
ASSGN SYSPCH,00D     IF 1A93D, CLOSE SYSPCH,00D
/&
* $$ EOJ

```

図 47. CICS/VSE 変換プログラム、COBOL コンパイラー、およびリンケージ・エディターの実行用のジョブ制御

VM の CMS のもとでのプログラムのコンパイルと実行

次のプログラムは、IBM COBOL 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コンパイルし実行します。

この例の REXX EXEC は QMF では出荷されませんが、ここからコピーして、インストール先に合うように変更することができます。

```

/*****/
/* Compile your COBOL program and run it.          */
/*****/
TRACE off
ADDRESS CMS
/*****/
/* Access COBOL product disk using a program, PRODUCT, that you */
/* write.                                           */
/*****/
"EXEC PRODUCT COBOL"
/*****/
/* Get QMF DSQCOMM into a macro library and set GLOBAL compile */
/* maclibs.                                         */
/*****/
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMMB"
Maclist = "TEMPP VSC2MAC COB2MLIB COB2PLIB DMSSP CMSLIB OSMACRO"
"GLOBAL MACLIB" Maclist
/*****/
/* Compile the program                             */
/*****/
"GLOBAL TXTLIB SCEELKED"
"COBOL2 yourname (LIB RESIDENT LIST RENT DYNAM"

```

図 48. CMS で COBOL をコンパイルし実行するためのプログラム (1/2)

```

/*****
/* Access DB2 Server for VM and initialize database          */
/*****
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"
/*****
/* Access GDDM product disk                                */
/*****
"EXEC PRODUCT GDDM"
/*****
/* Issue Filedefs for QMF product                          */
/*****
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT              */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT                  */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE                */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A (PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE                      */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 (PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE                          */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * (PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"
/*****
/* Provide access to QMF and COBOL program libraries      */
/*****
"GLOBAL LOADLIB DSQDLIB VSC2LOAD"
"GLOBAL TXTLIB VSC2LTXT ADMRLIB ADMPLIB ADMGLIB SCEELKED"
Say "Starting to run COBOL program"
"RUN yourname"
Exit 0

```

図 48. CMS で COBOL をコンパイルし実行するためのプログラム (2/2)

TSO での COBOL プログラムの実行

以下の節には、COBOL コンパイラーとリンケージ・エディターを実行するサンプル JCL、および ISPF を使用する場合と使用しない場合に、TSO 内でコンパイル済みプログラムを実行するためのサンプル・プログラムがあります。

TSO でのコンパイルとリンク・エディット

次のジョブは、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するために、COBOL コンパイラーを使用しています。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、インストール先によって異なること

があります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```
//samCOBOL JOB
//STEP1 EXEC PROC=IGYWCL
//* Provide Access to QMF Communications Macro DSQCOMM
//COBOL.SYSLIB DD DSN=QMF810.SAMPLIB,DISP=SHR
//COBOL.SYSIN DD *
        .
        Your program or copy of QMF sample DSQBFCO
        .
/*
//* Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
        INCLUDE QMFLOAD(DSQCIB)
        ENTRY samCOBOL
        MODE AMODE(31) RMODE(ANY)
        NAME samCOBOL(R)
/*
```

図 49. COBOL コンパイラーとリンケージ・エディターの実行のための JCL

ISPF を使用しない TSO でのプログラムの実行

プログラムが正常にコンパイルされた後、それを実行するには、次の例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 "QMF810.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('QMF810.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets                                     */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM)  DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command            */
/*****/
CALL samCOBOL
EXIT CODE(0)

```

図 50. COBOL コンパイラーとリンケージ・エディターの実行のための JCL

ISPF のもとの TSO でのプログラムの実行

プログラムが正常にコンパイルされた後、それを実行するには、次の例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('QMF810.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF810.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD',+
        'PRDUCT.COB2LIB')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF810.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF810.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF810.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(samCOBOL) NEWAPPL(DSQE)
EXIT CODE(4)

```

図 51. ISPF のもとで TSO でのプログラム実行のための CLIST

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

CICS ユーザーへの注

FORTRAN は CICS のもとでは使えないので、FORTRAN 用の QMF 呼び出し可能インターフェースは、CICS のもとでは機能しません。

FORTRAN 用のインターフェース連絡域マッピング (DSQCOMMF)

DSQCOMMF は FORTRAN 用の DSQCOMM マッピングであり、プロダクトと共に出荷されます。表 18 は、ユーザーが変更してはならない DSQCOMMF の情報を示しています。

表 18. インターフェース連絡域 DSQCOMMF

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。QMF START コマンドを出す前に、DSQ_CURRENT_COMM_LEVEL の値に設定する必要がある
DSQ_PRODUCT	CHARACTER(2)	使用中の IBM 照会プロダクトを識別する
DSQ_PRODUCT_RELEASE	CHARACTER(2)	使用中の照会プロダクトのリリース・レベルを識別する
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

表 18. 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_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。 DSQ_CANCEL_YES CHARACTER(1) DSQ_CANCEL_NO CHARACTER(1)
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 と 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、通信マクロ 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、大文字のストリングです。

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 3 つのコマンド (START および拡張フォーマットの 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、大文字のストリングです。

PNUM

コマンド・キーワード数。整数パラメーター。

KLTH

指定済みの各キーワードの長さ。整数パラメーターまたはパラメーターの配列です。

KWORD

QMF キーワード (単数または複数)。**KLTH** によって指定した長さと同じの、文字または文字の構造です。

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。

QMF は、キーワードが連続記憶域にあり、特殊な区切り文字によって区切られていないと想定します。

VLTH

キーワードに関連する各値の長さ。整数パラメーターまたはパラメーターの配列です。

VALUE

各キーワードに関連する値

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、文字の構造、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。文字値の場合、**QMF** は、値が連続記憶域にあり、特殊な区切り文字によって区切られていないと想定します。

VTTYPE

値ストリング VALUE の QMF データ・タイプ。VTTYPE の値は、通信マクロ DSQCOMMFF に提供されている以下の 2 つの値のいずれかです。

- 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、VTTYPE に指定する必要があります。

FORTRAN プログラミングの例

次のプログラム DSQABFF は、QMF と共に出荷され、VS FORTRAN を使用します。

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。z/OS の場合、サンプル・プログラムはライブラリー QMF810.SDSQSAPE のメンバーです。VM の場合、サンプル・プログラムはプロダクション・ディスクにあります。

FORTRAN 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 つのグローバル変数を設定する。
- 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。

この節では、呼び出し可能インターフェースを使用する FORTRAN プログラムをコンパイルし、リンク・エディットし、実行する方法も示します。この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

```
C*****
C Sample Program: dsqabff
C FORTRAN Version of 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
```

図 52. DSQABFF、サンプル FORTRAN プログラム (1/5)

```

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*****
PROGRAM DSQABFF

C*****
C Include and declare query interface communications area
C*****
INCLUDE (DSQCOMMF)

C*****
C Query interface command lengths and commands
C*****
INTEGER COMMAND_LENGTH
CHARACTER START_QUERY_INTERFACE*5,
+         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)

```

図 52. DSQABFF、サンプル FORTRAN プログラム (2/5)

```

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 /

```

図 52. DSQABFF、サンプル FORTRAN プログラム (3/5)

```

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)

```

図 52. DSQABFF、サンプル FORTRAN プログラム (4/5)

```
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
```

図 52. DSQABFF、サンプル FORTRAN プログラム (5/5)

FORTRAN 用の DSQCOMM

このファイルは DSQCOMM と呼ばれ、QMF と共に出荷されます。

```

C***** 00001000
C      FORTRAN include file for Callable Interface (MVS/VM) 00002000
C***** 00003000
C      Return codes 00004000
      INTEGER DSQ_SUCCESS, DSQ_WARNING, DSQ_FAILURE, DSQ_SEVERE 00005000
      PARAMETER( 00006000
+         DSQ_SUCCESS = 0, 00007000
+         DSQ_WARNING = 4, 00008000
+         DSQ_FAILURE = 8, 00009000
+         DSQ_SEVERE = 16) 00010000
      00011000
C      Communications level 00012000
      CHARACTER DSQ_CURRENT_COMM_LEVEL*12 00013000
      PARAMETER( 00014000
+         DSQ_CURRENT_COMM_LEVEL = 'DSQL>001002<') 00015000
      00016000
C      Query product IDs 00017000
      CHARACTER DSQ_QRW*2, DSQ_QMF*2, DSQ_QM4*2 00018000
      PARAMETER( 00019000
+         DSQ_QRW = '01', 00020000
+         DSQ_QMF = '02', 00021000
+         DSQ_QM4 = '03') 00022000
      00023000
C      Query product release levels 00024000
      CHARACTER DSQ_QRW_V1R2*2, DSQ_QRW_V1R3*2, 00025000
+         DSQ_QMF_V2R4*2, DSQ_QMF_V3R1*2, 00026000
+         DSQ_QMF_V3R1M1*2, DSQ_QMF_V3R2*2, 00027000
+         DSQ_QMF_V3R3*2, DSQ_QMF_V6R1*2, 00028000
+         DSQ_QM4_V1R1*2 00029000
      PARAMETER( 00030000
+         DSQ_QRW_V1R2 = '01', 00031000
+         DSQ_QRW_V1R3 = '02', 00032000
+         DSQ_QMF_V2R4 = '01', 00033000
+         DSQ_QMF_V3R1 = '02', 00034000
+         DSQ_QMF_V3R1M1 = '03', 00035000
+         DSQ_QMF_V3R2 = '04', 00036000
+         DSQ_QMF_V3R3 = '05', 00037000
+         DSQ_QMF_V6R1 = '06', 00038000
+         DSQ_QM4_V1R1 = '01') 00039000
      00040000
C      Host variable types 00041000
      CHARACTER DSQ_VARIABLE_CHAR*4, DSQ_VARIABLE_FINT*4 00042000
      PARAMETER( 00043000
+         DSQ_VARIABLE_CHAR = 'CHAR', 00044000
+         DSQ_VARIABLE_FINT = 'FINT') 00045000
      00046000
C      Cancel indicator 00047000
      CHARACTER DSQ_CANCEL_YES, DSQ_CANCEL_NO 00048000
      PARAMETER( 00049000
+         DSQ_CANCEL_YES = '1', 00050000
+         DSQ_CANCEL_NO = '0') 00051000
      00052000
      CHARACTER DSQCOMM(512) 00053000

```

図 53. DSQCOMM、FORTRAN 連絡域 (1/2)

INTEGER	DSQ_RETURN_CODE, DSQ_INSTANCE_ID	00054000
CHARACTER	DSQ_COMM_LEVEL*12,	00055000
+	DSQ_PRODUCT*2,	00056000
+	DSQ_PRODUCT_RELEASE*2,	00057000
+	DSQ_RESERVE1*28,	00058000
+	DSQ_MESSAGE_ID*8,	00059000
+	DSQ_Q_MESSAGE_ID*8,	00060000
+	DSQ_START_PARM_ERROR*8,	00061000
+	DSQ_CANCEL_IND*1,	00062000
+	DSQ_RESERVE2*23,	00063000
+	DSQ_RESERVE3*156,	00064000
+	DSQ_MESSAGE_TEXT*128,	00065000
+	DSQ_Q_MESSAGE_TEXT*128	00066000
		00067000
EQUIVALENCE	(DSQCOMM(1), DSQ_RETURN_CODE),	00068000
+	(DSQCOMM(5), DSQ_INSTANCE_ID),	00069000
+	(DSQCOMM(9), DSQ_COMM_LEVEL),	00070000
+	(DSQCOMM(21), DSQ_PRODUCT),	00071000
+	(DSQCOMM(23), DSQ_PRODUCT_RELEASE),	00072000
+	(DSQCOMM(25), DSQ_RESERVE1),	00073000
+	(DSQCOMM(53), DSQ_MESSAGE_ID),	00074000
+	(DSQCOMM(61), DSQ_Q_MESSAGE_ID),	00075000
+	(DSQCOMM(69), DSQ_START_PARM_ERROR),	00076000
+	(DSQCOMM(77), DSQ_CANCEL_IND),	00077000
+	(DSQCOMM(78), DSQ_RESERVE2),	00078000
+	(DSQCOMM(101), DSQ_RESERVE3),	00079000
+	(DSQCOMM(257), DSQ_MESSAGE_TEXT),	00080000
+	(DSQCOMM(385), DSQ_Q_MESSAGE_TEXT)	00081000
		00082000
C	Callable Interface Normal and Extended Calls	00083000
	EXTERNAL DSQCIF	00084000
	EXTERNAL DSQCIFE	00085000

図 53. DSQCOMM、FORTRAN 連絡域 (2/2)

VM の CMS のもとでのプログラムのコンパイルと実行

次のプログラムは、VS FORTRAN 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実行します。この例の REXX EXEC は QMF では出荷されませんが、ここからコピーして、インストール先に合うように変更することができます。

```

/*****/
/* Compile your program and run it. */
/*****/
TRACE off
ADDRESS CMS

/*****/
/* Access FORTRAN product disk using a program, PRODUCT, that you */
/* write. */
/*****/
"EXEC PRODUCT FORTRAN"

/*****/
/* Get QMF DSQCOMM into a macro library and set GLOBAL compile */
/* maclibs. */
/*****/
"ERASE TEMPP MACLIB A"
"MACLIB GEN TEMPP DSQCOMM"
Maclist = "TEMPP VSF2PLIB VSF2MLIB DMSSP CMSLIB OSMACRO"
"GLOBAL MACLIB" Maclist

/*****/
/* Compile the program */
/*****/
'FORTVS2 yourname (RENT OPT(0) XREF'

/*****/
/* Access DB2 Server for VM and initialize database */
/*****/
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"

/*****/
/* Access GDDM product disk */
/*****/
"EXEC PRODUCT GDDM"

```

図 54. ユーザーのプログラムをコンパイルし実行する REXX プログラム (1/2)

```

/*****
/* Issue Filedefs for QMF product */
/*****
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A (PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 (PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * (PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQLDLIB LOADLIB *"

/*****
/* Provide access to QMF and FORTRAN program libraries */
/*****
'GLOBAL LOADLIB VSF2LOAD DSQLDLIB'
'GLOBAL TXTLIB VSF2LINK VSF2FORT ADMRLIB ADMPLIB ADMGLIB'
Say "Starting to run FORTRAN program"
"RUN yourname"

Exit 0

```

図 54. ユーザーのプログラムをコンパイルし実行する REXX プログラム (2/2)

このプログラムは、インストール先に合うように変更する必要がある場合があります。

z/OS の TSO のもとでのプログラムの実行

プログラムを作成した後、実行する前に必要に応じてコンパイルし、リンク・エディットしなければなりません。この節にリストしたプログラムは、このために必要なステップを示しています。

この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

TSO でのコンパイルとリンク・エディット

次のジョブは、MVS 用の VS FORTRAN 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、インストール先によって異なることがあります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```
//samFORT    JOB
//STEP1      EXEC PROC=VSF2CL
//* Provide Access to QMF Communications Macro DSQCOMM
//FORT.SYSLIB DD DSN=QMF810.SAMPLIB,DISP=SHR
//FORT.SYSIN  DD *
              .
              Your program or copy of QMF sample DSQABFF
              .
/*
//* Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN   DD *
              INCLUDE QMFLOAD(DSQCIF)
              INCLUDE QMFLOAD(DSQCIFE)
              ENTRY  samFORT
              MODE   AMODE(31) RMODE(ANY)
              NAME   samFORT(R)
/*
```

図 55. FORTRAN コンパイラーとリンケージ・エディター実行のための JCL

ISPF を使用しない TSO でのプログラムの実行

次のプログラムは、VS FORTRAN 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実行します。いくつかのパラメーターは、インストール先によって異なることがあります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```

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 "QMF810.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('QMF810.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets                                       */
/*****/
ALLOC FI(ADMGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command              */
/*****/
CALL samFORT
EXIT CODE(0)

```

図 56. ISPF を使用しない TSO でのプログラム実行のための CLIST

ISPF のもとの TSO での実行

次のプログラムは、VS FORTRAN 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実行します。いくつかのパラメーターは、インストール先に

よって異なることがあります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```

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 UDUMPCB 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('QMF810.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****
/* Datasets used by ISPF */
/*****
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF810.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PRODUCT.VSF2LOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF810.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF810.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF810.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* QMF/GDDM Datasets */
/*****
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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(UDUMPCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF810.DSQPNLE') SHR REUSE
/*****
/* Start your program as the initial ISPF dialog */
/*****
ISPSTART PGM(samFORT) NEWAPPL(DSQE)
EXIT CODE(4)

```

図 57. ISPF のもとの TSO でのプログラム実行のための CLIST

EXIT CODE(4) は、ISPF 後処理パネルの表示を抑止します。

PL/I 言語インターフェース

CICS で QMF を使用するために必要な PL/I のリリース・レベルは、PL/I バージョン 2 以降です。PL/I バージョン 2 は VSE/ESA ではサポートされません。

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、プロダクトと共に出荷されます。表 19 は、DSQCOMML の各値を示しています。

表 19. 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 のレベルを識別する。QMF START コマンドを出す前に、DSQ_CURRENT_COMM_LEVEL の値に設定する必要がある
DSQ_PRODUCT	CHAR(2)	使用中の IBM 照会プロダクトを識別する。
DSQ_PRODUCT_RELEASE	CHAR(2)	使用中の照会プロダクトのリリース・レベルを識別する
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。 • 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。

表 19. 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_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、コミュニケーション・マクロ 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、大文字のストリングです。

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 3 つのコマンド (START および拡張フォーマットの 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。これは、整数 FIXED BIN(31) パラメーターです。

CMDSTR

実行する QMF コマンド。CMDLTH によって指定した長さの、大文字のストリングです。

PNUM

コマンド・キーワード数。これは、整数 FIXED BIN(31) パラメーターです。

KLTH

指定済みの各キーワードの長さ。これは、整数 FIXED BIN(31) パラメーターまたはパラメーターの配列です。

KEYWORD

QMF キーワード (単数または複数)。それぞれ、KLTH によって指定した長さと同じの、文字または文字の構造です。

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。

QMF は、キーワードが連続記憶域にあり、特殊な区切り文字によって区切られていないと想定します。

VLTH

キーワードに関連する各値の長さ。これは、整数 FIXED BIN(31) パラメーターまたはパラメーターの配列です。

VALUE

各キーワードに関連する値。

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、文字の構造、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。文字値の場合、QMF は、値が連続記憶域にあり、特殊な区切り文字によって区切られていないと想定します。

VTTYPE

値ストリング VALUE の QMF データ・タイプ。VTTYPE の値は、DSQCOMML 通信マクロに提供されている以下の 2 つの値のいずれかです。

- 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、VTTYPE に指定する必要があります。

PL/I プログラミングの例

次のサンプル・プログラム DSQABFP は、QMF と共に出荷され、IBM PL/I を使用します。

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。

- VM の場合、サンプル・プログラムはプロダクション・ディスクにあります。
- z/OS の場合、サンプル・プログラムはライブラリー QMF810.SDSQSAPE のメンバーです。
- CICS で QMF を使用するために必要な PL/I のリリース・レベルは、バージョン 2 以降です。PL/I バージョン 2 は VSE/ESA ではサポートされません。

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 つのグローバル変数を設定する。
- 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。

この節では、呼び出し可能インターフェースを使用する PL/I プログラムをコンパイルし、リンク・エディットし、実行する方法も示します。この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

```

DSQABFP: PROCEDURE OPTIONS(MAIN REENTRANT) REORDER;                00001000
/*****/ 00002000
/* Sample Program: DSQABFP */ 00003000
/* PL/I Version of the QMF Callable Interface */ 00004000
/*****/ 00005000
00006000
/*****/ 00007000
/* Include and declare query interface communications area */ 00008000
/*****/ 00009000
%INCLUDE SYSLIB(DSQCOMML); 00010000
00011000
/*****/ 00012000
/* Builtin function */ 00013000
/*****/ 00014000
DCL LENGTH BUILTIN; 00015000
00016000
/*****/ 00017000
/* Query interface command length and commands */ 00018000
/*****/ 00019000
DCL COMMAND_LENGTH FIXED BIN(31); 00020000
DCL START_QUERY_INTERFACE CHAR(5) INIT('START'); 00021000
DCL SET_GLOBAL_VARIABLES CHAR(10) INIT('SET GLOBAL'); 00022000
DCL RUN_QUERY CHAR(12) INIT('RUN QUERY Q1'); 00023000
DCL PRINT_REPORT CHAR(22) INIT('PRINT REPORT (FORM=F1)'); 00024000
DCL END_QUERY_INTERFACE CHAR(4) INIT('EXIT'); 00025000
00026000
/*****/ 00027000
/* Query command extension, number of parameters and lengths */ 00028000
/*****/ 00029000
DCL NUMBER_OF_PARAMETERS FIXED BIN(31);/* number of variables */ 00030000
DCL KEYWORD_LENGTHS(10) FIXED BIN(31);/* lengths of keyword names*/ 00031000
DCL DATA_LENGTHS(10) FIXED BIN(31);/* lengths of variable data*/ 00032000
00033000

```

図 58. DSQABFP、サンプル PL/I プログラム (1/3)

```

/*****/ 00034000
/* Keyword parameter and value for START command */ 00035000
/*****/ 00036000
DCL START_KEYWORDS CHAR(8) INIT('DSQSMODE'); 00037000
DCL START_KEYWORD_VALUES CHAR(11) INIT('INTERACTIVE'); 00038000
00039000
/*****/ 00040000
/* Keyword parameter and value for SET command */ 00041000
/*****/ 00042000
DCL 1 SET_KEYWORDS, 00043000
    3 SET_KEYWORDS_1 CHAR(7) INIT('MYVAR01'), 00044000
    3 SET_KEYWORDS_2 CHAR(5) INIT('SHORT'), 00045000
    3 SET_KEYWORDS_3 CHAR(7) INIT('MYVAR03'); 00046000
00047000
DCL 1 SET_VALUES, 00048000
    3 SET_VALUES_1 FIXED BIN(31), 00049000
    3 SET_VALUES_2 FIXED BIN(31), 00050000
    3 SET_VALUES_3 FIXED BIN(31); 00051000
00052000
/*****/ 00053000
/* Main program */ 00054000
/*****/ 00055000
DSQCOMM = ''; 00056000
DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL; 00057000
00058000
/*****/ 00059000
/* Start a query interface session */ 00060000
/*****/ 00061000
NUMBER_OF_PARAMETERS = 1; 00062000
COMMAND_LENGTH = LENGTH(START_QUERY_INTERFACE); 00063000
KEYWORD_LENGTHS(1) = LENGTH(START_KEYWORDS); 00064000
DATA_LENGTHS(1) = LENGTH(START_KEYWORD_VALUES); 00065000
00066000
CALL DSQCIPX(DSQCOMM, 00067000
    COMMAND_LENGTH, 00068000
    START_QUERY_INTERFACE, 00069000
    NUMBER_OF_PARAMETERS, 00070000
    KEYWORD_LENGTHS, 00071000
    START_KEYWORDS, 00072000
    DATA_LENGTHS, 00073000
    START_KEYWORD_VALUES, 00074000
    DSQ_VARIABLE_CHAR); 00075000
00076000

```

図 58. DSQABFP、サンプル PL/I プログラム (2/3)

```

/*****/ 00077000
/* Set numeric values into query using SET command */ 00078000
/*****/ 00079000
NUMBER_OF_PARAMETERS = 3; 00080000
COMMAND_LENGTH = LENGTH(SET_GLOBAL_VARIABLES); 00081000
KEYWORD_LENGTHS(1) = LENGTH(SET_KEYWORDS_1); 00082000
KEYWORD_LENGTHS(2) = LENGTH(SET_KEYWORDS_2); 00083000
KEYWORD_LENGTHS(3) = LENGTH(SET_KEYWORDS_3); 00084000
DATA_LENGTHS(1) = 4; 00085000
DATA_LENGTHS(2) = 4; 00086000
DATA_LENGTHS(3) = 4; 00087000
SET_VALUES_1 = 20; 00088000
SET_VALUES_2 = 40; 00089000
SET_VALUES_3 = 84; 00090000
00091000
CALL DSQCIPX(DSQCOMM, 00092000
              COMMAND_LENGTH, 00093000
              SET_GLOBAL_VARIABLES, 00094000
              NUMBER_OF_PARAMETERS, 00095000
              KEYWORD_LENGTHS, 00096000
              SET_KEYWORDS, 00097000
              DATA_LENGTHS, 00098000
              SET_VALUES, 00099000
              DSQ_VARIABLE_FINT); 00100000
00101000
/*****/ 00102000
/* Run a Query */ 00103000
/*****/ 00104000
COMMAND_LENGTH = LENGTH(RUN_QUERY); 00105000
00106000
CALL DSQCIPL(DSQCOMM, 00107000
             COMMAND_LENGTH, 00108000
             RUN_QUERY); 00109000
00110000
/*****/ 00111000
/* Print the results of the query */ 00112000
/*****/ 00113000
COMMAND_LENGTH = LENGTH(PRINT_REPORT); 00114000
00115000
CALL DSQCIPL(DSQCOMM, 00116000
             COMMAND_LENGTH, 00117000
             PRINT_REPORT); 00118000
00119000
/*****/ 00120000
/* End the query interface session */ 00121000
/*****/ 00122000
COMMAND_LENGTH = LENGTH(END_QUERY_INTERFACE); 00123000
00124000
CALL DSQCIPL(DSQCOMM, 00125000
             COMMAND_LENGTH, 00126000
             END_QUERY_INTERFACE); 00127000
00128000
END DSQABFP; 00129000

```

図 58. DSQABFP、サンプル PL/I プログラム (3/3)

PL/I 用の DSQCOMM

```

/*****/ 00001000
/* PL/I include for Query Callable Interface (MVS/VM) */ 00002000
/*****/ 00003000
00004000
/* Structure declare for Communications Area */ 00005000
DCL
1 DSQCOMM, 00007000
3 DSQ_RETURN_CODE FIXED BIN(31), /* function return code */ 00008000
3 DSQ_INSTANCE_ID FIXED BIN(31), /* start ID */ 00009000
3 DSQ_COMM_LEVEL CHAR(12), /* communications level */ 00010000
3 DSQ_PRODUCT CHAR(2), /* query product id */ 00011000
3 DSQ_PRODUCT_RELEASE CHAR(2), /* query product release */ 00012000
3 DSQ_RESERVE1 CHAR(28), /* reserved */ 00013000
3 DSQ_MESSAGE_ID CHAR(8), /* completion message ID */ 00014000
3 DSQ_Q_MESSAGE_ID CHAR(8), /* query message ID */ 00015000
3 DSQ_START_PARM_ERROR CHAR(8), /* start parms in error */ 00016000
3 DSQ_CANCEL_IND CHAR(1), /* cmd cancel indicator */ 00017000
/* 1 = cancelled, 0 = not cancelled*/ 00018000
3 DSQ_RESERVE2 CHAR(23), /* reserved */ 00019000
3 DSQ_RESERVE3 CHAR(156), /* reserved */ 00020000
3 DSQ_MESSAGE_TEXT CHAR(128), /* QMF command message */ 00021000
3 DSQ_Q_MESSAGE_TEXT CHAR(128); /* QMF query message */ 00022000
00023000
/* Return Codes */ 00024000
DCL 00025000
DSQ_SUCCESS FIXED BIN(31) INIT(0) STATIC, 00026000
DSQ_WARNING FIXED BIN(31) INIT(4) STATIC, 00027000
DSQ_FAILURE FIXED BIN(31) INIT(8) STATIC, 00028000
DSQ_SEVERE FIXED BIN(31) INIT(16) STATIC; 00029000
00030000
/* Communications Level */ 00031000
DCL 00032000
DSQ_CURRENT_COMM_LEVEL CHAR(12) INIT('DSQL>001002<') STATIC; 00033000
00034000
/* Query Product ID */ 00035000
DCL 00036000
DSQ_QRW CHAR(2) INIT('01') STATIC, 00037000
DSQ_QMF CHAR(2) INIT('02') STATIC, 00038000
DSQ_QM4 CHAR(2) INIT('03') STATIC; 00039000
00040000

```

図 59. DSQCOMM、PL/I 連絡域 (1/2)

```

/* Query Product Release ID                                */ 00041000
DCL                                                         00042000
    DSQ_QRW_V1R2          CHAR(2) INIT('01') STATIC,      00043000
    DSQ_QRW_V1R3          CHAR(2) INIT('02') STATIC,      00044000
    DSQ_QMF_V2R4          CHAR(2) INIT('01') STATIC,      00045000
    DSQ_QMF_V3R1          CHAR(2) INIT('02') STATIC,      00046000
    DSQ_QMF_V3R1M1       CHAR(2) INIT('03') STATIC,      00047000
    DSQ_QMF_V3R2          CHAR(2) INIT('04') STATIC,      00048000
    DSQ_QMF_V3R3          CHAR(2) INIT('05') STATIC,      00049000
    DSQ_QMF_V6R1          CHAR(2) INIT('06') STATIC,      00050000
    DSQ_QM4_V1R1          CHAR(2) INIT('01') STATIC;      00051000
                                                         00052000

/* Cancelled Indicator                                    */ 00053000
DCL                                                         00054000
    DSQ_CANCEL_YES       CHAR(1) INIT('1') STATIC,        00055000
    DSQ_CANCEL_NO        CHAR(1) INIT('0') STATIC;        00056000
                                                         00057000

/* Variable Types                                        */ 00058000
DCL                                                         00059000
    DSQ_VARIABLE_CHAR    CHAR(4) INIT('CHAR') STATIC,     00060000
    DSQ_VARIABLE_FINT    CHAR(4) INIT('FINT') STATIC;     00061000
                                                         00062000

/* Mode                                                  */ 00063000
DCL                                                         00064000
    DSQ_INTERACTIVE      CHAR(1) INIT('1') STATIC,        00065000
    DSQ_BATCH            CHAR(1) INIT('2') STATIC;        00066000
                                                         00067000

/* Yes or No                                             */ 00068000
DCL                                                         00069000
    DSQ_YES              CHAR(1) INIT('1') STATIC,        00070000
    DSQ_NO               CHAR(1) INIT('2') STATIC;        00071000
                                                         00072000

/* Query Interface Entry Point                          */ 00073000
DCL                                                         00074000
    DSQCIPL ENTRY (*,                                     /* interface block */ 00075000
                  FIXED BIN(31),                         /* length of command */ 00076000
                  CHAR(*))                               /* command string */ 00077000
                  EXTERNAL OPTIONS(ASSEMBLER);          00078000
                                                         00079000
DCL
    DSQCIPX ENTRY (*,                                     /* interface block */ 00080000
                  FIXED BIN(31),                         /* length of command */ 00081000
                  CHAR(*),                               /* command string */ 00082000
                  FIXED BIN(31),                         /* # of command keywords */ 00083000
                  *,                                     /* length of keyword */ 00084000
                  *,                                     /* keyword string */ 00085000
                  *,                                     /* length of value */ 00086000
                  *,                                     /* value of keyword */ 00087000
                  CHAR(4))                               /* data type of value */ 00088000
                  EXTERNAL OPTIONS(ASSEMBLER);          00089000

```

図 59. DSQCOMML、PL/I 連絡域 (2/2)

CICS のもとでのプログラムの実行

プログラムは、作成した後、実行するにはコンパイルする必要があります。この節でリストされている例は、必要なステップを示しています。

この例の REXX EXEC、JCL、または CLIST は QMF では出荷されませんが、ここからコピーしてインストール先に合うように変更することができます。

z/OS での CICS のもとでの変換、コンパイル、およびリンク・エディット

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し、コンパイルし、リンク・エディットする場合は、以下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・フェーズで 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。

次の例は CICS 提供のプロシージャ DFHEBTPL を使用しています。 .

```
//samPLI    JOB
//          EXEC PROC=DFHEBTPL
//TRN.SYSIN DD *
*PROCESS   XOPTS(CICS translator options ..... )
           .
           Your program or copy of QMF sample DSQABFP
           .
/*
/** Provide Access to QMF Communications Macro DSQCOMML
//PLI.SYSLIB DD DSN=QMF810.SDSQSAPE,DISP=SHR
/** Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN  DD *
              INCLUDE CICSLOAD(DFHPL10I)
              INCLUDE CICSLOAD(DFHEPI)
              INCLUDE QMFLOAD(DSQCIPL)
              INCLUDE QMFLOAD(DSQCIPX)
              ORDER DFHPL10I,DFHEPI
              ENTRY  sampPLI
              MODE   AMODE(31) RMODE(ANY)
              NAME   sampPLI(R)
/*
```

図 60. CICS 変換プログラム、PL/I コンパイラー、リンケージ・エディター実行のための JCL

VSE での CICS のもとでの変換、コンパイル、およびリンク・エディット

図 61 の VSE ジョブ制御は、PL/I プログラムを、VSE で実行中の CICS にインストールする例です。この例は QMF に含まれており、DSQ3CIP.Z という名前で QMF サブライブラリーに置かれています。詳細については、「CICS (VSE/ESA 版) システム定義の手引き」および「PL/I VSE プログラミングの手引き」を参照してください。

```

..* $$ JOB JNM=DSQ3CIP,DISP=D,CLASS=0
// JOB DSQ3CIP      Sample job to Install QMF Callable Interface (PL/I)
* -----
* Install QMF Callable Interface Example (PL/I)
* -----
// SETPARM VOLID=valid      *-- update valid for syspch
// SETPARM START=rtrk      *-- update start track/block (syspch)
// SETPARM SIZE=ntrks      *-- update number of tracks/blocks (syspch)
* -----
// DLBL  IJSYSPH,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSPCH,,1,0,&START,&SIZE
ASSGN SYSPCH,DISK,VOL=&VOLID,SHR
* Library search chain must contain the QMF, CICS and PL/I sublibrary
// LIBDEF *,SEARCH=(PRD2.PROD,PRD1.BASE,PRD2.CONFIG)
// LIBDEF PHASE,CATALOG=PRD2.PROD
* -----
* Step 1: Translate callable interface program (PL/I)
* -----
* You may need to update or remove the SLI statement for your program.
* -----
// EXEC  DFHEPP1$,SIZE=256K,PARM='XOPTS(CICS)'
..* $$ SLI MEM=DSQABFP.Z,S=PRD2.PROD
/*
* -----
* Step 2: Compile callable interface program (PL/I)
* -----
CLOSE SYSPCH,00D
// DLBL  IJSYSIN,'CICS.TRANSLAT.OUTPUT',0
// EXTENT SYSIPT
ASSGN SYSIPT,DISK,VOL=&VOLID,SHR
// OPTION NODECK,CATAL
    PHASE DSQABFP,*
        INCLUDE DFHPL1I
// EXEC PLIOPT
CLOSE SYSIPT,SYSRDR
/*

```

図 61. VSE 用 JCL のサンプル (1/2)

```
* -----  
* Step 3: Link-edit callable interface program (PL/I)  
* -----  
// EXEC LNKEDT,PARM='AMODE=31,RMODE=ANY'  
/*  
/&  
// JOB RESET  
ASSGN SYSIPT,SYSRDR      IF 1A93D, CLOSE SYSIPT,SYSRDR  
ASSGN SYSPCH,00D        IF 1A93D, CLOSE SYSPCH,00D  
/&  
..* $$ E0J
```

図 61. VSE 用 JCL のサンプル (2/2)

VM の CMS のもとでのプログラムのコンパイルと実行

以下のプログラムは、PL/I 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コンパイルし実行します。

この例の REXX EXEC は QMF では出荷されませんが、ここからコピーして、インストール先に合うように変更することができます。

```
/******  
/* Compile QMF PL/I program and run it. */  
/******  
TRACE off  
ADDRESS CMS  
  
/******  
/* Access PL/I product disk using a program, PRODUCT, that you */  
/* write. */  
/******  
"EXEC PRODUCT PLIV"  
  
/******  
/* Get QMF DSQCOMM into a macro library and set GLOBAL compile */  
/* maclibs. */  
/******  
"ERASE TEMPP MACLIB A"  
"MACLIB GEN TEMPP DSQCOMML"  
MacList = "TEMPP PLICOMP DMSSP CMSLIB OSMACRO"  
"GLOBAL MACLIB" MacList  
  
/******  
/* Compile the program */  
/******  
POPTS = '(INC SOURCE LIST LMSG M NAG NC(E) NIS NOESD NSTG OPT(2)'  
'PLIOPT' yourname popts  
  
/******  
/* Access DB2 Server for VM and initialize database */  
/******  
"EXEC PRODUCT SQLDS"  
"EXEC SQLINIT DBNAME(SQLDBA)"  
  
/******  
/* Access GDDM product disk */  
/******  
"EXEC PRODUCT GDDM"
```

図 62. ユーザーのプログラムをコンパイルし実行する REXX プログラム (1/2)

```
/* Issue Filedefs for QMF product */
FILEDEF ISPLLIB CLEAR
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
FILEDEF DSQPRINT PRINTER ( LRECL 133 BLKSIZE 133 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
FILEDEF DSQEDIT DISK QMFEDIT FILE A (PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
FILEDEF DSQSPILL DISK DSQSIDE DATA A1 (PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
FILEDEF DSQPNLE DISK DSQPNLE FILE * (PERM"
FILEDEF ISPLLIB CLEAR
FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *

/* Provide access to QMF and PL/I program libraries */
GLOBAL MACLIB TEMPP'
GLOBAL LOADLIB DSQDLIB PLILIB'
GLOBAL TXTLIB PLILIB IBMLIB ADMRLIB ADMPLIB ADMGLIB'

Say "Starting to run PL/I program"
"RUN yourname"

Exit 0
```

図 62. ユーザーのプログラムをコンパイルし実行する REXX プログラム (2/2)

このプログラムは、インストール先に合うように変更する必要がある場合があります。

TSO でのコンパイルとリンク・エディット

次のジョブは、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、インストール先によって異なることがあります。詳細については、インストール先の QMF 管理者にお問い合わせください。

```
//samPLI      JOB
//STEP1       EXEC IEL1CL
//* Provide Access to QMF Communications Macro DSQCOMML
//PLI.SYSLIB   DD DSN=QMF810.SAMPLIB,DISP=SHR
//PLI.SYSIN    DD *
              .
              Your program or copy of QMF sample DSQABFP
              .
/*
/* Provide Access to QMF Interface Module
//LKED.QMFLOAD DD DSN=QMF810.SDSQLOAD,DISP=SHR
//LKED.SYSIN   DD *
              INCLUDE QMFLOAD(DSQCIPL)
              INCLUDE QMFLOAD(DSQCIPX)
              ENTRY  sampPLI
              MODE   AMODE(31) RMODE(ANY)
              NAME   sampPLI(R)
/*
```

図 63. PL/I コンパイラーとリンケージ・エディター実行のための JCL

ISPF を使用しない TSO での実行

TSO 環境用にプログラムをコンパイルした後、次の CLIST によってプログラムを実行します。

```
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 "QMF810.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('QMF810.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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('QMF810.DSQPNLE') SHR REUSE
/*****/
/* Start your program using TSO CALL command */
/*****/
CALL sampPLI
EXIT CODE(0)
```

図 64. ISPF を使用しない TSO でのプログラム実行のための CLIST

ISPF のもとの TSO での実行

TSO 環境用にプログラムをコンパイルした後、次の CLIST によってプログラムを実行します。

```

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('QMF810.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF810.SDSQEXCE')
/*****/
/* Datasets used by ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF810.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PLI.PLILINK','PLI.SIBMLINK')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF810.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF810.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF810.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* QMF/GDDM Datasets */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF810.QMFMAPS') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF810.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF810.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(DSQPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF810.DSQPNLE') SHR REUSE
/*****/
/* Start your program as the initial ISPF dialog */
/*****/
ISPSTART PGM(sampPLI) NEWAPPL(DSQE)
EXIT CODE(4)

```

図 65. ISPF のもとの TSO でのプログラム実行のための CLIST

EXIT CODE(4) は ISPF 後処理パネルを押しします。

REXX 言語インターフェース

CICS ユーザーへの注

REXX は、QMF CICS のもとでは使えませんので REXX 用の QMF 呼び出し可能インターフェースは、CICS のもとでは機能しません。

REXX は解釈言語なので、コンパイルする必要がありません。しかし、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して作成したプログラムに比べて、解釈 REXX を使用して作成した同様のプログラムのパフォーマンスは劣ります。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。

TSO のもとで、REXX 呼び出し可能インターフェースを使用して QMF を呼び出すことによって、ロジックを持つプロシージャーおよび特定の書式機能（計算、定義列、および条件）を使用すれば、REXX サービスを使用するために必要なリソースを削減することができます。これらのすべての機能は、REXX を使用します。

たとえば、REXX 呼び出し可能インターフェースを使用して QMF セッションを開始すれば、REPORT パネルで PRINT REPORT または BOTTOM を実行するために必要なリソースは少なくて済みます。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。詳細については、37 ページの『第 5 章 ADDRESS QRW: QMF コマンド環境の使用』を参照してください。

REXX 用のインターフェース連絡変数

連絡変数は、以下の REXX 変数から構成されます。これらの変数は、各呼び出しの完了後に設定されます。

223 ページの表 20 は、呼び出し側プログラムで変更してはならないインターフェース連絡変数を示しています。

表 20. 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。可能な値は、以下のとおり。</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 重大エラー。QMF セッションは終了したので、QMF に対する追加の呼び出しを、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</p> <p>dsq_return_code の値は、REXX 変数 <i>rc</i> 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>使用中の照会マネージャー・プロダクト。可能な値は以下のとおり。</p> <p>dsq_qmf QMF</p> <p>dsq_qm4 OS/400 Query 管理機能</p>
dsq_product_release	<p>使用中の照会プロダクトのリリース・レベル。可能な値は以下のとおり。</p> <ul style="list-style-type: none"> • OS/400: <ul style="list-style-type: none"> dsq_qm4_v1r4 バージョン 1 リリース 1 • QMF: <ul style="list-style-type: none"> dsq_qmf_v3r3 QMF バージョン 3 リリース 3 dsq_qmf_v6r1 QMF バージョン 6 dsq_qmf_v7r2 QMF バージョン 7 リリース 2 dsq_qmf_v8r1 QMF バージョン 8 リリース 1
dsq_message_id	完了メッセージ ID
dsq_q_message_id	照会メッセージ ID

表 20. REXX 用のインターフェース連絡変数 (続き)

構造名	説明
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	コマンド取り消し標識。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。可能な値は以下のとおり。 dsq_cancel_yes ユーザーがコマンドを取り消した。 dsq_cancel_no ユーザーがコマンドを取り消さなかった。
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、すべての SAA 照会コマンドを実行するために使用できる、DSQCIX と呼ばれる外部サブルーチンが用意されています。

DSQCIX 線形構文

```
call DSQCIX cmd parmlist
```

- *cmd* は大文字の文字列として書く QMF コマンドです。
- *parmlist* は、以下の図に示すようなパラメーターと値のペアのリストです。



parmlist を含むコマンド全体を、文字列として書いた単一の REXX 変数として QMF に渡す必要があります。この文字列は、引用符 (') または (") で囲む必要があります。コマンド・文字列の一部として REXX 変数を使用する場合には、引き数を囲まないでください。以下に例を示します。

```
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

例:

```
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。

パラメーターの間のコンマ (,)

1 つのスペースでもかまいません。

右括弧 ())

必要ありません。

parmname と 値 の間の等号 (=)

1 つのスペースでもかまいません。

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

次のプログラム DSQABFX は QMF と共に出荷されます。

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、z/OS の場合は、ライブラリー QMF810.SDSQEXCE のメンバーであり、VM の場合はプロダクション・ディスクにあります。REXX は、QMF CICS では使用できません。

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 つのグローバル変数を設定する。
- 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
```

図 66. DSQABFX、サンプル REXX プログラム

VM の CMS のもとでのプログラムの実行

次のプログラムは、REXX CALL インターフェース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。

このプログラムは、インストール先に合うように変更する必要がある場合があります。

```

/***** */
/* Access DB2 Server for VM and initialize database */
/***** */
"EXEC PRODUCT SQLDS"
"EXEC SQLINIT DBNAME(SQLDBA)"
*/
*/

/***** */
/* Access GDDM product disk */
/***** */
"EXEC PRODUCT GDDM"
*/
*/

/***** */
/* Issue Filedefs for QMF product */
/***** */
/* DEBUG = DDNAME FOR QMF DIAGNOSTICS OUTPUT */
"FILEDEF DSQDEBUG PRINTER ( LRECL 80 BLKSIZE 80 RECFM FBA PERM"
/* PRINT = DDNAME FOR QMF PRINTED OUTPUT */
"FILEDEF DSQPRINT PRINTER ( LRECL 121 BLKSIZE 121 RECFM FBA PERM"
/* EDIT = DDNAME FOR QMF EDIT TRANSFER FILE */
"FILEDEF DSQEDIT DISK QMFEDIT FILE A (PERM"
/* DSQSIDE = DDNAME FOR QMF SPILL FILE */
"FILEDEF DSQSPILL DISK DSQSIDE DATA A1 (PERM"
/* DSQPNLE = DDNAME FOR PANEL FILE */
"FILEDEF DSQPNLE DISK DSQPNLE FILE * (PERM"
"FILEDEF ISPLLIB CLEAR"
"FILEDEF ISPLLIB DISK DSQDLIB LOADLIB *"

/***** */
/* Provide access to QMF and GDDM program libraries */
/***** */
"GLOBAL LOADLIB DSQDLIB "
"GLOBAL TXTLIB ADMRLIB ADMPLIB ADMGLIB"

/* The beginning of your REXX program ..... */
.
.
.
.
/* The end of your REXX program ..... */
*/

```

図 67. CMS でプログラムを実行する REXX プログラム

z/OS の TSO のもとでのプログラムの実行

ユーザーの REXX プログラムを実行するには、次の例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('QMF810.DSQCCLSTE',' ",
    "'DSN.DSNCLIST')"
```

```

"ALLOC FI(SYSEXEC) SHR REUSE ",
  "DA('QMF810.SDSQEXCE')"
```

```

"ALLOC FI(ISPLLIB) SHR REUSE ",
  "DA('QMF810.SDSQLOAD',' ",
    "'ADM.GDDM.GDDMLoad',' ",
    "'DSN.DSNLOAD')"
```

```

"ALLOC FI(DSQPNLE) DA('QMF810.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('QMF810.DSQMAPE') SHR REUSE"
"ALLOC FI(ADMCFORM) DA('QMF810.DSQCHART') SHR REUSE"
"ALLOC FI(DSQCFRM) DA('QMF810.DSQCFRM') 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 ..... */
```

図 68. TSO でプログラムを実行する REXX プログラム

INTERACT ループを使用する REXX の例

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。

次のロジックをプログラムに追加することによって、呼び出し可能インターフェース・プログラムから INTERACT コマンドによって開始した対話式セッションで END コマンドを、END が対話式 QMF で機能しているかのように機能させることができます。

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 */

exit dsq_return_code

```

図 69. INTERACT ループを使用する REXX プログラム

付録 B. エクスポート / インポート・フォーマット

本章では、データの QMF フォーマットについて説明し、以下の各エンコード・フォーマット・オブジェクトの表番号とフィールド番号をリストします。

- 書式
- 指示照会
- 報告書

これらのリストの説明と例については、79 ページの『第 9 章 QMF オブジェクトのインポートとエクスポート』を参照してください。

データの QMF フォーマット

EXPORT コマンド (DATAFORMAT=QMF) 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するヘッダー・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

外部データ・ファイルのレコード長は、233 ページの『データ・レコード』で説明されているように、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・レコードも、この長さに分割されます。表 21 は、ヘッダー・レコードに含まれる情報を示しています。

表 21. ヘッダー・レコード情報

バイト位置	情報とタイプ
1-8	QMF オブジェクト・フォーマット・レベル (8 文字のデータ) QMF バージョン 8.1 のオブジェクト・レベルは、「QMF 3.0」です。それ以前の QMF リリースのオブジェクト・レベルは、すべて「QMF 1.0」です。QMF バージョン 8.1 を使用してエクスポートされたデータは、以前の QMF リリースにはインポートできません。
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、37-54、...	列名 (30 文字のデータ)
31-32、55-56、...	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、232 ページの表 22 参照。

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
33-34、57-58、...	列幅 (ハーフワード符号付き整数) - 多くのデータ・タイプでは、バイト数で表した列幅。例外は以下のとおりです。 <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。
35、59、...	ヌルの許可。ヌル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、60、...	未使用バイト

バイト 11-12 は、列数を示します。これは、バイト 13-36 の情報が、ヘッダー・レコード内の列ごとに反復されることを意味します。ヘッダー・レコード内の列ごとに 24 バイトが必要です。

データ・タイプ・コードについては、表 22 に記載されています。

表 22. 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	可変文字
X'1C4'	452	CHAR	固定文字
X'1D0'	464	VARGRAPHIC	可変図形
X'1D4'	468	GRAPHIC	固定図形
X'1E0'	480	FLOAT	浮動小数点
X'1E4'	484	DECIMAL	10 進数
X'1F0'	496	INTEGER	整数
X'1F4'	500	SMALLINT	小整数

日付、時刻、およびタイム・スタンプのデータは常に ISO フォーマットでエクスポートされます。

データ・レコード

データ・レコードは固定長フォーマットであり、エクスポートされるデータが入っています。データ・レコードの最大長は 7,000 バイトです。データ・レコードの長さは、レコードを構成しているデータ・タイプの幅の合計です。各データ・タイプの幅を計算するためには、次の表を使用してください。

重要: 最大長が 254 を超えるような VARCHAR 列がある表をエクスポートすることはできません。

表 23. エンコード・フォーマット・データ・レコード内のデータ幅： 各列のバイト数を加算して個々のデータ・タイプの幅の計算をします。

データ・タイプ	ヌル 標識	長さ フィールド	SO/SI	データ
文字	2			ヘッダーの長さ (LIH)
日付	2			LIH
浮動小数点	2			8
整数	2			LIH
小整数	2			LIH
時刻	2			LIH
タイム・スタンプ	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)

注: LIH は、ヘッダー・レコード内でその列に与えられた幅です。

各データ・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、以下の値になる可能性があり、それぞれ対応する意味を持っています。

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

指示照会オブジェクトの表番号とフィールド番号

次の表には、指示照会エクスポート・フォーマットの各表について記述する T レコードごとに、指示照会の表番号とフィールド番号が入っています。DESCRIPTION 列内の情報は、指示照会基本パネル内の特定のフィールドを固有に識別します。

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場合、ファイルの表の T レコードの前に H レコードがなければなりません。表を指定しなくてもかまいません。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 24. 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	-	表定義の表
		1112	-- 表 ID (有効な表 ID は A-Z、および #、\$、@)
		1113	-- 表名 短い長さ (50) 拡張された長さ (280)
T	1150	-	結合条件表
		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)
T	1210	-	列表
		1212	-- 列のタイプ • 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	-- 列名、式、または合計機能 短い長さ (255) 拡張された長さ (560)
T	1310	-	行選択条件
		1312	-- 入力タイプ • 1 - 演算子の左 • 2 - 演算子 • 3 - 演算子の右 • 4 - コネクター
		1313	-- 入力タイプが '1' の場合、 <u>列タイプ</u> を識別する • 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)

表 24. 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
			-- 入力タイプが '2' の場合、動詞を識別する <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
			-- 入力タイプが '3' の場合 (使用されていない)
			-- 入力タイプが '4' の場合、コネクタを識別する <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)
		1314	-- 入力タイプが '1' の場合、このフィールドは以下のとおり <ul style="list-style-type: none"> • 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
		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)
V		1501	重複行の処理 <ul style="list-style-type: none"> • '保持 (keep)' を意味する K • '廃棄 (discard)' を意味する D

フィールド 1313 と 1314 の値の意味は、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。

書式オブジェクトの表番号とフィールド番号

表 25 は、書式オブジェクトに関する T レコードの表番号および V レコードのフィールド番号をリストしています。各番号は、書式の特典部分に対応しています。

V レコードのフィールド 3080 は、その後が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が記述する切れ目パネルの番号を示します。

表 25. エクスポートされた書式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 ²	FORM.COLUMNS
1113	R	列ヘッダー	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
1118	R	列順序	FORM.COLUMNS
1119	R	列ヘッダー位置合わせ ³	FORM.COLUMNS
1120	R	列データ位置合わせ ³	FORM.COLUMNS
1121	R	列定義 ³	FORM.COLUMNS
1122	R	列定義でのヌルの受け渡し ³	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	計算でのヌルの受け渡し ³	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

2. QMF は、このフィールドを書式パネルに表示しません。

3. このフィールドは、バージョン 3 で新しくできました。

表 25. エクスポートされた書式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
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	デフォルトの切れ目テキスト	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
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	列の並べ替え	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	明細ヘッダー・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

表 25. エクスポートされた書式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
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 ³	FORM.BREAKn
3101	V	切れ目ヘッダー用の改ページ ³	FORM.BREAKn
3102	V	切れ目ヘッダーの反復 ³	FORM.BREAKn
3103	V	切れ目ヘッダーの前にスキップする行数 ³	FORM.BREAKn
3104	V	切れ目ヘッダーの後にスキップする行数 ³	FORM.BREAKn
3110	T	切れ目ヘッダー・テキスト表 ³	FORM.BREAKn
3112	R	切れ目ヘッダー行番号 ³	FORM.BREAKn
3113	R	切れ目ヘッダー位置合わせ ³	FORM.BREAKn
3114	R	切れ目ヘッダー・テキスト ³	FORM.BREAKn
3201	V	切れ目テキスト用の改ページ ³	FORM.BREAKn
3202	V	切れ目テキスト合計行 ³	FORM.BREAKn
3203	V	切れ目テキストの前にスキップする行数 ³	FORM.BREAKn
3204	V	切れ目テキストの後にスキップする行数 ³	FORM.BREAKn
3210	T	切れ目テキスト表 ³	FORM.BREAKn
3212	R	切れ目テキスト行 ³	FORM.BREAKn
3213	R	切れ目テキスト位置合わせ ³	FORM.BREAKn
3214	R	切れ目テキスト ³	FORM.BREAKn
3310	T	条件表 ³	FORM.CONDITIONS
3312	R	条件識別番号 ³	FORM.CONDITIONS
3313	R	条件式 ³	FORM.CONDITIONS
3314	R	条件パネルでのヌルの受け渡し ³	FORM.CONDITIONS

表 26 は、バージョン 3.1 以前に作成されたオブジェクトに有効なフィールドを示しています。QMF は、これらのフィールドを入力時に受け入れますが、出力時に作成することはありません。切れ目パネルごとに、フィールド番号の固有のセットがあります。

表 26. QMF バージョン 3.1 以前の、エクスポートされた書式オブジェクトのフィールド番号

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
1601	V	BREAK1: ヘッダー用の改ページ	FORM.BREAK1
1602	V	BREAK1: 列ヘッダーの反復	FORM.BREAK1
1603	V	BREAK1: ヘッダーの前のブランク行数	FORM.BREAK1
1604	V	BREAK1: ヘッダーの後のブランク行数	FORM.BREAK1
1610	T	BREAK1: ヘッダー表	FORM.BREAK1
1612	R	BREAK1: ヘッダー行	FORM.BREAK1
1612	R	BREAK1: ヘッダー位置合わせ	FORM.BREAK1
1614	R	BREAK1: ヘッダー・テキスト	FORM.BREAK1

表 26. QMF バージョン 3.1 以前の、エクスポートされた書式オブジェクトのフィールド番号 (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
1701	V	BREAK1: 後書き用の改ページ	FORM.BREAK1
1702	V	BREAK1: 列後書きの反復	FORM.BREAK1
1703	V	BREAK1: 後書きの前のブランク行数	FORM.BREAK1
1704	V	BREAK1: 後書きの後のブランク行数	FORM.BREAK1
1710	T	BREAK1: 後書き表	FORM.BREAK1
1712	R	BREAK1: 後書き行	FORM.BREAK1
1713	R	BREAK1: 後書き位置合わせ	FORM.BREAK1
1714	R	BREAK1: 後書きテキスト	FORM.BREAK1
1801	V	BREAK2: ヘッダー用の改ページ	FORM.BREAK2
1802	V	BREAK2: 列ヘッダーの反復	FORM.BREAK2
1803	V	BREAK2: ヘッダーの前のブランク行数	FORM.BREAK2
1804	V	BREAK2: ヘッダーの後のブランク行数	FORM.BREAK2
1810	T	BREAK2: ヘッダー表	FORM.BREAK2
1812	R	BREAK2: ヘッダー行	FORM.BREAK2
1813	R	BREAK2: ヘッダー位置合わせ	FORM.BREAK2
1814	R	BREAK2: ヘッダー・テキスト	FORM.BREAK2
1901	V	BREAK2: 後書き用の改ページ	FORM.BREAK2
1902	V	BREAK2: 列後書きの反復	FORM.BREAK2
1903	V	BREAK2: 後書きの前のブランク行数	FORM.BREAK2
1904	V	BREAK2: 後書きの後のブランク行数	FORM.BREAK2
1910	T	BREAK2: 後書き表	FORM.BREAK2
1912	R	BREAK2: 後書き行	FORM.BREAK2
1913	R	BREAK2: 後書き位置合わせ	FORM.BREAK2
1914	R	BREAK2: 後書きテキスト	FORM.BREAK2
2001	V	BREAK3: ヘッダー用の改ページ	FORM.BREAK3
2002	V	BREAK3: 列ヘッダーの反復	FORM.BREAK3
2003	V	BREAK3: ヘッダーの前のブランク行数	FORM.BREAK3
2004	V	BREAK3: ヘッダーの後のブランク行数	FORM.BREAK3
2010	T	BREAK3: ヘッダー表	FORM.BREAK3
2012	R	BREAK3: ヘッダー行	FORM.BREAK3
2013	V	BREAK3: ヘッダー位置合わせ	FORM.BREAK3
2014	R	BREAK3: ヘッダー・テキスト	FORM.BREAK3
2101	V	BREAK3: 後書き用の改ページ	FORM.BREAK3
2102	V	BREAK3: 列後書きの反復	FORM.BREAK3

表 26. QMF バージョン 3.1 以前の、エクスポートされた書式オブジェクトのフィールド番号 (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
2103	V	BREAK3: 後書きの前のブランク行数	FORM.BREAK3
2104	V	BREAK3: 後書きの後のブランク行数	FORM.BREAK3
2110	T	BREAK3: 後書き表	FORM.BREAK3
2112	R	BREAK3: 後書き行	FORM.BREAK3
2113	R	BREAK3: 後書き位置合わせ	FORM.BREAK3
2114	R	BREAK3: 後書きテキスト	FORM.BREAK3
2201	V	BREAK4: ヘッダー用の改ページ	FORM.BREAK4
2202	V	BREAK4: 列ヘッダーの反復	FORM.BREAK4
2203	V	BREAK4: ヘッダーの前のブランク行数	FORM.BREAK4
2204	V	BREAK4: ヘッダーの後のブランク行数	FORM.BREAK4
2210	T	BREAK4: ヘッダー表	FORM.BREAK4
2212	R	BREAK4: ヘッダー行	FORM.BREAK4
2213	R	BREAK4: ヘッダー位置合わせ	FORM.BREAK4
2214	R	BREAK4: ヘッダー・テキスト	FORM.BREAK4
2301	V	BREAK4: 後書き用の改ページ	FORM.BREAK4
2301	V	BREAK4: 列後書きの反復	FORM.BREAK4
2303	V	BREAK4: 後書きの前のブランク行数	FORM.BREAK4
2304	V	BREAK4: 後書きの後のブランク行数	FORM.BREAK4
2310	T	BREAK4: 後書き表	FORM.BREAK4
2312	R	BREAK4: 後書き行	FORM.BREAK4
2313	R	BREAK4: 後書き位置合わせ	FORM.BREAK4
2314	R	BREAK4: 後書きテキスト	FORM.BREAK4
2401	V	BREAK5: ヘッダー用の改ページ	FORM.BREAK5
2402	V	BREAK5: 列ヘッダーの反復	FORM.BREAK5
2403	V	BREAK5: ヘッダーの前のブランク行数	FORM.BREAK5
2404	V	BREAK5: ヘッダーの後のブランク行数	FORM.BREAK5
2410	T	BREAK5: ヘッダー表	FORM.BREAK5
2412	R	BREAK5: ヘッダー行	FORM.BREAK5
2413	R	BREAK5: ヘッダー位置合わせ	FORM.BREAK5
2414	R	BREAK5: ヘッダー・テキスト	FORM.BREAK5
2501	V	BREAK5: 後書き用の改ページ	FORM.BREAK5
2502	V	BREAK5: 列後書きの反復	FORM.BREAK5
2503	V	BREAK5: 後書きの前のブランク行数	FORM.BREAK5
2504	V	BREAK5: 後書きの後のブランク行数	FORM.BREAK5
2510	T	BREAK5: 後書き表	FORM.BREAK5

表 26. QMF バージョン 3.1 以前の、エクスポートされた書式オブジェクトのフィールド番号 (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
2512	R	BREAK5: 後書き行	FORM.BREAK5
2513	R	BREAK5: 後書き位置合わせ	FORM.BREAK5
2514	R	BREAK5: 後書きテキスト	FORM.BREAK5
2601	V	BREAK6: ヘッダー用の改ページ	FORM.BREAK6
2602	V	BREAK6: 列ヘッダーの反復	FORM.BREAK6
2603	V	BREAK6: ヘッダーの前のブランク行数	FORM.BREAK6
2604	V	BREAK6: ヘッダーの後のブランク行数	FORM.BREAK6
2610	T	BREAK6: ヘッダー表	FORM.BREAK6
2612	R	BREAK6: ヘッダー行	FORM.BREAK6
2613	R	BREAK6: ヘッダー位置合わせ	FORM.BREAK6
2614	R	BREAK6: ヘッダー・テキスト	FORM.BREAK6
2701	V	BREAK6: 後書き用の改ページ	FORM.BREAK6
2702	V	BREAK6: 列後書きの反復	FORM.BREAK6
2703	V	BREAK6: 後書きの前のブランク行数	FORM.BREAK6
2704	V	BREAK6: 後書きの後のブランク行数	FORM.BREAK6
2710	T	BREAK6: 後書き表	FORM.BREAK6
2712	R	BREAK6: 後書き行	FORM.BREAK6
2713	R	BREAK6: 後書き位置合わせ	FORM.BREAK6
2714	R	BREAK6: 後書きテキスト	FORM.BREAK6

報告書オブジェクトの表番号とフィールド番号

以下の図は、T レコードの表番号と V レコードのフィールド番号を示しています。

表 27. 全般報告書。エクスポートされた報告書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
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	フォーマット設定済み報告書表
		報告書内のフォーマット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	データのフォーマットを設定する編集コード
1014	T	フォーマット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

表 27. 全般報告書。エクスポートされた報告書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
1015	T	フォーマット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	フォーマット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	フォーマット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

報告書が横方向スタイルの報告書の場合のフィールド 1014、1015、および 1016 の意味については、表 28 の後の注 2 を参照してください。

表 28. 横方向報告書。エクスポートされた報告書オブジェクトのフィールド番号

フィールド番号	レコード・ タイプ	説明
2001	V	横方向の値のフォーマット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	フォーマット設定済み横方向値の開始位置。(横方向値は、列ヘッダー行に現れる)
2013	T	フォーマット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

注:

1. 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。
2. 横方向報告書の統計列の場合、フィールド 1014、1015、および 1016 は、統計列の横方向値のセット内のフィールドの相対的な開始位置と終了位置について記述しません。(表 28 のフィールド 2014 を参照してください。)
3. 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。
4. 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。

QMF 報告書で使用される HTML タグ

表 29 は、QMF がワールド・ワイド・ウェブ (WWW) 上での表示用に報告書をフォーマット設定するために使用する HTML のタグ・セットを簡単に示しています。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 3.0 の資料を参照してください。

表 29. HTML 報告書で使用される HTML 3.0 のタグ

タグ・セット	説明
<HTML></HTML>	ファイルを HTML 文書として定義する。
<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 フォーマット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。

付録 C. 統合交換フォーマット (IXF)

DATAFORMAT=IXF オプションを指定した EXPORT コマンドを使用して、DATA または TABLE オブジェクトをエクスポートすると、ファイル、データ・セット、または CICS データ・キューは統合交換フォーマット (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。これについて、この節で説明します。統合交換フォーマットの詳細については、*Data Extract: Reference* を参照してください。

IXF フォーマットは、QMF 環境の外で表を作成してインポートする場合に、特に便利です。そのためには、OUTPUTMODE を CHARACTER に設定してください。

QMF では、エクスポート後の IXF ファイル、データ・セット、または CICS データ・キューは、以下のレコードから構成されます。

- 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)

エクスポート後のファイル、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。エクスポート後のファイル、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は、図 70 のようになります。

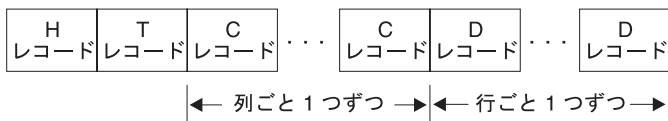


図 70. エクスポート後のデータ・ファイル、データ・セット、または CICS データ・キュー内のレコードの配置 (IXF フォーマット)

注: データベース・マネージャー PC/IXF ファイル・フォーマットは、システム /370 IXF フォーマットと同じではありません。IXF フォーマットのデータは、PC とシステム /370 プラットフォームの間で転送できません。

以下の節では、各レコードのフォーマットについて説明します。括弧内の値は、データのエクスポート時に QMF が提供する値です。

ヘッダー・レコード (H)

ヘッダー・レコード (必須レコード) は、ファイル、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。これは、文字データが入っている 42 バイトのレコードです。H レコードのフォーマットは次のとおりです。

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	ファイル、データ・セット、または CICS データ・キューID (IXF)
05-08	IXF のバージョン: <ul style="list-style-type: none"> • 0000 短い列名を含むデータ・オブジェクト。18 文字以内。 • 0001 少なくとも 1 つの長い列名を含むデータ・オブジェクト。19 文字以上。
09-14	送り元のプロダクト名 (QMF)
15-20	送り元のプロダクト・リリース・レベル (V8R1M0)
21-28	ファイル、データ・セット、または CICS データ・キューの作成日付。 フォーマットは YYYYMMDD
29-34	ファイル、データ・セット、または CICS データ・キューの作成時刻。 フォーマットは HHMMSS
35-39	ファイル、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。これは、文字書式で表現される 5 桁の数値です。
40	DBCS 標識。DBCS データの有無を示す。Y または N
41-42	予約済み

表レコード (T)

ヘッダー・レコードの後に表レコードがきます。各 IXF ファイル、データ・セット、または CICS 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、QMF EXPORT TABLE コマンドを使用してエクスポート中のファイル、データ・セット、または CICS データ・キューに関する表とデータ情報が入ります。ただし、RUN QUERY コマンドの結果である QMF データ・オブジェクトをエクスポートした場合、この表レコードには、所有者と名前はブランクになります。ロング・ネームの場合、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。

T レコードのフォーマットは次のとおりです。

バイト位置	情報とタイプ
01	表レコード標識 (T)

バイト位置	情報とタイプ
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・フォーマット。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 レコード数を示す、文字フォーマットの数値。
50-51	予約済み
52-81	ブランク

列レコード (C)

QMF バージョン 8.1 では、18 文字より長い列名が存在する場合、列レコードの列名フィールドを、18 文字から 30 文字に増やす必要があります。18 文字より長い列名がある場合、ヘッダー・レコード内の IXF のバージョン番号は、0001 に設定されます。18 文字よりも長い列名がない場合は、QMF の以前のリリースと互換性のある、古い IXF フォーマットが使用されます。その場合、ヘッダー・レコード内の IXF のバージョン番号は、0000 です。

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。列レコードのフォーマットは次のとおりです。

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

バイト位置	情報とタイプ
04-21	<p>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)</p> <p>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る。</p>
22 または 34	ヌル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、111 ページの表 12 参照)
29-33 または 41-45	コード・ページ (00000)
34-38 または 46-50	予約済み
39-43 または 51-55	<p>列データの長さ。文字フォーマットの数値</p> <p>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 INTEGER、SMALLINT、DATE、TIME、または TIMESTAMP の場合、このフィールドはブランクになります (長さは各データ・タイプに固有)。</p>
44-49 または 56-61	<p>列データの開始位置。データ・レコードの開始点からの、列データのオフセットを示す値 (文字フォーマット)。</p> <p>列でヌルが許可される場合、このフィールドはヌル標識を指す。列でヌルが使用されない場合、このフィールドはデータ自体を指す。列でヌルが許可されるかどうかに関係なく、ヌル標識用のスペースはレコード内に常に存在する。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、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。</p>
50-79 または 62-91	使用可能であれば、列ラベル情報。他の場合は、ブランク。
80-81 または 92-93	文字フォーマットの 2 バイトのゼロ (00)

データ・レコード (D)

表内の行ごとに 1 つのデータ・レコードがあります。以下にデータ・レコードのフォーマットを示します。

バイト位置	情報とタイプ
01	データ・レコード標識 (D)
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 M (機械語) または C (文字) のいずれであるかによって、2 進数フォーマットまたは文字フォーマットである行データ。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・フォーマット

n 個の列用の D レコードのデータは、隣り合わせに置かれます。

列 1	列 2	...	列 n
--------	--------	-----	----------

列ごとに、データはヌル標識に続くデータ自体から構成されます。ヌルが許可される場合 (C レコードのバイト 22 = Y)、各 C レコードのバイト 44-49 は、列のデータの前にあるヌル標識を指します。ヌルが許可されない場合 (バイト 22 = N)、バイト 44-49 はデータ自体を指します。ただし、後者の場合でも、ヌル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 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)。

ヌル標識の表示は、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・フォーマットが文字の場合、1 バイトがヌル標識に使用されます。

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

256 ページの図 71 には、1 つはヌルのデータを示し、もう 1 つはヌルでないデータを示す 2 つの D レコードが示されています。

データ・フォーマットが 2 進数の場合、2 バイトがヌル標識に使用されます。

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

258 ページの図 72 には、ヌル・データ標識と非ヌル・データ標識を示す 2 つの D レコードが示されています。

データ・タイプ別の列データのフォーマット

表 30 は、データ・タイプごとに、文字フォーマットと 2 進数フォーマットの D レコード内の列データの長さフォーマットを示しています。表では、IXFCLENG は C レコードのバイト 39-43 の内容 (列データの長さ) を示しています。

表 30. データ・タイプ別の IXF 列データのフォーマット

データ・タイプ・コード	データ・タイプ	データ長情報の文字フォーマット	データ長情報のバイナリー・フォーマット
384	DATE	<p>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>フォーマットは次のとおり。</p> <p style="text-align: center;">yyyy-mm-dd</p> <p>ここで yyyy は年、mm は月、dd は日を示す。yyyy、mm、および dd 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。yyyy の可能な範囲は 0001-9999、mm は 01-12。dd の範囲は月によって異なる。</p> <p>2002-02-28</p>	文字フォーマットと同じ

表 30. データ・タイプ別の IXF 列データのフォーマット (続き)

データ・タイプ・コード	データ・タイプ	データ長情報の文字フォーマット	データ長情報のバイナリー・フォーマット
388	TIME	<p>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>フォーマットは次のとおり。</p> <p style="text-align: center;">hh.mm.ss</p> <p>ここで hh は 24 時間フォーマットの時、mm は分、および ss は秒を示す。hh、mm、および ss 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"> • hh は 00 - 23 • mm は 00 - 59 • ss 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 is 10:37:42 AM 08.00.00 is 8 AM exactly 23.30.00 is 11:30 PM</p>	文字フォーマットと同じ

表 30. データ・タイプ別の IXF 列データのフォーマット (続き)

データ・タイプ・コード	データ・タイプ	データ長情報の文字フォーマット	データ長情報のバイナリー・フォーマット
392	TIMESTAMP	<p>IXFLENG の値は意味がない。長さ (26 バイト) はデータ・タイプによって決まる。</p> <p>フォーマットは次のとおり。</p> <pre>yyyy-mm-dd-hh .mm.ss.nnnnnn</pre> <p>ここで yyyy は年、最初の mm は月、dd は日、hh は 24 時フォーマットの時、2 番目の mm は分、ss は秒、および nnnnnn はマイクロ秒を示す。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。nnnnnn の範囲は 000000-999999 である。例:</p> <pre>1997-12-31-23 .59.59.999999 (the last microsecond in 1997) 1998-01-01-00 .00.00.000000 (the first microsecond in 1998)</pre> <p>24.00.00.000000 は、有効なタイム・スタンプの時刻部分である。</p>	文字フォーマットと同じ
448	VARCHAR	<p>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 0-254 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</p> <pre>If IXFLENG=00010 Data format is: 00005JONESxxxxx</pre> <p>ここで、各 x は空白文字 (X'40') である。</p>	<p>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 1-254 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</p> <pre>If IXFLENG=00010 Data format is: nnJONESxxxxx</pre> <p>ここで nn=X'0005' であり、各 x は空白文字 (X'40') である。</p>

表 30. データ・タイプ別の IXF 列データのフォーマット (続き)

データ・タイプ・コード	データ・タイプ	データ長情報の文字フォーマット	データ長情報のバイナリー・フォーマット
452	CHAR	IXFCLENG 文字ストリングの長さである。データ長は、IXFCLENG 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 If IXFCLENG=00005 Data format is: JONES ここで JONES は、C レコードのバイト 44-49 によって示される、5 バイトの文字ストリング	文字フォーマットと同じ
456	LONG VARCHAR	N の有効範囲が 0-32767 であることを除いて、VARCHAR と同じ	文字フォーマットと同じ
464	VARGRAPHIC	IXFCLENG は 2 バイト文字の最大数 (2xN 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、IXFCLENG 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 1-127 であり、カウント・フィールドの有効範囲は 0-N である。) 例 IXFCLENG = 00006 の場合のデータ・フォーマット 00003oZZYXXixxxxxx ここで、o はシフトアウト、i はシフトイン、x はブランク文字 (X'40') である。	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 IXFCLENG によって示されたバイト数の 2 倍から構成される。(IXFCLENG の有効範囲は 1-127 であり、カウント・フィールドの有効範囲は 0-IXFCLENG 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 IXFCLENG = 00008 の場合のデータ・フォーマット nnZZYXXWVxxxxxxx ここで nn=X'0004' であり、x はブランク文字 (X'40') である。
468	GRAPHIC	IXFCLENG は 2 バイト文字数 (2*N バイト)。データ長は、2*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 If IXFCLENG=00005 Data format is: oZZYXXWVVi ここで、o はシフトアウト、i はシフトインである。	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、文字フォーマットと同じである。例 If IXFCLENG=00005 Data format is: ZZYXXWVVV
472	LONG VARGRAPHIC	N の有効範囲が 0-16383 であることを除いて、VARGRAPHIC と同じ	文字フォーマットと同じ

表 30. データ・タイプ別の IXF 列データのフォーマット (続き)

データ・タイプ・コード	データ・タイプ	データ長情報の文字フォーマット	データ長情報のバイナリー・フォーマット
480	FLOAT	<p>IXFLENG 内の値は 8。データの長さフォーマットは、データ・タイプによって決まる。</p> <p>データは以下のような配置の 23 バイトの文字値から構成される。</p> <ul style="list-style-type: none"> • 1 文字の符号 • 18 文字の仮数 (17 桁と小数点) • 文字 E • 3 文字の符号付き指数 <p>例:</p> <pre>-1.2345678901234567E+14 +6.2345678901234567E-01 0.0000000000000000E+00</pre>	<p>IXFLENG 内の値は 8。データの長さフォーマットは、データ・タイプによって決まる。</p> <p>データは、長精度浮動小数点用の標準 IBM S/370 フォーマットの 8 バイトの浮動小数点値からなる。</p>
484	DECIMAL	<p>C レコードのバイト 39-43 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 (または S=0 の場合は P+1 バイト) の文字値として、右寄せされてフォーマット設定される。最初のバイトは符号用であり、小数点 (S によって暗示される位置) は、S がゼロ以外の場合にのみ存在する。例:</p> <pre>If P=005, S=00; Data format is: 12345 If P=006, S=02; Data format is: +2345.10 If P=004, S=03; Data format is: -8.515</pre>	<p>C レコードのバイト 39-43 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、標準 IBM S/370 パック 10 進数フォーマットの (P+2)/2 バイトのパック 10 進数値から構成される。P 桁のうちの S は、以下のように暗黙の小数点として解釈される。例:</p> <pre>If P=005, S=00; Data format is: X'12345C' If P=006, S=02; Data format is: X'0234510D'</pre>
496	INTEGER	<p>IXFLENG の値は意味がない。データの長さフォーマットは、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> <pre>0000000013 +1187642200 -0033588727</pre>	<p>IXFLENG の値は意味がない。データの長さフォーマットは、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 30. データ・タイプ別の IXF 列データのフォーマット (続き)

データ・タイプ・コード	データ・タイプ	データ長情報の文字フォーマット	データ長情報のバイナリー・フォーマット
500	SMALLINT	IXFLENG の値は意味がない。データの長さフォーマットは、データ・タイプによって決まる。 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: 00023 +00763 -21311	IXFLENG の値は意味がない。データの長さフォーマットは、データ・タイプによって決まる。 データは 2 バイトの 2 進数値から構成される。

IXF の例

82 ページの例に示した表 (QMF フォーマットを使用してエクスポートされた表) を、IXF フォーマットを使用して (OUTPUTMODE=CHARACTER を指定) エクスポートするとします。

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

エクスポート後のファイル、データ・セット、または CICS データ・キューは、以下に示すように、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 レコードで合計 7 つのレコードから構成されます。

```
HIXF0000QMF  V7R1M01998120409560000005N
T18                      データベース      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 で示してあります。256 ページの図 71 は、これらのレコードの詳細を示しています。

統合交換フォーマット (IXF)

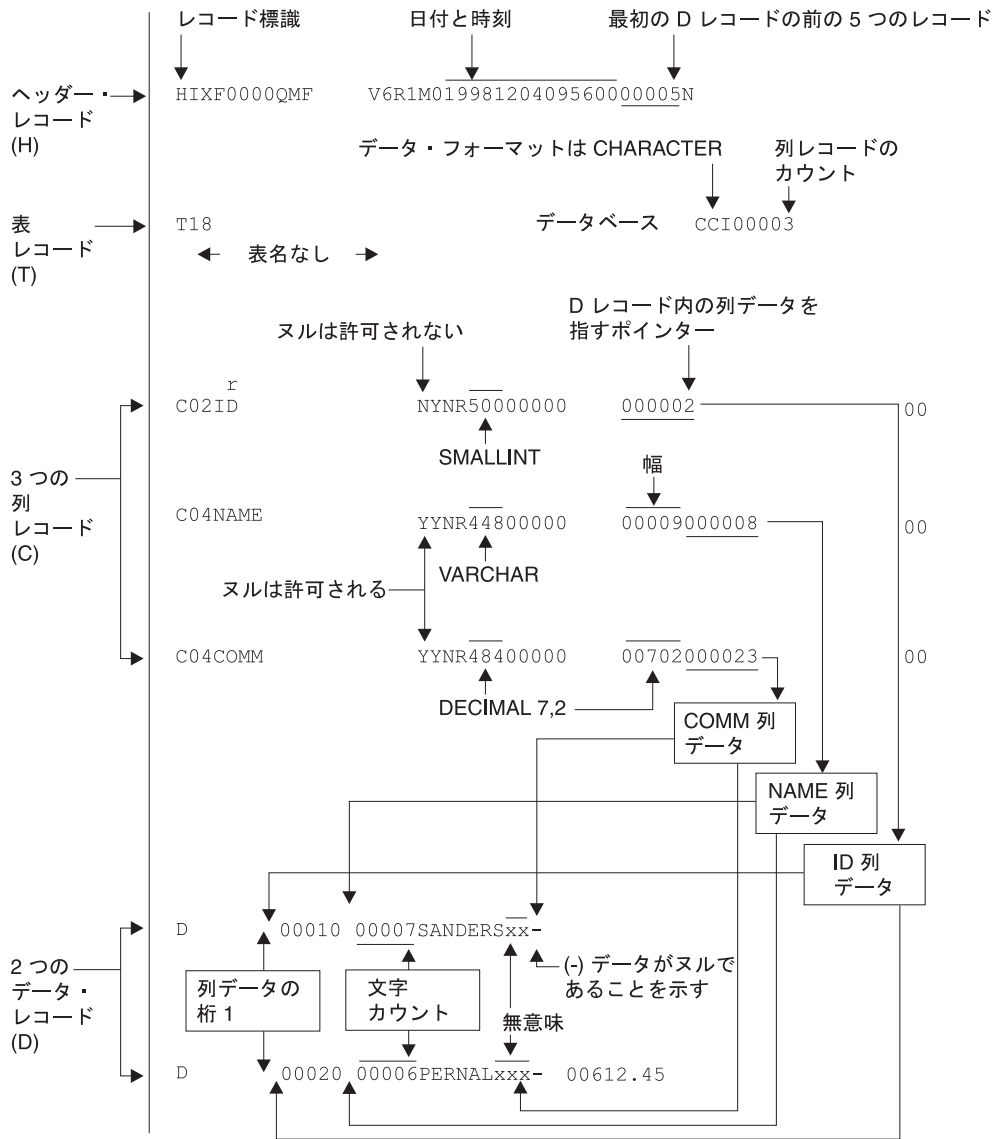


図 71. サンプル IXF レコードのフォーマット (OUTPUTMODE=CHARACTER)

IXF フォーマット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。前の例と同様に、エクスポート後のファイル、データ・セット、または CICS データ・キューは、以下に示すように 7 つのレコードから構成されま

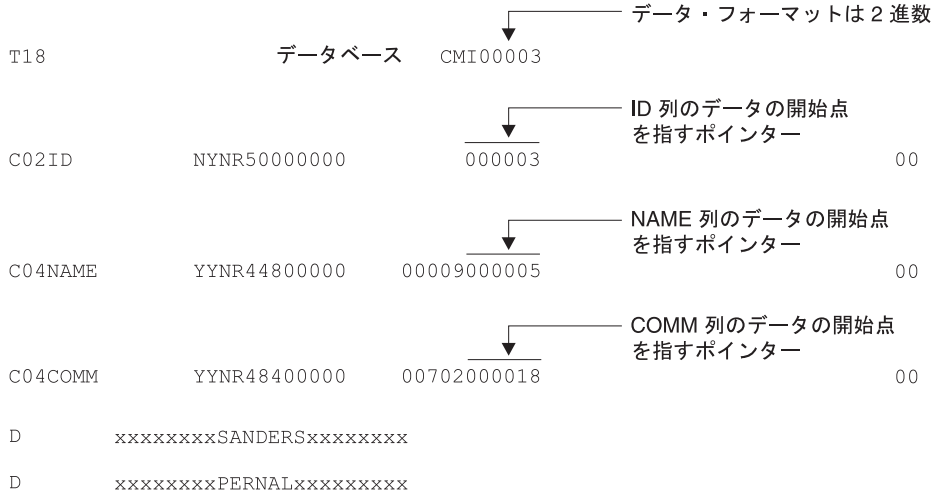
す。

```

HIXF0000QMF  V7R1M01998120409565000005N
T18           データベース      CMI00003
C02ID        NYNR50000000      000003      00
C04NAME      YYNR44800000      00009000005  00
C04COMM      YYNR48400000      00702000018  00
D   xxxxxxxxSANDERSxxxxxxxx
D   xxxxxxxxPERNALxxxxxxxx
    
```

バイト 44 から 49 まで (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。258 ページの図 72 は、エクスポート後のファイル、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。

HIXF0000QMF V6R1M01998120409565000005N



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。

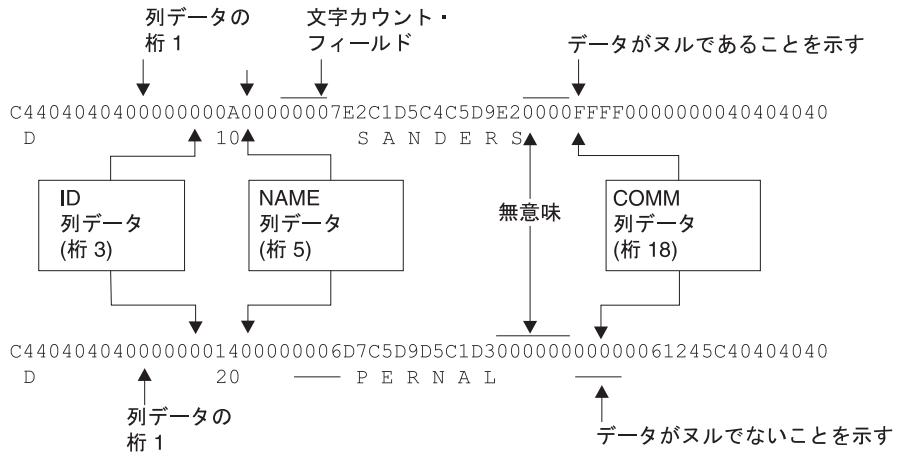


図 72. サンプル IXF レコードのフォーマット (OUTPUTMODE=BINARY)

付録 D. プロダクト・インターフェース・マクロ

この付録に示したマクロは、ユーザーのための汎用プログラミング・インターフェースとして、QMF によって提供されています。

警告: この付録に示したマクロ以外の 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てはなりません。

プロダクト・インターフェース・マクロ

- DSQQMF n

ここで、 n は NLF 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・マクロ

- アセンブラ
 - DSQCIA
 - DSQCOMMA
- C/370
 - DSQCIC
 - DSQCICE
 - DSQCOMM C
- COBOL
 - DSQCIB
 - DSQCOMMB
- FORTRAN
 - DSQCIF
 - DSQCIFE
 - DSQCOMMF
- PL/I
 - DSQCIPL
 - DSQCIPX
 - DSQCOMML
- REXX
 - DSQCIX

コマンド・インターフェース・マクロ

- DSQCCI

QMF 管理プログラム出口インターフェース・マクロ

- DXEGOVA
- DXEXCBA

QMF ユーザー編集出口マクロ

- DXEECS

付録 E. QMF グローバル変数表

QMF には、アプリケーションで使用できる多くの変数が用意されています。QMF バージョン 3 からは、呼び出し可能インターフェースの現在の命名規則が採用されました。対応するコマンド・インターフェースの変数名もそのまま使用できます。

呼び出し可能インターフェースのグローバル変数名の長さは、18 文字までです。呼び出し可能インターフェース・ユーザーは、古い名前 (8 文字) または新しい名前 (18 文字) のいずれも使用できますが、新しい名前の使用をお勧めします。コマンド・インターフェース・ユーザーは、古い名前を使用する必要があります。

新しい命名規則は、**DSQcc_XXXXXXXXXX** です。

cc 以下のカテゴリ ID のいずれかです。

- AP** プロファイル関連の状態情報
- AO** その他の (プロファイル関連でない) 状態情報
- CM** 先行コマンドで生じたメッセージに関する情報
- CP** 表エディターに関する情報
- DC** QMF が画面に情報を表示する方法の制御
- EC** QMF がコマンドとプロシージャを実行する方法の制御
- QC** CONVERT QUERY オプションで生じる変数
- QM** RUN QUERY エラー・メッセージ情報
- QW** QMF (Windows 版) に固有な変数

_ 下線文字

XXXXXXXXXX

最大 12 文字の記述的な名前

バージョン 3.3 から、QMF は、**Q.SYSTEM_INI** という名前の特殊プロシージャを提供します。このプロシージャを使用すると、ユーザーは、初期化時にグローバル変数をカスタマイズできます。

プロファイル関連状態情報の DSQ グローバル変数

これらのグローバル変数は、いずれも **SET GLOBAL** コマンドでは変更できません。

QMF グローバル変数

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。値は以下のいずれかになります。 1 UPPER 2 MIXED 3 STRING
DSQAP_CONFIRM	DSQAPRMP	01	CONFIRM パラメーター。値は以下のいずれかになります。 0 NO 1 YES
DSQAP_DECIMAL	DSQAPDEC	01	DECIMAL パラメーター。値は以下のいずれかになります。 1 PERIOD 2 COMMA 3 FRENCH
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' または 'CONT')
DSQAP_PFKEY_TABLE	DSQAPPFK	31	ファンクション・キー表の名前
DSQAP_PRINTER	DSQAPPRT	08	PRINTER パラメーター。値は以下のいずれかになります。 <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
DSQAP_QUERY_MODEL	DSQAMODP	01	MODEL パラメーター値は RELATIONAL の場合は '1' になります。
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	SYNONYMS パラメーター
DSQAP_TRACE	DSQAPTRC	18	TRACE パラメーター。値は以下のいずれかになります。 ALL (最大トレース) NONE (最小トレース) 各 QMF コンポーネントの指定 (たとえば、A2L2C1)

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_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')。

プロファイル関連でない状態情報の DSQ グローバル変数

これらのグローバル変数は、いずれも SET GLOBAL コマンドでは変更できません。

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	アプリケーション・トレース・レベル。値は以下のいずれかになります。 0 レベル A0 1 レベル A1 2 レベル A2
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。値は以下のいずれかになります。 1 対話式セッション 2 バッチ・モード・セッション
DSQAO_CONNECT_ID	DSQAAUTH	08	データベースとの接続に使用されるユーザー ID。作業を行う際のユーザー ID です。
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の名前。名前は 18 文字です (必要に応じて、右側がブランクで埋められます)。
DSQAO_CURSOR_OPEN	DSQACRSR	01	データベースのカーソルの状況。値は以下のいずれかになります。 1 カーソルがオープンされている 2 カーソルがクローズされている
DSQAO_DB_MANAGER	DSQADBMG	01	データベース・マネージャー。値は以下のいずれかになります。 1 DB2 (VM/ESA または VSE/ESA 用) 2 DB2 (MVS/ESA 用) 3 ワークステーション・データベース・サーバー
DSQAO_DBCS	DSQADBCS	01	DBCS サポート状況。値は以下のいずれかになります。 1 DBCS サポートあり 2 DBCS サポートなし

QMF グローバル変数

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_FORM_PANEL	DSQASUBP	02	<p>現在の書式パネル。値は以下のいずれかになります。</p> <table border="0"> <tr><td>1</td><td>FORM.MAIN</td></tr> <tr><td>2</td><td>FORM.COLUMN</td></tr> <tr><td>3</td><td>FORM.PAGE</td></tr> <tr><td>4</td><td>FORM.FINAL</td></tr> <tr><td>5</td><td>FORM.BREAK1</td></tr> <tr><td>6</td><td>FORM.BREAK2</td></tr> <tr><td>7</td><td>FORM.BREAK3</td></tr> <tr><td>8</td><td>FORM.BREAK4</td></tr> <tr><td>9</td><td>FORM.BREAK5</td></tr> <tr><td>10</td><td>FORM.BREAK6</td></tr> <tr><td>11</td><td>FORM.OPTIONS</td></tr> <tr><td>12</td><td>FORM.CALC</td></tr> <tr><td>13</td><td>FORM.DETAIL</td></tr> <tr><td>14</td><td>FORM.CONDITIONS</td></tr> </table> <p>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。</p>	1	FORM.MAIN	2	FORM.COLUMN	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
1	FORM.MAIN																														
2	FORM.COLUMN																														
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																														
DSQAO_INTERACT	DSQAIACT	01	<p>対話フラグの設定。値は以下のいずれかになります。</p> <table border="0"> <tr><td>0</td><td>対話式実行が許可されない</td></tr> <tr><td>1</td><td>対話式実行が許可される</td></tr> </table>	0	対話式実行が許可されない	1	対話式実行が許可される																								
0	対話式実行が許可されない																														
1	対話式実行が許可される																														
DSQAO_LOCAL_DB2	なし	18	<p>ローカルの DB2 データベースのロケーション名。変数 DSQAO_SUBSYS_ID で指定されたサブシステムのロケーション名です。</p> <p>リモート作業単位環境で、DSQ_LOCAL_DB2 はアプリケーション・リクエスターの名前。名前は 16 文字 (必要に応じて、右側がブランクで埋められる)。</p> <p>QMF が VM 環境または VSE 環境で実行している場合、このフィールドはブランク。</p>																												
DSQAO_LOCATION	DSQAITLO	16	<p>現行オブジェクトのロケーション名 (ある場合)。この値を適用できるのは、3 パート名が使用された場合だけです。</p>																												
DSQAO_NLF_LANG	DSQALANG	01	<p>ユーザーの国の言語。英語言語環境の場合、これは 'E' になります。</p>																												
DSQAO_NUM_FETCHED	DSQAROWS	16	<p>取り出されたデータの行数。DATA オブジェクトが空の場合は、'0' を含みます。</p>																												

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に示されている表 (報告書に入っている)、照会、プロシージャー、または書式の名前 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に示されている表 (報告書に入っている)、照会、プロシージャー、または書式の所有者 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_PANEL_TYPE	DSQAITEM	01	現行パネルのタイプ。値は以下のいずれかになります。 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 のリリース番号を表す数値。QMF バージョン 8.1 の場合は '14' になります。
DSQAO_QMF_VER_RLS	DSQAQMF	10	QMF のバージョン番号とリリース番号。 <ul style="list-style-type: none"> • QMF 8.1 の場合 • これは '14' です。
DSQAO_QRY_SUBTYPE	DSQASUBI	01	照会のサブタイプ。値は以下のとおり。 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。値は RELATIONAL の場合は '1' になります。

QMF グローバル変数

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_SAME_CMD	DSQACMDM	01	値は以下のとおり。 0 2 つのコマンドは同じでない 1 2 つのコマンドは同じである
DSQAO_SUBSYS_ID	なし	04	QMF が TSO で実行している場合に、QMF が接続しているローカル DB2 UDB サブシステムの ID。 CMS または CICS から DSQSUBS プログラム・パラメーターの値を指定すると、その値はこのグローバル変数に入ります。これは、そのパラメーターが黙認され、値が処理されないからです。つまり、その値はグローバル変数フィールドに置かれ、何も処理されません。このロジックでは、同一の EXEC を複数の環境で使用することができます。
DSQAO_SYSTEM_ID	DSQASYST	01	現行のオペレーティング・システム・システム値。値は以下のいずれかになります。 1 VM/SP 2 TSO/MVS 3 TSO またはネイティブの z/OS 4 VM/XA または VM/ESA 5 CICS
DSQAO_TERMINATE	DSQCSESC	01	QMF 終了フラグ。値は以下のいずれかになります。 0 セッションはマークされなかった。 1 セッションはマークされた。
DSQAO_VARIATION	DSQAVARN	02	書式パネル・バリエーション番号。ブランクの場合、FORM.DETAIL は現行パネルではありません。

CICS に関連付けられている DSQ のグローバル変数

表中の変数のうち、SET GLOBAL コマンドで変更できるのは DSQAP_CICS_PQNAME と DSQAP_CICS_PQTYPE だけです。

キュー・タイプが TD の場合、対応するキュー名の最大長は 4 です。たとえば、DSQAO_CICS_SQTYPE が TD の場合、DSQAO_CICS_SQNAME の最大長は 4 です。

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 印刷を入れる CICS データ・キューの名前

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_PQTYPE	なし	02	QMF 印刷を入れる 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込む。これがデフォルトです。 TD QMF 印刷を CICS 一時データ・キューに書き込む。
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 予備ファイルを補助記憶装置上の CICS 一時記憶域キューに書き込む。これがデフォルトです。 TD QMF 予備ファイルを CICS 一時データ・キューに書き込む。
DSQAO_CICS_TQNAME	なし	08	QMF トレースを入れる 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込む。 TD QMF トレースを CICS 一時データ・キューに書き込む。これがデフォルトです。

先行コマンドで作成されたメッセージに関連する DSQ グローバル変数

これらのグローバル変数は、いずれも SET GLOBAL コマンドでは変更できません。

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	DSQCIMSG	80	メッセージ・テキスト
DSQCM_MESSAGE_ALL	DSQCIMSA	360	完全なメッセージ・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>
DSQCM_SUBST_VARS	DSQCIM00	04	メッセージ中にある置換変数の個数

表編集機能に関連付けられている DSQ のグローバル変数

これらのグローバル変数のいずれも 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 の場合、表エディターは、この表に示されているグローバル変数の値を検査することによって、有効にする確認のカテゴリーを決定します。

表エディターのデフォルトは、EDIT TABLE コマンドの SAVE キーワードによって、以下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が有効です。
- SAVE=END なら、DELETE、MODIFY、END/CANCEL カテゴリーのデフォルトが有効です。ADD カテゴリーと CHANGE カテゴリーのデフォルトは無効です。

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_TEADD	なし	01	ADD サブコマンドの後に確認パネルを表示します。値は以下のいずれかになります。 0 パネルを表示しない。 1 パネルを表示する。 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によって決まる。
DSQCP_TECHG	なし	01	CHANGE サブコマンドの後に確認パネルを表示します。値は以下のいずれかになります。 0 パネルを表示しない。 1 パネルを表示する。 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によって決まる。
DSQCP_TEEND	なし	01	ユーザーが表エディター・サブセッションを終了するために END サブコマンドまたは CANCEL サブコマンドを出した場合に、確認パネルを表示する。 出されたのが END か CANCEL か、データベースへの変更が行なわれたのか、END または CANCEL が出されたとき画面に変更後のデータが含まれていたかどうかによって、いくとおりかのパネルが表示されます。値は以下のとおり。 0 パネルを表示しない。 1 パネルを表示する。 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によって決まる。

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 サブコマンドの後に確認パネルを表示します。値は以下のいずれかになります。 0 パネルを表示しない。 1 パネルを表示する。 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によって決まる。
DSQCP_TEDFLT	なし	01	表編集機能内の列のデフォルト値を示すの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	表エディター内の、漢字ストリング列用のデフォルト値を示す予約 DBCS 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。このグローバル変数は DBCS 環境でのみ使用される点に注意してください。
DSQCP_TEMOD	なし	01	表示されたデータが変更され、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します。表示されるパネルには、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。以下の値を指定できます。 0 パネルを表示しない。 1 パネルを表示する。 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によって決まる。
DSQCP_TENULL	なし	01	表編集機能内の、列の空白値を示すの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	表エディター内の、漢字ストリング列用のヌル値を示す (あるいは、検索基準のコンテキストでは無視することを示す) 予約 DBCS 文字 この値は、1 つの DBCS 文字から構成される 4 バイトの混合ストリングで、先頭にシフトアウト文字が付き、最後にシフトイン文字が付いていなければなりません。最初は DBCS のハイフン (-) 文字に設定されています。このグローバル変数は DBCS 環境でのみ使用される点に注意してください。

画面への情報表示方法を制御する DSQ のグローバル変数

これらのグローバル変数のいずれも SET GLOBAL コマンドで変更できます。

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_COST_EST	なし	01	<p>オプションとして、データベースのコスト見積もりを抑止します。以下の値を指定できます。</p> <p>0 NO — コスト見積もりを表示しない</p> <p>1 YES — コスト見積もりを表示する。これがデフォルトです。</p>
DSQDC_CURRENCY	なし	18	<p>DC 編集コードが指定されると通貨記号が使用される</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>
DSQDC_DISPLAY_RPT	DSQADPAN	01	<p>RUN QUERY の後に報告書を表示します。以下の値を指定できます。</p> <p>0 RUN 照会コマンドからの結果を QMF に表示させない</p> <p>QMF を DSQQMFE で対話式に起動したとき、または BATCH モードで起動した場合、これがデフォルト値です。QMF が BATCH モードで起動している場合にこの変数を変更しても、QMF 画面は表示されません。</p> <p>1 QMF に報告書を自動的に表示させる</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、これがデフォルト値です。START コマンドの DSQADPAN プログラム・パラメーターがこのグローバル変数に優先します。</p> <p>このグローバル変数は、アプリケーション専用です。コマンド行から入力された RUN QUERY コマンドには影響がありません。</p>

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・オブジェクトの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。最初の文字の値には、以下を指定できます。</p> <p>1 リストはデフォルトの順を使用する</p> <p>2 リストはオブジェクト所有者によってソートされる</p> <p>3 リストはオブジェクト名によってソートされる</p> <p>4 リストはオブジェクト・タイプによってソートされる</p> <p>5 リストは変更された日付によってソートされる</p> <p>6 リストは最後に使用された日付によってソートされる</p> <p>2 番目の文字の値は以下のとおりです。</p> <p>A リストは昇順でソートされる</p> <p>D リストは降順でソートされる</p> <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・パネルからなどの他の文脈でつく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>

QMF グローバル変数

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_SCROLL_AMT	なし	04	<p>「QMF」パネルのスクロール移動量を設定します。値は以下のいずれかになります。</p> <p>Csr カーソルに対するスクロール移動量を設定する</p> <p>後方、前方、左、または右にスクロールするかどうかによって、QMF は、カーソルがある位置の行または列をスクロール可能域の最下部、最上部、左端、または右端にスクロールします。</p> <p>Half スクロール移動量をスクロール可能域の半分に設定する</p> <p>Page スクロール移動量を全ページに設定する。これがデフォルトです。</p> <p>n スクロール移動量を n 行数または列数に設定する。n は、1 から 9999 の任意の値に設定できます。</p>
DSQDC_SHOW_PANID	DSQCPDSP	01	<p>CUA に類似したパネルにパネル ID を表示します。値は以下のいずれかになります。</p> <p>0 パネル ID を抑止する。これがデフォルトです。</p> <p>1 パネル ID を表示する</p>

コマンドとプロシーチャーの実行を制御する DSQ グローバル変数

これらのグローバル変数のいずれも SET GLOBAL コマンドで変更できます。

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	<p>ユーザーが表のリストを DB2 UDB (z/OS 版) のロケーションに要求したとき、または現行サーバーが DB2 UDB (z/OS 版) またはワークステーション・データベース・サーバーの場合に、表と視点の別名のリストを取り出すための視点。</p>

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_CC	なし	01	紙送り制御文字が報告書出力フォーマットに出力されないようにします。値は以下のいずれかになります。 0 紙送り制御文字を 1 桁目に出力しない。 1 紙送り制御文字が有効。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 のとき、そこにある表の列情報を取り出すための視点。
DSQEC_COLS_RDB2	なし	31	リモート DB2 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取り出すための視点。
DSQEC_COLS_SQL	なし	31	Database 2 Server (VM/ESA または VSE/ESA 版) データベースの表の列情報を取り出すための視点。
DSQEC_FORM_LANG	なし	01	保存またはエクスポートされたフォームのデフォルトの NLF 言語を設定します。値は以下のいずれかになります。 0 書式に主要 NLF 言語を使用する 1 書式に英語を使用する。これがデフォルトです。
DSQEC_ISOLATION	なし	01	デフォルトの照会分離レベル - 値は以下のいずれかになります。 0 分離レベル UR (非コミット読み取り) 1 分離レベル CS (カーソル固定)。これがデフォルトです。 重要: 値を '0' に設定すると、存在しないデータが QMF 報告書に入ることがあります。QMF 報告書に存在しないデータが入らないようにするには、値を '0' に設定しないでください。
DSQEC_NLFCMD_LANG	なし	01	コマンドに対して予想されている NLF を設定します。値は以下のいずれかになります。 0 コマンドに主要 NLF 言語を使用する。これがデフォルトです。 1 コマンドに英語を使用する

QMF グローバル変数

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_RERUN_IPROC	なし	01	<p>END コマンドの実行後に、呼び出しプロシージ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、呼び出しプロシージャーを再実行しない</p> <p>1 END コマンドの実行後、呼び出しプロシージャーを再実行する。これがデフォルトです。</p> <p>QMF を呼び出しプロシージャーで開始しておいてから、この変数を '0' に設定すると、プロシージャーを再実行する代わりに 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 DATA オブジェクトがパフォーマンスに影響すると思われる場合に、QMF がユーザーにプロンプトを出すかどうかを決定します。値は以下のいずれかになります。</p> <p>0 「報告書プロンプトのリセット (Reset Report Prompt)」パネルは表示されず、QMF が実行中の報告書を完了する。これがデフォルト値です。</p> <p>1 「報告書プロンプトのリセット (Reset Report Prompt)」パネルが表示される。このパネルは、新しいコマンドを実行する前に、現在実行中の報告書の完了、またはリセットを要求します。</p> <p>2 報告書リセット・プロンプト・パネルは表示されず、QMF が現在実行中の報告書をリセットする。</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 他のユーザーとデータを共用しない</p> <p>1 他のユーザーとデータを共用する</p>
DSQEC_TABS_LDB2	なし	31	<p>現行サーバーが DB2 UDB (z/OS 版)、またはワークステーション・データベース・サーバーである場合に、現行サーバーにある表と視点のリストを取り出すための視点。</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>

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_TABS_SQL	なし	31	Database 2 Server (VM/ESA 版または VSE/ESA 版) 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。

CONVERT QUERY の結果を示す DSQ グローバル変数

これらのグローバル変数は、いずれも SET GLOBAL コマンドでは変更できません。

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_nnn	DSQCLnnn	05	変換結果の長さ <i>nnn</i>
DSQQC_QRY_COUNT	DSQCQCNT	03	変換された結果内の照会の数。元の照会が QBE I. または U. 照会以外の場合、値は常に '1' でなければならない。
DSQQC_QRY_LANG	DSQCQLNG	01	変換された照会の言語。値は以下のいずれかになります。 1 SQL 2 QBE 3 入力をプロンプト指示する。
DSQQC_QRY_TYPE	DSQCQTYP	規定なし	変換結果内の最初の語
DSQQC_RESULT_nnn	DSQCQnnn	規定なし	変換結果 <i>nnn</i>

RUN QUERY エラー・メッセージ情報を示す DSQ のグローバル変数

これらのグローバル変数は、いずれも SET GLOBAL コマンドでは変更できません。

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	完全な照会メッセージ・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	SQLSTATE がデータベース・マネージャーから戻された場合の、DSQQM_SQL_RC 内の SQLCODE に関連する SQLSTATE
DSQQM_SUB_TXT_nm	DSQCIQnn	20	置換値 <i>nn</i>
DSQQM_SUBST_VARS	DSQCIQ00	04	置換変数の数

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

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

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

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

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

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

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

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1099
U.S.A.

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

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

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

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

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

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

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

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

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。

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

商標

以下は、IBM Corporation の商標です。

AIX	iSeries
C/370	MVS
CICS	MVS/ESA
CICS/ESA	OS/390
CICS/MVS	PL/I
CICS/VSE	QMF
COBOL/370	RACF
DataJoiner	S/390
DB2	SQL/DS
DB2 Information Integrator	Virtual Machine/Enterprise
DB2 Universal Database	Systems Architecture
Distributed Relational Database Architecture	VM/XA
DRDA	VM/ESA
GDDM	VSE/ESA
IBM	VTAM
IBMLink	WebSphere
IMS	z/OS

Java、またはすべての Java ベースの商標およびロゴ、および Solaris は、米国、およびその他の国における Sun Microsystems, Inc.の商標です。

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。

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

参考文献

以下の資料リストは、個々のライブラリーの全資料を示しているものではありません。これらの資料の注文、または個々のライブラリーの詳細については、IBM 担当員にお問い合わせください。

CICS の資料

CICS Transaction Server for OS390

CICS User's Handbook

CICS Transaction Server for z/OS CICS アプリケーション・プログラミング・リファレンス

CICS Transaction Server for z/OS CICS アプリケーション・プログラミング・ガイド

CICS Transaction Server for z/OS CICS DB2 ガイド

CICS Transaction Server for z/OS CICS リソース定義ガイド

CICS 問題判別の手引き

CICS システム定義の手引き

CICS Transaction Server for z/OS CICS 相互通信ガイド

CICS Performance Guide

CICS Transaction Server (VSE/ESA 版)

User's Handbook

アプリケーション・プログラミング解説書

アプリケーション・プログラミングの手引き

リソース定義ガイド

問題判別の手引き

システム定義の手引き

相互通信の手引き

パフォーマンスの手引き

COBOL の資料

COBOL VSE/ESA 版 言語解説書

COBOL VSE/ESA 版 プログラミングの手引き

DB2 ユニバーサル・データベース (z/OS 版) の資料

DB2 Universal Database (z/OS 版)

インストール・ガイド

管理の手引き

SQL 解説書

コマンド解説書

アプリケーション・プログラミングおよび SQL の手引き

メッセージとコード

ユーティリティー・ガイドおよび解説書

リモート DRDA リクエスターおよびサーバー解説書

IBM DB2 Server (VSE および VM 版)

診断の手引きおよび解説書

DB2 Server (VSE 版) メッセージおよびコード

DB2 Server (VM 版) メッセージおよびコード

DB2 Server (VSE 版) システム管理

DB2 Server (VM 版) システム管理

DB2 サーバー (VSE および VM 版) オペレーション

DB2 サーバー (VSE および VM 版) SQL 解説書

DB2 サーバー (VSE および VM 版) アプリケーション・プログラミング

DB2 サーバー (VSE および VM 版) ISQL ガイドおよびリファレンス

DB2 サーバー (VSE および VM 版) データベース・サービス・ユーティリティー

DB2 サーバー (VSE および VM 版) パフォーマンス・チューニング・ハンドブック

DB2 Universal Database for iSeries

SQL 解説書

SQL Programming with Host Languages

DB2 Universal Database

コマンド解説書

SQL 解説書

メッセージ解説書

DB2 DataJoiner

DataJoiner Application Programming and SQL Reference Supplement

Document Composition Facility (DCF) の資料

DCF and DLF General Information

Distributed Relational Database Architecture (DRDA) の資料

Every Manager's Guide

接続の手引き

Graphical Data Display Manager (GDDM) の資料

GDDM 概説書

GDDM 基本適用業務プログラミングの手引き

GDDM 使用者の手引き

GDDM/VSE Program Directory

GDDM メッセージ

GDDM システムのカスタマイズおよび管理

High Level Assembler (HLASM) の資料

High-Level Assembler for MVS, VM and VSE Programming Guide

High-Level Assembler for MVS, VM and VSE Language Reference

Interactive System Productivity Facility (ISPF) の資料

OS/390

対話式システム生産性向上機能(ISPF) 計画とカスタマイズ

対話式システム生産性向上機能(ISPF) ダイアログ開発者 ガイドとリファレンス

VM

ISPF for VM Dialog Management Guide and Reference

OS/390 の資料

JCL

OS/390 MVS JCL 解説書

OS/390 MVS JCL 使用者の手引き

ページング可能リンク・パック域 (PLPA)

OS/390 Extended Architecture Initialization and Tuning

OS/390 SPL: Initialization and Tuning

VSAM

OS/390 VSAM Administration Guide

OS/390 VSAM Catalog Administration Access Method Services

TSO/E

TSO/E 入門

TSO/E ユーザーズ・ガイド

参考文献

SMP/E

OS/390 SMP/E メッセージおよびコード

OS/390 SMP/E 解説書

OS/390 SMP/E 使用者の手引き

OS PL/I の資料

OS PL/I Programming Language Reference

OS PL/I Programming Guide

REXX の資料

OS/390 環境

TSO/E REXX/MVS User's Guide

TSO/E REXX/MVS Reference

VM 環境

System Product Interpreter Reference

REXX/VM 使用者の手引き

VM/ESA の資料

VM/ESA 計画および管理

VM/ESA コマンド解説書

VSE/ESA の資料

計画

システム・ユーティリティー

問題解決の手引き

用語集

この用語集は、QMF ライブラリーの中で使用されている用語を定義したものです。探している用語が見つからない場合は、本書の索引を参照してください。

[ア行]

値 (value). 表で行と列を割り当てられているデータ・エレメント。

宛先管理テーブル (destination control table - DCT). CICS で、それぞれの一時データ・キューの定義を収めているテーブル。

アプリケーション (application). QMF ライセンス・プログラムを変更せずに QMF の機能を拡張する、QMF ユーザー作成のプログラム。QMF プロシージャー、インストール先定義コマンド、あるいは EXEC を呼び出す CMS または CLIST を呼び出す TSO のコマンドのための RUN コマンドを発行することによって QMF セッションから開始される。

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。(1) アプリケーション・プロセスが接続されるローカルまたはリモートのデータベース・マネージャー。アプリケーション・サーバーは、所要のデータをもっているシステムで実行される。(2) DRDA では、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。DB2 UDB (OS/390 版) では、アプリケーション・サーバーは、完全な DB2 UDB (OS/390 版) サブシステムの一部である。

DB2 (VM および VSE 版) では、アプリケーション・サーバーは、DB2 (VM および VSE 版) データベース・マシンの一部である。

アプリケーション・サポート・コマンド (application-support command). アプリケーション・プログラム内で使用され、アプリケーション・プログラムと QMF 間で情報を交換できるようにする、QMF コマンドの 1 つ。この種のコマンドには、INTERACT、MESSAGE、STATE、および QMF がある。

アプリケーション・リクエスター (application requester). (1) アプリケーション・プロセスからデータベース要求を受け入れ、それをアプリケーション・サーバーに渡す機能。(2) DRDA において、リモート・リレーショナル・データベース管理システムへの送信元。

アプリケーション・リクエスターは、分散接続の QMF 終端を扱う DBMS コードである。QMF が接続するローカル DB2 UDB (OS/390 版) サブシステムは、QMF に対するアプリケーション・リクエスターとして知られる。DB2 UDB (OS/390 版) のアプリケーション・リクエスターはローカル・データベース・マネージャー内でインストールされるからである。したがって、DB2 UDB (OS/390 版) サブシステム全体 (データを含む) がアプリケーション・リクエスターに関連するが、SQL ステートメントは現行ロケーションで処理される。このサブシステムは、「ローカル DB2 UDB (OS/390 版)」と呼ばれる。

用語集

DB2 (VM および VSE 版) では、アプリケーション・リクエスターは QMF と同じ仮想計算機で実行される。すなわち、どのデータベースも本来は、DB2 (VM および VSE 版) アプリケーション・リクエスターに関連付けられていない。

異種 (unlike). 複数の異なる IBM 操作環境を指す語。たとえば、異種分散は DB2 (VM および VSE 版) と DB2 UDB (OS/390 版) 間の分散である。同種 (*like*) と対比。

異常終了 (abend). タスクの異常終了。

一時記憶域 (temporary storage). 現在使用している照会、書式、プロシージャー、プロファイル、報告書、図表、およびデータ・オブジェクトを保管する区域。データ・オブジェクト以外はすべて表示できる。

一時記憶域キュー (temporary storage queue). CICS において、QMF とアプリケーションまたはシステム・サービスの間のオブジェクト転送に用いられる一時記憶域。

一時データ・キュー (transient data queue). CICS において、宛先管理テーブル (DCT) で定義されている名前をもつ記憶域。ここには、後続の内部処理または外部処理に備えてオブジェクトが保管される。

インストール先定義コマンド (installation-defined command). インストール先で作成されたコマンド。QMF は、このコマンドを QMF 自身のコマンドの 1 つとして、またはそのコマンドの組み合わせとして処理する。

インストール先定義フォーマット (installation-defined format). インストール先で定義 (または作成) される日付 / 時刻フォーマット。LOCAL フォーマットとも呼ばれる。

ウィンドウ (window). パネルの全部または一部が表示される画面の長方形の部分。ウィンドウは、画面のサイズ以下に小さくできる。

永続記憶域 (permanent storage). すべての表および QMF オブジェクトが保管されているデータベース。

オブジェクト (object). QMF 照会、書式、プロシージャー、プロファイル、報告書、図表、データ、または表。報告書、図表、およびデータ・オブジェクトは、一時記憶域だけに存在し、データベースに保管できない。表オブジェクトは、データベースにだけ存在する。

オブジェクト名 (object name). QMF ユーザーが所有するオブジェクトを識別する文字ストリング。文字ストリングは最大長が 18 バイトで、英字で始まっていなければならない。用語「オブジェクト名」には、「所有者名」接頭部は含まれない。ユーザーは許可を得た場合にだけ、他のユーザーのオブジェクトにアクセスできる。

オブジェクト・パネル (object panel). 1 つの QMF コマンドの実行後、別の QMF コマンドの実行前に、オンラインで表示できる QMF パネル。そのようなパネルには、ホーム・パネル、報告書パネル、図表パネル、および QMF オブジェクトを表示するすべてのパネルがある。リスト・パネル、ヘルプ・パネル、プロンプト・パネル、および状況パネルは含まれない。

折り返し (wrapping). 『列の折り返し (column wrapping)』および『行の折り返し (line wrapping)』を参照。

オンライン実行 (online execution). オブジェクト・パネルからの、またはファンクション・キーを押すことによるコマンドの実行。

[カ行]

会話 (conversation). LU 6.2 セッション上で、トランザクションを処理する一方で相互に通信することを可能にする 2 つのプログラム間の論理接続。

拡張仮想記憶 (Virtual Storage Extended). ディスク・オペレーティング・システム / 仮想記憶 (DOS/VS) の拡張版であるオペレーティング・システム。VSE は、(1) VSE/ 拡張機能サポート、および (2) ユーザーのデータ処理ニーズを満たす必要のある IBM 提供およびユーザー作成のプログラムから成り立っている。VSE と VSE が制御するハードウェアは、両方で完全なコンピューター・システムを構成する。

拡張構文 (extended syntax). QMF 呼び出し可能インターフェースで使用する QMF コマンド構文。この構文は、呼び出し可能インターフェース・アプリケーションで獲得され、QMF と共用される記憶域に保管される変数を定義する。

拡張対等通信ネットワーク機能 (Advanced Peer-to-Peer Networking). ネットワークを形成する複数のコンピューターがそれぞれ対等のものとして動的に通信できるようにする、分散ネットワークおよびセッション制御のアーキテクチャー。拡張プログラム間通信 (Advanced Program-to-Program Communication (APPC)) と比較。相互接続した複数のシステムが互いに通信でき、プログラムの処理を共用できるようにする SNA 同期データ・リンク制御 LU 6.2 プロトコルのインプリメンテーション。

拡張プログラム間通信 (APPC, Advanced Program-to-Program Communication). 相互接続した複数のシステムが互いに通信でき、プログラムの処理を共用できるようにする SNA 同期データ・リンク制御 LU 6.2 プロトコルのインプリメンテーション。

確認域 (echo area). 指示照会を作成する指示照会基本パネルの一部。

画面 (screen). ユーザーに情報を表示するディスプレイの物理的な表示面。

キーワード・パラメーター (keyword parameter). キーワードと割り当て値から構成される QMF コマンドの 1 要素。

期間 (duration). 数値の後に次の 7 つのキーワード、すなわち、YEARS、MONTHS、DAYS、HOURS、MINUTES、SECONDS、MICROSECONDS の 1 つを続けて表される時間の量。

基本 QMF 環境 (base QMF environment). QMF のインストール時に設定される QMF の英語環境。他の言語環境は、インストール後に設定される。

基本 QMF セッション (primary QMF session). QMF の外部から開始される対話式セッション。このセッションの中で、他のセッションを開始するときは、INTERACT コマンドを使用する。

用語集

基本パネル (primary panel). ユーザーの照会を入れる指示照会のメイン・パネル

行 (row). 表データの水平方向の集合。

行演算子域 (row operator area). QBE ターゲット表または例示表の最左端の列。

行の折り返し (line wrapping). 報告書内の表の行が複数行を占めることができるようにする、表中の行のフォーマット設定。列名の行と列値の各行が、報告書の行の長さに必要なだけの行数に分割される。

区域分離記号 (area separator). 表示される報告書の固定域とその他の区域を区分するバリア。

組み込み関数 (built-in function). スカラー関数または列関数の総称。「関数」と呼ばれることもある。

クライアント (client). サーバーから共用サービスを受ける機能単位。

グループ化行 (grouped row). G 関数または組み込み関数のいずれかで合計される QBE ターゲット表または例示表のデータ行。

グローバル変数 (global variable). 1 回セットすると、1 つの QMF セッションの間使用できる変数。グローバル変数はプロシージャー、照会、または書式で使用できる。ランタイム変数 (run-time variable) と対比。

ゲートウェイ (gateway). 異なるネットワーク・アーキテクチャーをもつ 2 つのコンピューター・ネットワークを接続する機能単位。同一または類似のアーキテクチャーでネットワークまたはシステムを接続するブリッジと異なり、ゲートウェイは、異なるアーキテクチャーのネットワークまたはシステムを接続する。

計算変数 (calculation variable). CALCid は、ユーザー定義の計算値が入る書式用の特殊変数である。CALCid は FORM.CALC パネルで定義される。

結合 (join). リレーショナル操作の 1 つで、同じデータ・タイプの値をもつ列を突き合わせることによって、複数の表からデータを検索することができるようにするもの。

現行オブジェクト (current object). 一時記憶域にあって、現在表示されているオブジェクト。保管オブジェクト (saved object) と対比。

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。CONNECT などの接続タイプのステートメント (これらはアプリケーション・リクエスターによって処理される) を除いて、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 始動プログラム・パラメーターにより指示される。(そのパラメーターが指定されていない場合は、ローカル DB2 UDB (OS/390 版) サブシステム)

構造化照会言語 (Structured Query Language (SQL)). DB2 UDB (OS/390 版) および DB2 (VSE または VM 版) と通信を行なうために使用する言語。記述文で照会を書くために使用される。

顧客情報管理システム (Customer Information Control System (CICS)). リモート端末で入力されるトランザクションをユーザー作成アプリケーション・プログラムによって並行して処理できるようにする IBM ライセンス・プログラム。これには、データベースの構築、使用、維持管理の機能が含まれる。

固定域 (fixed area). 固定列を入れる報告書の部分。

固定列 (fixed columns). ユーザーが水平方向にスクロールしても、その位置から移動しない報告書の列。複数ページの印刷報告書では、これらの列は各ページの左側に繰り返される。

コマンド同義語 (command synonym). インストール先定義コマンドの動詞または動詞 / 目的語の部分。ユーザーは、コマンドの代わりにこれを入力し、その後他の必要情報を続ける。

コマンド同義語表 (command synonym table). それぞれの行にインストール先定義コマンドを記述する表。各ユーザーに、これらの表の 1 つを割り当てることができる。

コマンド・インターフェース (command interface). QMF コマンドを実行するためのインターフェース。QMF コマンドは、アクティブ QMF セッションからしか出せない。呼び出し可能インターフェース (callable interface) と対比。

コミット (commit). データに永続的な変更を加える処理。コミットされると、データ・ロックは解除され、コミットされたばかりのデータを、他のアプリケーションが使用できるようになる。ロールバック (rollback) も参照。

[サ行]

サーバー (server). 共用サービスをネットワーク上でワークステーションに提供する機能単位。

作業単位 (unit of work). (1) アプリケーション・プロセス内の回復可能な操作シーケンス。どの時点でも、アプリケーション・プロセスは、単一の操作単位であるが、アプリケーション・プロセスの寿命には、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係する場合がある。(2) DRDA において、データベース・マネージャーが単一のエンティティーとして扱う一連の SQL コマンド群。データベース・マネージャーは、ある作業単位の間に行われたすべてのデータ変更が実行されたか、あるいはデータ変更が 1 つも行われなかったかのどちらかを確認することによって、データの整合性を確保する。

索引 (index). 特定のキーをもつレコードに迅速にアクセスできるようにする、表中のレコード位置に関するデータの集合。

サブstring (substring). SUBSTR 関数で開始位置と長さを指定する、stringの一部。

サンプル表 (sample tables). QMF に添えて出荷される表。サンプル表のデータは、QMF を初めて体験するユーザーがプロダクト学習のために使用できる。

時刻 (time). 時刻を時、分、必要に応じて秒 (2 つまたは 3 つの部分値) で指定する。

用語集

指示照会 (Prompted Query). 1 組のダイアログ・パネルに対するユーザーの応答に従って作成される照会。

システム・ネットワーク体系 (Systems Network Architecture). ネットワークの構成および操作を通し、またそれを制御することによって情報単位を伝送するための論理構造、フォーマット、プロトコル、および操作手順の記述。

システム・ログ (System Log (SYSLOG)). オペレーターとの間での、ジョブ関連情報、操作上のデータ、通常でないオカレンスの記述、コマンド、およびオペレーターとの間のメッセージを保管できるデータ・セットまたはファイル。

視点 (view). 1 つまたは複数の表にあるデータの代替表示。視点には、視点が定義されるもとになっている 1 つの表または複数の表に入っているすべての列、またはいくつかの列を組み込むことができる。

(2) 照会用に検索されるデータの範囲を定義する 1 つまたは複数のエンティティ。

修飾子 (qualifier). QMF オブジェクトについて用いられているときは、所有者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。たとえば、'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. データの照会または操作を実行する SQL ステートメントまたは QBE ステートメント、もしくはプロンプト指示によって作成されるステートメント。保管照会とは、データベースに保管されている SQL 照会、QBE 照会、または指示照会のことである。一時記憶域にある照会は、QUERY という名前をもつ。

初期化プログラム (initialization program). QMF プログラム・パラメーターを設定するプログラム。このプログラムは、呼び出し可能インターフェースの DSQSCMD で指定される。対話式 QMF のデフォルト・プログラムは DSQSCMD n である。ここで、 n は主要言語の修飾子である (英語の場合は 'E')。

初期プロシージャ (initial procedure). QMF の呼び出し直後に実行される QMF 開始コマンドの DSQSRUN パラメーターで指定される QMF プロシージャの 1 つ。

書式. 報告書または図表の、印刷または表示のための仕様が入っているオブジェクトの 1 つ。一時記憶域にある書式は、FORM という名前をもつ。

ジョブ制御 (job control). VSE で、各ジョブまたはジョブ・ステップの実行準備のために記憶域に呼び出されるプログラム。その機能の例としては、入出力装置を記号名に割り当てたり、スイッチをプログラムで使用するために設定したり、制御ステートメントをログ (または印刷) したり、各ジョブ・ステップの第 1 段階を取り出すことなどがある。

所有者名 (owner name). 特定のオブジェクトを作成したユーザーの許可 ID。

スカラー (scalar). 列、リテラルの値、または他のスカラーを含む式の値。

スカラー関数 (scalar function). 1 つの値を別の値から作成し、関数名の後ろに括弧で囲んだ引き数リストを付けた形で表す操作。

スクロール可能域 (scrollable area). 上、下、左、右に移動可能な表示オブジェクトの部分。

図形データ表示管理プログラム (Graphical Data Display Manager). ピクチャーが、グラフィック・プリミティブに対応する機能ルーチンを通してプロシージャののっとり定義および表示できるようにするルーチンのグループ。

ストリング (string). 1 組の連続した同じタイプの項目。たとえば、文字ストリング。

図表. 報告書の情報のグラフィック表示。

スレッド (thread). アプリケーションの接続の記述、その進行のトレース、リソース機能処理能力の提供、および DB2 UDB (OS/390 版) リソースとサービスへのアクセス可能性の限度の決定を行なう DB2 UDB (OS/390 版) の構造。ほとんどの DB2 UDB (OS/390 版) 機能は、スレッド構造のもとで実行される。

セッション (session). ユーザーがログオンしてからログオフするまでの、ユーザーと QMF 間のすべての対話。

接続性 (connectivity). 異なるシステムが互いに通信できるようにすること。たとえば、DB2 UDB (OS/390 版) アプリケーション・リクエスターと DB2 (VM および VSE 版) アプリケーション・サーバー間の接続性によって、DB2 UDB (OS/390 版) ユーザーは、DB2 (VM および VSE 版) データベースのデータを要求できる。

線形構文 (linear syntax). あるプログラムまたはプロシージャの 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。

線形プロシージャ (linear procedure). REXX コメントで始まっていないプロシージャ。線形プロシージャには、QMF コマンド、コメント、ブランク行、RUN コマンド、置換変数を入れることができる。『ロジックを持つプロシージャ (procedure with logic)』も参照。

ソート優先順位 (sort priority). 検索されたある列のソート値が別の検索列の値のソートを決定する、検索照会の仕様の 1 つ。

関連名 (correlation name). SELECT 照会の FROM 文節で指定された表名の別名。列名と連結させると、列が属する表を識別できる。

総計関数 (aggregation function). 列中のデータを総計する関数の任意のグループ。書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 を使って要求される。

総計変数 (aggregation variable). FORM.BREAK、FORM.CALC、FORM.DETAIL、または FORM.FINAL パネルのいずれかを使って報告書の中に入れられる総計関数。その値は、作成された報告書に、切れ目後書き、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現われる。

[タ行]

ターゲット表 (target table). 例示エレメントを使用して、列を組み合わせたたり、行を組み合わせたたり、報告書に定数値を組み込んだりする空の表。

ダイアログ・パネル (dialog panel). 指示照会基本パネルの一部を重ね書きし、照会の作成を援助するダイアログを拡張するパネル。

タイム・スタンプ (timestamp). 日付および時刻、必要に応じてマイクロ秒 (6 または 7 部分値)。

対話式実行 (interactive execution). コマンドの実行の間にユーザーと QMF との間に発生すべきすべてのダイアログが実際に行われる、QMF の実行。

対話式スイッチ (interactive switch). オンになっていると、アプリケーション・プログラムで QMF コマンドを対話的に実行できるようにする概念上のスイッチ。

対話式セッション (interactive session). ユーザーと QMF が対話できる任意の QMF セッション。QMF INTERACT コマンドを使用すれば、ある対話式セッションから別の対話式セッションを開始できる。

多重仮想記憶 (Multiple Virtual Storage). MVS/ESA プロダクトを意味する。

置換変数 (substitution variable). (1) 値が、グローバル変数またはランタイム変数のいずれかで指定される、プロシーチャー内または照会内の変数。 (2) 値が、グローバル変数で指定される書式の変数。

データベース (database). 複数のユーザーの求めに応じてデータを受け入れ、保管し、提供するための所定の構造をもつデータの集合。 DB2 UDB (OS/390 版) において、表スペースおよび索引スペースが入っている作成済みオブジェクト。 DB2 (VM および VSE 版) では、システムで維持管理される表、索引、サポート情報 (制御情報およびデータ回復情報など) の集合。 OS/2 では、表、視点、索引など、情報の集合。

データベース管理システム (database management system (DBMS)). データベースを定義、作成、操作、制御、管理、使用するためのコンピューター・ベースのシステム。データベース管理システムはまた、データの保全性を保護するためのトランザクション管理およびデータ管理機能をもつ。

データベース管理者 (database administrator). データベースの内容とデータベースへのアクセスを管理する担当者。

データベース・サーバー (database server). (1) DRDA において、アプリケーション・サーバーから受け取った要求のターゲット。 (2) OS/2 において、そのローカル・データベースのデータベース・サービスをデータベースのクライアントに提供するワークステーション。

データベース・マネージャー (database manager). データベースを作成し維持管理するため、またデータベースをアクセスする必要のあるプログラムと通信するために使用されるプログラム。

定位置パラメーター (positional parameter). コマンド内の決まった位置に指定しなければならない QMF コマンドの 1 要素。

デフォルト書式 (default form). 照会が実行されるときに、QMF が作成する書式。保管されている書式が照会で実行されるときは、デフォルト書式は作成されない。

同種 (like). 複数の類似または同一の IBM 操作環境を表現する語。たとえば、同種分散は、互換性のあるサーバー属性レベルによる 2 つの DB2 UDB (OS/390 版) 間の分散である。『異種 (unlike)』 と対比。

トランザクション (transaction). 「作業単位の開始」から「コミット」または「ロールバック」までに発生する作業。

トランザクション・プログラム (transaction program). SNA ネットワークでトランザクションを処理するプログラム。2 種類のトランザクション・プログラムがある。アプリケーション・トランザクション・プログラムとサービス・トランザクション・プログラムである。

トランザクション・プログラム名 (transaction program name). LU 6.2 会話に参加する各プログラムがそれによって識別される名前。通常、接続のイニシエーターは、他の LU で接続を希望するプログラムの名前を識別する。LU 名とともに使用される場合、ネットワークでの特定のトランザクション・プログラムを識別する。

[ナ行]

ヌル (null). ある行のある列に値がないときに使用される特殊な値。ヌル は、ゼロと同じではない。

ヌル値 (null value). *ヌル (null)* を参照。

ネットワーク制御プログラム (Network Control Program (NCP)). 単一ドメイン、複数ドメイン、相互接続ネットワーク機能に対する通信コントローラー・サポートを提供する IBM のライセンス・プログラム。

ノード (node). SNA で、リンクの終点またはネットワークの複数のリンクに共通の接続点。ノードは、ホスト・プロセッサ、通信コントローラー、クラスター・コントローラー、または端末に分散できる。ノードは、ルーティングや他の機能の点でさまざまである。

[ハ行]

パートナー論理装置 (partner logical unit). SNA において、セッションのリモート・システム。

バインド (bind). DRDA で、アプリケーション・プログラム内の SQL ステートメントを、アプリケーション・サポート・プロトコル (およびデータベース・サポート・プロトコル) フロー上でデータベース管理システムに認識させるプロセス。バインドの際に、プリコンパイラーまたはプリプロセッサからの

用語集

出力は、パッケージと呼ばれる制御構造に変換される。さらに、参照されたデータへのアクセス・パスが選択され、一部の許可検査が実行される。(オプションで、DB2 UDB (OS/390 版) では、出力がアプリケーション・プランとなることがある。)

パッケージ (package). アプリケーション・プログラムの SQL ステートメントがリレーショナル・データベース管理システムにバインドされる場合につくられる制御構造。データベース管理システムは、この制御構造を使用して、ステートメントの実行時に見出される SQL ステートメントを処理する。

バッチ QMF セッション (batch QMF session). バックグラウンドで実行する QMF セッション。指定された QMF プロシージャが呼び出されたときに開始し、そのプロシージャが終了したときに終了する。バックグラウンド QMF セッション時には、ユーザー対話もパネル表示対話も行えない。

パネル (panel). ウィンドウに表示するためにグループ分けした情報の特定の配列。パネルには、通知用のテキスト、入力フィールド、ユーザーが選択できるオプション、またはそれらの組み合わせを入れることができる。

パラメーター (parameter). QMF コマンドの 1 要素。この用語は QMF 資料では キーワード・パラメーター と定位置パラメーター の総称として用いられる。

バリエーション (variation). FORM.DETAIL パネルに指定されて、条件に応じて、1 つの報告書または報告書の部分のフォーマットの設定に使用できる、データ・フォーマット設定定義。

引き数 (argument). 独立変数。

日付 (date). 日、月、および年 (3 部分から構成される値) を指す。

日付 / 時刻データ (date/time data). データ・タイプが DATE、TIME、または TIMESTAMP である、表の列内のデータ。

日付 / 時刻デフォルト・フォーマット (date/time default formats). データベース・マネージャーのインストール・オプションで指定される日付 / 時刻フォーマット。そのフォーマットには、EUR、ISO、JIS、USA、または LOC (LOCAL) がある。

表. リレーショナル・データベース・マネージャーの制御下にある名前の付いたデータの集合。表は、固定数の行と列から構成される。

表エディター (Table Editor). 許可ユーザーが照会を作成しなくてもデータベースに変更を加えることができるようにする QMF の対話式エディター。

表データ (tabular data). 列のデータ。データの内容および書式は、FORM.MAIN および FORM.COLUMNS で指定される。

表名域 (table name area). QBE 例示表の最左端の列。

ファンクション・キー表 (function key table). キーを記述するテキストが付いている 1 つまたは複数の QMF パネルのファンクション・キー定義が入っている表。各ユーザーに、これらの表の 1 つを割り当てることができる。

副照会 (subquery). 別の照会 (主照会またはより高位の副照会) の WHERE または HAVING 文節の中に現れる完結した SQL 照会。

プラン (plan). 複数のプログラムの SQL ステートメントがバインド時に一緒に集められてプランが作られる、パッケージの 1 つの形式。

プロシージャ (procedure). QMF コマンドが入っているオブジェクト。プロシージャは単一の RUN コマンドで実行できる。一時記憶域にあるプロシージャは、PROC という名前をもつ。『線形プロシージャ (linear procedure)』 および『ロジックを持つプロシージャ (procedure with logic)』 も参照。

プロシージャ終了スイッチ (procedure termination switch). QMF MESSAGE コマンドでオンにできる概念上のスイッチ。オンの状態になっていると、制御が戻る QMF プロシージャはすべて直ちに終了する。

プロトコル (protocol). 通信をおこなう場合に順守する必要がある、通信システムの機能を律する規則。

プロファイル (profile). ユーザーのセッションの特性情報が入っているオブジェクト。保管プロファイルとは、永続記憶域に保管されているプロファイルのことである。一時記憶域にあるプロファイルは、PROFILE という名前をもつ。ユーザーは、それぞれプロファイルを 1 つしかもつことができない。

プロンプト・パネル (prompt panel). 不完全なまたは正しくない QMF コマンドが出された後で表示されるパネル。

分散作業単位 (distributed unit of work). 分散リレーショナル・データにアクセスする方式。単一の作業単位内で、ユーザーまたはアプリケーションが SQL ステートメントを複数のロケーションに実行依頼できる (ただし、SQL ステートメントと RDBMS は 1 対 1 に対応する)。

DB2 UDB (OS/390 版) では V2R2 において、QMF がサポートするシステム指示アクセスと呼ばれる、制限された形の分散作業単位サポートが採用されている。

分散データ (distributed data). ネットワーク内の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な全体として見ることができ、ローカルにアクセスできるが、実際は複数のロケーションにあるデータベースから構成されているデータベースの 1 つ。

分散リレーショナル・データベース (distributed relational database). すべてのデータがリレーショナル・モデルに従って保管されている分散データベースの 1 つ。

分散リレーショナル・データベース体系 (Distributed Relational Database Architecture). IBM とベンダーのリレーショナル・データベース・プロダクトで使用される分散リレーショナル・データベース処理の接続プロトコル。

別名 (alias). DB2 UDB (OS/390 版) で、同じ、またはリモートの DB2 UDB (OS/390 版) サブシステム中の表あるいは視点を参照する SQL ステートメント中で使用できる代替名。OS/2 で、オブジェクト

用語集

ト、データベース、または LU などのネットワーク・リソースを識別するために使用される代替名。QMF で、ローカルまたはリモート DB2 UDB (OS/390 版) サブシステムで保管されている QMF 表または視点にアクセスするために使用されるローカルに定義された名前。

報告書 (report). データを検索するための照会が出されたとき、または表か視点に対する DISPLAY コマンドが入力されたときに作成されるフォーマット設定されたデータ。

保管オブジェクト (saved object). データベースに保管されているオブジェクト。現行オブジェクト (current object) と対比。

保管オブジェクト (stored object). 永続記憶域に保管されているオブジェクト。現行オブジェクト (current object) と対比。

ホスト (host). ネットワークでのサービスをワークステーションに提供するメインフレームまたは中間サイズのプロセッサ。

[マ行]

無名列 (unnamed column). 例示表に追加される空の列。無名列は、ターゲット表と同様に、列を組み合わせて、行を組み合わせて、あるいは定数値を報告書に組み込んだりするのに使用される。

明細ブロック・テキスト (detail block text). 個々のデータ行に対応している、報告書の本文中のテキスト。

明細ヘッダー・テキスト (detail heading text). 報告書のヘッダーのテキスト。ヘッダーを印刷するかどうかは、FORM.DETAIL で指定する。

[ヤ行]

ヨーロッパ・フォーマット (EUR (European) format). 以下のように日付 / 時刻値を表すフォーマット。

- 日付:dd.mm.yyyy
- 時刻:hh.mm.ss

呼び出し CLIST または EXEC (invocation CLIST または EXEC). QMF を呼び出す (開始する) プログラム。

呼び出し可能インターフェース (callable interface). 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。アプリケーションは、QMF セッション外で実行している場合でも、これらのサービスにアクセスできる。コマンド・インターフェース (command interface) と対比。

[ラ行]

ランタイム変数 (run-time variable). プロシージャーまたは照会が実行される時にユーザーがその値を指定するプロシージャーまたは照会の変数。ランタイム変数の値は、現行のプロシージャーまたは照会でしか使用できない。グローバル変数 (global variable) と対比。

リテラル (literal). プログラミング言語において、1 つの値を直接表す字句単位。文字自体によって値が与えられている文字ストリング。

リモート (remote). ローカル・リレーショナル DBMS 以外のリレーショナル DBMS を指す語。

リモート作業単位 (remote unit of work). (1) アプリケーションがリレーショナル・データベースとは異なるシステム上にあり、単独アプリケーション・サーバーが単一の論理作業単位内ですべてのリモート作業単位要求にこたえるという、SQL 分散処理の形。(2) SQL ステートメントのリモートでの準備と実行を可能にする作業単位。

リモート・データ (remote data). データをアクセスしようとするサブシステム以外のサブシステムが維持管理するデータ。ローカル・データ (local data) と対比。

リモート・データ・アクセス (remote data access). リモートからデータを取り出すための方法。QMF で使用する 2 つのリモート・データ・アクセス機能は、リモート作業単位 と DB2 UDB (OS/390 版) 専用分散作業単位 (システム指示アクセス と呼ばれる) である。

リレーショナル・データベース (relational database). ユーザーが表の集合として認識するデータベース。

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・データベースを定義、作成、操作、制御、管理、使用するためのコンピューター・ベースのシステム。

例示エレメント (example element). QBE 照会において、計算または条件の中で使用される値のシンボル。

例示照会 QBE (Query-By-Example). 照会を図形的に作成するときに使用する言語。詳細については、*QMF 使用の手引き*

例示表 (example table). QBE 照会の枠組み。

列 (column). 表データの垂直方向の集合。特定のデータ・タイプ (たとえば、文字または数値) と名前をもつ。列内の値は、すべて同じデータ特性をもっている。

列関数 (column function). 列内のすべての値に 1 回ずつ適用され、結果として単一の値を戻す操作であり、関数名の後に、括弧で囲んだ 1 つまたは複数の引き数を続けた形で表すもの。

列の折り返し (column wrapping). 1 つの列が数行にまたがることを可能にする報告書内の書式設定の値。長さが列幅を超える値を列に入れるときに、しばしば使用される。

用語集

列ヘッダー (column heading). ユーザーが書式に指定できる列名の代替。列名やラベルとは異なり、データベースには保管されない。

列ラベル (column label). データベースに保管される、データの列の代替記述子。使用すると、列ラベルは書式にデフォルトとして現われるが、ユーザーはそれを変更することができる。

連結 (concatenation). 2 番目のストリングを 1 番目のストリングに付加して、2 つのストリングを 1 つに結合すること。

ローカル DB2 UDB (OS/390 版) (local DB2 UDB (OS/390 版)). DB2 UDB (OS/390 版) と併用する場合は、アプリケーション・リクエスターは QMF と同じ MVS システムで実行される DB2 UDB (OS/390 版) サブシステムの一部である。したがって、DB2 UDB (OS/390 版) サブシステム全体 (データを含む) がアプリケーション・リクエスターに関連するが、SQL ステートメントは現行ロケーションで処理される。このサブシステムは、QMF プランがバインドされる個所である。

QMF が TSO で実行するときには、このサブシステムは、DSQSSUBS 始動プログラム・パラメーターを使用して指定される。QMF が CICS で実行するときには、このサブシステムはリソース管理テーブル (RCT) で識別される。ローカル DB2 UDB (OS/390 版) は、CICS 領域で開始された DB2 UDB (OS/390 版) のサブシステム ID である。

ローカル (local). ユーザーのプロセッサに所在するリレーショナル・データベース、データ、またはファイルを形容する用語。『ローカル DB2 UDB for OS/390』も参照。リモート (*remote*) と対比。

ローカル・エリア・ネットワーク (local area network (LAN)). (1) ローカル・リソース共用のために接続された複数のプロセッサ。 (2) 限定された地域 (たとえば、単独のオフィス・ビル、倉庫、キャンパスなど) 内のネットワーク。

ローカル・データ (local data). データをアクセスしようとするサブシステムが維持管理するデータ。リモート・データ (*remote data*) と対比。

ロールバック (rollback). アプリケーションまたはユーザーが行った、コミットされていないデータベース変更を除去するプロセス。ロールバックが起ると、ロックは解除され、変更が加えられたリソースの状態は、前回コミット、ロールバック、または開始されたときの状態に戻される。コミット (*commit*) も参照。

ロケーション (location). 分散リレーショナル・データベース・システムの特定のリレーショナル・データベース管理システム。各 DB2 UDB (OS/390 版) サブシステムは 1 つのロケーションと見なされる。

ロジックを持つプロシージャー (procedure with logic). REXX のコメントで始まる QMF プロシージャー。ロジックを持つプロシージャーでは、条件つきロジックの実行、計算、ストリングの作成、およびホスト環境へのコマンドの戻しが可能である。『線形プロシージャー (*linear procedure*)』も参照。

論理装置 (logical unit - LU). エンド・ユーザーが他のエンド・ユーザーと通信するために SNA ネットワークにアクセスし、これを通してエンド・ユーザーがシステム・サービス制御点から提供される機能にアクセスするポート。

論理装置タイプ 6.2 (Logical Unit type 6.2 (LU 6.2)). 分散処理環境におけるプログラム間の一般的な通信をサポートする SNA 論理装置タイプ。

[ワ行]

ワークステーション・データベース・サーバー (Workstation Database Server). UNIX および Intel のプラットフォーム上の IBM ファミリーの DRDA データベース・プロダクト。(たとえば、DB2 ユニバーサル・データベース (UDB)、DB2 コモン・サーバー、DB2 パラレル・エディション、および DataJoiner。)

[数字]

1 バイト文字 (single-byte character). 内部表示が 1 バイトから構成される文字。英字は 1 バイト文字の例である。

2 バイト文字 (double-byte character). 2 個の文字バイトを必要とするエンティティ。

2 バイト文字セット (double-byte character set (DBCS)). 各文字が 2 バイトで表現される文字セット。日本語、中国語、韓国語など、256 個のコード・ポイントでは表現できないほど多くの記号をもつ言語には、2 バイト文字セットが必要である。各文字が 2 バイトを必要とするため、DBCS 文字のタイプ、表示、印刷には、DBCS をサポートするハードウェアとプログラムが必要となる。1 バイト文字セット (single-byte character set) と対比。

2 フェーズ・コミット (two-phase commit). 参加しているリレーショナル・データベース管理システムが 1 作業単位を矛盾なくコミットまたはロールバックできるようにするために分散作業単位で使用されるプロトコル。

3 部分名 (three-part name). ロケーション名、所有者 ID、およびオブジェクト名から構成される表または視点の完全修飾名。アプリケーション・サーバー (つまり DB2 UDB (OS/390 版)) でサポートされる場合は、3 部分名を SQL ステートメントで使って、指定されたロケーションの指定された表または視点を検索または更新することができる。

A

ABENDx. 異常終了問題のキーワード。

APAR. プログラム診断依頼書 (Authorized Program Analysis Report)。

APPC. 拡張プログラム間通信 (Advanced program-to-program communication)。

C

CICS. 顧客情報管理システム (Customer Information Control System)。

CMS. 会話型モニター・システム (Conversational Monitor System)。

用語集

CP. VM 用制御プログラム (Control Program)。

CSECT. 制御セクション (Control section)。

D

DATA. 検索照会で戻された情報を収容する一時記憶域内のオブジェクト。表に含まれ、報告書でフォーマット設定された英数字で表現される情報。

DB2 (AIX 用). AIX 用の DATABASE2。QMF のリレーショナル・データのデータベース・マネージャ。

DB2 UDB for OS/390. DB2 ユニバーサル・データベース (OS/390 版) (IBM のリレーショナル・データベース管理システム)。

DBCS. 2 バイト文字 セット (double-byte character set)。

DBMS. データベース管理システム (Database management System (DBMS))。

DOC. 文章問題のキーワード。

DRDA. 分散リレーショナル・データベース体系 (Distributed Relational Database Architecture (DRDA))。

E

EBCDIC. 拡張 2 進化 10 進コード (Extended Binary-Coded Decimal Interchange Code)。

G

GDDM. 図形データ表示管理プログラム (Graphical Data Display Manager)。

H

HELP . エラー・メッセージ、QMF パネル、または QMF コマンドとそのオプションについての追加情報。

HTML. ハイパーテキスト・マークアップ言語 (Hypertext Markup Language)。WWW で表示される文書用の標準化されたマークアップ言語。

I

ICU. 対話式図表ユーティリティー (Interactive Chart Utility)。

INCORROUT. 正しくない出力のキーワード。

ISO (国際標準化機構) フォーマット (International Standards organization format). 以下のように日付 / 時刻値を表すフォーマット。

- 日付:yyyy-mm-dd
- 時刻:hh.mm.ss

ISPF. 対話式システム生産性機能 (Interactive System Productivity Facility)。

IXF. 統合交換フォーマット (Integration Exchange Format)。各種のソフトウェア・プロダクト間で表データを転送するためのプロトコル。

J

JCL. OS/390 用のジョブ制御言語。

JIS (日本工業規格) フォーマット (Japanese Industrial Standard format). 以下のように日付 / 時刻値を表すフォーマット。

- 日付:yyyy-mm-dd
- 時刻:hh:mm:ss

L

LOOP. 無限ループの問題のキーワード。

LU. 論理装置 (Logical Unit)。

LU 6.2. 論理装置タイプ 6.2。

M

MSGx. メッセージ問題のキーワード。

MVS/ESA. 多重仮想記憶 / エンタープライズ・システム体系 (IBM のオペレーティング・システム)。

N

NCP. ネットワーク制御プログラム。

NLF. 各国語機能 (National Language Feature)。米国英語以外の言語を 1 つ選択できるようにする、QMF とともに使用できるいくつかのオプション・フィーチャーの 1 つ。

NLS. 各国語サポート (National Language Support)。

P

PERFM. パフォーマンス問題のキーワード。

PSW. プログラム状況ワード (Program status word)。

PTF. プログラム一時修正 (Program temporary fix)。

Q

QMF 管理権限 (QMF administrative authority). 最低でも、Q.PROFILES 制御表に対する挿入または削除の権限。

QMF 管理者 (QMF administrator). QMF 管理権限を持つ QMF ユーザー。

QMF コマンド (QMF command). QMF 言語の一部である任意のコマンドを指す。インストール先定義コマンドを含まない。

QMF セッション (QMF 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 間で行なうすべての対話。

R

RDBMS. リレーショナル・データベース管理システム (Relational database management system)。

REXX. 再構造化拡張実行プログラム (Restructured extended executor)。

S

SBCS. 1 バイト文字セット (single-byte character set)。

SNA. システム・ネットワーク体系 (Systems Network Architecture (SNA))。

SNAP ダンプ (SNAP dump). 異常終了時に QMF が生成する 1 つまたは複数の記憶域の内容の動的ダンプ。

SQL. SQL (構造化照会言語)。

SQLCA. 構造化照会言語連絡域 (Structured Query Language Communication Area)。

SSF. ソフトウェア・サポート機能 (Software Support Facility)。現行のすべての APAR および PTF に関する情報の保管および検索を可能にする IBM オンライン・データベースの 1 つ。

T

- TP.** トランザクション・プログラム (Transaction Program)。
- TPN.** トランザクション・プログラム名 (Transaction program name)。
- TSO.** タイム・シェアリング・オプション (Time Sharing Option)。

U

- USA (米国) フォーマット (United States of America format).** 以下のように日付 / 時刻値を表すフォーマット。
- 日付: mm/dd/yyyy
 - 時刻: hh:mm xM

V

- VM.** 仮想計算機 (Virtual Machine) (IBM のオペレーティング・システム)。VM/ESA 環境の総称。
- VSE.** 拡張仮想記憶 (Virtual Storage Extended) (IBM のオペレーティング・システム)。VSE/ESA 環境を総称的指す語。

W

- WAIT.** 無限待ち状態問題のキーワード。

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

- 関数呼び出し 133
- 言語インターフェース 131
- 高水準アセンブラー (HLASM) 131
- サンプル・プログラム 134
- マクロ 259
- 連絡域 145
- CICS 131
 - サンプル・プログラム 135
 - MVS 146
 - VSE 147
- CMS サンプル・プログラム 140
- TSO サンプル・プログラム 140, 151

アプリケーション 2, 3

- インプリメンテーション・メソッド 3
- 開発 1
- コマンド 1
 - 概要 51
 - 処理 20
- INTERACT 58
- コマンド同義語 3
- 制御 2
- タイプ 1
- データ・レコード 100
- デバッグ 125
- プロシージャ 7
- ロジックを持つプロシージャ 4
- 2 か国語使用の 45
- CICS 環境 4
- ISPF 要件 39

アプリケーション (続き)

- QMF
 - インプリメンテーション・メソッド 3
 - 制御 2
 - もとの実行 3
- QMF とのプログラミング・インターフェース 4
- QMF もとの実行 3
- アプリケーションのデバッグ
 - ISPF の使用 43
 - PDF ダイアログ・テスト 43
- 移行
 - C 157
 - COBOL 175
- 移行情報 89
- 移植性 21
- 一時記憶域 25
 - キュー 124
 - 制約 4
 - 変更 25
- 一時データ・キュー
 - 一時記憶域キューとの対比 124
- インポート 79
- オブジェクト・レベル情報 89
- 指示照会オブジェクト 104
- 書式オブジェクト 109
- QMF の外で作成された表 82
- エクスポート
 - オブジェクトのタイプ 79
 - 書式オブジェクト 79
 - 日付 / 時刻情報 109
 - フォーマット 79
- エクスポート後のオブジェクトでの制御域 92
 - 書式ファイルのレコード 92
 - 報告書ファイルのレコード 92
 - T レコード 95
 - V レコード 93
- エラーのある命令でのシグナル 13

オブジェクトの終わりレコード (E) 99

[カ行]

- 拡張接続機能 (ECF) 6
- 環境 4
- 関数呼び出し
 - C 155
 - DSQCIC 155
 - DSQCICE 156
- 関連プロダクト 6
- キーワード 68
 - START コマンド 68
- 切れ目
 - パネル 108, 236
- 切れ目パネル 236
- グローバル変数
 - 規則 67
 - 作成 64
 - 設定 65
 - SET GLOBAL コマンド 65
 - 変数の作成 64
 - DSQEC_RERUN_IPROC 54
 - RUW を介して使用する QMF 261
- 言語変数
 - (DSQEC_NLFCMD_LANG) 46
- 戻りコード 26
- 呼び出し可能インターフェース 4
- ID 39
- 現行ロケーション 51
- コマンド
 - アプリケーション 51
 - インターフェース 3
 - サンプル・プログラム 30
 - 説明 29
 - プログラムからの呼び出し 31

コマンド (続き)

インターフェース (続き)

戻りコード 33

要件 4

SELECT サービス 31

環境 37

言語変数 46

システム固有 10

長さ 20

戻りコード 13

2 か国語使用のアプリケーション
47

INTERACT 58

QMF の開始 54

RUN 10

SET GLOBAL 64

コマンド同義語

作成 74

説明 6

表 75

例 3

IRM 6

NLF 表 49

コメント

アプリケーション・データ・レコ
ード 100, 109

エクスポート・フォーマット
109

[サ行]

指示照会

インポート / エクスポート・ファ
イル仕様 122

エクスポート 102

エクスポート・フォーマット
233

データ・フォーマット 231

表番号 233

フィールド番号 233

指示照会オブジェクト 88, 90, 102,
231, 233

システム・アプリケーション体系
(SAA)

アプリケーション 4

言語サポート 4

システム・アプリケーション体系
(SAA) (続き)

プログラムの移植性 21

呼び出し可能インターフェース
19, 131

QUERY コマンド 21, 224

START コマンドのキーワード
68

初期プロシージャ 54

作成 8

名前の指定 8

保管 9

2 か国語使用のアプリケーション
47

CONNECT コマンド 9

書式

サンプル・ヘッダー 90

データ・フォーマット 231

表番号 236

フィールド番号 236

変換 110, 113

書式オブジェクト 88, 90, 106, 108,
112, 231, 236

図形データ・フォーマット

(GDF) 87

図表オブジェクト 87

セッション環境 48

線形プロシージャ

抑止 62

STOPPROC オプション 62

[夕行]

対話の切り替え

(DSQAO_INTERACT) 60

対話モード

初期プロシージャ 8

GDDM ICU 58

QMF 58

置換変数 10

値の割り当て 10

グローバル変数の設定 10

構文 10

REXX プログラム呼び出し 16

データ

オブジェクト

フォーマット 231

IXF のエクスポート後のフォ
ーマット 245

タイプの幅 233

転送速度 85

レコードのエクスポート 233

2 進数 85

D レコード 249

データベース・リモート接続 9

同義語 74

特記事項 277

トレース

オフにする 127

トレース定義の作成 43

例 128

A オプション 126

ISPF コマンド 43

L オプション 125

トレース・データ出力 125, 127

[ハ行]

パネル

現行の 58

対話式 58

ホーム 54

バリエーション・パネル 110, 113

引き数 12

表

オブジェクト

インポート 81

インポート / エクスポートの
規則 84

インポート / エクスポート・
ファイル仕様 122

処理 81

記述レコード (T) 94, 246

行レコード (R) 98

作成、QMF の外での 245

指示照会の番号 233

書式番号 236

報告書番号 241

表レコード 246

フォーマット
 インポート 231
 エクスポート 231
 エンコード
 定義 79
 指示照会オブジェクト 233
 書式オブジェクト 236
 データのエクスポート 233
 表 81
 ヘッダー・レコード 231, 232
 報告書オブジェクト 241
 列データ 249
 IXF 79, 245
 不完全データ・プロンプト 85
 複数言語環境 48
 プログラム呼び出し 20
 プロダクト・インターフェース・マ
 クロ 259
 ヘッダー・レコード
 オブジェクト・レベル 89
 書式オブジェクト 90
 長さの計算 82
 フィールド 88
 フォーマット 231, 232
 IXF 246
 変換可能なアプリケーション 50
 変数
 エラー処理 26
 規則 67
 グローバル 10, 261
 置換 10
 設定 10
 置換 10
 入力プロンプト 11
 プール 19
 QMF 32
 rc 13
 ホーム・パネル 8, 54
 報告書
 エクスポート 81
 エクスポートの使用 113
 エクスポート例 113
 エクスポート・フォーマット
 241
 エクスポート・レコード 115
 オブジェクト 3

報告書 (続き)
 エクスポート・フォーマット
 241
 表番号 241
 フィールド番号 241
 横方向 241
 行 (L) レコード 116
 行データ 98
 サンプル・ヘッダー 91
 データ・フォーマット 231
 テキストの表示 62
 パネル 3
 表示 59
 表データ 94
 表番号 241
 フィールド番号 241
 ミニ・セッション 76
 横方向 241
 横方向エクスポート 119
 レコード 88
 HTML 243

[マ行]

マクロ、プロダクト・インターフェ
 ース 259
 ミニ・セッション
 報告書 76
 無効なコマンド 77
 有効なコマンド 77
 戻りコード
 コマンド・インターフェース 34,
 35
 非ゼロ 13
 メッセージ 14
 呼び出し可能インターフェース
 23

[ヤ行]

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 25
 アプリケーションのデバッグ
 125
 言語 4, 19, 131
 コマンド 23

呼び出し可能インターフェース (続
 き)
 コマンド処理情報 19
 サンプル・プログラム 4
 アセンブラー 134
 C 158
 COBOL 175
 説明 19
 プログラム 4
 マクロ 259
 戻りコード 23
 連絡域 4
 エラー処理 26
 定義 21
 フィールドの設定 22
 C 162
 COBOL 178
 ロジックを持つプロシージャか
 らの呼び出し 16
 CICS のもとでの実行 27
 COBOL 172
 FORTRAN 189
 GET GLOBAL コマンド 57
 ISPF 4
 PL/I 205
 REXX
 使用 4
 説明 222
 で呼び出す 7
 連絡変数 222
 QMF の始動 24
 START コマンド 4
 構文 67, 68
 QMF の開始 24

[ラ行]

リソース管理テーブル 27
 リポジトリ・マネージャー
 /OS/390 6
 リモート作業単位
 コマンドの動作 51
 レコード
 アプリケーション・データ
 (*) 100
 可変長フォーマット 91

レコード (続き)

- 固定フォーマット 88
 - データ (D) 249
 - データ値 (V) 92
 - データ継続 (C) 118
 - 表の記述 (T) 246
 - フォーマット 233
 - ヘッダー 88
 - 列 (C) 247
 - 列 247, 249
 - データ・フォーマット 249
 - C レコード 247
- ## 連絡域
- 定義 21
 - COBOL 172, 178
 - FORTRAN 197
 - PL/I 112

[数字]

- 2 か国語使用のオブジェクト 45

A

- A オプション、デバッグ用の 126
- ADDRESS コマンド 15, 37
- ARG ステートメント 12
- AS (アプリケーション・システム) 6

C

C 言語

- 移行情報 157
 - インターフェースの要件 157
 - 関数呼び出し 155
 - サンプル・プログラム 158
 - 呼び出し可能インターフェース 154
 - 連絡域 154
 - マッピング 155
 - DSQCOMM 162
 - CICS 163
 - ISPF 168
 - TSO 167
- ### CALL 命令 15

CICS

- アセンブラー 4
 - MVS の要件 146
 - VSE の要件 147
- データ・キュー 4
 - 一時データ・キュー 123
 - IXF フォーマット 245
 - QMF オブジェクトを転送するための使用 81
- プログラム開始パラメーターの指定変更 27
- 呼び出し可能インターフェース 4
- 領域 27
- COBOL プログラム 181
- CONNECT コマンド 9
- DB2 対話 27
- VSE/ESA
 - アセンブラー 147
 - C プログラム 164
 - HLASM プログラム 147

COBOL

- 移行情報 175
 - 関数呼び出し 173
 - 区切り文字 180
 - サンプル・プログラム 175
 - 実行の要件 180
 - マクロ 259
 - 呼び出し可能インターフェース 172
 - 連絡域 172
 - CICS 181
 - DSQCOMM 178
 - ISPF 186
 - TSO 185
 - VM 184
 - VSE 182
- ### CONNECT コマンド
- 初期プロシージャ 9
 - 説明 51
 - プロシージャ 9
 - 例 51
 - SQL/DS 9
 - VM 10

D

- DB2 (IBM DATABASE 2)
 - CICS 要件 27
- DSQABFA 140
- DSQABFAC 135
- DSQADPAN 70
- DSQALANG 70
- DSQCIA 133
- DSQCIX サブルーチン 224
- DSQCOMM
 - アセンブラー 145
 - エラー処理 26
 - 定義 21
 - フィールドの設定 22
 - メッセージ・テキスト 129
- C 155, 162
- COBOL 172
- DSQCOMM 145
- DSQCOMM 162
- DSQDC_DISPLAY_RPT グローバル変数 76
- DSQRUN 54
- DSQSBSTG 71
- DSQSCMD 71
- DSQSDBCS 72
- DSQSDBNM 72
- DSQSQDBQN 72
- DSQSQDBQT 72
- DSQSQDBUG 72
- DSQSDCSS 73
- DSQSIROW 73
- DSQSMODE 73
- DSQSPILL 73
- DSQSPRID 73
- DSQSRSTG 73
- DSQSRUN 74
- DSQSSPQN 74
- DSQSSUBS 74
- DSQSUSER 74

E

- EDIT コマンド 42
- END コマンド
 - コマンド・インターフェース 31

END コマンド (続き)

初期プロシージャの再実行 8
セッション・タイプ 54

初期プロシージャ 54

初期プロシージャを伴わない 55

バッチ・モード 56

呼び出し可能インターフェースによる始動 54

INTERACT コマンド 55

説明 54

対話式セッション 228

EXIT コマンド 56

EXPORT コマンド

データ・オブジェクト 81

表オブジェクト 81

CICS の使用 123

DATA 81

IXF オプション 245

F

FORTRAN

関数呼び出し 190

サンプル・プログラム 192

マクロ 259

呼び出し可能インターフェース 189

連絡域 189

CMS 199

DSQABFF 192

DSQCOMM 197

ISPF 202

MVS 201

TSO 201

G

GDDM (図形データ表示管理プログラム) 58

対話式図表ユーティリティ 6

GET GLOBAL コマンド 23, 57

H

HTML 報告書 81, 120, 243

I

ICU (対話式図表ユーティリティ) 6, 58

IFX

OUTPUTMODE=BINARY 256

IMPORT コマンド

実行中のエラーと警告 112

定義 79

CICS の使用 123

DATA オプション 84

INTERACT コマンド

コマンド形式 60

セッション

始動 55

終了 55, 56, 60

書式 58

説明 58

ISPF (対話式システム生産性向上機能)

パネル生成 6

呼び出し可能インターフェース 41

IXF

サンプル・レコード 255

OUTPUTMODE=CHARACTER 255

IXF (統合交換フォーマット)

文字 249

2 進数 85, 249

IXF ロング・ネーム・サポート 246

L

L オプション、デバッグ用の 125

LAYOUT コマンド 106

Lotus 1-2-3/M 6

M

MESSAGE コマンド

オプション 61

説明 61

線形プロシージャ実行の抑止 62

テキストの表示 62

トレース 128

MESSAGE コマンド (続き)

例 64

ISPF パネル 61

QMF ヘルプ・パネル 61

N

NLF (各国語フィーチャー)

言語 49

言語 ID 39

セッション環境 48

定義 45

パネル要件 49

複数言語環境 48

P

PARSE ARG ステートメント 12

PDF 43

PL/I

関数呼び出し 206

サンプル・プログラム 208

マクロ 259

呼び出し可能インターフェース 205

連絡域 205

CICS 214, 215

DSQABFF 208

DSQCOMM 212

ISPF 219

MVS 214

TSO 218

VM 216

Q

QBE (例示照会プログラム)

インポート / エクスポート・ファイル仕様 122

エクスポート・フォーマット 122

R

- RUN コマンド
 - 組み込まれた置換変数 16
 - 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

- SAA (システム・アプリケーション体系)
 - アプリケーション 4
 - 言語サポート 4
 - プログラムの移植性 21
 - 呼び出し可能インターフェース 19, 131
- QUERY コマンド 21, 224
- START コマンドのキーワード 68
- SAVE DATA コマンド 81
- SELECT コマンド 29, 41
- SET GLOBAL コマンド
 - 構文 65
 - 変数値の入力プロンプト 12
 - 呼び出し可能インターフェース 23, 65
- SQL 照会オブジェクト 86
- SQL/DS
 - リモート接続 9
 - CICS/VSE 要件 27
 - CONNECT コマンド 10
- START コマンド
 - インターフェース連絡域 21
 - エラーのデバッグ 129
 - キーワード
 - リスト 68
 - DSQADPAN 70
 - DSQALANG 70
 - DSQSBSTG 71
 - DSQSCMD 71
 - DSQSDBCS 72
 - DSQSDBNM 72
 - DSQSDBQN 72
 - DSQSDBQT 72
 - DSQSDBUG 72

START コマンド (続き)

キーワード (続き)

- DSQSDCSS 73
- DSQSIROW 73
- DSQSMODE 73
- DSQSPILL 73
- DSQSPRID 73
- DSQSRSTG 73
- DSQSRUN 74
- DSQSSPQN 74
- DSQSSUBS 74
- DSQSUSER 74
- 構文 67, 68
- QMF の始動 24, 67
- STATE コマンド 32

T

TSO

- アセンブラー呼び出し可能インターフェース・プログラム 151
- アセンブラー・プログラム 151
- C プログラム 167
- C 呼び出し可能インターフェース・プログラム 167
- REXX プログラム 228
- REXX 呼び出し可能インターフェース・プログラム 228

V

- VSE CICS 122



プログラム番号: 5625-DB2

Printed in Japan

SC88-9836-00



日本アイ・ビー・エム株式会社

〒106-8711 東京都港区六本木3-2-12