

DB2 Query Management Facility
Version 12.1

*Développement d'applications DB2
QMF*



DB2 Query Management Facility
Version 12.1

*Développement d'applications DB2
QMF*



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques» à la fin du document.

Première édition - octobre 2016

Réf. US: SC27-8878-00

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.ibm.com/ca/fr> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette édition s'applique à la version 12.1 d'IBM DB2 Query Management Facility (QMF) Classic Edition et Enterprise Edition, qui sont des fonctions d'IBM DB2 12 for z/OS (5650-DB2) et IBM DB2 11 for z/OS (5615-DB2). Elle s'applique aussi à la version 12.1 d'IBM DB2 QMF for z/OS (5697-QM2), qui est un outil IBM DB2 for z/OS autonome. Ces informations s'appliquent à toutes les versions et modifications ultérieures, sauf indication contraire dans de nouvelles éditions.

© Copyright IBM Corporation 1982, 2016.

Table des matières

Avis aux lecteurs canadiens	vii
A propos de ces informations	ix
Connaissances préalables requises	ix
Mises à jour du service et informations de support	x
Conventions pour la mise en évidence.	x
Lecture des diagrammes syntaxiques	x
Comment envoyer vos commentaires.	xii
Chapitre 1. Présentation du développement d'applications QMF	1
Présentation du développement d'applications dans QMF	1
Utilisation de l'application	1
Interaction principale avec l'application	1
Démarrage de l'application à partir d'une session QMF	2
Outils de développement d'applications QMF disponibles	3
Procédures QMF	3
Interfaces de programmation d'applications dans QMF	4
Conventions pour les informations NLF (National Language Feature)	5
Chapitre 2. Procédures utilisées en tant qu'applications	7
Procédures de lancement	7
Commande QMF CONNECT dans une procédure.	9
Variables de substitution dans les procédures	9
Spécification de valeurs dans la commande RUN.	10
Spécification de valeurs dans l'écran d'invite de la commande RUN	10
Variables REXX dans les procédures navigationnelles	11
Transmission d'arguments à une procédure navigationnelle	12
Instructions de traitement d'erreurs REXX dans les procédures navigationnelles	12
Bifurcation vers des sous-routines de traitement d'erreurs	12
Messages avec l'instruction REXX EXIT	13
Appel de programmes REXX à partir d'une procédure navigationnelle	14
Appel de programmes REXX sans variables de substitution	14
Appel de programmes REXX contenant des variables de substitution	15
Chapitre 3. Interface callable et applications QMF	17
Présentation de l'interface callable	17
Considérations relatives à l'utilisation de l'interface callable de QMF	17
Zone de communication d'interface (DSQCOMM)	18
Codes retour	20
Commandes pour l'utilisation de l'interface callable	21
Démarrage de QMF à partir d'une application.	21
Exécution de l'application d'interface callable	22
Interface callable dans QMF	22
Traitement des erreurs.	22
Exécution de programmes d'interface callable sous CICS	23
Chapitre 4. Emission de commandes QMF à partir d'une boîte de dialogue ISPF	25
Exemple d'écriture d'un programme utilisant l'interface de commande	25
Appel de l'interface de commande.	26
Commande END	26
Variables dans l'interface de commande	27
Codes retour de l'interface de commande	27

Chapitre 5. ADDRESS QRW et environnement de commande QMF	31
Chapitre 6. Ecriture d'applications QMF utilisant des services ISPF	33
Démarrage et exécution de QMF à partir d'une application ISPF.	33
Exécution de requêtes contenant des variables.	34
Démarrage d'un programme utilisant des services ISPF à partir de QMF	34
Services ISPF dans une procédure navigationnelle	35
Commande EDIT avec ISPF	35
ISPF et débogage d'applications	36
Chapitre 7. Ecriture d'applications bilingues	39
Comparaison des environnements en anglais et NLF	39
Création d'objets à utiliser dans les applications bilingues	41
Variable de langue de commande	41
Procédures de lancement dans les applications bilingues	42
Commandes en anglais uniquement	43
Environnements multilingues	43
Création d'applications traduisibles	44
Chapitre 8. Commandes QMF dans des applications	45
Commandes conçues pour des applications.	45
CONNECT	45
END.	46
EXIT.	46
GET GLOBAL (syntaxe étendue)	47
INTERACT	48
MESSAGE.	50
SET GLOBAL (syntaxe étendue)	53
START	55
TRACE.	63
Commandes utilisables dans une mini-session de rapport RUN QUERY	68
Synonymes de commande	70
Chapitre 9. Exportation et importation d'objets	71
Opérations possibles avec un fichier UNIX, une file d'attente de données CICS ou un ensemble de données TSO exporté.	71
Exportation et sauvegarde de données	72
Exportation d'objets données et de tables de base de données	72
Exportation de données ou de tables au format QMF	73
Exportation de données ou de tables au format IXF	80
Exportation de données ou de tables au format XML	97
Exportation de données ou de tables au format CSV	100
Règles et informations relatives à l'exportation et à l'importation d'objets données et de tables	101
Exportation de formats, de rapports et de requêtes créées de façon interactive.	102
Format général du fichier exporté	102
Exportation d'un format	113
Considérations relatives aux objets format QMF dans des applications	122
Exportation d'un rapport standard	124
Exportation d'un rapport au format HTML	127
Exportation d'un rapport sans informations de contrôle	129
Exportation d'un rapport de style transversal.	129
Exportation d'une requête créée de façon interactive	131
Vérification de la validité du format de la requête créée de façon interactive exportée	139
Importation de formats et de requêtes créées de façon interactive	140
Procédures et requêtes SQL.	141
Exportation de diagrammes basés sur des formats et de requêtes QBE	142
Spécifications de taille pour les objets externalisés	142
Considérations relatives au stockage.	145
Files d'attente de données CICS	145
Ensembles de données TSO	146

Chapitre 10. Débogage des applications QMF	149
Débogage de vos applications d'interface appelable.	149
Option L pour le traçage	149
Option A pour le traçage	149
Désactivation du traçage	150
Allocation de la sortie de données de trace QMF	150
Commande QMF MESSAGE pour le traçage	151
Erreurs relatives à la commande START ou à d'autres commandes QMF	152
Chapitre 11. Spécifications de langage de programmation pour l'utilisation de l'interface appelable	153
Interface en langage Assembler	153
Mappage de la zone de communication d'interface pour le langage Assembler (DSQCOMMA)	153
Appels de fonction pour le langage Assembler	154
Exemple de programmation Assembler	156
DSQCOMM pour Assembler	162
Exécution de vos programmes Assembler dans CICS	164
Exécution de vos programmes Assembler dans TSO	164
Interface en langage C	167
Mappage de la zone de communication d'interface pour le langage C (DSQCOMMC)	167
Appels de fonction pour le langage C	168
Exemple de programmation en langage C	170
DSQCOMM pour le langage C	172
Exécution de vos programmes en langage C dans CICS	174
Exécution de vos programmes en langage C dans TSO	175
Interface en langage COBOL	177
Mappage de la zone de communication d'interface pour le langage COBOL (DSQCOMMB)	177
Appels de fonction pour le langage COBOL	178
Service ISPF LIBDEF avec le langage COBOL.	179
Exemple de programmation COBOL	179
DSQCOMM pour COBOL	181
Considérations relatives à l'exécution du programme d'interface appelable COBOL	183
Exécution de vos programmes en COBOL dans CICS	183
Exécution de vos programmes en COBOL dans TSO	184
Interface en langage FORTRAN	186
Mappage de la zone de communication d'interface pour le langage FORTRAN (DSQCOMMF)	186
Appels de fonction pour le langage FORTRAN	188
Exemple de programmation en langage FORTRAN.	189
DSQCOMM pour le langage FORTRAN	192
Exécution de vos programmes en langage FORTRAN	194
Interface en langage PL/I	196
Mappage de la zone de communication d'interface pour le langage PL/I (DSQCOMML)	196
Appels de fonction pour le langage PL/I	197
Exemple de programmation en langage PL/I.	199
DSQCOMM pour le langage PL/I	201
Exécution de vos programmes sous CICS	203
Exécution de vos programmes sous TSO	204
Interface en langage REXX	206
Variables de communication d'interface pour REXX	207
Appel de fonction pour REXX.	208
Exemple de programmation en langage REXX	208
Exécution de vos programmes REXX	209
Exemple REXX d'utilisation d'une boucle INTERACT	210
Annexe A. Macros d'interface de produit	213
Annexe B. Variables globales QMF	215
Convention de dénomination des variables globales QMF	215
Définition et affichage des valeurs des variables globales.	216
Variables globales relatives aux informations d'état non liées au profil	216

VARIABLES GLOBALES STOCKANT LES INFORMATIONS D'ÉTAT ASSOCIÉES AUX PROFILS	222
VARIABLES GLOBALES ASSOCIÉES À CICS	223
VARIABLES GLOBALES ASSOCIÉES À UN MESSAGE GÉNÉRÉ PAR LA COMMANDE LA PLUS RÉCENTE	224
VARIABLES GLOBALES ASSOCIÉES À L'ÉDITEUR DE TABLE	224
VARIABLES GLOBALES QUI CONTRÔLENT LES DIFFÉRENTS AFFICHAGES	227
VARIABLES GLOBALES CONTRÔLANT LE MODE D'EXÉCUTION DES COMMANDES ET DES PROCÉDURES	237
VARIABLES GLOBALES QUI STOCKENT LES RÉSULTATS DE LA COMMANDE CONVERT QUERY	249
VARIABLES GLOBALES QUI AFFICHENT DES INFORMATIONS SUR LE MESSAGE D'ERREUR RUN QUERY	249
VARIABLES GLOBALES QUI STOCKENT LES VALEURS D'ENTRÉE D'ÉCRAN	250
REMARQUES	261
Documentation sur l'interface de programmation	263
Marques	263
GLOSSAIRE DES TERMES ET ACRONYMES.	265
INDEX	279

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos de ces informations

IBM® DB2 Query Management Facility for TSO and CICS est un outil intégré, puissant et fiable qui offre des fonctions de requête et de génération de rapports destinées à vous aider à accéder et à présenter les données à partir des bases de données relationnelles suivantes :

- DB2 for z/OS
- DB2 for Linux, UNIX and Windows
- DB2 for iSeries
- DB2 Server for VSE et VM

Ces informations sont écrites pour les programmeurs d'applications chargés de développer des applications faisant appel aux fonctions QMF. Ces rubriques vous aident à :

- Prendre des décisions de conception de programmation d'application
- Faire un choix parmi différentes techniques de programmation
- Comprendre le mode d'utilisation de la commande QMF et des interfaces appelables
- Ecrire des applications bilingues

Des exemples de programmation spécifiques sont fournis pour Assembler, C, FORTRAN, COBOL, PL/I et REXX.

Connaissances préalables requises

Vous devez connaître les composants qui constituent votre environnement, ainsi que certains concepts et termes, avant de commencer à programmer des applications pour QMF.

Produits


Pour développer des applications pour QMF, il peut s'avérer utile de connaître certains ou tous les produits suivants, selon votre environnement et vos besoins métier :

- Système d'exploitation z/OS.
- DB2, gestionnaire de base de données pour QMF.
- Time Sharing Option (TSO), environnement prenant en charge DB2 QMF et les produits associés.
- Interactive System Productivity Facility (ISPF), gestionnaire de dialogues pour DB2 QMF.
- Customer Information Control System (CICS), système de communication de données à caractère général et de traitement des transactions en ligne. CICS fournit l'interface entre DB2 QMF et z/OS.
- Produit Graphical Data Display Manager (GDDM) de base permettant d'afficher les panneaux et de créer des diagrammes. Vous pouvez également utiliser GDDM pour fournir des services d'impression à partir de QMF.
- Assembler, C, COBOL, FORTRAN, PL/I ou REXX permettant de créer des applications d'interfaces appelables pour QMF.

Concepts

Les applications QMF vous permettent de travailler avec les objets QMF et d'exécuter les fonctions QMF à partir d'une application écrite dans l'un des langages pris en charge par QMF. Pour aborder ces informations, vous devez savoir écrire des requêtes et des procédures, mettre en forme des rapports et modifier la base de données.

Information associée:

 The IBM Publications Center
Recherche de publications décrivant les produits.

Mises à jour du service et informations de support

Pour obtenir des mises à jour du service et des informations de support (par exemple, les groupes de correctifs, PTF, questions fréquemment posées (FAQ), notes techniques, informations sur la résolution des problèmes ou encore téléchargements), consultez la page Web suivante :

Site Web du service de support logiciel IBM

Conventions pour la mise en évidence

Les conventions utilisées sont mises en évidence ci-après :

- **Gras**. Indique des commandes ou contrôles de l'interface utilisateur tels que des noms de zones, de dossiers, d'icônes ou d'options de menus.
- Espacement fixe : correspond aux exemples de texte à saisir telles quelles.
- *Italique* : Indique les titres d'autres publications ou la mise en évidence de termes clés. Indique également des variables devant être remplacées par une valeur.

Lecture des diagrammes syntaxiques

Les règles ci-après s'appliquent aux diagrammes de syntaxe utilisés dans ces informations.

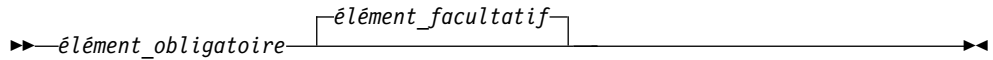
- Les diagrammes de syntaxe se lisent de la gauche vers la droite, du haut vers le bas, en suivant la ligne. Les conventions suivantes sont utilisées :
 - Le symbole >>--- indique le début d'un diagramme de syntaxe.
 - Le symbole ---> indique que le diagramme de syntaxe se poursuit à la ligne suivante.
 - Le symbole >--- indique qu'un diagramme de syntaxe a été commencé sur la ligne précédente.
 - Le symbole --->< indique la fin du diagramme de syntaxe.
- Les éléments obligatoires apparaissent sur la ligne horizontale (chemin principal).

▶▶—*élément_obligatoire*—————▶▶

- Les éléments facultatifs apparaissent au-dessous du chemin principal.

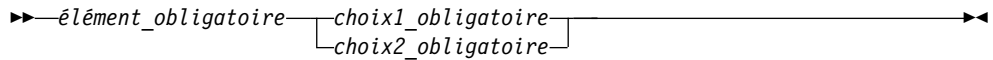
▶▶—*élément_obligatoire*—————▶▶
 └─*élément_facultatif*—┘

Si un élément facultatif apparaît au-dessus du chemin principal, il n'a aucun effet sur l'exécution de l'élément de syntaxe et il est utilisé uniquement pour la lisibilité.

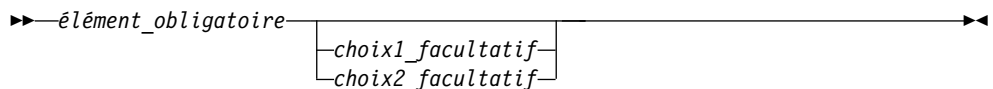


- Si vous avez le choix entre plusieurs éléments, ces derniers apparaissent verticalement, sous forme de pile.

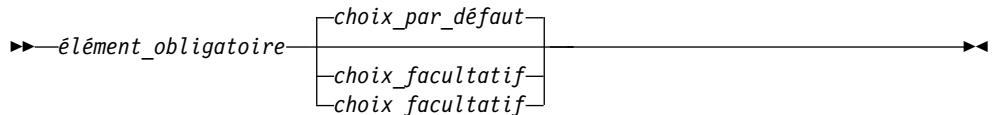
Si le choix de l'un des éléments est *obligatoire*, un élément de la pile apparaît sur le chemin principal.



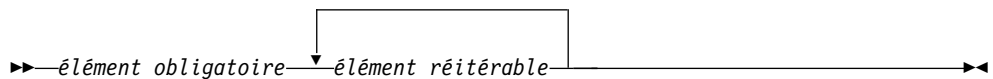
Si le choix parmi les éléments est facultatif, la totalité de la pile apparaît sous le chemin principal.



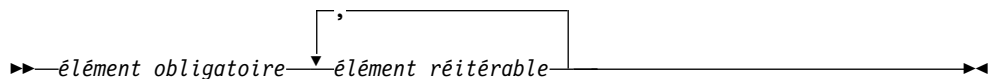
Si l'un des éléments est celui par défaut, il apparaît au-dessus du chemin principal et les autres choix au-dessous.



- Une flèche pointant vers la gauche, au-dessus de la ligne principale, indique un élément pouvant être répété.



Si la flèche de répétition contient une virgule, vous devez séparer les éléments répétés par une virgule.



Une flèche de répétition au-dessus d'une pile indique que vous pouvez répéter les éléments dans la pile.

- Les mots clés et leurs éventuelles abréviations minimales, apparaissent en majuscules. Ils doivent être saisis exactement comme indiqué. Les variables apparaissent en caractères italiques minuscules (par exemple, *nom-colonne*). Elles représentent des noms ou des valeurs fournis par l'utilisateur.
- Séparez les mots clés et les paramètres par un espace en l'absence d'autres signes de ponctuation dans le diagramme.

- Entrez des signes de ponctuation, des parenthèses, des guillemets, des opérateurs arithmétiques et d'autres symboles tels qu'ils apparaissent dans le diagramme.
- Les notes de bas de page sont signalées par un nombre entre parenthèses ; par exemple, (1).

Comment envoyer vos commentaires

Vos commentaires nous sont utiles car ils nous permettent d'améliorer la qualité et la pertinence des informations fournies. Pour tout commentaire sur le présent document ou sur toute autre documentation, utilisez l'une des options suivantes :

- Utilisez le formulaire de commentaires en ligne, qui se trouve à l'adresse suivante :
<http://www.ibm.com/software/data/rcf>
- Envoyez vos commentaires par courrier électronique à l'adresse comments@us.ibm.com. Indiquez bien le nom du document, son numéro de référence, la version de votre produit et, éventuellement, l'emplacement spécifique du texte sur lequel repose votre commentaire (par exemple, un numéro de page ou de tableau).

Chapitre 1. Présentation du développement d'applications QMF

Le terme *développement d'applications* fait référence au processus de création d'une procédure ou d'une application dans QMF.

Vous pouvez utiliser autant de fonctions de QMF que vous le souhaitez dans vos applications. Par exemple, vous pouvez écrire des applications qui :

- Exécutent des requêtes ou des procédures
- Exportent ou importent des tables et des objets QMF
- Affichent ou impriment des rapports ou des diagrammes
- Permettent à l'utilisateur de saisir ou de modifier des données dans la base de données
- Permettent à l'utilisateur d'apporter simultanément des modifications globales à plusieurs objets

Vous pouvez également écrire des applications qui fournissent des fonctions utiles aux utilisateurs dans QMF. Par exemple, écrivez une commande qui permet d'imprimer des rapports QMF à un emplacement distant ou une touche de fonction qui permet de générer un diagramme des résultats de vente hebdomadaires.

Présentation du développement d'applications dans QMF

Le mot *application* peut revêtir plusieurs significations. Dans QMF, une application est une procédure ou un programme qui émet des commandes QMF et utilise des services QMF pour accomplir une tâche métier spécifique.

Le développement d'applications inclut les actions suivantes :

- Compréhension du problème que votre procédure ou votre application doit résoudre
- Conception de la procédure ou de l'application
- Ecriture du code, des messages associés et des écrans d'aide

Utilisation de l'application

Il existe deux principaux types d'applications QMF : celles dans lesquelles les utilisateurs interagissent avec l'application et celles dans lesquelles l'application peut être démarrée à partir de QMF.

Interaction principale avec l'application

Si votre application est destinée à des utilisateurs qui ne sont pas familiarisés avec QMF, vous voudrez probablement qu'ils interagissent principalement avec votre application.

Il se peut que vous ne souhaitiez pas qu'ils sachent que QMF est actif. Dans ce cas, votre application utilisera les services QMF, mais s'exécutera en dehors de QMF. Votre programme émettra des commandes QMF uniquement si nécessaire.

Supposons que vous écriviez une application utilisant des services QMF. Cette application fournira à l'utilisateur une interface pilotée par menu. Dans l'exemple suivant, votre application contrôle QMF. L'utilisateur interagit uniquement avec l'interface utilisateur et ne sait pas que QMF est actif.

Si l'utilisateur sélectionne l'option 1, l'application exécute une procédure QMF qui lance une requête et imprime le rapport généré.

```
J & H Fournisseurs
Système d'information

Sélectionnez l'une des options de recherche suivantes :

1. Imprimer le rapport de ventes mensuel
2. Créer un rapport
3. Modifier les informations dans la base de données
4. Fermer l'application

====> 1
```

Figure 1. Exemple d'écran défini par une application

Démarrage de l'application à partir d'une session QMF

Si vos utilisateurs maîtrisent QMF, vous voudrez peut-être qu'ils considèrent votre application comme une extension ou une personnalisation de QMF. Dans ce cas, vous devrez configurer votre application pour qu'elle s'exécute dans QMF.

Supposons que vous écriviez une application appelée SEND_TO qui envoie un rapport QMF d'un utilisateur à un autre.

Vous vous attendez à ce que vos utilisateurs exécutent l'application à partir de l'environnement QMF. Les utilisateurs peuvent utiliser la ligne de commande pour émettre un synonyme de commande QMF appelé SEND_TO (que vous créez). Vous pouvez également affecter l'application à une touche de fonction qui exécute automatiquement votre application.

Après avoir généré un rapport, l'utilisateur peut l'envoyer à l'utilisateur Smith en saisissant la commande QMF personnalisée SEND_TO SMITH sur la ligne de commande QMF.


```

REPORT                                     LINE 1   POS 1   79

NOM      DEPT  POSTE      SALAIRE      COMM
-----
DANIELS   10  DIR      19260.25      -
JONES     10  DIR      21234.00      -
LU        10  DIR      20010.00      -
MOLINARE  10  DIR      22959.20      -
HANES     15  DIR      20659.80      -
KERMISCH  15  EMPL     12258.50      110.10
NGAN      15  EMPL     12508.20      206.60
ROTHMAN   15  VENTES   16502.83     1152.00
JAMES     20  EMPL     13504.60      128.20
PERNAL    20  VENTES   18171.25      612.45
SANDERS   20  DIR      18357.50      -
SNEIDER   20  EMPL     14252.75      126.50
ABRAHAMS  38  EMPL     12009.75      236.50
MARENGHI  38  DIR      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

```

Figure 2. Exemple de saisie par un utilisateur d'une commande QMF personnalisée

Outils de développement d'applications QMF disponibles

Vous pouvez écrire des applications qui utilisent des procédures QMF et des interfaces de programmation d'applications dans QMF.

Vous pouvez créer des synonymes de commande pour appeler vos procédures et applications. Un synonyme de commande est une commande qui exécute une commande QMF, TSO ou CICS. Pour créer un synonyme de commande, saisissez la commande et sa définition dans une table de synonymes de commande. Lors de l'initialisation, QMF charge la table de synonymes de commande qui est spécifiée dans le profil QMF de l'utilisateur qui a démarré QMF.

Procédures QMF

Les procédures QMF sont des objets QMF qui s'exécutent dans QMF et émettent des commandes QMF. Les procédures QMF peuvent s'exécuter de façon interactive ou non. Vous pouvez exécuter une procédure de façon non interactive en démarrant QMF en mode de traitement par lots. Si vous utilisez QMF for TSO, vous pouvez également démarrer QMF en tant que procédure mémorisée DB2 for z/OS.

Les procédures QMF peuvent exécuter les commandes QMF disponibles sur votre site. QMF fournit deux types de procédures : les procédures linéaires et les procédures navigationnelles.

- Les *procédures linéaires* contiennent uniquement des commandes QMF et des commentaires. Vous pouvez utiliser des procédures linéaires dans tous les environnements pris en charge dans QMF.
- Les *procédures navigationnelles* combinent des commandes QMF avec la logique REXX qui vous aide à créer des programmes plus puissants. Vous pouvez utiliser des procédures navigationnelles dans tous les environnements pris en

charge dans QMF, sauf CICS. Les procédures navigationnelles peuvent contenir des commandes QMF et toute instruction valide dans un programme REXX, notamment les commandes système.

QMF fournit une procédure d'initialisation du système et d'autres méthodes qui vous permettent d'exécuter des commandes et de définir des variables globales avant que l'utilisateur ne voie s'afficher l'écran d'accueil QMF.

Concepts associés:

Chapitre 2, «Procédures utilisées en tant qu'applications», à la page 7

Vous pouvez écrire des applications entièrement sous forme de procédures. Si vous utilisez QMF dans l'environnement CICS, vous pouvez écrire des *procédures linéaires*, qui sont des procédures incluant uniquement des commandes QMF ou des synonymes qui émettent des commandes QMF. Si vous utilisez QMF dans l'environnement TSO, vous pouvez écrire des *procédures navigationnelles* en plus des procédures linéaires. Les procédures navigationnelles peuvent inclure non seulement des commandes QMF et des synonymes de commande, mais aussi des fonctions et des instructions REXX.

Référence associée:

«Variables globales contrôlant le mode d'exécution des commandes et des procédures», à la page 237

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

Interfaces de programmation d'applications dans QMF

Il existe deux interfaces de programmation d'applications dans QMF : l'interface callable et l'interface de commande.

Interface callable

Vous pouvez utiliser l'interface callable de QMF pour créer une application qui est appelée en dehors de QMF. Cette application démarre ensuite une session QMF et envoie des commandes à QMF en vue de leur exécution.

L'interface callable est une interface de programmation qui permet d'accéder aux services QMF. Elle est disponible pour les langages de programmation et les environnements présentés dans le tableau ci-dessous.

Tableau 1. Prise en charge de l'interface callable

	CICS	TSO	Mode de traitement par lots z/OS natif
Assembler	×	×	×
C	×	×	×
COBOL	×	×	×
FORTRAN		×	×
PL/I	×	×	×
REXX		×	×

QMF prend en charge toutes les versions de ces langages de programmation qui sont acceptées par DB2 for z/OS.

Interface de commande (TSO uniquement)

Vous pouvez créer des applications qui envoient des commandes à QMF à partir d'une boîte de dialogue ISPF à l'aide de l'interface de commande de QMF. QMF communique avec la boîte de dialogue ISPF via le pool de variables ISPF par le biais de cette interface de commande. QMF doit être démarré avant le lancement de l'application ISPF.

L'interface de commande est disponible uniquement avec ISPF ; elle ne l'est pas dans CICS. Vous pouvez écrire une application d'interface de commande dans n'importe quel langage de programmation pris en charge par ISPF.

Concepts associés:

Chapitre 3, «Interface callable et applications QMF», à la page 17

Les langages de programmation peuvent utiliser l'interface callable QMF pour exécuter des commandes QMF.

Chapitre 4, «Emission de commandes QMF à partir d'une boîte de dialogue ISPF», à la page 25

Vous pouvez émettre des commandes QMF à partir d'une boîte de dialogue ISPF qui s'exécute sous QMF à l'aide de l'interface de commande QMF.

Information associée:

➡ Langages de programmation et méthodes pour développer des programmes d'application

Consultez les informations relatives aux langages de programmation pris en charge par DB2 for z/OS.

Conventions pour les informations NLF (National Language Feature)

DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Utilisez les fonctions NLF pour permettre aux utilisateurs d'entrer des commandes QMF, d'afficher l'aide et de réaliser des tâches QMF dans des langues autres que l'anglais. Les fonctions NLF sont installées en tant que fonctions distinctes de DB2 QMF.

Toutes les tâches décrites dans ces informations peuvent être réalisées pour le produit QMF de base (en anglais) et pour une fonction NLF. Les procédures pour les sessions de base et NLF sont identiques ; toutefois, des considérations spéciales relatives aux utilisateurs NLF sont identifiées.

Certains noms de programmes ou d'ensembles de données présentés dans ces informations comportent la variable *n*, qui indique que ce caractère peut varier dans le nom. Dans le tableau suivant, remplacez la variable *n* par un identificateur de langue nationale d'un caractère (NLID) qui correspond à la fonction de langue que vous utilisez. Le tableau contient également les noms à l'aide desquels QMF reconnaît chaque langue.

Tableau 2. Fonctions NLF de QMF et informations d'identification

Fonction de langue nationale (NLF)	Identificateur (NLID)	Nom utilisé par QMF pour cette fonction NLF
Anglais	F	ENGLISH
Anglais en lettres capitales	U	UPPERCASE

Tableau 2. Fonctions NLF de QMF et informations d'identification (suite)

Fonction de langue nationale (NLF)	Identificateur (NLID)	Nom utilisé par QMF pour cette fonction NLF
Français (Canada)	C	FRANCAIS CANADIEN
Danois	Q	DANSK
Français	F	FRANCAIS
Allemand	D	DEUTSCH
Italien	I	ITALIANO
Japonais (Kanji)	K	NIHONGO
Coréen (Hangeul)	H	HANGEUL
Portugais (Brésil)	P	PORTUGUES
Espagnol	S	ESPAÑOL
Suédois	V	SVENSKA
Français (Suisse)	Y	FRANCAIS (SUISSE)
Allemand (Suisse)	Z	DEUTSCH (SCHWEIZ)

La fonction Anglais en lettres capitales utilise l'anglais, mais convertit l'ensemble du texte en majuscules. Les majuscules permettent aux utilisateurs qui travaillent avec des écrans Katakana d'utiliser le produit et d'obtenir une aide en ligne et des messages en anglais.

Chapitre 2. Procédures utilisées en tant qu'applications

Vous pouvez écrire des applications entièrement sous forme de procédures. Si vous utilisez QMF dans l'environnement CICS, vous pouvez écrire des *procédures linéaires*, qui sont des procédures incluant uniquement des commandes QMF ou des synonymes qui émettent des commandes QMF. Si vous utilisez QMF dans l'environnement TSO, vous pouvez écrire des *procédures navigationnelles* en plus des procédures linéaires. Les procédures navigationnelles peuvent inclure non seulement des commandes QMF et des synonymes de commande, mais aussi des fonctions et des instructions REXX.

Si vous écrivez une application qui fonctionne sur une procédure dans la mémoire temporaire QMF, vous ne pouvez pas écrire votre application sous forme de procédure. Lorsque vous exécutez une procédure, elle devient la procédure en cours dans la mémoire temporaire QMF.

Concepts associés:

«Services ISPF dans une procédure navigationnelle», à la page 35

Vous devez effectuer un transfert de la boîte de dialogue du programme QMF vers une boîte de dialogue de commande ISPF pour exécuter des commandes ISPF à partir d'une procédure QMF navigationnelle qui s'exécute sous ISPF.

Procédures de lancement

Une procédure de lancement est une procédure qui s'exécute immédiatement après le démarrage de votre session QMF. Utilisez le paramètre DSQSRUN pour spécifier le nom de cette procédure et comprendre le comportement des procédures de lancement dans des situations spécifiques.

Vous pouvez utiliser le paramètre DSQSRUN :

- Avec la commande DSQQMF n lorsque QMF est démarré en mode interactif (où n représente un identificateur de langue nationale d'un caractère qui correspond à la fonction de langue que vous utilisez).
- Avec la commande QMF START lorsque QMF est démarré via l'interface appelable.

En mode par lots TSO, ISPF et z/OS natif, les applications peuvent également définir des paramètres de programme à l'aide d'un programme REXX. Le programme est spécifié par le paramètre DSQSCMD de la commande QMF START. Etant donné que QMF for CICS ne prend pas en charge le langage REXX, vous devez indiquer tous les paramètres de programme dans la commande START à l'aide de DSQSMODE=I. Cette valeur pour DSQSMODE spécifie une opération interactive, dans CICS. Le mode par défaut à partir de l'interface appelable est B (mode par lots).

Considérations relatives à l'écriture de procédures de lancement

Tenez compte des points suivants lors de l'écriture et de l'utilisation d'une procédure de lancement :

- Par défaut, QMF réexécute la procédure de lancement lorsque l'utilisateur émet la commande END dans une session interactive de QMF démarré à l'aide de DSQQMF n . (La variable n est un identificateur de langue nationale d'un

caractère qui correspond à la fonction de langue que vous utilisez). La variable globale DSQEC_RERUN_IPROC indique si la procédure de lancement est réexécutée. La valeur par défaut de cette variable est 1 pour réexécuter la procédure ; un paramètre 0 empêche la réexécution de la procédure de lancement.

Dans les programmes d'interface callable, la procédure de lancement n'est jamais réexécutée afin que cette variable globale n'affecte pas vos programmes d'interface callable.

- Lorsque vous écrivez des procédures de lancement à utiliser dans une session QMF interactive, évitez d'utiliser l'écran d'accueil comme écran en cours à la fin de la procédure. Dans ce cas, QMF n'affiche pas d'écran en mode interactif à la fin de la procédure. Si aucune erreur grave ne se produit et que DSQEC_RERUN_IPROC est défini sur 1, QMF réexécute la procédure de lancement sans interaction avec l'utilisateur. Cela provoque une boucle ininterrompue qui peut apparaître même sans démarrage de QMF.

Pour éviter la création d'une boucle ininterrompue, prenez en compte l'une des options suivantes :

- Vérifiez que l'écran en cours à la fin de la procédure n'est pas l'écran d'accueil.
 - Assurez-vous que la procédure contient une commande QMF EXIT ou INTERACT.
 - Définissez DSQEC_RERUN_IPROC sur zéro (0).
- Le nombre de perluètes (&) que vous devez utiliser avant le nom des variables de substitution dans les procédures de lancement peut varier en fonction de votre environnement. Par exemple, vous pouvez spécifier DSQSRUN sous la forme suivante :

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

Le nombre de perluètes que vous devez spécifier avec VAR1 dépend du fait que QMF s'exécute en mode par lots CICS, TSO ou z/OS natif. Leur nombre varie si ISPF est présent et si le programme qui démarre QMF est écrit en langage REXX.

Procédures de lancement et unité d'oeuvre éloignée

La procédure de lancement doit être stockée sur le système sur lequel vous démarrez QMF (système local).

Lorsque vous utilisez la commande QMF CONNECT dans des procédures de lancement, vous devez vous déconnecter de l'emplacement distant. En d'autres termes, codez l'application pour qu'elle se reconnecte à votre emplacement d'origine avant de coder une commande END pour rappeler votre procédure de lancement. Si vous êtes toujours connecté à l'emplacement distant, une erreur est renvoyée. La déconnexion est également requise si la commande QMF CONNECT est émise à partir de la ligne de commande lors d'une session interactive configurée par une procédure de lancement.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5
DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

«Mots clés de la commande START», à la page 56

Vous pouvez spécifier des mots clés dans la commande START.

Commande QMF CONNECT dans une procédure

Pour vous connecter à un autre ID utilisateur ou à une base de données DB2 éloignée afin d'utiliser le support d'unité d'oeuvre éloignée, émettez la commande QMF CONNECT. Vous pouvez utiliser cette commande dans une procédure linéaire ou navigationnelle.

Lorsque vous écrivez des procédures qui utilisent la commande QMF CONNECT pour accéder à des bases de données éloignées, tenez compte des points suivants :

- Si vous êtes connecté à une base de données éloignée et que vous émettez une commande RUN PROC, cette procédure et tous les objets qu'elle utilise doivent être stockés sur la base de données éloignée.
- Toutes les commandes QMF de la procédure sont exécutées dans la mémoire temporaire QMF sur le système où QMF est en cours d'exécution (système local). Toutefois, tous les objets utilisés par ces commandes QMF (requêtes, procédures ou formats) doivent être définis dans la base de données à l'emplacement en cours (système distant).
- Les commandes qui affectent la base de données (instructions SQL, requêtes QMF ou mises à jour EDIT TABLE) s'exécutent à l'emplacement en cours.
- Si la procédure contient des commandes spécifiques au système (CICS ou TSO), ces commandes s'exécutent sur le système où QMF est en cours d'exécution (système local). Si vos procédures contiennent des commandes spécifiques au système qui ne s'exécutent pas sur le système où QMF est en cours d'exécution, votre procédure ne peut pas s'exécuter correctement.
- Les fichiers ou files d'attente de données utilisés dans une commande spécifique au système doivent se trouver sur le système où QMF est en cours d'exécution (système local).
- Si votre site utilise TSO et bénéficie de la prise en charge RACF pour les mots de passe en majuscules/minuscules, vérifiez que l'option CASE de votre profil QMF est définie sur MIXED. Sinon, QMF convertit toutes les entrées en majuscules, provoquant ainsi l'échec de la commande CONNECT. Lorsque l'option est définie sur CASE=MIXED, n'oubliez pas d'indiquer aux utilisateurs de l'application QMF qu'ils doivent saisir toutes les entrées en majuscules, car QMF ne reconnaît que les commandes en majuscules.
- Si la procédure est transmise sous forme de paramètre dans une instruction CALL qui démarre QMF for TSO à l'aide de l'interface de procédure mémorisée, la procédure ne peut pas accéder aux bases de données éloignées. Les commandes de la procédure qui tentent d'accéder à une base de données éloignée doivent être supprimées ou mises en commentaire avant l'exécution de la procédure avec cette interface.

VARIABLES DE SUBSTITUTION DANS LES PROCÉDURES

Vous pouvez utiliser des variables de substitution QMF dans les procédures linéaires et les procédures navigationnelles.

Une variable de substitution est toute variable utilisable dans une commande QMF. Une variable de substitution est toujours précédée d'une perluète (&). Vous pouvez affecter une valeur à une variable de substitution de l'une des façons suivantes :

- Définition de variables globales à l'aide de la commande SET GLOBAL
- Spécification de valeurs dans la commande RUN
- Spécification de valeurs dans l'écran d'invite de la commande RUN

Référence associée:

«SET GLOBAL (syntaxe étendue)», à la page 53

Pour créer vos propres variables globales et les utiliser dans des commandes QMF en tant que variables de substitution, émettez la commande SET GLOBAL. Vous pouvez également utiliser la commande SET GLOBAL pour définir des valeurs pour les variables globales QMF prédéfinies, qui commencent par "DSQ".

Spécification de valeurs dans la commande RUN

Vous pouvez affecter une valeur à une variable de substitution à l'aide de la commande RUN :

Si la procédure est linéaire, affectez la valeur de variable dans la commande RUN PROC comme suit :

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

Si la procédure est navigationnelle, affectez la valeur de variable dans la commande RUN PROC comme suit :

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

Seule la procédure nommée SCHEDULE a accès à la valeur de &TYPE.

Dans cet exemple :

- La valeur de variable VACATION est placée entre apostrophes car la valeur est une chaîne de caractères.
- TYPE est précédé de doubles perluètes (&&) pour indiquer que la valeur est définie dans l'instruction RUN à transmettre à la procédure nommée SCHEDULE. Si l'instruction RUN spécifie &TYPE, la procédure qui contient cette instruction invite l'utilisateur à saisir la valeur.

Cette valeur pour la variable de substitution est active uniquement au sein de la procédure qui la définit. La valeur n'est pas active dans une procédure ou un module appelé à partir de la procédure qui la définit.

Spécification de valeurs dans l'écran d'invite de la commande RUN

Lorsque vous exécutez une requête ou une procédure avec une variable de substitution, vous pouvez lui affecter une valeur dans la commande RUN ou via une variable globale. Toutefois, si la variable figurant dans la requête ou procédure ne contient pas de valeur, QMF affiche un écran d'invite de commande RUN. Vous pouvez alors indiquer la valeur pour cette variable dans cet écran.

Cette valeur pour la variable de substitution est active uniquement au sein de la procédure qui la définit. La valeur n'est pas active dans une procédure ou un module appelé à partir de la procédure qui la définit.

Saisie de variables dans les procédures linéaires

Dans une procédure linéaire, QMF analyse la procédure à la recherche de variables de substitution et les résout avant de traiter les commandes. L'utilisateur est invité à saisir toutes les variables avant l'exécution de la procédure.

Saisie de variables dans les procédures navigationnelles

Dans une procédure navigationnelle, l'utilisateur n'est pas invité à saisir de variables tant que REXX ne détecte pas l'instruction contenant les variables. Pour

que l'utilisateur ne voie s'afficher qu'une seule invite, vous pouvez exécuter une procédure distincte qui demande la saisie des variables.

Par exemple, vous ne souhaitez être invité à saisir les variables de substitution LASTNAME et DEPT_NUM qu'une seule fois. Ces variables peuvent figurer sur deux lignes différentes dans votre procédure navigationnelle :

```
/* Cette procédure exécute deux requêtes, qui affichent le rapport après chaque */  
/* exécution de requête. */  
  
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";  
"INTERACT"  
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";
```

Figure 3. Saisie de valeurs de variable dans une procédure navigationnelle

Ajoutez cette ligne au début de votre procédure navigationnelle, immédiatement après les lignes de commentaire :

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

Important : Toutes les procédures navigationnelles doivent commencer par au moins une ligne de commentaire.

Dans cette commande, PROMPT_ME est une procédure navigationnelle comme celle présentée ci-dessous, qui contient une ligne de commentaire et aucune instruction :

```
/* Cette procédure est une procédure fictive de saisie */
```

La procédure complète inclut la commande RUN PROC pour la procédure PROMPT_ME qui invite à saisir des variables :

```
/* Cette procédure exécute deux requêtes, qui affichent le rapport après chaque */  
/* exécution de requête */  
  
"RUN PROC PROMPT_ME (&LASTNAME, &DEPT_NUM";  
"RUN QUERY REG_QUERY (&&LASTNAME=&LASTNAME";  
"INTERACT"  
"RUN QUERY REG2_QUERY (&&DEPT_NUM=&DEPT_NUM";
```

Figure 4. Procédure navigationnelle invitant à saisir des variables

Vous pouvez également utiliser la commande SET GLOBAL pour la saisie simultanée de toutes les valeurs de la procédure, comme dans l'exemple suivant :

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

VARIABLES REXX DANS LES PROCÉDURES NAVIGATIONNELLES

Vous pouvez utiliser des variables REXX dans une procédure navigationnelle. Les valeurs pour ces variables ne sont reconnues qu'au sein de la procédure dans laquelle vous les avez définies.

Vous pouvez utiliser des variables REXX dans une procédure navigationnelle de l'une des façons suivantes :

- Copie d'une variable REXX vers une variable QMF à l'aide de la commande SET GLOBAL
- Copie d'une variable globale vers une variable REXX à l'aide de la commande GET GLOBAL

- Utilisation de variables REXX dans vos instructions REXX

Transmission d'arguments à une procédure navigationnelle

Pour les procédures navigationnelles, QMF fournit une option ARG sur la commande RUN PROC. Utilisez cette option pour transmettre des arguments ou des valeurs, à une procédure navigationnelle.

Utilisez l'option ARG lorsque vous exécutez une procédure qui contient une instruction REXX PARSE ARG ou ARG, comme dans l'exemple présenté ci-dessous.

```

PROC                WILDE.SHOW_ARGS                MODIFIED    LINE 1

/*****/
/* Cette procédure montre comment utiliser l'option 'ARG=' sur la      */
/* commande RUN                                                         */
/* PROC.                                                                */
/*****/
parse upper arg query_name form_name
"RUN QUERY" query_name "(FORM="form_name

```

Figure 5. Transmission de valeurs de variable à l'aide de l'option ARG de la commande RUN PROC

La commande RUN pour cette procédure se présente comme suit :

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

Dans cette commande, *query_name* et *form_name* sont des noms de variable REXX qui décrivent les paramètres transmis à la procédure navigationnelle. Utilisez ces variables, qui contiennent les noms d'objet pour une requête et un format, pour référencer les paramètres qui ont été transmis à la procédure navigationnelle.

Instructions de traitement d'erreurs REXX dans les procédures navigationnelles

Vous pouvez utiliser des techniques de traitement d'erreurs REXX, telles que l'instruction REXX SIGNAL, dans une procédure navigationnelle. Les commandes et variables QMF employées avec l'instruction REXX EXIT peuvent également vous permettre de clarifier les codes retour différents de zéro.

Bifurcation vers des sous-routines de traitement d'erreurs

L'instruction REXX SIGNAL ON ERROR demande à REXX de quitter la ligne en cours et de bifurquer vers un libellé intitulé erreur lorsqu'un code retour différent de zéro est détecté.

Cette instruction requiert deux éléments :

- Instruction SIGNAL ON ERROR
Après chaque commande, REXX place le code retour de la commande dans une variable appelée *rc*.
Si une commande comporte un code retour différent de zéro, REXX bifurque vers le libellé erreur.
SIGNAL ON ERROR renvoie des erreurs provenant de l'environnement de commande de la procédure QMF REXX (ADDRESS QRW), mais pas de l'interface appellable REXX.
- Libellé d'erreur

L'instruction SIGNAL ON ERROR nécessite que vous fournissiez un libellé vers lequel la procédure peut bifurquer si elle détecte un code retour différent de zéro. Le libellé précède votre code de traitement d'erreurs. Le code retour figure dans la variable *rc*. Vous pouvez utiliser cette variable pour bifurquer vers une autre sous-routine ou l'employer dans votre instruction EXIT, comme dans l'exemple suivant :

```
/* code de traitement d'erreurs pour une procédure navigationnelle */
erreur:
  exit rc
```

Messages avec l'instruction REXX EXIT

Vous pouvez utiliser l'instruction REXX EXIT pour quitter une procédure navigationnelle. QMF émet toujours un message à la fin de l'exécution d'une procédure navigationnelle.

Si vous utilisez l'instruction EXIT, le message qui s'affiche dépend des facteurs suivants :

- Si la dernière commande QMF a détecté une erreur.
- Si le code retour était zéro.

Le tableau ci-dessous indique le message qui s'affiche, en fonction des conditions.

Tableau 3. Messages renvoyés par QMF

Code retour de la dernière commande QMF	Code retour de la procédure	Exemples de messages à la fin de la procédure
0	0	OK, your procedure was run
0	Différent de zéro	The return code from your procedure was 8
Différent de zéro	0	Message d'erreur fourni par QMF
Différent de zéro	Différent de zéro	Message d'erreur fourni par QMF

Un message d'erreur QMF est prioritaire sur le message de code retour si la commande QMF est incorrecte et le code retour différent de zéro.

Si vous voulez afficher le message d'erreur de la dernière commande et que l'opération se termine avec un code retour QMF, utilisez la commande MESSAGE et l'instruction EXIT DSQ_RETURN_CODE. Par exemple :

```
⋮
"MESSAGE (TEXT = '"dsq_message_text"'"
exit dsq_return_code
```

Les variables *dsq_message_text* et *dsq_return_code* sont des variables REXX fournies par QMF. Vous pouvez utiliser la commande MESSAGE et la variable *dsq_message_text* pour stocker et afficher un message ultérieurement, comme présenté dans cet exemple :

```
/* Rapport mensuel */
signal on error
"DISPLAY TABLE JUNE_INFO"
"PRINT REPORT"
exit(0);
error:  original_msg = dsq_message_text
/* Sauvegarde le message d'erreur. */
"RUN PROC GENERAL_RECOVERY"
```

```

/* Cette procédure génère */
/* un nouveau texte dsq_message_text. */
"MESSAGE (TEXT=' original_msg '"
/* Affiche le message d'erreur d'origine. */
exit dsq_return_code;

```

Référence associée:

«Variables de communication d'interface pour REXX», à la page 207

Les variables de communication d'interface sont composées de plusieurs variables REXX. Elles sont définies à la fin de chaque appel et ne doivent pas être modifiées par le programme appelant.

«MESSAGE», à la page 50

Lorsque vous créez des applications, vous voulez souvent envoyer à vos utilisateurs des messages spécifiques sur les informations qui s'affichent à leur propos ou sur l'action suivante. Vous pouvez écrire vos propres messages et les afficher sur les écrans QMF via la commande MESSAGE. Dans ISPF, vous pouvez également demander à QMF d'afficher l'aide relative à un message d'erreur ISPF.

Appel de programmes REXX à partir d'une procédure navigationnelle

Utilisez différentes méthodes pour appeler votre programme REXX selon que vous vous servez ou non de variables de substitution.

Appel de programmes REXX sans variables de substitution

Si votre programme REXX ne contient pas de commande RUN imbriquée incluant des variables de substitution, appelez-le à l'aide de l'instruction ADDRESS ou CALL. Vous pouvez également appeler le programme en tant que fonction.

Utilisez l'une des commandes suivantes pour appeler votre programme :

- L'instruction ADDRESS

Cette instruction définit un environnement de commande. Par exemple, si votre programme s'appelle PANDA et que vous voulez l'appeler à partir de l'environnement TSO, la commande se présente comme suit :

```
ADDRESS TSO "PANDA"
```

- L'instruction CALL

Cette instruction appelle un programme. Par exemple, pour le programme nommé PANDA, la commande se présente comme suit :

```
CALL PANDA
```

- Une fonction

Vous pouvez également appeler le programme PANDA en tant que fonction :

```
answer = PANDA()
```

Vous pouvez envisager de supprimer les variables de substitution de la commande RUN si vous souhaitez appeler vos programmes à l'aide de l'un des appels REXX. Dans ce cas, QMF invite l'utilisateur à entrer les variables.

Concepts associés:

Chapitre 5, «ADDRESS QRW et environnement de commande QMF», à la page 31

Lorsque QMF est démarré dans TSO, ISPF ou z/OS natif, QMF crée un environnement de commande REXX appelé QRW. Lorsque vous exécutez un programme REXX, vous pouvez définir l'environnement de commande par défaut sur QRW en émettant la commande REXX ADDRESS appelée ADDRESS QRW. Grâce à ADDRESS QRW, QMF reste l'environnement de commande par défaut jusqu'à ce que vous émettiez une autre commande ADDRESS.

Appel de programmes REXX contenant des variables de substitution

Si votre application REXX contient une commande QMF RUN avec une variable de substitution, vous devez l'appeler à l'aide de la commande TSO *nom_programme*.

Que vous exécutiez une procédure navigationnelle ou un programme d'interface appellable appelé par une procédure navigationnelle, les commandes sont transmises à QMF de la même façon. Dans ce contexte, le programme d'interface appellable devient une extension logique de la procédure elle-même.

Par exemple, prenons la commande suivante :

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

Dans une procédure navigationnelle, utilisez deux perluètes dans la variable de substitution pour transmettre la variable à la requête :

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

Si une variable de substitution ne contient qu'une perluète, QMF résout la variable pour la procédure elle-même et ne peut pas transmettre la variable à la requête.

Si vous appelez votre application d'interface appellable REXX à partir d'une procédure navigationnelle et que cette application contient la commande RUN QUERY WEEKLY_Q (&DEPT=58, QMF résout la variable comme il le ferait pour la procédure appelante. Etant donné qu'une seule perluète est utilisée, la variable n'est pas transmise à la requête.

Pour transmettre des variables à QMF à partir d'une application d'interface appellable REXX appelée par une procédure navigationnelle, vous avez trois options :

- Utilisez la commande TSO pour appeler l'application.
Lorsque vous appelez l'application, QMF ne traite pas les variables de substitution qu'il rencontre. Dans la commande RUN QUERY précédente, &DEPT=58 est transmis à la requête, dans laquelle la variable de substitution est résolue.
- Traitez toutes les variables de substitution figurant dans votre application comme si vous les utilisiez dans une procédure navigationnelle :
Ajoutez une perluète à chaque variable de substitution afin que la procédure navigationnelle ne la résolve pas.
- Utilisez des variables globales.
Vous pouvez définir des variables globales au début de votre application et les utiliser dans votre session QMF. Vous pouvez également définir la variable globale DSQEC_USERGLV_SAV pour sauvegarder les valeurs de variable globale d'une session dans une autre.

Référence associée:

«Variables globales contrôlant le mode d'exécution des commandes et des procédures», à la page 237

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

Chapitre 3. Interface callable et applications QMF

Les langages de programmation peuvent utiliser l'interface callable QMF pour exécuter des commandes QMF.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Présentation de l'interface callable

L'interface callable de QMF fournit des interfaces standard pour différents langages de programmation et offre un stockage commun et un accès aux variables de programme.

Lorsqu'un programme d'application doit exécuter une commande QMF, il doit démarrer la communication entre le programme et QMF. Cette communication est établie via l'émission d'un appel à une routine d'interface QMF. QMF fournit une routine pour chaque langage pris en charge.

Le programme d'application peut émettre une ou plusieurs commandes QMF après l'appel START initial. Il appelle la routine pour émettre chaque commande QMF.

Une fois le traitement de la commande QMF terminé, QMF fournit un code retour qui indique l'état de QMF. L'interface callable regroupe d'autres informations sur le traitement de la commande et stocke ces informations dans des variables accessibles à la fois par QMF et par le programme d'application. Ces variables se trouvent, soit dans un *pool de variables*, soit dans une *zone de communication d'interface*. Lorsque l'interface callable renvoie au programme d'application appelant, l'application peut faire référence à ces variables, mais ne les modifie pas.

Lorsque le programme d'application n'a plus besoin d'utiliser QMF, le programme émet un appel pour mettre fin à la communication entre le programme et QMF. Cet appel est envoyé à la routine QMF.

Considérations relatives à l'utilisation de l'interface callable de QMF

Le flux de contrôle entre votre application et QMF à l'aide de l'interface callable est soumis à certaines règles.

Tenez compte des points suivants lorsque vous écrivez des programmes d'application à utiliser avec l'interface callable de QMF :

- Un appel à QMF renvoie au programme d'application appelant uniquement lorsque QMF a fini de traiter la commande QMF.
- QMF est à l'état inactif lorsqu'il ne traite pas d'appel.
- Le programme d'application et QMF communiquent avec les codes retour et les données de variable stockés dans le pool de variables ou la zone de communication de l'interface.
- Toutes les commandes QMF doivent être codées en majuscules, en anglais.

Si vous utilisez une fonction de langue nationale (NLF) QMF, vos commandes QMF doivent être écrites dans la langue NLF dominante. Vos commandes doivent également être écrites ou converties en majuscules par QMF. Les commandes sont converties en majuscules par QMF si l'option CASE de votre profil QMF est paramétrée sur UPPER. La langue dominante est définie au démarrage de QMF, en indiquant une valeur pour le paramètre DSQALANG sur la commande START. Cette valeur est enregistrée dans la variable globale DSQEC_NLFCMD_LANG.

- La longueur maximale des commandes transmises est 2 048 octets pour les programmes REXX et 32 768 octets pour tous les autres langages.

Ce diagramme montre la façon dont l'application transmet les commandes à QMF via l'interface appelable.

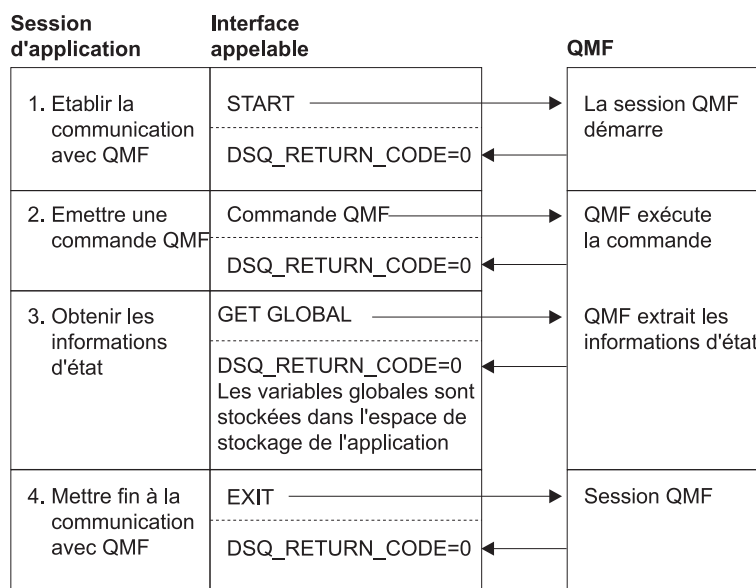


Figure 6. Utilisation de l'interface appelable de QMF par une application pour communiquer avec QMF

Référence associée:

«Mots clés de la commande START», à la page 56

Vous pouvez spécifier des mots clés dans la commande START.

Zone de communication d'interface (DSQCOMM)

QMF fournit une zone de communication d'interface pour chaque langage de programmation pris en charge. Cette zone contient des définitions de codes retour et de codes anomalie, ainsi que les définitions des appels de fonction à QMF.

La zone de communication d'interface définit le stockage pour les variables de communication d'interface. Les variables stockées dans cette zone sont accessibles à la fois par QMF et par l'application d'interface appelable. Toutefois, autorisez uniquement la modification des valeurs par QMF. Vérifiez que le programme d'application traite ces variables en lecture seule.

L'interface appelable REXX utilise les variables de communication d'interface prises en charge par QMF plutôt qu'une zone de communication.

La zone de communication d'interface appellable QMF est requise pour tous les appels d'interface appellable. Le stockage pour la zone de communication d'interface appellable est alloué par le programme qui utilise QMF.

La commande START établit une instance ou une occurrence unique d'une session QMF. La commande START ne peut établir qu'une seule session QMF :

- Dans un espace adresse TSO
- A partir d'une transaction CICS unique

Lors de l'exécution de la commande START, QMF met à jour les variables dans la zone de communication d'interface.

Ces variables ne doivent jamais être modifiées par le programme d'application, à l'exception des cas suivants :

DSQ_COMM_LEVEL

Affectez à DSQ_COMM_LEVEL la valeur DSQ_CURRENT_COMM_LEVEL pour identifier le niveau de DSQCOMM. Cette exception ne s'applique pas à REXX.

DSQ_INSTANCE_ID

Si vous appelez un programme d'interface appellable à partir de QMF, vous devez définir DSQ_INSTANCE_ID sur zéro (0) lors du premier appel. Avec ce paramètre, QMF réinitialise la variable à la valeur définie par la commande START initiale.

Tous les appels qui suivent la commande START doivent transmettre l'adresse de la zone de communication d'interface qui correspond à l'instance QMF. Le programme d'application est responsable du pointage vers la zone de communication d'interface correcte.

Chaque langage de programmation pris en charge comporte une zone de communication d'interface unique. Les programmes d'application doivent référencer les variables par leur nom plutôt que par leur valeur s'ils doivent être portables, car les valeurs peuvent être différentes sur d'autres systèmes.

Les variables dans la zone de communication d'interface contiennent les informations présentées dans ce tableau :

Tableau 4. Zones DSQCOMM qui ne doivent pas être modifiées

Informations fournies par la variable	Description
Code retour	Indique l'état du traitement QMF une fois que QMF a traité une commande
Identificateur d'instance	Identifie l'instance de QMF qui a été démarrée par la commande START
ID de message d'achèvement	Contient l'ID du message affiché par QMF Cette zone est définie lors de l'achèvement de chaque commande QMF. Elle contient le message que QMF affiche à la fin d'une commande.

Tableau 4. Zones DSQCOMM qui ne doivent pas être modifiées (suite)

Informations fournies par la variable	Description
ID de message de requête	Contient l'ID du message qui s'affiche sur l'écran de requête lors de l'achèvement d'une commande RUN QUERY Cette zone est définie lorsqu'une erreur se produit pendant qu'une requête est en cours d'exécution. Elle contient le message que QMF affiche dans l'objet de requête à la fin d'une commande.
Paramètre de commande START erroné	Contient le nom du paramètre erroné lorsque la commande START échoue en raison d'une erreur de paramètre
Indicateur d'annulation	Indique si l'utilisateur a annulé le traitement pendant que QMF exécutait la commande
Message d'achèvement	Contient le message d'achèvement affiché par QMF
Message de requête	Contient le texte du message de requête qui s'affiche sur l'écran de requête lors de l'achèvement d'une commande RUN QUERY Par exemple, si vous exécutez un objet de requête contenant une erreur, QMF affiche un message décrivant l'erreur qui a empêché l'exécution de la requête. La zone de message de requête contient ensuite le texte de ce message d'erreur.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153
L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Codes retour

Des codes retour sont renvoyés après chaque appel à l'interface callable de QMF. Les valeurs de code retour sont décrites par la zone de communication de l'interface fournie avec QMF.

Les valeurs des codes retour peuvent être différentes sur d'autres systèmes. Si vous voulez que vos applications soient portables sur les divers systèmes, elles doivent faire référence aux valeurs de ces codes à l'aide des noms de variable. Les noms des variables de code retour dans la zone de communication de l'interface sont documentées avec la spécification de langage de programmation.

Le tableau ci-dessous présente les codes retour possibles pour les conditions d'interface callable.

Tableau 5. Codes retour de l'interface callable

Valeur	Explication
0	Exécution réussie
4	Session QMF marquée en vue de son arrêt par une commande EXIT ou END
8	L'exécution a échoué, mais l'erreur n'a pas marqué la session en vue de son arrêt
16	Erreur grave : session marquée en vue de son arrêt

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Commandes pour l'utilisation de l'interface callable

Vous pouvez vous servir de l'interface callable pour émettre une commande QMF qui sera utilisée dans une procédure. Toutefois, certaines commandes ont une syntaxe spéciale pour l'interface callable : START, GET GLOBAL, SET GLOBAL et TRACE.

Les commandes START et TRACE fonctionnent uniquement dans l'interface callable. Pour utiliser les commandes GET GLOBAL et SET GLOBAL dans une application d'interface callable écrite dans un autre langage que REXX, utilisez la *syntaxe étendue* pour ces commandes.

Pour obtenir des exemples de commandes START et SET GLOBAL dans un langage de programmation, consultez la spécification pour ce langage.

Concepts associés:

Chapitre 8, «Commandes QMF dans des applications», à la page 45

Certaines commandes sont conçues pour être utilisées dans des applications et vous pouvez créer vos propres synonymes de commande.

Référence associée:

«GET GLOBAL (syntaxe étendue)», à la page 47

Vous pouvez utiliser la commande GET GLOBAL pour accéder aux variables globales QMF dans votre application. Pour les autres langages que REXX, QMF fournit une syntaxe étendue pour la commande GET GLOBAL.

«SET GLOBAL (syntaxe étendue)», à la page 53

Pour créer vos propres variables globales et les utiliser dans des commandes QMF en tant que variables de substitution, émettez la commande SET GLOBAL. Vous pouvez également utiliser la commande SET GLOBAL pour définir des valeurs pour les variables globales QMF prédéfinies, qui commencent par "DSQ".

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Démarrage de QMF à partir d'une application

Avant d'exécuter toute commande à partir d'une application, vous devez démarrer QMF. Lors de l'utilisation de l'interface callable, démarrez QMF en émettant la commande START dans votre application. Vous ne pouvez disposer que d'une seule session QMF à la fois.

Votre application peut émettre une commande START pour vérifier si QMF est démarré. Si tel n'est pas le cas, QMF démarre. Si QMF est démarré, le code retour est différent de zéro et vous recevez le numéro de message et le message suivants :
DSQ50720 QMF already active; secondary session not permitted.

Si votre commande START génère une erreur qui n'est pas grave (code retour 4 ou 8), QMF démarre et une session est établie. Dans ce cas, vous devez émettre une

commande QMF EXIT pour arrêter QMF. Examinez le contenu de la zone de communication d'interface ou la sortie de données de trace QMF pour rechercher l'origine de l'erreur.

Pour transmettre des paramètres à QMF, indiquez les mots clés de commande souhaités sur la commande START.

Référence associée:

«START», à la page 55

Lorsque vous démarrez QMF via l'interface callable, vous devez utiliser la commande START.

Exécution de l'application d'interface callable

Lorsque vous exécutez votre application d'interface callable, vous devez configurer votre environnement comme pour une exécution interactive de QMF.

Pour plus d'informations sur la configuration de votre environnement, ainsi que sur la compilation et l'exécution de votre application d'interface callable, consultez l'exemple de codage dans la spécification de langage appropriée.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Interface callable dans QMF

Si vous avez besoin de modifier un objet QMF à partir d'un programme utilisateur, vous pouvez utiliser l'interface callable dans QMF. Par exemple, vous pouvez exporter ou importer des objets via l'interface callable au cours d'une session QMF interactive. Vous pouvez utiliser l'interface callable dans QMF en vous servant de la commande TSO pour appeler l'application. Vous pouvez exécuter toute commande QMF valide à partir de l'application.

Restriction : Vous ne pouvez pas utiliser l'interface callable dans QMF lorsque vous vous trouvez dans l'environnement CICS.

Vous devez définir l'identificateur d'instance DSQCOMM (DSQ_INSTANCE_ID) sur zéro (0) avant votre premier appel à QMF. QMF détermine l'instance en cours et met à jour DSQ_INSTANCE_ID en vue de son utilisation dans des appels QMF ultérieurs.

Traitement des erreurs

À la fin de chaque commande QMF, la zone de communication DSQCOMM contient un texte de message dans la variable dsq_message_text et un code retour dans la variable dsq_return_code.

L'une des valeurs suivantes est affectée au code retour :

dsq_success

La commande s'est achevée avec succès

dsq_warning

Achèvement normal avec des avertissements

dsq_failure

La commande ne s'est pas exécutée correctement

dsq_severe

Erreur grave ; la session QMF a pris fin

Les variables et zones figurant dans chaque zone DSQCOMM sont documentées avec les spécifications du langage de programmation.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Exécution de programmes d'interface callable sous CICS

Pour exécuter des programmes qui utilisent l'interface callable QMF, installez-les sous CICS en suivant votre méthode habituelle d'installation des programmes CICS.

Outre la configuration normale requise pour CICS, les considérations suivantes s'appliquent à tous vos programmes d'interface callable QMF qui s'exécutent sous CICS :

Environnement

Lorsque votre programme appelle le produit QMF, il prend les mêmes caractéristiques que le produit QMF interactif et devient un vaste programme conversationnel.

QMF est un programme en langage assembleur qui contient des commandes CICS. Ses liens peuvent être édités avec d'autres programmes en langage assembleur ou avec des programmes pris en charge par l'interface de programmation d'application callable. Lorsque vous appelez QMF à l'aide d'un langage évolué, les liens du programme doivent d'abord être édités. En outre, la définition de programme de définition des ressources en ligne doit spécifier ce langage évolué. Des considérations et des restrictions CICS spécifiques sont définies pour chaque programme évolué.

Dans CICS, si vous voulez remplacer l'un des paramètres par défaut du programme QMF, vous devez spécifier les valeurs de substitution en tant que paramètres de la commande START. Par exemple, le mode de fonctionnement par défaut à partir de l'interface callable est le mode de traitement par lots. Pour exécuter une session QMF interactive, vous devez émettre la commande START avec l'option DSQSMODE=I.

CICS considérations relatives à la région

Le programme utilisateur qui contient le module de communications de l'interface QMF et le module QMF principal doit s'exécuter dans la même région ou partition. Les ressources QMF, telles qu'elles ont été décrites lors de l'installation de QMF, doivent également être allouées à la région ou à la partition CICS qui exécute QMF.

Base de données

La transaction CICS qui appelle votre programme doit être décrite à DB2.

Concepts associés:

«Interfaces de programmation d'applications dans QMF», à la page 4

Il existe deux interfaces de programmation d'applications dans QMF : l'interface

appelable et l'interface de commande.

Chapitre 4. Emission de commandes QMF à partir d'une boîte de dialogue ISPF

Vous pouvez émettre des commandes QMF à partir d'une boîte de dialogue ISPF qui s'exécute sous QMF à l'aide de l'interface de commande QMF.

Restriction : L'interface de commande QMF nécessite l'exécution d'ISPF, mais ISPF ne s'exécute pas dans l'environnement CICS. Vous devez donc utiliser l'interface callable QMF pour le développement d'applications sous CICS.

L'interface de commande QMF permet à QMF de communiquer avec la boîte de dialogue via le pool de variables ISPF, comme indiqué dans le diagramme ci-dessous.

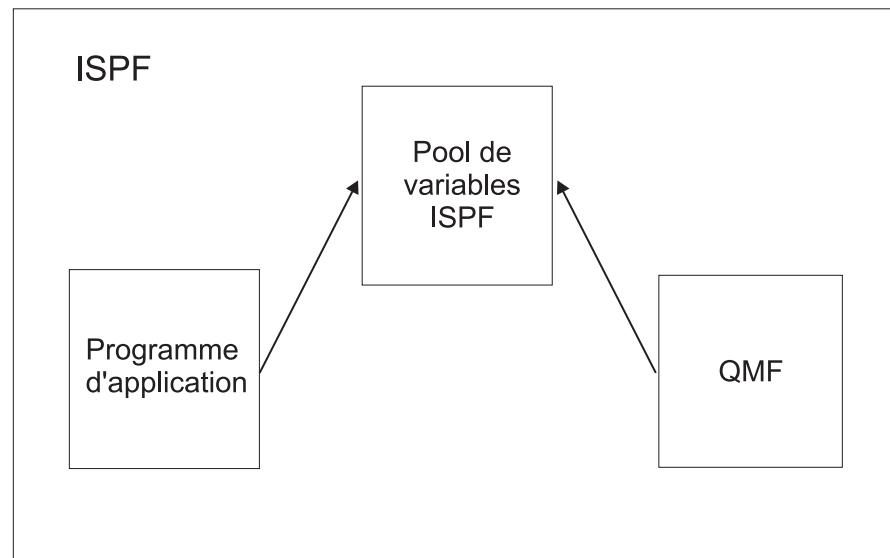


Figure 7. Application d'interface de commande QMF en interaction avec QMF

Pour utiliser l'interface de commande de façon efficace, vous devez également comprendre les services ISPF et les pools de variables.

Exemple d'écriture d'un programme utilisant l'interface de commande

Dans cet exemple, vous écrivez un programme qui permet d'afficher un écran ISPF invitant l'utilisateur à spécifier un nom de requête, à exécuter la requête indiquée et à afficher un rapport.

Procédure

1. Ecrivez votre programme REXX d'interface de commande :
 - a. Affichez l'écran ISPF avec les services DISPLAY : Dans cet exemple, le nom d'écran est QRYNAME
`ADDRESS ISPEXEC "DISPLAY PANEL(QRYNAME)"`
 - b. Exécutez une requête QMF basée sur les données saisies par l'utilisateur dans le service DISPLAY précédent. La variable ISPF QNAME contient le nom de la requête QMF

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

c. Affichez le résultat de la requête :

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

2. Appelez votre programme avec la commande TSO à partir de la ligne de commande QMF. Par exemple, si votre programme s'appelle GETINFO, utilisez cette commande :

```
TSO GETINFO
```

Appel de l'interface de commande

L'interface de commande est un programme nommé DSQCCI. Vous pouvez l'appeler à partir d'un programme via le service ISPF SELECT.

Pour démarrer l'interface de commande, commencez par lancer ISPF. Démarrez ensuite QMF à l'aide du service ISPF SELECT pour appeler l'interface de commande QMF (DSQCCI). Vous pouvez transmettre les commandes QMF en utilisant l'option PARM de la commande ISPF SELECT PGM.

Après avoir démarré l'interface de commande, vous pouvez transmettre les commandes QMF à exécuter en utilisant le paramètre PARM de la commande ISPF SELECT PGM :

```
SELECT PGM(DSQCCI) PARM(commande_qmf)
```

Toutes les commandes QMF spécifiées sous forme de paramètres à l'interface de commande doivent être en majuscules, quel que soit le paramètre de profil QMF. ISPF ne convertit pas les commandes minuscules en majuscules. Si vous spécifiez vos commandes QMF en minuscules, QMF ne les reconnaît pas.

Si vous voulez utiliser des invites de commande QMF à partir de votre application ISPF, vous pouvez émettre la commande INTERACT. Faites suivre la commande INTERACT de la commande QMF pour laquelle vous souhaitez ouvrir une invite et terminez l'instruction par un point d'interrogation. Par exemple, pour afficher l'écran d'invite RUN QUERY, émettez la commande suivante :

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

Le service SELECT requiert l'utilisation de perluètes doubles sur une commande RUN QUERY. L'utilisation de perluètes doubles empêche ISPF d'interpréter la variable comme l'une des siennes.

Lors de l'appel, n'indiquez pas l'option NEWPOOL ou NEWAPPL. L'omission des options NEWPOOL et NEWAPPL permet de s'assurer que l'interface de commande peut accéder aux variables de votre application. L'interface de commande utilise le pool partagé pour communiquer entre QMF et votre application.

Commande END

La commande END met fin au programme DSQCCI et renvoie à l'application appelante lorsqu'elle est émise pendant l'exécution de l'interface de commande. La session QMF reste active.

QMF définit la variable globale DSQCSESC pour marquer la session en vue de son arrêt si elle détecte une commande EXIT ou une erreur grave lors de l'appel de l'interface de commande. Lorsque le programme qui a appelé DSQCCI se termine et renvoie à QMF, la session QMF prend fin.

Variables dans l'interface de commande

La commande STATE indique la valeur en cours pour chaque variable fournie par QMF. La commande STATE peut être utilisée uniquement dans l'interface de commande. Vous pouvez placer les variables QMF dans le pool de variables ISPF par le biais de la commande VPUT.

Référence associée:

Annexe B, «Variables globales QMF», à la page 215

QMF fournit de nombreuses variables globales qui vous aident à contrôler divers aspects de votre session QMF, de vos commandes QMF et de l'affichage d'écran. Les variables globales vous aident également à gérer le comportement des fonctions QMF dans des procédures et des applications.

Codes retour de l'interface de commande

Les codes retour pour l'interface de commande peuvent être positifs ou avoir une valeur de zéro. Une valeur de zéro indique que l'exécution a réussi. Une valeur positive indique que l'exécution a échoué ou s'est déroulée de façon anormale. Les codes retour sont identiques quel que soit le langage de votre application.

Les codes retour s'affichent dans une variable dans la commande exec ou CLIST de l'utilisateur. Si vous exécutez une commande exec REXX, le code retour se trouve dans la variable REXX appelée RC ; si vous exécutez une commande CLIST, le code retour se trouve dans la variable CLIST &LASTCC.

L'exemple suivant présente une commande exec qui examine un code retour. Il montre comment exécuter une requête et tester une erreur avec la variable REXX RC :

```
ADDRESS ISPEXEC SELECT PGM(DSQCCI) PARM(RUN QUERYA (FORM=FORMA))
Select
  When (RC = 0) Then nop
  When (RC = 64) Then
    Say "Vous devez exécuter QMF avec ISPF pour utiliser l'interface de commande."
  When (RC = 100) Then
    Say "Vous devez démarrer QMF avant de lancer votre application"
  Otherwise
    Say "Erreur inattendue ("RC") de l'interface de commande QMF."
End
```

Vous pouvez placer le code de traitement des erreurs dans les modules de programme et les commandes exec ou CLIST.

Codes retour de 0 à 16

Les codes retour de 0 à 16 décrivent le traitement QMF de la commande transmise avec l'interface de commande. L'interface de commande renvoie le code, mais également les valeurs des variables de message de la commande QMF dans le pool partagé ISPF de l'application.

Tableau 6. Codes retour de 0 à 16

Valeur	Explication
0	Exécution réussie
4	Session QMF marquée en vue de son arrêt par une commande EXIT ou END

Tableau 6. Codes retour de 0 à 16 (suite)

Valeur	Explication
8	L'exécution a échoué, mais l'erreur n'a pas provoqué le marquage de la session en vue de son arrêt
16	Erreur grave : session marquée en vue de son arrêt

Un code retour de 4 se produit uniquement sur la commande qui a provoqué le marquage de la session en vue de son arrêt. Si l'application tente ensuite d'exécuter une autre commande, QMF renvoie une autre valeur de code retour à l'application.

Codes retour de 20 ou plus

Ces codes traduisent généralement un incident au niveau de l'interface de commande (DSQCCI). L'incident empêche l'interface de copier une variable dans le pool partagé de l'application. Par conséquent, les variables QMF risquent d'être non valides ou non définies. Le même résultat peut se produire pour les variables STATE si votre programme utilise la commande STATE. (Une variable est définie si elle est copiée dans le pool partagé de l'application.)

Ces codes retour indiquent généralement des erreurs plus graves que les codes retour de 0 à 16. Certains codes retour peuvent nécessiter l'assistance du service de support logiciel IBM.

Dans ce tableau, le terme *variables partagées* fait référence aux variables QMF (et aux variables STATE, si la commande en cours est STATE). Certains codes indiquent que la commande a été exécutée mais que les variables partagées n'ont pas été définies. QMF a exécuté la commande STATE correctement, mais l'interface de commande n'a pas réussi à définir les variables partagées QMF et STATE mises à jour. La cause de l'incident est décrite dans l'explication du code d'erreur.

Tableau 7. Codes retour de 20 ou plus

Valeur	Explication
20	Une routine de sortie utilisateur a appelé l'interface de commande ; ces appels ne sont jamais valides. La commande transmise à l'interface de commande n'a pas été exécutée. Les variables partagées n'ont pas été définies.
24	Une erreur s'est produite dans une commande ISPF VCOPY. La commande transmise à l'interface de commande a été exécutée. Les variables partagées n'ont pas été définies.
32	Une erreur s'est produite dans une commande ISPF VREPLACE. La commande transmise à l'interface de commande a été exécutée. Les variables partagées n'ont pas été définies.
36	Une erreur s'est produite dans une commande ISPF VPUT. La commande transmise à l'interface de commande a été exécutée. Les variables partagées n'ont pas été définies.
40	Une erreur s'est produite dans une commande ISPF VREPLACE. Ce code s'applique uniquement à l'exécution de la commande STATE. La commande transmise à l'interface de commande a été exécutée, mais les variables partagées n'ont pas été définies.
44	Une erreur s'est produite dans une commande ISPF VPUT. Ce code s'applique uniquement à l'exécution de la commande STATE. Les variables QMF ont été définies, mais pas les variables STATE.

Tableau 7. Codes retour de 20 ou plus (suite)

Valeur	Explication
60	Un appel non valide a été envoyé à l'interface de commande. La commande transmise à l'interface de commande n'a pas été exécutée. Les variables partagées n'ont pas été définies.
64	Cette erreur est générée lorsque DSQCCI est exécuté et qu'ISPF n'est pas actif. Par exemple, l'utilisateur a appelé DSQCCI sans utiliser une commande ISPF SELECT PGM.
100	Cette erreur se produit lorsqu'une application tente d'émettre une commande QMF alors que QMF n'est pas actif. Démarrez QMF avant de lancer votre application. La commande transmise à l'interface de commande n'a pas été exécutée. Les variables partagées n'ont pas été définies.
104	L'ancrage n'a pas été localisé. La commande transmise à l'interface de commande n'a pas été exécutée. Les variables partagées ont été définies, mais ne sont pas valides.

Chapitre 5. ADDRESS QRW et environnement de commande QMF

Lorsque QMF est démarré dans TSO, ISPF ou z/OS natif, QMF crée un environnement de commande REXX appelé QRW. Lorsque vous exécutez un programme REXX, vous pouvez définir l'environnement de commande par défaut sur QRW en émettant la commande REXX ADDRESS appelée ADDRESS QRW. Grâce à ADDRESS QRW, QMF reste l'environnement de commande par défaut jusqu'à ce que vous émettiez une autre commande ADDRESS.

Restriction : L'instruction ADDRESS QRW n'est pas prise en charge dans l'environnement CICS.

Vous pouvez également demander qu'une seule commande soit exécutée par l'environnement QRW en émettant la commande REXX ADDRESS suivie de la commande QMF :

```
ADDRESS QRW commande_qmf
```

Dans cette situation, QMF est l'environnement de commande uniquement pour la commande qui suit l'instruction ADDRESS QRW.

Lorsque vous utilisez une procédure navigationnelle QMF, QRW est l'environnement de commande par défaut.

L'exemple suivant montre comment utiliser l'environnement de commande QMF :

```
⋮  
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  
⋮
```

Chapitre 6. Ecriture d'applications QMF utilisant des services ISPF

Vous pouvez ignorer les écrans QMF en écrivant des applications qui possèdent leurs propres interfaces utilisateur. Vous pouvez utiliser soit l'interface callable, soit l'interface de commande, pour écrire des applications utilisant ISPF.

Restriction : ISPF ne s'exécutant pas dans l'environnement CICS, les services ISPF ne sont pas disponibles sous CICS.

Concepts associés:

Chapitre 3, «Interface callable et applications QMF», à la page 17

Les langages de programmation peuvent utiliser l'interface callable QMF pour exécuter des commandes QMF.

Chapitre 4, «Emission de commandes QMF à partir d'une boîte de dialogue ISPF», à la page 25

Vous pouvez émettre des commandes QMF à partir d'une boîte de dialogue ISPF qui s'exécute sous QMF à l'aide de l'interface de commande QMF.

Démarrage et exécution de QMF à partir d'une application ISPF

Lorsque vous démarrez et exécutez QMF à l'aide de l'interface callable à partir d'une application ISPF, vous devez respecter certaines règles.

Lorsque vous écrivez une application d'interface callable qui utilise ISPF, vous devez vous assurer que vous respectez les conditions suivantes :

- L'application d'interface callable doit correspondre au langage de votre boîte de dialogue ISPF.

Par exemple, si votre boîte de dialogue ISPF est un programme PL/I, vous devez utiliser l'interface callable QMF pour PL/I pour l'écriture de votre application.

- Vous devez utiliser l'identificateur de langue nationale correct.

Vous devez démarrer votre application ISPF avec un ID de type DSQ n , où n représente un identificateur de langue nationale. Cet ID d'application empêche QMF de remplacer votre environnement ISPF, tels que les libellés et les paramètres de touche de fonction. L'ID vérifie également que l'environnement ISPF reste intact même après le démarrage de QMF. Par exemple, cette instruction lance un programme PL/I appelé MONPROG qui démarre QMF à l'aide de la commande START de l'interface callable :

```
SELECT PGM(MONPROG) NEWAPPL(DSQn)
```

- Utilisez les commandes GET GLOBAL ou SET GLOBAL dans votre application à la place de la commande STATE pour définir et extraire des valeurs de variable.

La commande STATE fonctionne uniquement pour les variables contenant des informations d'état. Les commandes GET GLOBAL et SET GLOBAL fonctionnent pour toutes les variables globales QMF. Toutefois, vous ne pouvez pas définir de variables globales en lecture seule.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

«Mots clés de la commande START», à la page 56

Vous pouvez spécifier des mots clés dans la commande START.

«Variables globales relatives aux informations d'état non liées au profil», à la page 216

Les variables globales DSQAO contiennent des informations d'état, des paramètres ou des indicateurs. La commande SET GLOBAL ne permet pas de les modifier.

«Variables globales stockant les informations d'état associées aux profils», à la page 222

Les variables globales DSQAP stockent des informations associées aux paramètres de profil QMF. Aucune d'elles n'est modifiable à l'aide de la commande SET GLOBAL.

Exécution de requêtes contenant des variables

Les applications qui utilisent des services ISPF peuvent exécuter des requêtes contenant des variables.

Vous pouvez exécuter ces requêtes à partir d'une application qui utilise des services ISPF, en procédant de l'une des trois façons suivantes :

- Utilisation des services de personnalisation de fichier ISPF.

Cette technique permet de représenter la requête par le biais d'un squelette de personnalisation de fichier ISPF. Dans ce squelette, les parties de la requête qui peuvent être modifiées s'affichent sous forme de variables de dialogue ISPF. Après avoir affecté les valeurs appropriées à ces variables, votre programme démarre certains services de personnalisation de fichier ISPF. Un fichier séquentiel contenant la requête est alors généré.

Le programme peut ensuite importer la requête dans la mémoire temporaire QMF pour que QMF l'exécute. Les commandes IMPORT et RUN requises peuvent être exécutées via l'interface callable ou l'interface de commande.

Pour utiliser cette technique, vous devez savoir comment définir des variables de dialogue ISPF dans le programme qui utilise le service de dialogue ISPF.

- Utilisation de l'éditeur PDF (Program Development Facility) pour créer des objets QMF

Vous pouvez utiliser l'éditeur PDF avec des macros d'édition PDF pour concevoir et contrôler la saisie de données dans des requêtes, procédures, formats et profils. Vous pouvez utiliser le langage REXX pour écrire des macros PDF.

- Création d'une requête utilisant une boîte de dialogue ISPF.

Votre programme peut utiliser des services d'affichage ISPF pour afficher un écran et créer un fichier basé sur les données saisies par l'utilisateur. Ce fichier, qui contient ensuite une requête SQL, est alors importé dans QMF et exécuté.

Démarrage d'un programme utilisant des services ISPF à partir de QMF

Si vous voulez démarrer votre programme ISPF à partir de QMF, vous devez appeler le programme à partir d'une procédure linéaire ou navigationnelle.

Pour appeler votre programme, utilisez le service ISPF SELECT PGM en incluant la commande suivante dans votre procédure :

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


Utilisez le mot clé CMD pour indiquer à ISPF que vous exécutez votre programme en tant que fonction de boîte de dialogue ISPF. La syntaxe de cette commande est la suivante :

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

ou

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

Dans cette instruction, *cmdname* représente le nom de votre commande d'interface appelable.

Services ISPF dans une procédure navigationnelle

Vous devez effectuer un transfert de la boîte de dialogue du programme QMF vers une boîte de dialogue de commande ISPF pour exécuter des commandes ISPF à partir d'une procédure QMF navigationnelle qui s'exécute sous ISPF.

Pour définir l'environnement ISPF correct et exécuter un programme contenant vos commandes ISPF, utilisez l'instruction ISPF SELECT CMD suivante avec le mot clé CMD :

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

Dans cette instruction, *userprogram* correspond au programme qui contient vos commandes ISPF.

Par exemple, si le programme contenant vos commandes ISPF s'appelle DIALOG, incluez la commande suivante dans votre procédure navigationnelle :

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

Vous pouvez également utiliser une commande QMF TSO pour exécuter le programme contenant les commandes ISPF (par exemple, TSO DIALOG). Dans ce cas, QMF émet l'instruction ISPF SELECT CMD à votre place.

Lors de l'exécution de QMF sous ISPF, une procédure navigationnelle qui démarre un programme nécessitant des services ISPF doit utiliser l'environnement ISPF SELECT CMD. Par exemple, supposons que vous exécutiez QMF sous ISPF et que votre procédure navigationnelle émette la commande DB2 DSN. Etant donné que la commande DSN se sert des services ISPF, utilisez l'une des commandes suivantes pour émettre la commande DSN :

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

ou

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

Dans la deuxième instruction, DSNEXEC est un programme qui contient l'instruction ADDRESS TSO DSN.

Commande EDIT avec ISPF

Lorsque vous exécutez votre application QMF sous ISPF, vous pouvez éditer votre procédure ou requête SQL QMF à l'aide de la commande EDIT QUERY ou EDIT PROC.

Si vous émettez la commande QMF EDIT à partir d'un écran PROC ou QUERY, vous n'avez pas besoin de spécifier les types d'objet PROC ou QUERY. EDIT

suppose qu'il s'agit de ces valeurs lorsque vous l'appellez à partir des écrans respectifs. Par défaut, la commande QMF EDIT place votre procédure ou requête dans une session d'éditeur PDF. QMF démarre l'éditeur PDF à l'aide de l'ID d'application QMF DSQ n , où n correspond à l'identificateur NLF. QMF définit également les touches de fonction et l'emplacement de la ligne de commande pour qu'ils correspondent à votre application QMF.

Pour remplacer l'éditeur par défaut, utilisez les commandes EDIT QUERY et EDIT PROC comme suit :

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

Dans ces instructions, *name* peut être l'une des valeurs suivantes :

- Un éditeur auquel vous avez accès
- Le nom d'un programme REXX qui spécifie un ID d'application différent de celui par défaut. L'ID d'application par défaut est DSQ n , où n correspond à l'identificateur de langue nationale pour la fonction NLF que vous utilisez). Utilisez un ID d'application différent de l'ID d'application QMF par défaut si vous voulez des touches de fonction différentes de celles fournies par QMF.

Si vous utilisez des options PDF EDIT qui requièrent des membres d'ensemble de données PDF PROFILE, vous devez créer ces membres. Par exemple, l'option PDF EDIT RECOVERY nécessite un membre d'ensemble de données DSQ n EDRT PROFILE (où n correspond au caractère NLF approprié). La fonction NLF doit être installée avant l'émission de la commande.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

ISPF et débogage d'applications

La fonction de trace QMF peut vous aider à tracer l'activité QMF à divers niveaux de détail. Pour déboguer plus efficacement les applications qui utilisent ISPF, vous pouvez également utiliser le service de journalisation ISPF et le service de test de dialogue PDF. Ces services viennent compléter la fonction de trace QMF.

Service de journalisation ISPF

Vous pouvez utiliser le service de journalisation ISPF pour écrire un message dans le fichier journal ISPF. Par exemple, dans REXX, la commande ISPF pour écrire un message dans le journal ISPF est la suivante :

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

Dans cette instruction, *id-message* correspond au code d'identification du message qui sera extrait de la bibliothèque des messages et écrit dans le journal.

Service de test de dialogue PDF

Si votre site comporte un PDF, vous pouvez utiliser le service de test de dialogue PDF pour consigner les appels de service de l'application ISPF dans le fichier journal ISPF. En outre, vous pouvez utiliser l'option de journalisation du service de

test de dialogue PDF pour parcourir le contenu du fichier journal ou de l'ensemble de données. Vous pouvez également imprimer le fichier journal ou l'ensemble de données lorsque vous quittez ISPF.

Le service de test de dialogue comporte de nombreuses autres options utiles pour le débogage de votre application. Par exemple, vous pouvez effectuer un débogage interactif. Vous pouvez exécuter tout ou partie de votre application, examiner les résultats, modifier votre application et la réexécuter. Vous pouvez également utiliser les services de test de dialogue pour atteindre les objectifs suivants :

- Lancer des écrans de sélection, des procédures de commande et des programmes
- Afficher des écrans
- Ajouter des variables et modifier des valeurs de variable
- Exécuter des services de dialogue ISPF
- Ajouter, modifier et supprimer des définitions de point d'arrêt
- Ajouter, modifier et supprimer des définitions de trace de fonction et de variable

Pour créer, modifier et supprimer des définitions de trace, utilisez l'option de trace (TRACES) du service de test de dialogue. Utilisez également cette option pour surveiller les appels de service de dialogue et l'utilisation des variables de dialogue. Lors du traitement, si des définitions de trace sont satisfaites, la sortie de trace est écrite dans le journal ISPF. Vous pouvez utiliser l'option LOG des services de test de dialogue pour parcourir le journal ISPF ou examiner la sortie imprimée lorsque vous quittez ISPF.

Concepts associés:

Chapitre 10, «Débogage des applications QMF», à la page 149

Outre le traitement des erreurs, QMF offre des fonctions de débogage pour vos applications d'interface callable.

Chapitre 7. Ecriture d'applications bilingues

De nombreuses applications métier doivent s'exécuter dans plusieurs langues différentes. Vous pouvez écrire une application en anglais et l'exécuter dans n'importe quelle langue prise en charge par QMF.

Chaque langue prise en charge par QMF est appelée fonction de langue nationale (National Language Feature ou NLF). Une fonction NLF fournit à l'utilisateur une session QMF adaptée à une langue spécifique.

QMF offre un support bilingue des commandes et des formats. Vous pouvez exécuter les commandes QMF en anglais et afficher des formats dans n'importe quelle langue.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Comparaison des environnements en anglais et NLF

Bien que certains aspects de l'environnement de session QMF soient identiques quelle que soit la fonction NLF utilisée, il existe des différences. Lorsqu'aucune fonction NLF n'est installée, le seul environnement de session QMF disponible est l'environnement en anglais.

Similarités environnementales

Ces similarités sont les plus importantes dans les environnements de session QMF, quelle que soit la fonction NLF utilisée :

Fonctions

En général, vous pouvez effectuer les mêmes opérations dans une session NLF que dans une session en anglais. Vous pouvez créer des objets dans une mémoire temporaire et les sauvegarder dans la base de données, formater et imprimer des rapports, et émettre des commandes SQL. Vous pouvez également exécuter des requêtes créées de façon interactive, des requêtes SQL et QBE, ainsi que des procédures QMF. La différence entre les environnements ne réside pas dans les opérations que vous pouvez exécuter, mais dans la façon dont vous devez saisir vos entrées et dans les langues qui s'affichent.

SQL et QBE

Les instructions, opérateurs et mots clés des langages SQL et QBE ne sont pas traduits.

Codes usage pour les formats

Les codes sont identiques ; ils ne sont pas traduits.

Commandes système

Les commandes TSO et CICS peuvent toujours être émises à partir de QMF via la commande TSO ou CICS. Ces commandes ne sont pas affectées par la traduction : saisissez TSO ou CICS suivi de la commande à exécuter et entrez la commande exactement comme si vous l'exécutiez en dehors de QMF.

Différences environnementales

Voici certaines différences entre l'environnement NLF et l'environnement en anglais :

Langage de commande QMF

Chaque fonction NLF dispose d'un ensemble complet d'instructions et de mots clés QMF traduits. Ces instructions et ces mots clés traduits doivent apparaître dans vos commandes QMF lorsque vous vous trouvez dans l'environnement régional d'une fonction NLF. Pour une fonction NLF donnée, ces mots peuvent être traduits. Par exemple, supposons que dans la fonction NLF en français, l'instruction DISPLAY et le mot clé PROC soient traduits par AFFICHER et PROC. Dans une session NLF en français, vous pouvez émettre la commande AFFICHER PROC, mais pas la commande DISPLAY PROC.

Certains éléments du langage QMF sont des synonymes de commande et peuvent être traduits. Par conséquent, chaque fonction NLF comporte sa propre table de synonymes de commande dotée d'un nom unique. Lorsque la fonction NLF est installée, sa table de synonymes de commande est créée et le profil pour la fonction NLF indique le nom de la table de synonymes de commande pour cette fonction NLF.

Ecrans et messages QMF

Chaque fonction NLF dispose d'un ensemble complet de messages et d'écrans QMF traduits. Comme les instructions et les mots clés pour les commandes QMF, ces messages et écrans peuvent être traduits ou non. Dans la plupart des cas, ils le sont. Dans les écrans et les messages, les parties fixes du texte peuvent être traduites. Les informations qui peuvent varier au sein de chaque écran ou message, tels que les noms de requête, ne sont pas traduites.

Saisie autorisée à l'écran

De nombreux écrans QMF qui nécessitent une saisie par l'utilisateur limitent la portée de certaines entrées à un petit ensemble de mots clés, qui sont traduits. Ces écrans incluent par exemple les invites et les formats. Par exemple, les réponses YES et NO en anglais sont OUI et NON en français.

Valeurs de paramètre de profil

Dans un environnement multilingue, les utilisateurs disposent d'un profil distinct pour chaque fonction NLF disponible qu'ils peuvent utiliser pour une session QMF. Pour chacun de ces profils, les paramètres sont identiques et ont la même signification, mais les noms de paramètre sont traduits. Certaines valeurs de paramètre sont également traduites.

Par exemple, dans un profil en anglais, le paramètre CASE peut avoir la valeur UPPER, STRING ou MIXED. Dans un profil en français, le paramètre CASE devient le paramètre CASSE et les valeurs admises sont MAJ, CHAINE ou MIXTE.

Objets format exportés et sauvegardés

Utilisez les commandes SAVE, EXPORT et IMPORT pour spécifier la langue dans laquelle vous souhaitez que les objets format soient sauvegardés. Vous pouvez les sauvegarder en anglais ou dans la langue dominante de votre session en cours.

Exemples de tables et de requêtes

IBM fournit les versions traduites des exemples de tables et de requêtes en anglais, sauf pour les fonctions Suédois et Majuscules. Aucun exemple de table n'est fourni pour ces fonctions.

Création d'objets à utiliser dans les applications bilingues

Les objets d'une application bilingue sont identiques aux autres objets QMF. L'essentiel consiste à les créer ou les sauvegarder en anglais.

Le mode de création ou de sauvegarde des applications bilingues en anglais dépend de l'objet spécifique :

Requêtes

Vous pouvez générer des requêtes créées de façon interactive et QBE dans toute langue prise en charge par les fonctions NLF de QMF ou vous pouvez créer des requêtes SQL en anglais.

Formats

Créez toujours des formats dans la langue dominante. Sauvegardez-les en utilisant la langue par défaut dans la commande SAVE (ANGLAIS) ou utilisez la langue dominante.

La variable globale DSQEC_FORM_LANG contrôle la langue dans laquelle un format est sauvegardé, importé ou exporté. La valeur par défaut est 1 (anglais). La valeur zéro indique que les formats doivent être sauvegardés, importés et exportés dans la langue dominante de la session.

Procédures

Vous pouvez créer des procédures en anglais ou dans la langue dominante.

Analyse

Vous pouvez créer des objets d'analyse en anglais ou dans la langue dominante.

Vous pouvez traduire en anglais un format que vous créez et sauvegardez dans une fonction de langue nationale en émettant une commande SAVE. Par exemple, la commande en français pour sauvegarder un format appelé SEMAINE_F en anglais sous la forme WEEKLY_F est la suivante :

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

Cette commande convertit votre format NLF en un format anglais que vous pouvez utiliser dans votre application bilingue.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Variable de langue de commande

Pour utiliser des commandes en anglais dans une session en langue nationale (NLF), définissez la variable de langue dominante, DSQEC_NLFCMD_LANG, sur l'anglais. Utilisez cette variable pour basculer entre l'anglais et la langue dominante de la session NLF.

Par exemple, supposons que votre application soit une procédure nommée WEEKLY_P. Les commandes présentées ici montrent comment basculer entre l'anglais et la langue NLF dominante.

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

Ces commandes peuvent faire partie de n'importe quelle application QMF valide, qu'il s'agisse d'une procédure de lancement ou d'un programme en langage évolué, mais elles doivent figurer dans l'ordre suivant :

- 1** Cette ligne de la procédure sauvegarde la valeur de langue dominante dans une variable.
La commande GET GLOBAL sauvegarde la valeur pour la langue dominante dans une variable appelée CURR_LANG.
- 2** Cette ligne de la procédure définit la langue dominante sur la langue pour laquelle l'application a été écrite.
Dans cet exemple, l'application WEEKLY_P a été écrite avec des commandes en anglais. C'est la raison pour laquelle la commande SET GLOBAL définit la langue dominante sur l'anglais en affectant la valeur 1 à la variable DSQEC_NLFCMD_LANG.
- 3** Cette ligne de la procédure exécute l'application.
Une fois que la session QMF est définie sur l'anglais, l'application figurant dans l'exemple peut être exécutée. Les commandes utilisateur doivent être en anglais. Toutefois, si un utilisateur appuie sur une touche de fonction, la commande sous-jacente est censée être dans la langue dominante.
QMF suppose que les écrans d'invite sont dans la langue dominante de l'utilisateur. Pour les écrans d'invite de commande EXPORT et IMPORT, le type d'ensemble de données, le type de file d'attente de données ou le nom de chemin par défaut est également dans la langue dominante.
Le profil QMF en vigueur pour la session est le profil de l'utilisateur sous la langue nationale définie au démarrage de l'application. Le profil QMF de la langue dominante n'est pas le profil en vigueur. Par exemple, un utilisateur qui exécute QMF à la fois en anglais et en français dispose de profils QMF en anglais et en français. Si l'utilisateur démarre une session QMF sous la fonction NLF en français, les options figurant dans le profil QMF en français sont en vigueur. L'utilisateur définit ensuite la variable DSQEC_NLFCMD_LANG sur l'anglais pour exécuter une procédure écrite avec des commandes en anglais. Dans ce cas, les options figurant dans le profil QMF en français demeurent en vigueur tout au long de la session.
- 4** Cette ligne de la procédure renvoie à la langue dominante.
A la fin de l'application, restaurez la valeur d'origine de la variable de langue de commande comme indiqué dans l'exemple.

Procédures de lancement dans les applications bilingues

Si votre application démarre QMF et exécute une procédure de lancement, QMF lance cette procédure chaque fois que l'utilisateur émet la commande END. QMF prend fin si cette procédure détecte une erreur.

Par exemple, un utilisateur qui exécute QMF en anglais émet une commande END dans la langue dominante. QMF interprète la commande comme une erreur et prend fin.

Vous pouvez empêcher cela en procédant de l'une des deux façons suivantes :

- Modifiez la procédure de lancement pour qu'elle traite les applications bilingues.

Une procédure de lancement bilingue inclut les commandes présentées ici :

```
"GET GLOBAL (CURR_LANG=DSQEC_NLFCMD_LANG"
"SET GLOBAL (DSQEC_NLFCMD_LANG=0"
:
:
/* Commandes QMF dans la langue dominante */
:
:
"SET GLOBAL (DSQEC_NLFCMD_LANG=CURR_LANG"
```

- Evitez d'exécuter la procédure de lancement après la commande END.

Vous pouvez définir la variable DSQEC_RERUN_IPROC sur 0 afin que QMF n'exécute pas la procédure de lancement lorsque l'utilisateur émet la commande END.

Commandes en anglais uniquement

Pour la plupart des commandes QMF, vous devez modifier la variable de langue dominante avant d'exécuter la commande en anglais. Toutefois, certaines commandes doivent être émises en anglais même lorsque la variable de langue dominante n'est pas définie sur l'anglais.

Par exemple, vous disposez d'une application interactive que vous souhaitez écrire en anglais et exécuter dans une langue nationale. Vous devez utiliser la commande MESSAGE pour envoyer des messages personnalisés à l'utilisateur. En outre, vous avez besoin de la commande INTERACT pour afficher le message, comme dans l'exemple ci-dessous (qui peut être exécuté dans une session NLF en français) :

```
proceed_text = 'Continuer... '
"RUN WEEKLY_Q" /* Utilisez la commande RUN en anglais */
"FIXER GLOBALES (DSQEC_NLFCMD_LANG=0" /* Basculez à nouveau vers le français */
"MESSAGE (TEXT='\"proceed_text\"'" /* Message en français */
"INTERACT" /* Afficher le rapport avec le message */
```

Figure 8. Utilisation des commandes MESSAGE et INTERACT pour afficher des messages

Les commandes en anglais suivantes fonctionnent dans n'importe quel environnement en langue nationale :

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

Environnements multilingues

Lorsque plusieurs fonctions NLF sont installées, un environnement multilingue est créé. Si vous disposez de l'autorisation appropriée, vous pouvez choisir une langue dominante pour chaque session QMF.

Par exemple, vous pouvez choisir l'anglais pour une session et le français pour une autre, à condition que la fonction NLF en français soit installée. Bien que vous ne puissiez pas passer d'une langue à l'autre au cours d'une session QMF, vous

pouvez changer la variable de la langue de commande. Mettez fin à la session en cours et démarrez-en une autre pour accéder à l'environnement de langue approprié.

Création d'applications traduisibles

Pour gagner du temps, adaptez une application à de nouvelles langues en utilisant des variables pour autant d'objets sensibles à la langue que vous le souhaitez. Utilisez des variables pour exécuter le même programme dans plusieurs fonctions NLF.

Ces variables peuvent inclure :

- Les expressions, noms d'objet et options d'une commande QMF
- Des noms d'écran définis par l'utilisateur

Si vous créez vos propres écrans ISPF pour votre application, vous avez besoin d'un ensemble d'écrans traduits pour chaque langue dans laquelle l'application est exécutée. Affectez un nom unique à ces écrans et rendez-les accessibles aux utilisateurs de l'application. L'application peut ensuite utiliser des variables pour les noms d'écran.

- Identificateurs de message définis par l'utilisateur

Si vous créez vos propres écrans ISPF, vous pouvez également créer les messages qui seront émis par ISPF. Des ID uniques sont affectés à ces écrans de message et vous pouvez utiliser des variables pour y faire référence. Traduisez le texte des messages dans les langues NLF appropriées. L'application peut utiliser des variables pour les noms de message.

Chapitre 8. Commandes QMF dans des applications

Certaines commandes sont conçues pour être utilisées dans des applications et vous pouvez créer vos propres synonymes de commande.

Commandes conçues pour des applications

Toute commande qui est valide sur la ligne de commande QMF dans un environnement donné est valide dans une application. Toutefois, certaines commandes sont spécialement conçues pour être utilisées dans des applications.

Les commandes qui sont conçues pour des applications peuvent être utilisées à la fois dans des applications d'interface appelable et dans des applications d'interface de commande, à deux exceptions près. Vous pouvez utiliser les commandes START et TRACE uniquement avec l'interface appelable.

CONNECT

Utilisez la commande QMF CONNECT pour accéder à des données et objets sur un serveur distant. Lorsque vous vous connectez au système distant, il devient l'emplacement en cours.

Lorsque vous écrivez des applications, vous pouvez émettre la commande CONNECT à partir :

- De l'interface appelable
- De l'interface de commande
- D'une procédure (linéaire ou navigationnelle)

Toutefois, vous ne pouvez pas émettre la commande CONNECT si la procédure est un paramètre sur l'instruction CALL qui démarre QMF for TSO en tant que procédure mémorisée DB2 for z/OS. La connectivité avec les serveurs distants n'est pas prise en charge lorsque QMF est démarré en tant que procédure mémorisée.

Certains aspects de vos applications peuvent être affectés par l'utilisation de la commande QMF CONNECT pour accéder à un serveur distant. Vous devez tenir compte des considérations suivantes :

- Lorsque votre application se connecte à un nouvel emplacement, les valeurs présentes sur le nouvel emplacement (en cours) sont restaurées pour le profil QMF, les synonymes de commande et les touches de fonction.
- Au démarrage de QMF, le programme peut émettre une commande QMF CONNECT pour se connecter à un serveur distant. Les commandes QMF ou instructions SQL ultérieures qui affectent les objets de base de données sont exécutées sur le serveur distant.
- Les différents types de commandes se comportent différemment avec une unité d'oeuvre éloignée. Lorsque vos applications utilisent une unité d'oeuvre, toutes les commandes spécifiques au système et la plupart des commandes QMF s'exécutent sur le système où QMF est en cours d'exécution (système local). Toutefois, lorsqu'une commande QMF effectue l'une des opérations suivantes, les commandes affectent la base de données sur le serveur distant :
 - Envoi de commandes SQL à la base de données

- Utilisation ou modification des données et des objets QMF stockés dans la base de données
- Si votre site utilise TSO et bénéficie de la prise en charge RACF pour les mots de passe en majuscules/minuscules, vérifiez que l'option CASE de votre profil QMF est définie sur MIXED. Sinon, QMF convertit toutes les entrées en majuscules, provoquant ainsi l'échec de la commande CONNECT car la casse du mot de passe est incorrecte. Lorsque l'option est définie sur CASE=MIXED, n'oubliez pas d'indiquer aux utilisateurs de l'application QMF qu'ils doivent saisir toutes les entrées en majuscules, car QMF ne reconnaît que les commandes en majuscules.

Les instructions de l'exemple suivant montrent l'utilisation de la commande QMF CONNECT par un programme d'interface appelable REXX. Le programme se connecte à un serveur distant, exécute une tâche prédéfinie et quitte QMF :

1. L'instruction suivante démarre une session QMF :

```
CALL DSQCIX "START"
```

2. L'instruction suivante se connecte à la base de données DB2 distante (DALLAS) :

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

3. Cette instruction exécute une procédure navigationnelle (EARNINGS) qui recherche des données sur le serveur distant, les formate et imprime le rapport :

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

La procédure EARNINGS contient la logique suivante :

```

:
"RUN QUERY EARNQ (FORM=EARNF"
"PRINT REPORT"
:

```

4. Cette instruction met fin à la session QMF :

```
CALL DSQCIX "EXIT"
```

END

Vous pouvez utiliser la commande END pour définir l'écran d'accueil QMF comme écran en cours.

Par exemple, un rapport QMF est l'écran QMF en cours. L'émission de la commande END à partir d'un programme d'interface appelable ou d'un programme d'interface de commande définit l'écran d'accueil QMF en tant qu'écran en cours. Lorsque l'écran d'accueil QMF est l'écran en cours, l'émission de la commande END n'a aucun effet sur la session QMF.

EXIT

La commande EXIT fonctionne de façon identique quel que soit le mode de démarrage de la session QMF : elle marque toutes les sessions de l'utilisateur en vue de leur arrêt.

Lorsque la commande EXIT est saisie sur la ligne de commande, la session dans laquelle elle est entrée prend fin immédiatement. Chaque session lancée à l'aide de la commande INTERACT prend fin en même temps que l'application qui l'a démarrée. Lorsque la commande EXIT est émise dans une application, la session se termine en même temps que la session QMF d'origine. Toutes les sessions interactives lancées à l'aide de la commande INTERACT doivent se terminer avant QMF.

Dans un programme d'interface callable, il est important d'inclure la commande QMF EXIT lorsque l'application n'a plus besoin de QMF. Si vous oubliez d'inclure cette commande, votre session QMF reste active tant que vous ne vous déconnectez pas ou tant que votre travail par lots n'est pas terminé.

Lorsque l'utilisateur ou une application émet la commande EXIT, QMF définit DSQAO_TERMINATE sur 1 (marqué en vue de son arrêt). Seule une application qui s'exécute dans QMF peut tester et utiliser cette variable globale. Si DSQAO_TERMINATE est défini sur 1 lorsque QMF renvoie à la session QMF principale, QMF prend fin immédiatement et libère les ressources.

Référence associée:

«Variables globales relatives aux informations d'état non liées au profil», à la page 216

Les variables globales DSQAO contiennent des informations d'état, des paramètres ou des indicateurs. La commande SET GLOBAL ne permet pas de les modifier.

GET GLOBAL (syntaxe étendue)

Vous pouvez utiliser la commande GET GLOBAL pour accéder aux variables globales QMF dans votre application. Pour les autres langages que REXX, QMF fournit une syntaxe étendue pour la commande GET GLOBAL.

► GET Global—(—| Définitions de variable |—————►

Définitions de variable :

|—nombre de noms de variable—,—longueurs des noms de variable—,—noms de variable—,—
►—longueurs des valeurs—,—valeurs—,—type de valeur—————|

Vous pouvez utiliser la commande GET GLOBAL pour copier la valeur d'une variable globale QMF dans une variable définie par une application afin qu'elle soit utilisée par l'application. Les paramètres que vous spécifiez dans la commande GET GLOBAL définissent la variable d'application.

nombre de noms de variable
Nombre de variables requis.

longueurs des noms de variable
Liste de longueurs pour chaque nom de variable spécifié.

Longueur du nom de variable. Une zone de 18 caractères remplie à l'aide de blancs de fin est autorisée.

noms de variable
Liste des noms de variables QMF.

N'indiquez pas de blancs de fin dans les noms de variable globale, car QMF les supprime.

longueurs des valeurs
Liste des longueurs des valeurs de variables.

Les règles suivantes s'appliquent à la valeur de variable :

- Si la longueur de valeur que vous fournissez est inférieure à la valeur stockée dans QMF, QMF la tronque à droite et renvoie une valeur tronquée.

- Si la longueur de valeur que vous fournissez est supérieure à la valeur stockée dans QMF, QMF renvoie une valeur remplie à l'aide de blancs de fin.

valeurs

Liste de valeurs de variable.

type de valeur

Type de données de la zone de stockage qui contient les valeurs ; il doit s'agir du type caractère ou entier.

INTERACT

Vous pouvez utiliser la commande INTERACT pour afficher l'écran QMF en cours et permettre aux utilisateurs d'interagir avec QMF à différents stades de votre application. La commande INTERACT possède deux formats : session et commande.

Lorsque l'utilisateur émet la commande END à partir d'un écran QMF, QMF renvoie à l'application. Lorsque l'utilisateur émet la commande EXIT à partir d'un écran QMF, la session QMF est marquée en vue de son arrêt et QMF renvoie à l'application.

Format de session d'INTERACT

Lorsque vous émettez la commande INTERACT, QMF place l'utilisateur sur l'écran en cours et l'autorise à émettre des commandes QMF en mode interactif. La commande INTERACT fournit une autre session QMF dans le cadre de votre session en cours.

La commande INTERACT peut placer l'utilisateur dans une session QMF interactive ou une session ICU GDDM interactive.

- Pour une session QMF interactive :
Émettez la commande INTERACT après une commande QMF qui doit normalement afficher un écran QMF. Dans cette session, l'utilisateur peut saisir des commandes valides pour une session QMF interactive.
- Pour une session ICU GDDM interactive :
Émettez la commande INTERACT après une commande qui entraîne normalement le démarrage d'ICU GDDM par QMF et l'affichage de l'écran ICU. Dans cette session, l'utilisateur peut saisir des commandes valides pour l'utilitaire ICU.

Scénario

Cet exemple montre une procédure qui ne nécessite qu'une seule étape pour la génération d'un rapport.

```
/* Cette procédure permet d'imprimer le rapport de ventes hebdomadaire. */
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
"PRINT REPORT"
```

Figure 9. Procédure simple sans commande INTERACT

QMF affiche l'écran REPORT qui contient vos données formatées ainsi que le message suivant : "Votre procédure a été exécutée."

Ecrivez une procédure impliquant diverses étapes si vous voulez voir les résultats intermédiaires de la procédure. Par exemple, supposons que vous souhaitez voir les résultats intermédiaires d'une procédure exécutant plusieurs requêtes. Utilisez

la commande INTERACT aux stades de la procédure où vous voulez voir les résultats d'une commande. Dans ce cas, insérez une commande INTERACT immédiatement après la première commande RUN QUERY dans la procédure suivante :

```
/* Cette procédure permet de générer un rapport affichant les ventes annuelles. */  
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"  
"INTERACT"  
"RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"
```

Figure 10. Utilisation d'INTERACT dans une procédure

Lorsque vous exécutez cette procédure à partir de l'écran d'accueil, QMF affiche l'écran REPORT qui contient vos données formatées. Saisissez ensuite la commande END à partir de l'écran REPORT. La procédure exécute la deuxième requête et affiche le rapport final. Si vous omettez la commande INTERACT, QMF affiche uniquement le rapport final sans montrer le résultat de la première requête.

Suppression de l'affichage des rapports

Si vous exécutez une requête dans une application d'interface appellable QMF, QMF affiche le rapport obtenu. Toutefois, vous pouvez demander à QMF de ne pas afficher automatiquement le rapport en définissant la variable globale DSQDC_DISPLAY_RPT sur zéro (0). Vous pouvez également définir cette variable globale dans la commande START en spécifiant DSQADPAN=0.

Cette variable globale est valide uniquement lorsque la commande RUN QUERY est émise à partir d'une application. Elle n'affecte pas l'affichage des rapports lorsque la commande RUN QUERY est émise à partir de la ligne de commande QMF.

Arrêt d'une session démarrée à l'aide de la commande INTERACT

Lorsque l'utilisateur émet la commande END, il est renvoyé au processus qui a émis la commande INTERACT ; toutefois, les deux sessions ne sont pas indépendantes. Toute opération effectuée au cours de la session INTERACT demeure en vigueur à la reprise de l'ancienne session. Par exemple, l'utilisateur modifie l'objet format en cours dans la nouvelle session interactive. Dans ce cas, l'objet format en cours dans l'ancienne session contient ces modifications lorsque la nouvelle session prend fin.

Vous pouvez faire en sorte que votre application affiche l'écran d'accueil QMF lorsque l'utilisateur émet une commande END à partir d'un écran d'objet QMF. Ajoutez la logique d'une boucle INTERACT.

Référence associée:

«Exemple REXX d'utilisation d'une boucle INTERACT», à la page 210
Vous pouvez faire en sorte que la commande END dans une session interactive se comporte de la même façon que dans une session interactive QMF.

Format de commande d'INTERACT

L'interface de commande (DSQCCI) exécute les commandes QMF en mode interactif uniquement lorsque l'application d'interface de commande utilise le format de commande d'INTERACT et que QMF exécute une session interactive (DSQSMODE=I).

Le format de commande d'INTERACT n'a aucun effet sur une commande émise via l'interface appellable. Dans l'interface appellable, la seule façon de contrôler si

les commandes sont exécutées de façon interactive consiste à définir DSQSMODE=I dans la commande START.

Utilisez la syntaxe de commande suivante pour demander l'exécution interactive d'une commande donnée :

INTERACT *commande*

Dans cette instruction, *commande* représente la commande que vous voulez exécuter en mode interactif. Divers écrans d'état et d'invite QMF peuvent apparaître dans cette boîte de dialogue.

Par exemple, la commande suivante affiche l'écran d'invite de commande pour les options de commande RUN QUERY :

INTERACT RUN QUERY ABC ?

Si l'exécution interactive n'est pas autorisée, le format de commande d'INTERACT n'a aucun effet sur la commande qu'il précède. Une session interactive n'est pas autorisée dans une session de traitement par lots QMF ou lorsque QMF for TSO est démarré en tant que procédure mémorisée DB2 for z/OS.

Vous pouvez vérifier si l'exécution interactive est autorisée dans la session en cours en examinant la variable globale DSQAO_INTERACT. Une valeur 1 pour la variable globale DSQAO_INTERACT signifie que la commande INTERACT est autorisée.

Référence associée:

«Mots clés de la commande START», à la page 56

Vous pouvez spécifier des mots clés dans la commande START.

«Variables globales relatives aux informations d'état non liées au profil», à la page 216

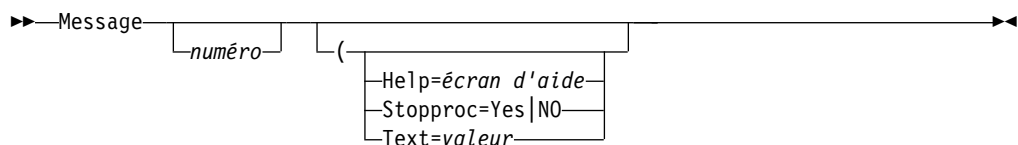
Les variables globales DSQAO contiennent des informations d'état, des paramètres ou des indicateurs. La commande SET GLOBAL ne permet pas de les modifier.

MESSAGE

Lorsque vous créez des applications, vous voulez souvent envoyer à vos utilisateurs des messages spécifiques sur les informations qui s'affichent à leur propos ou sur l'action suivante. Vous pouvez écrire vos propres messages et les afficher sur les écrans QMF via la commande MESSAGE. Dans ISPF, vous pouvez également demander à QMF d'afficher l'aide relative à un message d'erreur ISPF.

Syntaxe

La syntaxe de la commande MESSAGE est la suivante :



numéro (avec ISPF uniquement)

numéro n'est valide que sous ISPF. Ce paramètre représente le numéro d'identification d'une définition de message dans une bibliothèque de messages ISPF.

HELP

Utilisez ce paramètre pour spécifier un écran d'aide différent de celui défini avec le message qui s'affiche normalement dans cette situation. Remplacez *écran d'aide* par l'ID écran approprié.

Vous ne pouvez pas modifier un écran QMF à afficher si sa définition se trouve dans DSQPNLE.

Dans ISPF, si vous voulez créer et afficher votre propre écran, la définition de l'écran doit se trouver dans une bibliothèque d'écrans ISPF. Cette bibliothèque doit être concaténée à votre ensemble de données ISPLIB. L'écran ne doit être ni un menu, ni un écran de saisie de données, mais un écran d'aide.

Dans ISPF, si vous avez spécifié *numéro*, le paramètre *écran d'aide* prend par défaut la valeur de l'indicateur d'écran d'aide pour la définition de message indiquée par *numéro*.

Dans ISPF, si la définition de message spécifiée par *numéro* ne fait pas référence à un indicateur d'écran d'aide, la commande MESSAGE ne fournit pas d'aide sur le message. L'aide QMF relative à l'écran objet s'affiche à la place sur l'écran de l'utilisateur lorsque l'utilisateur demande de l'aide.

STOPPROC

Utilisez **Stopproc** pour supprimer l'exécution de procédures linéaires en définissant la bascule de fin de procédure. La commande suivante définit la bascule de fin de procédure :

Message (Stopproc=Yes

Lorsque Stopproc=Yes, la bascule de fin de procédure est activée. La valeur par défaut est No (bascule désactivée). Cette bascule n'affecte que les procédures linéaires.

Lorsque cette bascule est activée, toute procédure QMF à laquelle vous êtes renvoyé prend immédiatement fin. Lorsque la bascule est désactivée, les procédures s'exécutent normalement.

Lorsque la bascule est désactivée, une commande MESSAGE suffit à la réactiver. Lorsque la bascule est activée, elle n'est pas désactivée tant que l'une des conditions suivantes ne se produit pas :

- Une autre commande QMF est émise. Il peut s'agir de n'importe quelle commande QMF, à l'exception d'une commande MESSAGE avec l'option permettant d'activer la bascule.
- L'utilisateur reprend le contrôle à la fin de l'application. Un utilisateur peut toujours émettre des commandes en ligne qui exécutent des procédures QMF.

Vous pouvez vérifier si la bascule de fin de procédure est activée en examinant la variable DSQCM_MESSAGE. Si l'option de fin est en vigueur, cette variable contient le message relatif à la commande MESSAGE qui a activé la bascule de fin.

TEXT

Utilisez l'option TEXT pour définir un message ou pour remplacer le texte dans une définition de message ISPF. Remplacez *valeur* par la chaîne de caractères à utiliser pour le message. Une valeur comportant des espaces doit être encadrée de délimiteurs : apostrophes, parenthèses et guillemets. Lorsque les délimiteurs sont des guillemets, ces derniers sont considérés comme faisant partie du message. La longueur maximale pour une valeur de message est 360 caractères mono-octet. Le nombre de caractères affichés dans le message est déterminé par la largeur de l'écran que vous utilisez. Une valeur supérieure à

78 caractères est tronquée pour ne contenir que les 78 premiers caractères. QMF ne convertit pas le texte en majuscules ; toutefois, ISPF peut réaliser cette opération si la commande MESSAGE est émise via l'interface de commande.

Si votre message contient des guillemets, vous devez les doubler dans la spécification TEXT=.

Dans ISPF, le paramètre par défaut est le texte long du message ISPF spécifié par *numéro*, qui devient le message généré. Le texte demeure en l'état ; aucune conversion n'a lieu, quelle que soit la valeur du paramètre CASE dans le profil QMF de l'utilisateur.

Supposons que vous vouliez écrire une application à l'aide d'une procédure exécutant deux requêtes et affichant deux rapports. Lorsque QMF affiche le premier rapport, vous voulez indiquer aux utilisateurs comme afficher le second. Vous pouvez écrire une procédure linéaire comme celle présentée ici, qui inclut sur l'écran REPORT message défini par la commande MESSAGE. Pour que votre message s'affiche sur l'écran REPORT, placez la commande MESSAGE immédiatement avant la commande INTERACT, comme indiqué ici :

```
⋮
  RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F
  MESSAGE (TEXT='Appuyez sur FIN quand vous avez fini de visualiser ce rapport.'
  INTERACT
  RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F
⋮
```

Figure 11. Exemple d'utilisation de la commande MESSAGE

Si vous utilisez une procédure navigationnelle, vous pouvez employer une variable REXX à la place de la chaîne de texte que vous spécifiez pour la commande MESSAGE, comme indiqué ici. Lorsque vous utilisez des variables REXX, vous devez entourer le nom de variable de guillemets dans la chaîne de texte du message.

```
oktext = 'Appuyez sur FIN quand vous avez fini de visualiser ce rapport.'
"RUN QUERY WEEKLY_SALES_Q (FORM=WEEKLY_SALES_F"
"MESSAGE (TEXT='oktext'"
"INTERACT"
"RUN QUERY YEAR_TOTAL_Q (FORM=YEAR_TOTAL_F"
```

Figure 12. Utilisation de variables REXX avec la commande MESSAGE dans une procédure

Ce message s'étend sur plusieurs lignes avec des caractères de continuation REXX.

```
/* PROCEDURE QMF REXX */
MSGTEXT="Vous avez saisi une valeur de données incompatible avec "||,
"le type de données de colonne ; vérifiez le type de données de la "||,
"colonne et réessayez."
"MESSAGE(TEXT=(MSGTEXT))"
EXIT
```

Exemples

Voici des exemples d'émission de la commande MESSAGE dans diverses conditions.

- Exemple d'émission de la commande MESSAGE à partir d'une procédure linéaire QMF :

Ce message s'étend sur plusieurs lignes en utilisant des caractères de continuation pour les procédures linéaires :

```
MESSAGE(TEXT='Vous avez saisi une valeur de données incompatible avec  
+le type de données de colonne ; vérifiez le type de données de la  
+colonne et réessayez.')
```

- Exemples d'utilisation de commandes MESSAGE avec ISPF :

Voici des exemples d'utilisation de la commande MESSAGE si vous créez une application utilisant ISPF :

– MESSAGE MSG011X

- Le texte du message est le message long dans MSG011X.
- L'écran d'aide du message est l'écran identifié (le cas échéant) dans MSG011X.
- L'activation ou non de la bascule de fin de procédure après le traitement de la commande par QMF est déterminée par la bascule de fin de procédure dans MSG011X.

– MESSAGE MSG011X (HELP=ANELX STOPPROC=YES

- Le texte du message est le message long dans MSG011X.
- L'écran d'aide du message est un écran appelé ANELX.
- La bascule de fin de procédure est activée, ce qui supprime l'exécution des procédures linéaires QMF dans l'application.

SET GLOBAL (syntaxe étendue)

Pour créer vos propres variables globales et les utiliser dans des commandes QMF en tant que variables de substitution, émettez la commande SET GLOBAL. Vous pouvez également utiliser la commande SET GLOBAL pour définir des valeurs pour les variables globales QMF prédéfinies, qui commencent par "DSQ".

Restriction : N'utilisez pas la commande SET GLOBAL pour définir les variables globales qui sont définies comme étant en lecture seule.

Syntaxe de la commande SET GLOBAL (syntaxe étendue)

Vous pouvez utiliser la syntaxe étendue de la commande SET GLOBAL pour modifier les valeurs des variables dans les langages d'interface appelable autres que REXX. Les autres langages sont les suivants : Assembler, C, COBOL, Fortran et PL/I.

Le nom de variable peut comprendre 18 caractères au maximum pour les variables utilisées avec les applications d'interface appelable. Si la variable doit être utilisée comme variable de substitution, son nom peut comporter jusqu'à 17 caractères. La longueur maximale de la commande, dont la syntaxe de commande, est 2 000 octets.

La syntaxe de la commande est la suivante :

```
►►—SET GLOBAL—(—| Définitions de variable |—————►
```

Définitions de variable :

```
|—nombre de noms de variable—,—longueurs des noms de variable—,—noms de variable—,—————►
```

```
►—longueurs des valeurs—,—valeurs—,—type de valeur—————|
```

nombre de noms de variable

Nombre de variables requis.

longueurs des noms de variable

Liste de longueurs pour chaque nom de variable spécifié.

Vérifiez que la longueur du nom de variable globale est égale à la longueur réelle du nom global dans votre programme. Une zone de 18 caractères remplie à l'aide de blancs de fin est autorisée.

noms de variable

Liste des noms de variables QMF.

longueurs des valeurs

Liste des longueurs des valeurs de variables. Si la longueur de valeur que vous indiquez est inférieure à la longueur de la valeur stockée dans votre zone de stockage, la valeur est tronquée à droite lorsqu'elle est stockée dans QMF.

QMF utilise la valeur provenant de votre programme, en commençant par l'adresse affectée à la longueur que vous avez définie. Si la longueur est excessive, QMF risque de prendre fin de façon anormale.

valeurs

Liste de valeurs de variable.

type de valeur

Type de données de la zone de stockage qui contient les valeurs. Il doit s'agir du type caractère ou entier.

Des exemples d'utilisation de la syntaxe étendue de la commande SET GLOBAL sont documentés avec les spécifications de langage de programmation.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

Instructions pour la définition et l'utilisation de variables globales

Les noms de variables globales sont soumis à certaines règles.

Lorsque vous définissez et utilisez des noms de variables globales, gardez à l'esprit les règles suivantes :

- Dans la commande SET GLOBAL, les noms de variables ne sont pas précédés d'une perluète, contrairement aux commandes RUN et CONVERT.
- Si vous créez une variable globale portant le même nom qu'une variable de format ou une variable d'agrégation, QMF n'utilise pas la variable globale dans le format. QMF utilise la valeur de la variable de format (ou variable d'agrégation) dans le format plutôt que la valeur de la variable globale.
- Le format QMF ne reconnaît pas les variables globales dont les noms contiennent des points d'interrogation.
- Le nom d'une variable globale est limité à 18 caractères, sauf si cette dernière doit être utilisée en tant que variable de substitution. Le nom d'une variable de substitution est limité à 17 caractères.
- Un nom de variable globale peut contenir des caractères numériques, mais le premier caractère d'un nom de variable globale ne doit pas être numérique.
- Les noms de variables globales ne doivent pas commencer par DSQ car QMF réserve ce préfixe aux variables globales QMF prédéfinies.

- Le premier caractère d'un nom de variable globale doit être un caractère alphabétique (de A à Z) ou l'un des caractères spéciaux suivants :

¢ ! \$ ~ { } ? @ # % \

- Un nom de variable globale ne peut contenir ni blanc, ni l'un des caractères suivants :

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

- QMF retire les blancs de fin des noms de variables globales.
- Par défaut, une valeur de variable globale est conservée tant que vous ne la réinitialisez pas ou que vous ne mettez pas fin à la session QMF. Cependant, vous pouvez définir la variable globale DSQEC_USERGLV_SAV pour sauvegarder les valeurs des variables globales d'une session dans une autre.

Référence associée:

«Variables globales contrôlant le mode d'exécution des commandes et des procédures», à la page 237

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

Annexe B, «Variables globales QMF», à la page 215

QMF fournit de nombreuses variables globales qui vous aident à contrôler divers aspects de votre session QMF, de vos commandes QMF et de l'affichage d'écran. Les variables globales vous aident également à gérer le comportement des fonctions QMF dans des procédures et des applications.

START

Lorsque vous démarrez QMF via l'interface callable, vous devez utiliser la commande START.

Syntaxe générale

La syntaxe de la commande START dépend du langage de programmation que vous utilisez pour votre application callable.

Vous ne pouvez disposer que d'une seule session QMF active à la fois. Lorsque vous démarrez QMF à partir d'une application, émettez une commande START pour tester si QMF est démarré.

Des exemples de syntaxe pour chaque langage de programmation sont documentés avec les spécifications du langage de programmation.

Voici la syntaxe générale pour la commande START :

►►—START—(—| Définitions de mot clé |—————►►

Définitions de mot clé :

|—nombre de mots clés—,—longueurs des mots clés—,—mots clés—,—————►

►—longueurs des valeurs—,—valeurs—,—type de valeur—————|

Les langages Assembler, C, COBOL, Fortran et PL/I utilisent les spécifications suivantes pour la commande START :

nombre de mots clés

Nombre de mots clés de commande START que vous utilisez dans votre commande START.

longueurs des mots clés

Longueur de chaque mot clé de commande START spécifié.

mots clés

Noms des mots clés de commande START.

longueurs des valeurs

Liste contenant les longueurs des valeurs pour chaque mot clé de commande START.

valeurs

Liste de valeurs pour les mots clés de commande START spécifiés dans cette commande.

type de valeur

Type de données de la valeur. Les données doivent être de type caractère pour la commande START.

Concepts associés:

«Démarrage de QMF à partir d'une application», à la page 21

Avant d'exécuter toute commande à partir d'une application, vous devez démarrer QMF. Lors de l'utilisation de l'interface callable, démarrez QMF en émettant la commande START dans votre application. Vous ne pouvez disposer que d'une seule session QMF à la fois.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

«Mots clés de la commande START»

Vous pouvez spécifier des mots clés dans la commande START.

Mots clés de la commande START

Vous pouvez spécifier des mots clés dans la commande START.

Les mots clés suivants sont disponibles dans la commande START :

- DSQADPAN
- DSQALANG
- DSQSBSTG
- DSQSCMD (TSO uniquement)
- DSQSDBCS
- DSQSDBNM
- DSQSDBQN (CICS uniquement)
- DSQSDBQT (CICS uniquement)
- DSQSDEBUG
- DSQSIROW
- DSQSMODE
- DSQSMRFI
- DSQSPILL
- DSQSPLAN (TSO uniquement)
- DSQSPRID (TSO uniquement)

- DSQSPTYP (TSO uniquement)
- DSQSRSTG (TSO uniquement)
- DSQSRUN
- DSQSSPQN (CICS uniquement)
- DSQSSUBS (TSO uniquement)

Ces mots clés sont décrits dans le tableau ci-dessous.

Vous pouvez spécifier les mots clés de la commande START en respectant les conventions suivantes :

- Vous pouvez indiquer n'importe quel mot clé dans la commande START. Dans TSO, vous pouvez également spécifier n'importe quel mot clé dans le programme REXX nommé par le mot clé DSQSCMD. Etant donné que QMF for CICS ne prend pas en charge le langage REXX, vous devez indiquer tous les mots clés dans la commande START.
- Si votre application ou la procédure de lancement (spécifiée par le mot clé DSQSRUN) indique des mots clés qui ne sont pas pris en charge dans un environnement donné, ces mots clés sont ignorés. Cette convention vous permet de compiler un programme unique pour qu'il s'exécute dans plusieurs environnements QMF sans modifier les mots clés spécifiques à l'environnement.
- Si vous ne spécifiez aucun mot clé, QMF utilise les valeurs des mots clés de la commande START telles qu'elles apparaissent dans le programme indiqué par le mot clé DSQSCMD. Si vous n'utilisez pas ce programme, QMF utilise les valeurs par défaut de chaque mot clé affiché dans ce tableau.

Tableau 8. Mots clés, descriptions et valeurs par défaut de la commande START

Mots clés de la commande START	Description	Valeur par défaut
DSQADPAN	A utiliser uniquement avec l'interface callable, ce paramètre de commande START définit la variable globale DSQDC_DISPLAY_RPT. Cette variable détermine si QMF doit afficher le rapport lorsqu'une requête est exécutée à partir d'un programme d'application. La valeur 1 affiche le rapport lors de l'exécution d'une requête. Définissez la valeur sur 0 pour ne pas afficher le rapport.	1 (affichage du rapport)

Tableau 8. Mots clés, descriptions et valeurs par défaut de la commande START (suite)

Mots clés de la commande START	Description	Valeur par défaut
DSQALANG	<p>Détermine la langue nationale dominante pour la session que vous démarrez. Vous pouvez spécifier ce paramètre dans vos applications afin que les utilisateurs puissent saisir ou spécifier des commandes QMF dans une langue nationale. La valeur de ce paramètre est un identificateur de langue nationale d'un caractère, qui est indiqué ici. Si vous voulez saisir des commandes en anglais alors que la langue dominante est une autre langue, vous pouvez utiliser le support bilingue de QMF. La fonction de langue nationale pour la langue que vous spécifiez doit être installée.</p> <ul style="list-style-type: none"> C - Français (Canada) D - Allemand E - Anglais F - Français H - Coréen (Hangeul) I - Italien K - Japonais (Kanji) P - Portugais (Brésil) Q - Danois S - Espagnol U - Anglais en lettres capitales V - Suédois Y - Français (Suisse) Z - Allemand (Suisse) 	E (Anglais)

Tableau 8. Mots clés, descriptions et valeurs par défaut de la commande START (suite)

Mots clés de la commande START	Description	Valeur par défaut
DSQSBSTG	<p>Indique la quantité maximale de mémoire virtuelle par utilisateur qui sera utilisée pour la génération de rapports QMF. Les valeurs admises sont les suivantes :</p> <p>0 - 2 147 483 647 Spécifie la mémoire en octets.</p> <p>0 ko - 2 097 152 ko Spécifie la mémoire en ko.</p> <p>0 Mo - 2 048 Mo Spécifie la mémoire en Mo.</p> <p>0 Go - 2 Go Spécifie la mémoire en Go.</p> <p>1 % - 100 % Spécifie un pourcentage de mémoire disponible. Les pourcentages sont valides dans TSO uniquement.</p> <p>Si vous utilisez CICS, vous pouvez indiquer une quantité fixe de mémoire en octets, ko, Mo ou Go.</p> <p>Si vous utilisez TSO, QMF détermine par défaut la quantité maximale de mémoire à utiliser pour la génération de rapports QMF. Si vous affectez une valeur valide différente de zéro au paramètre DSQSBSTG, la quantité de mémoire que vous spécifiez pour le paramètre est utilisée à la place. Si vous utilisez TSO, vous pouvez indiquer la valeur du paramètre DSQSBSTG en tant que quantité de mémoire fixe ou en tant que pourcentage de la mémoire virtuelle disponible. Si vous définissez le paramètre DSQSBSTG sur 0 et que vous affectez une valeur différente de zéro au paramètre DSQSRSTG, le paramètre DSQSRSTG est utilisé pour déterminer la quantité de mémoire maximale pour les rapports. Si vous spécifiez des valeurs valides différentes de zéro à la fois pour DSQSBSTG et DSQSRSTG, la valeur de DSQSBSTG est utilisée.</p> <p>Si vous définissez pour DSQSBSTG une valeur inférieure à la quantité minimale de stockage requise pour la génération d'un rapport, QMF affecte automatiquement la quantité minimale de mémoire requise. Dans TSO, la quantité minimale de mémoire est 15 000 octets.</p> <p>La disponibilité de la mémoire est réévaluée au cours de la session QMF.</p> <p>Restrictions :</p> <ul style="list-style-type: none"> • Les valeurs de pourcentage doivent inclure le signe pourcent (%) et ne doivent pas contenir d'espaces. • Les valeurs au format Go, Mo ou ko ne doivent pas contenir d'espaces (par exemple, vous ne pouvez pas saisir 2 Go). • Les valeurs au format Go, Mo ou ko ne doivent pas contenir de caractères après les unités (par exemple, vous ne pouvez pas saisir 2Goxyz). • Les Go, Mo ou ko qui ne sont pas précédés d'un nombre sont traités comme la valeur 0. 	<p>Dans CICS : 500 000</p> <p>Dans TSO : 0</p>

Tableau 8. Mots clés, descriptions et valeurs par défaut de la commande START (suite)

Mots clés de la commande START	Description	Valeur par défaut
DSQSCMD (TSO uniquement)	<p>Ce mot clé spécifie le programme REXX qui définit les paramètres de programme QMF dans QMF for TSO.</p> <p>Lorsque QMF reçoit la commande START en provenance d'une application d'interface appelable, TSO appelle le programme REXX spécifié par ce mot clé. Ce programme REXX fournit des valeurs pour les paramètres de programme QMF sauf si vous avez indiqué leurs valeurs directement dans la commande START. Le programme par défaut livré avec TSO est DSQSCMDE, qui fournit par défaut des valeurs de paramètre de programme en anglais pour tous les mots clés présentés dans ce tableau. La valeur NULL pour un paramètre donné indique que TSO utilise la valeur par défaut pour ce paramètre.</p> <p>Si vous utilisez une fonction NLF, vous pouvez modifier le nom du programme par défaut en DSQSCMD<i>n</i>. La variable <i>n</i> représente l'identificateur de langue nationale (NLID) pour la langue que vous utilisez.</p> <p>Bien qu'il ne soit pas présenté dans ce tableau, le paramètre DSQSDBLG est également défini par défaut dans le programme DSQSCMDE fourni avec TSO. Ce paramètre est défini lorsque vous démarrez QMF for TSO en tant que procédure mémorisée DB2 for z/OS. Il ne peut pas être défini en externe en dehors du contexte de l'interface de procédure mémorisée.</p>	DSQSCMDE
DSQSDBCS	<p>Détermine si QMF autorise les caractères à double octet lorsque l'écran ne prend pas en charge le jeu de caractères codé sur deux octets (DBCS). Les valeurs sont YES ou NO.</p> <p>Définissez la valeur sur YES si vous voulez imprimer des données DBCS à partir d'un écran non DBCS ou exécuter un travail par lots QMF imprimant des données DBCS. Sinon, définissez la valeur sur NO.</p>	NO
DSQSDBNM	Indique le serveur distant auquel se connecter lors du démarrage d'une session QMF. Une valeur NULL signifie que QMF se connecte à la base de données par défaut (base de données à laquelle il se connecte normalement sans unité d'oeuvre éloignée).	NULL
DSQSDBQN (CICS uniquement)	Spécifie le nom de la file d'attente de stockage CICS à utiliser pour les données de trace QMF. Le nom doit se conformer aux spécifications de nom CICS pour le type de file d'attente CICS spécifié par DSQSDBQT.	DSQD
DSQSDBQT (CICS uniquement)	<p>Spécifie le type de stockage CICS à utiliser pour les données de trace QMF.</p> <p>Les valeurs sont les suivantes :</p> <p>TD Utilise une file d'attente de données transitoires CICS.</p> <p>TS Utilise une file d'attente de mémoire temporaire auxiliaire CICS. Faites attention lors de la spécification de la mémoire temporaire, car QMF peut générer une grande quantité de données de trace.</p>	TD

Tableau 8. Mots clés, descriptions et valeurs par défaut de la commande START (suite)

Mots clés de la commande START	Description	Valeur par défaut
DSQSDEBUG	<p>Indique si l'activité du produit est tracée ou non lors de l'initialisation de QMF. Les valeurs sont les suivantes :</p> <p>ALL Indique le traçage QMF le plus détaillé.</p> <p>NONE N'indique aucun traçage QMF.</p> <p>Lorsque vous démarrez QMF en mode de traitement par lots, tous les messages et commandes sont tracés (ce qui équivaut au niveau de traçage L2), quelle que soit la définition de DSQSDEBUG.</p>	NONE
DSQSIROW	<p>Indique le nombre de lignes extraites par QMF avant l'affichage du premier écran de données pour une commande RUN QUERY, IMPORT DATA ou DISPLAY.</p>	100
DSQSMODE	<p>Indique le mode de démarrage de QMF.</p> <p>I Indique le mode interactif.</p> <p>B Indique le mode de traitement par lots.</p> <p>Lorsque la valeur de DSQSMODE est B, l'affichage de l'écran est bloqué afin que QMF puisse s'exécuter en tant que travail en arrière-plan.</p>	B (traitement par lots)
DSQSMRFI	<p>Indique si la session QMF que vous démarrez utilise ou non l'extraction et l'insertion multilignes DB2. L'extraction et l'insertion multilignes DB2 permettent d'augmenter les performances de nombreuses commandes QMF (comme DISPLAY TABLE, EXPORT DATA ou EXPORT TABLE, IMPORT TABLE, PRINT REPORT ou PRINT TABLE, RUN QUERY ou RUN PROC) lorsque ces commandes extraient des données. L'extraction et l'insertion multilignes DB2 augmentent également les performances de commandes telles que SAVE DATA, DPRE et BOTTOM ou FORWARD lors de la navigation dans un rapport.</p> <p>Ce paramètre définit la variable globale DSQAO_DSQSMRFI.</p> <p>YES Indique que QMF utilise l'extraction et l'insertion multilignes DB2.</p> <p>Lorsque MR est défini sur YES et que vous utilisez une commande QMF incluant un nom tripartite, les serveurs doivent exécuter DB2 for z/OS. Le demandeur où la commande est initiée et le serveur vers lequel la commande est acheminée doivent s'exécuter à ce niveau de version. Les commandes avec des noms tripartites ne peuvent pas être acheminées vers des serveurs DB2 for VSE et VM.</p> <p>NO Indique que QMF n'utilisera pas les fonctions d'extraction et d'insertion multilignes DB2.</p> <p>Lors de l'extraction de données XML ou LOB, QMF utilise l'extraction multiligne, quelle que soit la définition du paramètre DSQSMRFI.</p>	NO

Tableau 8. Mots clés, descriptions et valeurs par défaut de la commande START (suite)

Mots clés de la commande START	Description	Valeur par défaut
DSQSPILL	Indique si QMF utilise ou non une mémoire auxiliaire lorsqu'une mémoire supplémentaire est nécessaire pour les rapports. Les valeurs possibles sont YES ou NO. Si vous spécifiez la valeur YES pour DSQSPILL et que vous utilisez QMF for TSO, vérifiez que le paramètre DSQSPTYP est défini pour s'adapter au type de mémoire auxiliaire que vous utilisez. Si vous utilisez CICS, consultez l'explication du mot clé DSQSSPQN pour savoir comment appeler la file d'attente de mémoire temporaire contenant les données temporaires.	Pour CICS : NO Pour TSO : YES
DSQSPLAN (TSO uniquement)	Indique l'ID de plan d'application DB2 affecté à QMF.	QMF12
DSQSPRID (TSO uniquement)	Indique si l'ID de connexion TSO ou l'ID autorisation de base de données principale doit être utilisé ou non pour sélectionner la ligne appropriée dans Q.PROFILES et pour qualifier les entrées Q.ERROR_LOG. Les valeurs admises sont PRIMEID ou TSOID.	PRIMEID
DSQSPTYP (TSO uniquement)	Lorsque le paramètre DSQSPILL est défini sur YES, indique le type de mémoire pour les données temporaires. La valeur FILE indique un fichier, tandis que la valeur 64BIT indique une mémoire étendue. Dans ce dernier cas, QMF acquiert de la mémoire à chaque appel émis vers le gestionnaire de mémoire étendue conformément à la quantité spécifiée dans la variable globale DSQEC_EXTND_STG.	FILE
DSQSRSTG (TSO uniquement)	Affecte dynamiquement la mémoire virtuelle disponible pour les rapports. Vous pouvez également utiliser le mot clé DSQSBSTG pour définir la quantité de mémoire maximale en tant que quantité fixe ou pourcentage de la mémoire disponible. Si DSQSBSTG est défini sur une valeur différente de zéro, QMF ignore la valeur DSQSRSTG.	0
DSQSRUN	Indique le nom de la procédure de lancement QMF à exécuter après le démarrage de QMF. La procédure de lancement ne s'exécute qu'une seule fois avec l'interface appelable. Dans cette procédure, vous pouvez inclure des commandes permettant de définir des variables globales et des valeurs de profil pour personnaliser la session de l'utilisateur.	NULL
DSQSSPQN (CICS uniquement)	Spécifie le nom de la file d'attente de mémoire temporaire CICS qui est utilisée pour les données temporaires QMF. Lorsque la valeur YES est affectée au paramètre de programme DSQSPILL, cette zone de sauvegarde est utilisée pour contenir les données de rapport.	DSQSvid, où <i>id</i> correspond à l'ID de terminal CICS
DSQSSUBS (TSO uniquement)	Indique l'ID de la base de données DB2 dans laquelle QMF a démarré. L'ID de base de données que vous spécifiez dans ce mot clé doit être configuré en tant que demandeur d'application.	DSN

Concepts associés:

Chapitre 7, «Ecriture d'applications bilingues», à la page 39

De nombreuses applications métier doivent s'exécuter dans plusieurs langues différentes. Vous pouvez écrire une application en anglais et l'exécuter dans n'importe quelle langue prise en charge par QMF.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

«Variables globales contrôlant le mode d'exécution des commandes et des procédures», à la page 237

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

TRACE

Vous pouvez utiliser la commande TRACE pour ajouter des informations de trace provenant d'applications d'interface appelable à la sortie de données de trace QMF. Cette commande peut être utilisée dans des applications en langage Assembleur, C, COBOL, FORTRAN et PL/I. Elle ne peut pas être émise à partir de la ligne de commande QMF.

La syntaxe de la commande TRACE est la suivante :

►►—TRACE—(—| Définitions de zone de trace |—————►

Définitions de zone de trace :

|—nombre de zones de trace—,—longueurs de titre de trace—,—adresses de titre de trace—,—
►—longueurs de zone de trace—,—adresses de zone de trace—,—type de valeur—————|

nombre de zones de trace

Nombre de définitions de zone de trace que vous utilisez dans votre commande TRACE. Ce nombre doit être compris entre 1 et 10.

longueurs de titre de trace

Liste de longueurs pour chaque titre de trace spécifié.

adresses de titre de trace

Liste d'adresses pointant vers les titres de trace qui seront utilisés pour chaque zone de trace. Un titre de trace peut comprendre 40 caractères au maximum. Les blancs de fin sont enlevés. Lorsque le premier titre de trace est SNAPREGS, tous les autres titres de trace et adresses de zone de trace sont ignorés et les valeurs du registre QMF sont écrites dans la trace QMF.

longueurs de zone de trace

Liste contenant le nombre d'octets de stockage qui s'afficheront en commençant à l'adresse de zone de trace correspondante. Les longueurs de zone de trace doivent être intégrées dans des valeurs d'entier FIXED(31).

adresses de zone de trace

Liste d'adresses qui seront affichées dans la sortie de trace. Le nombre d'octets de stockage qui sont affichés sur chaque adresse de zone de trace est déterminé par la longueur de zone de trace. Lorsque le premier titre de trace est «SNAPREGS,» toutes les adresses et longueurs de zone de trace sont ignorées. Lorsque l'adresse de zone de trace est 0, la longueur de zone de trace est également considérée comme 0.

type de valeur

Il doit s'agir de la valeur «FINT.» «FINT» est une valeur constante qui est fournie avec la zone de communication d'interface (DSQCOMM) pour chaque langage de programmation. La constante est un nom semblable à

DSQ_VARIABLE_FINT. Vérifiez votre langage de programmation dans la zone de communication d'interface pour confirmer la valeur de variable.

La commande TRACE écrit des définitions de zone de trace dans la sortie de données de trace QMF, quels que soient les paramètres de trace QMF. Si vous voulez écrire une sortie de trace uniquement lorsque les paramètres de trace QMF sont actifs, utilisez les variables globales QMF DSQAO_APPL_TRACE ou DSQAP_TRACE.

Exemple de codage PL/I pour TRACE

Vous pouvez utiliser la commande TRACE dans une application PL/I pour écrire des informations de trace pour l'application dans la sortie de trace QMF.

L'exemple de codage suivant vérifie d'abord que l'utilisateur trace l'activité de l'application QMF en s'assurant que la variable globale DSQAO_APPL_TRACE n'est pas définie sur '0'. Si la trace de l'application est activée, la commande TRACE est émise. La commande TRACE spécifie trois définitions de zone de trace à écrire dans la sortie de trace QMF.

Cet exemple n'est pas inclus dans le fichier DSQABFP qui est fourni avec QMF.

```

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

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

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

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

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

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

```

```

/* displayed. */

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

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

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

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

DCL 1 GET_TRACE_VALUE,
    3 GET_TRACE_VALUE_1 CHAR(1);

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

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

CALL DSQCIPX(DSQCOMM,
            COMMAND_LENGTH,
            START_QUERY_INTERFACE,
            NUMBER_OF_PARAMETERS,
            KEYWORD_LENGTHS,
            START_KEYWORDS,
            DATA_LENGTHS,
            START_KEYWORD_VALUES,
            DSQ_VARIABLE_CHAR);

```

```

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

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

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

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

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

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             RUN_QUERY);

/*****/
/* Trace command */
/*****/
IF GET_TRACE_VALUE_1 ^= '0' THEN DO;
  NUMBER_OF_PARAMETERS = 3;
  COMMAND_LENGTH = LENGTH(TRACE_COMMAND);
  AREA_DESCRIPTION(1) = 'DSQAO_APPL_TRACE: ';
  AREA_DESCRIPTION_LENGTH(1) = LENGTH(AREA_DESCRIPTION(1));
  AREA_PTR(1) = ADDR(GET_TRACE_VALUE_1);
  AREA_LENGTH(1) = LENGTH(GET_TRACE_VALUE_1);
  AREA_DESCRIPTION(2) = 'DSQ_COMM_LEVEL: ';
  AREA_DESCRIPTION_LENGTH(2) = LENGTH(AREA_DESCRIPTION(2));
  AREA_PTR(2) = ADDR(DSQ_COMM_LEVEL);
  AREA_LENGTH(2) = LENGTH(DSQ_COMM_LEVEL);

```



```

AREA_DESCRIPTION(3) = 'DSQ_CURRENT_COMM_LEVEL: ';
AREA_DESCRIPTION_LENGTH(3) = LENGTH(AREA_DESCRIPTION(3));
AREA_PTR(3) = ADDR(DSQ_CURRENT_COMM_LEVEL);
AREA_LENGTH(3) = LENGTH(DSQ_CURRENT_COMM_LEVEL);

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

END;

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

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             PRINT_REPORT);

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

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             END_QUERY_INTERFACE);

END      DSQABFP;

```

Lorsque le programme détecte que l'utilisateur a activé le traçage, la sortie de trace suivante est écrite dans la sortie de trace QMF :

```

-----
          DSQDTRC :TRACE COMMAND OUTPUT                (14534)
          TRACE_AREA_NUMBER
341033B8: 00000001                                     *....      *
          TRACE_AREA_TITLE
341036A4: C4E2D8C1 D66DC1D7 D7D36DE3 D9C1C3C5      *DSQA0_APPL_TRACE*
341036B4: 7A                                         *:          *
          TRACE_AREA_CONTENTS
340D0A14: F2                                         *2          *
-----
          DSQDTRC :TRACE COMMAND OUTPUT                (14535)
          TRACE_AREA_NUMBER
341033B8: 00000002                                     *....      *
          TRACE_AREA_TITLE
341036A4: C4E2D86D C3D6D4D4 6DD3C5E5 C5D37A      *DSQ_COMM_LEVEL: *
          TRACE_AREA_CONTENTS
340D0500: C4E2D8D3 6EF0F0F1 F0F0F24C              *DSQL>001002<  *
-----
          DSQDTRC :TRACE COMMAND OUTPUT                (14536)
          TRACE_AREA_NUMBER
341033B8: 00000003                                     *....      *
          TRACE_AREA_TITLE
341036A4: C4E2D86D C3E4D9D9 C5D5E36D C3D6D4D4      *DSQ_CURRENT_COMM*
341036B4: 6DD3C5E5 C5D37A                          *_LEVEL:      *
          TRACE_AREA_CONTENTS
33F00C50: C4E2D8D3 6EF0F0F1 F0F0F24C              *DSQL>001002<  *
-----

```

Commandes utilisables dans une mini-session de rapport RUN QUERY

Certaines commandes que vous utilisez dans les applications QMF provoquent l'affichage d'un rapport pendant l'exécution de l'application. Cet environnement est appelé mini-session de rapport. Vous pouvez limiter l'accès des utilisateurs à QMF en utilisant des mini-sessions de rapport. Dans une mini-session de rapport, QMF limite les commandes qu'un utilisateur peut émettre lors de la visualisation d'un rapport.

Une mini-session de rapport se comporte comme une session imbriquée (session au sein d'une session). Dans les mini-sessions, votre session QMF initiale reste intacte, mais devient temporairement inaccessible pendant que vous visualisez un rapport. La mini-session devient votre session active en cours tant que vous n'émettez pas la commande END (ou que vous n'appuyez pas sur la touche de fonction Fin). Lorsque vous mettez fin à une mini-session, vous pouvez revenir soit à la session QMF initiale, soit à l'application appelante, selon la façon dont l'application a été écrite. L'application ne peut pas continuer à émettre de commandes ultérieures tant que la mini-session de rapport n'a pas pris fin.

La variable globale QMF DSQDC_DISPLAY_RPT détermine si QMF démarre une mini-session de rapport. Cette situation se produit parce que DSQDC_DISPLAY_RPT détermine si QMF affiche un rapport après l'exécution d'une requête. Définissez cette variable sur 1 pour afficher le rapport ou sur 0 pour supprimer l'affichage.

Lorsque vous démarrez QMF à l'aide de l'interface callable :

- La valeur par défaut pour la variable globale DSQDC_DISPLAY_RPT est 1. Lorsque QMF est démarré avec DSQQMF n (soit en mode interactif, soit en mode de traitement par lots), la valeur par défaut de cette variable globale est 0. La variable n représente ici l'identificateur de langue nationale.
- Si vous exécutez une procédure ou une application qui exécute une requête, QMF démarre une mini-session de rapport. QMF affiche le rapport généré par la requête dans cette mini-session.
- Si votre procédure ou application n'exécute pas de requête ou si vous exécutez une requête à partir de l'écran Requête SQL, QMF ne démarre pas de mini-session de rapport.

Si vous ne voulez pas que QMF démarre une mini-session de rapport, exécutez l'une des opérations suivantes :

- Affectez la valeur 0 à DSQDC_DISPLAY_RPT.
- Définissez le paramètre DSQADPAN sur 0 lorsque vous démarrez QMF à partir de l'interface callable.

A partir d'une mini-session de rapport, vous pouvez émettre les commandes et synonymes suivants pour ces commandes. Les restrictions sont indiquées entre parenthèses.

- BACKWARD
- BOTTOM
- CANCEL (lorsque la fenêtre en incrustation est active)
- CICS
- DISPLAY REPORT
- DISPLAY CHART
- END

- FORWARD
- GET GLOBAL
- HELP
- INTERACT
- ISPF
- LEFT
- MESSAGE
- PRINT REPORT
- PRINT CHART
- QMF
- RETRIEVE
- RIGHT
- SAVE DATA
- SET PROFILE
- SET GLOBAL
- SHOW REPORT
- SHOW CHART
- SWITCH (lorsque l'aide en ligne est active)
- TOP
- TSO

Les commandes suivantes *ne sont pas* valides dans une mini-session :

- ADD
- CANCEL
- CHANGE
- CHECK
- CLEAR
- CONNECT
- CONVERT
- DELETE
- DESCRIBE
- DISPLAY (QUERY, PROC, PROFILE, FORM, ANALYTICS)
- DRAW
- EDIT
- ENLARGE
- ERASE
- EXIT
- EXPORT
- EXTRACT
- GETQMF
- IMPORT
- INSERT
- INTERACT
- LIST
- NEXT

- PREVIOUS
- PRINT (QUERY, PROC, PROFILE, FORM)
- REDUCE
- REFRESH
- RESET GLOBAL
- RESET (Query, Proc, Form)
- RUN
- SAVE
- SEARCH
- SHOW
- SORT
- SPECIFY
- START
- SWITCH
- TRACE

QMF renvoie un message d'erreur lorsque vous exécutez une commande CLIST ou une procédure qui émet une commande restreinte.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5
 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

«SET GLOBAL (syntaxe étendue)», à la page 53

Pour créer vos propres variables globales et les utiliser dans des commandes QMF en tant que variables de substitution, émettez la commande SET GLOBAL. Vous pouvez également utiliser la commande SET GLOBAL pour définir des valeurs pour les variables globales QMF prédéfinies, qui commencent par "DSQ".

Synonymes de commande

Vous pouvez créer des synonymes de commande, qui sont des commandes qui ressemblent aux commandes QMF et peuvent exécuter un large éventail de fonctions.

Les synonymes de commande vous offrent une grande souplesse et s'avèrent utiles pour les utilisateurs. Par exemple, les synonymes de commande peuvent exécuter la fonction d'une commande ou démarrer une application.

Chapitre 9. Exportation et importation d'objets

Vous pouvez écrire des applications qui exécutent des commandes QMF EXPORT et IMPORT pour placer des objets en dehors de l'environnement QMF.

Vos applications peuvent exporter des tables, ainsi que les objets QMF suivants :

- DATA
- QUERY
- PROC
- FORM
- REPORT
- CHART

Lorsque vous exportez des objets, à l'exception des rapports ou des objets au format CSV, QMF convertit l'objet dans un format externalisé. QMF stocke le format externalisé de l'objet dans un fichier UNIX (dans le cas de données ou de tables uniquement), un ensemble de données TSO ou une file d'attente de données CICS. Le format externalisé des objets QMF est un élément majeur du développement d'applications QMF.

La commande IMPORT lit le format externalisé et place l'objet dans la mémoire temporaire QMF ou dans la base de données. L'emplacement dépend du mode d'émission de la commande.

Opérations possibles avec un fichier UNIX, une file d'attente de données CICS ou un ensemble de données TSO exporté

Les commandes QMF EXPORT et IMPORT sur des objets données sont utiles dans plusieurs cas.

Par exemple, vous pouvez utiliser les commandes IMPORT et EXPORT pour atteindre ces objectifs :

- Fournir des résultats de requête à votre application

Utilisez la commande QMF EXPORT pour extraire des données de la base de données et les placer dans votre application.

- Créer des objets dans votre application et les utiliser dans QMF

Vous pouvez créer un objet en dehors de l'environnement QMF en utilisant le format approprié pour l'objet. Par exemple, dans le cas de données ou de tables, lorsque vous importez le fichier UNIX dans QMF, un nouvel objet QMF est créé. Un nouvel objet est également créé lorsque vous importez un ensemble de données TSO ou une file d'attente de données CICS qui contient l'objet.

Vous ne pouvez pas importer des rapports, des diagrammes ou des fichiers CSV dans QMF. Pour les rapports et les diagrammes, vous pouvez importer à la place les données et formats qui ont été utilisés pour les créer.

- Mettre des objets QMF à la disposition d'autres environnements ou produits

Avertissement : Faites attention lorsque vous transférez des objets exportés entre des systèmes disposant de CCSID ou d'ensembles de caractères différents (entre des systèmes EBCDIC et ASCII, par exemple) ou entre des environnements NLF différents. Ce type de transfert d'objets entre systèmes risque de les rendre inutilisables.

Si vous avez besoin d'importer une requête créée de façon interactive ou une requête QBE dans un autre programme que QMF, vous devez d'abord utiliser la commande CONVERT QUERY. Cette commande convertit la requête en requête SQL que vous pouvez exporter et utiliser dans d'autres produits.

Vous pouvez transférer des objets QMF entre QMF sous TSO, ISPF ou z/OS natif en mode de traitement par lots. Vous pouvez également transférer des objets QMF sous CICS à l'aide de files d'attente extrapartition de données transitoires CICS.

- Sauvegarder des objets et données en dehors de la base de données
Par exemple, au milieu d'un programme, vous pouvez exporter vos données afin qu'un programme externe puisse les manipuler.
- Créer des applications bilingues
Vous pouvez créer un format QMF dans votre langue dominante et le traduire en anglais en utilisant l'option LANGUAGE dans la commande EXPORT. Vous pouvez également utiliser l'option LANGUAGE dans les commandes IMPORT et EXPORT pour traduire un format en anglais dans votre langue dominante.

Exportation et sauvegarde de données

La différence entre les commandes EXPORT DATA et SAVE DATA réside dans l'emplacement et le mode de stockage de l'objet.

Cette différence dans le mode de stockage des objets affecte les opérations que vous pouvez effectuer sur les résultats :

- L'exportation d'un objet données génère un fichier UNIX, un ensemble de données TSO ou une file d'attente de données CICS. Vous pouvez lire, modifier ou imprimer chaque élément de façon séquentielle via des programmes d'application QMF ou d'autres applications externes.
- La commande SAVE DATA génère une table de base de données. Les actions qui utilisent les données sauvegardées doivent être effectuées par le biais de la base de données.

Exportation d'objets données et de tables de base de données

Vous pouvez exporter des objets données et table au format QMF, IXF (Integrated Exchange Format), XML ou CSV. Le format QMF est le format par défaut.

Lorsque vous exécutez une requête, QMF affiche le résultat dans un rapport. Les données brutes pour le rapport sont stockées dans une zone de stockage temporaire sous forme d'objet données. Les tables relationnelles et les vues qui sont stockées dans la base de données sont appelées objets table.

Vous pouvez exporter les objets données et table vers des zones de stockage externes à QMF. Les formats exportés d'une table dans une mémoire temporaire (DATA) sont identiques aux formats exportés d'une table stockée dans la base de données (TABLE). Un objet qui est exporté sous forme de données peut être importé sous forme de table et inversement, sauf si les données sont au format CSV.

Vous pouvez créer vos propres tables en dehors de QMF. Utilisez le format QMF, IXF ou XML et importez le contenu du fichier UNIX, de l'ensemble de données TSO ou de la file d'attente de données CICS qui contient la table. Incluez les zones requises et ajoutez vos propres données si nécessaire. Importez ensuite le fichier UNIX, l'ensemble de données TSO ou la file d'attente de données CICS dans QMF sous forme d'objet table.

Concepts associés:

«Règles et informations relatives à l'exportation et à l'importation d'objets données et de tables», à la page 101

QMF exporte des objets données et table vers la mémoire temporaire et définit des règles concernant l'allocation de cette mémoire. QMF dispose également de méthodes spécifiques de traitement des erreurs d'importation.

Exportation de données ou de tables au format QMF

Le fichier de données que vous exportez à l'aide de la commande EXPORT avec la clause DATAFORMAT=QMF se compose de deux parties : les enregistrements d'en-tête, qui décrivent les données figurant dans les enregistrements, et les enregistrements de données, qui contiennent les données.

Enregistrements d'en-tête

Les enregistrements d'en-tête décrivent les données exportées dans le fichier de données.

La longueur d'enregistrement d'un fichier de données externe est la longueur d'une ligne de données, conformément à la description figurant dans l'enregistrement de données. Les enregistrements d'en-tête qui précèdent les enregistrements de données sont de même longueur que les enregistrements de données. Si les informations d'en-tête dépassent la longueur de l'enregistrement de données, plusieurs enregistrements d'en-tête sont écrits.

Deux formats sont utilisés pour les enregistrements d'en-tête : l'un pour les noms de colonne abrégés et l'autre pour les noms de colonne longs. Les tableaux ci-dessous présentent les informations contenues dans chaque format des enregistrements d'en-tête.

Tableau 9. Informations relatives à l'enregistrement d'en-tête pour les noms de colonne abrégés

Position de l'octet	Informations et type
1-8	Niveau de format d'objet QMF (8 caractères de données) Ces positions d'octet indiquent REL 1.0 lorsque tous les noms de colonne sont des noms abrégés et que la variable globale DSQDC_SHORT_EXPT est définie sur 1.
9-10	Nombre d'enregistrements d'en-tête (entier signé demi-mot)
11-12	Nombre de colonnes de données (entier signé demi-mot)
13-30	Nom de colonne La largeur de colonne maximale est 18 octets.

Tableau 9. Informations relatives à l'enregistrement d'en-tête pour les noms de colonne abrégés (suite)

Position de l'octet	Informations et type
31-32	Type de données (entier signé demi-mot) Les codes de type de données sont affichés dans le tableau 11, à la page 75. Cette zone stocke l'équivalent hexadécimal des codes décimaux affichés dans le tableau pour chaque type de données.
33-34	Largeur de colonne (entier signé demi-mot) ; pour la plupart des types de données, cette largeur représente la largeur de la colonne en octets, avec les exceptions suivantes : <ul style="list-style-type: none"> • Dans les colonnes DECIMAL, le premier octet du demi-mot représente la précision, tandis que le second représente l'échelle. • Dans les colonnes GRAPHIC et VARGRAPHIC, cette valeur reflète la largeur des caractères double octet. • Dans les colonnes FLOAT, cette valeur est soit 4, qui indique un nombre à virgule flottante à simple précision, soit 8, qui indique un nombre à virgule flottante à double précision. • Dans les colonnes DECFLOAT, cette valeur est 8 pour le format long et 16 pour le format étendu.
35	Les caractères NULL sont autorisés : Y si les caractères NULL sont autorisés et N s'ils ne le sont pas (1 caractère de données)
36	Octet inutilisé

Le bloc qui est décrit par les octets 13 à 36 se répète pendant le nombre de colonnes qui figurent dans les données.

Tableau 10. Informations relatives à l'enregistrement d'en-tête pour les noms de colonne longs

Position de l'octet	Informations et type
1-8	Niveau de format d'objet QMF (8 caractères de données) Ces positions d'octet indiquent REL 3.0 lorsque la variable globale DSQDC_SHORT_EXPT est définie sur 0.
9-10	Nombre d'enregistrements d'en-tête (entier signé demi-mot)
11-12	Nombre de colonnes de données (entier signé demi-mot)
13-42	Nom de colonne La longueur maximale par défaut pour le nom est 30 octets. Toutefois, vous pouvez utiliser la variable globale DSQDC_SHORT_EXPT pour définir une largeur de colonne maximale de 18 octets avant d'exporter les données. Dans ce cas, le format d'enregistrement d'en-tête pour les noms de colonne abrégés est utilisé.
43-44	Type de données (entier signé demi-mot) Les codes de type de données sont affichés dans le tableau 11, à la page 75. Cette zone stocke l'équivalent hexadécimal des codes décimaux affichés dans le tableau pour chaque type de données.

Tableau 10. Informations relatives à l'enregistrement d'en-tête pour les noms de colonne longs (suite)

Position de l'octet	Informations et type
45-46	<p>Largeur de colonne (entier signé demi-mot) ; pour la plupart des types de données, cette largeur représente la largeur de la colonne en octets, avec les exceptions suivantes :</p> <ul style="list-style-type: none"> • Dans les colonnes DECIMAL, le premier octet du demi-mot représente la précision, tandis que le second représente l'échelle. • Dans les colonnes GRAPHIC et VARGRAPHIC, cette valeur reflète la largeur des caractères double octet. • Dans les colonnes FLOAT, cette valeur est soit 4, qui indique un nombre à virgule flottante à simple précision, soit 8, qui indique un nombre à virgule flottante à double précision. • Dans les colonnes DECFLOAT, cette valeur est 8 pour le format long et 16 pour le format étendu. <p>La largeur de colonne maximale par défaut est 30 octets. Toutefois, vous pouvez utiliser la variable globale DSQDC_SHORT_EXPT pour définir une largeur de colonne maximale de 18 octets avant d'exporter les données. Dans ce cas, le format d'enregistrement d'en-tête pour les noms de colonne abrégés est utilisé.</p>
47	Les caractères NULL sont autorisés : Y si les caractères NULL sont autorisés et N s'ils ne le sont pas (1 caractère de données)
48	Octet inutilisé

Le bloc qui est décrit par les octets 13 à 48 se répète pendant le nombre de colonnes qui figurent dans les données.

Les codes de type de données s'affichent dans ce tableau. Les codes hexadécimaux figurant dans la première colonne sont utilisés pour indiquer chaque type de données affiché dans la troisième colonne.

Tableau 11. Codes de type de données

Code en notation hexadécimale	Code en notation décimale	Type de données	Signification
X'180'	384	DATE	Date
X'184'	388	TIME	Heure
X'188'	392	TIMESTAMP	Horodatage
X'1C0'	448	VARCHAR	Caractère de longueur variable
X'1C4'	452	CHAR	Caractère de longueur fixe
X'1D0'	464	VARGRAPHIC	Graphique de longueur variable
X'1D4'	468	GRAPHIC	Graphique de longueur fixe
X'1E0'	480	FLOAT	Virgule flottante
X'1E4'	484	DECIMAL	Décimal
X'1EC'	492	BIGINT	Entier long
X'1F0'	496	INTEGER	Entier

Tableau 11. Codes de type de données (suite)

Code en notation hexadécimale	Code en notation décimale	Type de données	Signification
X'1F4'	500	SMALLINT	Chiffre entier
X'38C'	908	VARBINARY	Binaire de longueur variable
X'390'	912	BINARY	Binaire de longueur fixe
X'3E4'	996	DECFLOAT(16) et DECFLOAT(34)	Virgule flottante décimale de format long et virgule flottante décimale de format étendu
X'990'	2448	TIMESTAMP WITH TIME ZONE	Horodatage avec fuseau horaire

Les colonnes contenant les types de données DATE, TIME, TIMESTAMP ou TIMESTAMP WITH TIME ZONE sont toujours exportées au format ISO.

Enregistrements de données

Les enregistrements de données sont au format bloc fixe (FB) et contiennent les données à exporter.

La longueur maximale autorisée pour un enregistrement de données est 7 000 octets. La longueur d'un enregistrement de données est la somme des largeurs des types de données inclus dans l'enregistrement. Utilisez le tableau suivant pour calculer les largeurs de chaque type de données.

Tableau 12. Largeurs des enregistrements de données. Calculez la largeur d'un enregistrement de données spécifique en ajoutant le nombre d'octets figurant dans chaque colonne.

Type de données	Indicateur de type NULL	Zone Longueur	SO/SI	Données
Caractère 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			Longueur dans l'en-tête (LIH)
Date 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			LIH
Virgule flottante 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			8

Tableau 12. Largeurs des enregistrements de données (suite). Calculez la largeur d'un enregistrement de données spécifique en ajoutant le nombre d'octets figurant dans chaque colonne.

Type de données	Indicateur de type NULL	Zone Longueur	SO/SI	Données
Virgule décimale flottante	2			8 pour les valeurs de format long ; 16 pour les valeurs de format étendu
Entier long 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			LIH
Entier 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			LIH
Petit entier 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			LIH
Heure 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			LIH
Horodatage 8 pour les valeurs de format long ; 16 pour les valeurs de format étendu	2			LIH
Horodatage avec fuseau horaire	2	2		LIH
Décimal	2			(Précision + 2) // 2
Graphique	2		2	(LIH × 2)
Caractère variable	2	2		LIH
Graphique variable	2	2	2	(2 × LIH)
Binaire	2			LIH
Binaire variable	2	2		LIH

Important : La longueur dans l'en-tête (LIH) est la largeur donnée pour cette colonne dans l'enregistrement d'en-tête.

Chaque enregistrement de données comporte 2 octets d'informations d'indicateur, qui peuvent avoir les valeurs et significations indiquées dans ce tableau :

Tableau 13. Valeurs hexadécimales indiquant la validité des enregistrements de données

Value	Signification
X'0000'	La colonne contient des données valides.
X'FFFF' ou X'FFFE'	La colonne contient une valeur NULL ; les données de la colonne sont sans importance.

Interprétation d'un objet données au format QMF

Vous pouvez calculer la longueur de l'enregistrement d'en-tête lorsque vous disposez de la longueur des enregistrements de données.

Par exemple, supposons que vous exportiez les données suivantes à partir de la table Q.PERS :

ID	NOM	COMM
10	SANDERS	-
20	PERNAL	612,45

Dans cet exemple, la longueur de chaque enregistrement de données est 23 octets. Le tableau 10, à la page 74 montre que les 12 premiers octets contiennent des informations de niveau et de numéro.

Calculez les largeurs de chaque colonne comme indiqué dans ce tableau :

Tableau 14. Calcul des largeurs de colonne

Nom de colonne	Type de données	Largeur de la colonne (longueur de l'en-tête)	Largeur de la colonne
ID	SMALLINT	2	2 + 2 = 4
NOM	VARCHAR	9	2 + 2 + 9 = 13
COMM	DECIMAL (7,2)	7	(7 + 1)/2 + 2 = 6
		Longueur de l'enregistrement de données :	23

Il y a 24 octets pour chaque colonne de données et 3 colonnes. Pour cet objet données à trois colonnes, la longueur de l'en-tête est donc 84 octets :

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

Chaque enregistrement d'en-tête est de la même longueur que les enregistrements de données, à savoir 23 octets. Les 84 octets sont répartis sur quatre enregistrements d'en-tête de 23 octets ; le dernier enregistrement est rempli à l'aide de blancs.

Cet exemple d'en-tête présente l'en-tête provenant du rapport et sa représentation hexadécimale. 40 est le code hexadécimal pour un caractère blanc. Les numéros de caractères noir au blanc sont associés aux remarques qui suivent l'exemple.

```

      R E L      3 . 0           I D
1  D9 C5 D3 40 F1 4B F0 40 0004 0003 C9 C4 40 40 40 40 40 40 40 40
      2 3 4
      N O M
2  40 40 40 40 40 40 40 01F4 0002 D5 00 D5 C1 D4 C5 40 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 40
      Y
4  40 40 40 40 40 40 40 40 40 01E4 07 02 E8 00 40 40 40 40 40 40

```

Figure 13. Exemple d'enregistrements d'en-tête pour un objet données exporté au format QMF. 40 est le code hexadécimal pour un caractère blanc.

L'exemple suivant présente les données provenant du rapport et leur représentation hexadécimale. Pour plus d'informations sur la signification des positions d'octet, consultez le tableau 10, à la page 74.

```

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

```

Figure 14. Exemple d'enregistrements de données pour un objet données exporté au format QMF

1 REL 3.0

Niveau de format d'objet : 3.0

Le niveau de format d'objet indique à QMF quelle version du format d'objet est utilisée par cet objet. Chaque fois qu'un format d'objet QMF est modifié, le numéro du niveau l'est également ; les formats d'objet ne sont pas modifiés à chaque nouvelle édition.

2 X'0004'

Nombre d'enregistrements d'en-tête : 4

3 X'0003'

Nombre de colonnes de données : 3

4 X'C9 C4'

Nom de colonne : ID

5 X'1F4'

Type de données : SMALLINT

6 X'0002'

Largeur de colonne : 2

7 X'D5'

Valeurs NULL autorisées : N signifie non

8 X'0A'

Valeur pour la première colonne du premier enregistrement de données : 10

9 X'07'

Longueur du nom dans la deuxième colonne du premier enregistrement de données : 7

10 X'FFFF'

Informations relatives à l'indicateur : la colonne contient une valeur NULL

Exportation de données ou de tables au format IXF

Lorsque vous utilisez la commande EXPORT pour exporter un objet données ou une table avec l'option DATAFORMAT=IXF, les données sont exportées au format IXF (Integrated Exchange Format). QMF prend en charge un sous-ensemble d'IXF.

L'ensemble de données TSO ou la file d'attente de données CICS qui contient la table ou les données exportées se compose des enregistrements suivants :

- Enregistrement d'en-tête (H)
- Enregistrement de table (T)
- Enregistrements de colonne (C)
- Enregistrements de données (D)

L'ensemble de données ou file d'attente CICS exporté est composé d'un enregistrement H, suivi d'un enregistrement T. L'enregistrement T contient un comptage du nombre d'enregistrements C qui suivent l'enregistrement T. Il existe un enregistrement C pour chaque colonne de la table. Les enregistrements D suivent les enregistrements C. Il existe un enregistrement D pour chaque ligne de la table. La disposition des enregistrements s'affiche dans le graphique suivant :

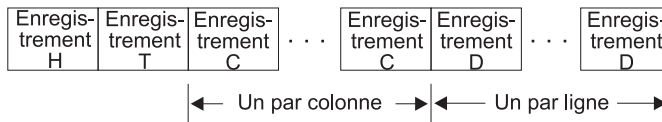


Figure 15. Disposition des enregistrements dans un ensemble de données ou file d'attente CICS exporté (format IXF)

Enregistrement d'en-tête (H)

Un enregistrement d'en-tête (qui est obligatoire) au format IXF est le premier enregistrement de l'ensemble de données ou file d'attente CICS.

L'enregistrement d'en-tête est un enregistrement de 42 octets qui contient des données de type caractères. Le format de l'enregistrement H est présenté dans le tableau suivant :

Tableau 15. Composants d'un enregistrement d'en-tête dans un ensemble de données ou une file d'attente de données IXF contenant une table de base de données ou un objet données exporté

Position de l'octet	Informations et type
01	Indicateur d'enregistrement d'en-tête (H)
02-04	Identificateur d'ensemble de données TSO ou de file d'attente de données CICS

Tableau 15. Composants d'un enregistrement d'en-tête dans un ensemble de données ou une file d'attente de données IXF contenant une table de base de données ou un objet données exporté (suite)

Position de l'octet	Informations et type
05-08	Version IXF ; il peut s'agir de l'un des types suivants : <ul style="list-style-type: none"> • 0000, qui prend en charge les données ou tables contenant des noms de colonne abrégés (18 caractères au maximum) et des colonnes <code>TIMESTAMP WITH TIME ZONE</code> • 0001, qui prend en charge les données ou tables contenant au moins un nom de colonne long (19 caractères au minimum) et des colonnes <code>TIMESTAMP WITH TIME ZONE</code> • 0002, qui prend en charge les données ou tables contenant des noms de colonne abrégés (18 caractères au maximum) et au moins une colonne <code>TIMESTAMP WITH TIME ZONE</code> • 0003, qui prend en charge les données ou tables contenant au moins un nom de colonne long (19 caractères au minimum) et au moins une colonne <code>TIMESTAMP WITH TIME ZONE</code>
09-14	Nom de produit d'origine (QMF)
15-20	Niveau d'édition du produit d'origine (VBR2M0)
21-28	Date de création de l'ensemble de données ou file d'attente CICS, au format <code>AAAAMMJJ</code>
29-34	Heure de création de l'ensemble de données ou file d'attente CICS, au format <code>HHMMSS</code>
35-39	Nombre d'enregistrements précédant le premier enregistrement D (données) dans l'ensemble de données ou file d'attente CICS ; cette valeur est une valeur numérique à 5 chiffres exprimée au format caractère
40	Indicateur DBCS indiquant si les données DBCS sont autorisées ; Y ou N
41-42	Blancs

Enregistrement de table (T)

Un enregistrement de table au format IXF suit l'enregistrement d'en-tête. Chaque ensemble de données ou file d'attente de données contenant un objet au format IXF doit comporter un enregistrement T. Un enregistrement de table contient des informations de table et de données relatives à l'objet qui a été exporté à l'aide de la commande `EXPORT TABLE` ou `EXPORT DATA`.

Les noms de tables qui sont exportés au format IXF sont tronqués à 18 caractères et les noms de propriétaire à 8. Si vous exécutez une requête et que vous exportez l'objet `DATA` obtenu, l'enregistrement de table contient un nom et un propriétaire vides.

Le format d'un enregistrement T est présenté dans le tableau suivant :

Tableau 16. Composants d'un enregistrement de table dans un ensemble de données ou une file d'attente de données IXF contenant une table de base de données ou un objet données exporté

Position de l'octet	Informations et type
01	Indicateur d'enregistrement de table (T)
02-03	Longueur du nom de données (18)
04-21	Nom de la table à partir de laquelle les données sont extraites ; il est aligné à gauche et rempli à l'aide de blancs de fin La totalité de la zone de 18 octets est vide si la table ne porte pas de nom.
22-29	Qualificatif de nom de données ; nom du propriétaire de la table de base de données à partir de laquelle les données sont extraites La zone de 8 octets est vide si la table n'a pas de propriétaire.
30-41	Source de données (base de données)
42	Convention utilisée pour décrire les données : C pour les données de colonne
43	Format de données : C pour caractère (OUTPUTMODE=CHARACTER) ; M pour machine (OUTPUTMODE=BINARY)
44	Emplacement des données : I pour interne
45-49	Nombre d'enregistrements de colonne (C) : valeur numérique au format caractère qui indique le nombre d'enregistrements C avant le premier enregistrement de données (D)
50-51	Réservé
52-81	Blancs

Enregistrements de colonne (C)

Un enregistrement de colonne au format IXF décrit les caractéristiques de données de la colonne. Il existe un enregistrement pour chaque colonne de la table.

Lorsqu'un nom de colonne contient plus de 18 caractères, la zone de nom de colonne dans l'enregistrement de colonne doit être augmentée pour comprendre de 18 à 30 caractères. Le numéro de version IXF qui est utilisé dans l'enregistrement d'en-tête dépend non seulement du fait qu'un nom de colonne contient plus de 18 caractères, mais également que les données contiennent une colonne `TIMESTAMP WITH TIME ZONE`. Le tableau 15, à la page 80 présente les numéros de version IXF utilisés dans chaque cas.

Le tableau suivant indique le format d'un enregistrement de colonne pour les données ou tables contenant des colonnes `TIMESTAMP WITH TIME ZONE` (numéros de version IXF 0000 ou 0001). Pour plus d'informations sur les données ou tables contenant une ou plusieurs colonnes `TIMESTAMP WITH TIME ZONE` (numéros de version IXF 0002 ou 0003), voir le tableau 18, à la page 84.

Tableau 17. Format IXF sans colonnes *TIMESTAMP WITH TIME ZONE* (numéros de version IXF 0000 ou 0001)

Position de l'octet	Informations et type
01	Indicateur d'enregistrement de colonne (C)
02-03	Longueur du nom de colonne
04-21	Nom de colonne, obtenu à partir de la base de données ou généré par QMF (lorsque la colonne ne portait pas de nom à l'origine) Le nom est aligné à gauche et rempli à l'aide de blancs à droite, si nécessaire.
22 ou 34	Indicateur précisant si les valeurs NULL sont autorisées ; Y ou N
23 ou 35	Indicateur de sélection de colonne (Y)
24 ou 36	Indicateur de colonne clé (Y)
25 ou 37	Classe de données (R)
26-28 ou 38-40	Type de données (pour les codes de type de données, voir le tableau 20, à la page 86)
29-33 ou 41-45	Page de codes
34-38 ou 46-50	Réservé
39-43 ou 51-55	Longueur des données de colonne ; valeur décimale au format caractère Si le type de données est DECIMAL, les 3 premiers octets représentent la précision des données, tandis que les 2 octets suivants représentent l'échelle. Si le type de données est BIGINT, INTEGER ou SMALLINT, cette zone est vide car la longueur est inhérente au type de données.
44-49 ou 56-61	Position de départ des données de colonne ; valeur décimale au format caractère Cette valeur reflète le décalage des données pour une colonne à partir du début de l'enregistrement de données. Si la colonne autorise les valeurs NULL, cette zone pointe vers l'indicateur de type NULL. Dans le cas contraire, elle pointe vers les données elles-mêmes. L'espace pour l'indicateur de type NULL est toujours présent dans l'enregistrement, que la colonne autorise les valeurs NULL ou non. La position de départ se fonde sur le premier octet contenant des données. Les cinq premiers octets de l'enregistrement de données (D) ne doivent donc pas être pris en compte pour la position de départ des données réelles. (La première position de données est la position 1 et non la position 0.)
50-79 ou 62-91	Informations de libellé de colonne, le cas échéant (si ces informations ne sont pas disponibles, ces positions d'octet contiennent des blancs)
80-81 ou 92-93	Deux octets de zéros au format caractère (00)

Si les données ou la table contiennent une ou plusieurs colonnes *TIMESTAMP WITH TIME ZONE*, le format de l'enregistrement de colonne est le suivant :

Tableau 18. Format IXF avec une ou plusieurs colonnes *TIMESTAMP WITH TIME ZONE* (numéros de version IXF 0002 ou 0003)

Position de l'octet	Informations et type
01	Indicateur d'enregistrement de colonne (C)
02-03	Longueur du nom de colonne
04-21	Nom de colonne, obtenu à partir de la base de données ou généré par QMF (lorsque la colonne ne portait pas de nom à l'origine) Le nom est aligné à gauche et rempli à l'aide de blancs à droite, si nécessaire.
22 ou 34	Indicateur précisant si les valeurs NULL sont autorisées ; Y ou N
23 ou 35	Indicateur de sélection de colonne (Y)
24 ou 36	Indicateur de colonne clé (Y)
25 ou 37	Classe de données (R)
26-29 ou 38-41	Type de données (voir le tableau 20, à la page 86 pour les codes de type de données)
30-34 ou 42-46	Page de codes
34-38 ou 47-50	Réservé
39-43 ou 51-55	Longueur des données de colonne ; valeur décimale au format caractère Si le type de données est DECIMAL, les 3 premiers octets représentent la précision des données, tandis que les 2 octets suivants représentent l'échelle. Si le type de données est BIGINT, INTEGER ou SMALLINT, cette zone est vide car la longueur est inhérente au type de données.
44-49 ou 56-61	Position de départ des données de colonne ; valeur décimale au format caractère Cette valeur reflète le décalage des données pour une colonne à partir du début de l'enregistrement de données. Si la colonne autorise les valeurs NULL, cette zone pointe vers l'indicateur de type NULL. Dans le cas contraire, elle pointe vers les données elles-mêmes. L'espace pour l'indicateur de type NULL est toujours présent dans l'enregistrement, que la colonne autorise les valeurs NULL ou non. La position de départ se fonde sur le premier octet contenant des données. Les cinq premiers octets de l'enregistrement de données (D) ne doivent donc pas être pris en compte pour la position de départ des données réelles. (La première position de données est la position 1 et non la position 0.)
50-79 ou 62-91	Informations de libellé de colonne, le cas échéant (si ces informations ne sont pas disponibles, ces positions d'octet contiennent des blancs)
80-81 ou 92-93	Deux octets de zéros au format caractère (00)

Enregistrements de données (D)

Les enregistrements de données au format IXF sont au format bloc variable (VB). Il existe un enregistrement de données pour chaque ligne de la table.

Le tableau ci-dessous présente le format d'un enregistrement de données :

Tableau 19. Format d'un enregistrement de données dans un ensemble de données ou une file d'attente de données IXF contenant une table ou un objet données exporté

Position de l'octet	Informations et type
01	Indicateur d'enregistrement de données (D)
02-04	Réservé
05	Blanc
De 06 à la fin de l'enregistrement	Données de ligne au format binaire ou caractère, selon que l'octet 43 de l'enregistrement de table est M (machine) ou C (caractère) L'octet 6 représente le début (position 1) des données de ligne pour la première colonne.

Format des données de colonne

Les données des enregistrements D pour n colonnes sont placées côte à côte, comme indiqué dans cette figure.

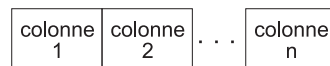


Figure 16. Format des données de colonne dans les enregistrements D

Pour chaque colonne, les données sont composées d'un indicateur de type NULL suivi des données elles-mêmes. Si les valeurs NULL sont autorisées (l'octet 22 ou 34 de l'enregistrement C a la valeur Y), les octets 44 à 49 ou 56 à 61 de chaque enregistrement C pointent vers l'indicateur de type NULL qui précède les données pour cette colonne. Si les valeurs NULL ne sont pas autorisées (l'octet 22 ou 34 de l'enregistrement C a la valeur N), les octets 44 à 49 ou 56 à 61 pointent vers les données elles-mêmes. Toutefois, dans ce dernier cas, l'espace pour l'indicateur de type NULL est conservé dans l'enregistrement de données. La première position dans les octets 44 à 49 ou 56 à 61 est représentée par la valeur 1, qui pointe vers l'octet 6 d'un enregistrement D (les octets 1 à 5 sont ignorés).

La représentation de l'indicateur de type NULL dépend des informations spécifiées pour le paramètre OUTPUTMODE : caractère ou binaire. La valeur du paramètre OUTPUTMODE est reflétée dans l'octet 43 de l'enregistrement T : C pour caractère ou M pour machine (binaire). Lorsque le format de données est caractère, 1 octet est utilisé pour l'indicateur de type NULL.

- Un tiret (-) indique que les données sont de type NULL
- Un blanc indique que les données ne sont pas de type NULL

Lorsque le format de données est binaire, 2 octets sont utilisés pour l'indicateur de type NULL :

- X'FFFF' indique que les données sont de type NULL
- X'0000' indique que les données ne sont pas de type NULL

Pour plus d'informations sur l'indicateur de type NULL, consultez les exemples ci-dessous qui vous aident à interpréter un objet au format IXF.

Format des données de colonne par type de données

La longueur et le format des données de colonne IXF dans les enregistrements D peuvent varier en fonction de la valeur du paramètre OUTPUTMODE.

Le tableau ci-dessous indique la longueur et le format des données de colonne dans les enregistrements D pour chaque type de données aux formats d'exportation caractère et binaire. Dans ce tableau, IXFLENG fait référence au contenu des octets 39 à 43 ou 51 à 55 d'un enregistrement C (longueur des données de colonne).

Tableau 20. Format des données de colonne IXF par type de données

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
384	DATE	<p>La valeur figurant dans IXFLENG n'est pas significative pour ce type de données. La longueur (10 octets) est inhérente au type de données.</p> <p>Le format est le suivant :</p> <p style="text-align: center;"><i>yyyy-mm-dd</i></p> <p>où <i>aaaa</i> représente l'année, <i>mm</i> le mois et <i>jj</i> le jour. <i>aaaa</i>, <i>mm</i> et <i>jj</i> doivent être des caractères numériques. Les zéros en tête ne peuvent pas être omis. La plage autorisée pour <i>aaaa</i> est 0001 à 9999 et 01 à 12 pour <i>mm</i>. La plage pour <i>jj</i> dépend du mois. Par exemple, la valeur suivante spécifie la date du 28 février 2002 :</p> <p>2002-02-28</p>	Identique au format caractère
388	TIME	<p>La valeur figurant dans IXFLENG n'est pas significative pour ce type de données. La longueur (8 octets) est inhérente au type de données.</p> <p>Le format est le suivant :</p> <p style="text-align: center;"><i>hh.mm.ss</i></p> <p>où <i>hh</i> représente l'heure au format 24 heures, <i>mm</i> les minutes et <i>ss</i> les secondes. <i>hh</i>, <i>mm</i> et <i>ss</i> doivent tous être des caractères numériques. Les zéros en tête ne peuvent pas être omis. Les plages autorisées sont les suivantes :</p> <ul style="list-style-type: none"> • 00 à 23 pour <i>hh</i> • 00 à 59 pour <i>mm</i> • 00 à 59 pour <i>ss</i> <p>La valeur spéciale 24.00.00 pour minuit est valide. Exemples :</p> <p>10.37.42 correspond à 10 heures 37 minutes 42 secondes du matin 08.00.00 correspond à 8 heures du matin précises 23.30.00 correspond à 23 heures 30</p>	Identique au format caractère

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
392	TIMESTAMP	<p>La longueur pour TIMESTAMP(0) est 19 ; la longueur pour TIMESTAMP(<i>n</i>) est 20+<i>n</i>, où <i>n</i> représente un nombre compris entre 1 et 12. Par exemple, une colonne définie en tant que TIMESTAMP(12) a une longueur de 32.</p> <p>Le format est le suivant :</p> <p><i>aaaa-mm-jj-hh.mm.ss.nnnnnnnnnnn</i></p> <p>où <i>aaaa</i> représente l'année, <i>mm</i> le mois, <i>jj</i> le jour, <i>hh</i> l'heure au format 24 heures, <i>mm</i> les minutes, <i>ss</i> les secondes et <i>nnnnnnnnnnnn</i> les fractions de seconde. Les plages valides pour l'année, le mois, le jour, l'heure, les minutes et les secondes sont les mêmes que pour les types de données DATE et TIME.</p> <p>Exemples :</p> <p>2010-12-31-23.59.59.999999999999 (dernière fraction de seconde de 2010)</p> <p>2010-01-01-00.00.00.000000000001 (première fraction de seconde de 2010)</p> <p>24.00.00.000000000000 est valide pour la partie heure d'un horodatage.</p>	Identique au format caractère

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
448	VARCHAR LONG VARCHAR	<p>IXFLENG est la longueur maximale de la chaîne de caractères. La longueur de données se compose de n octets indiqués par IXFLENG et précédés d'une zone spécifiant le nombre de caractères à 5 octets. (La plage autorisée pour n est comprise entre 0 et 32 704 et 0 et n pour la zone de nombre.) Le nombre de caractères indiqué par la zone de nombre est valide ; le reste est sans importance. Par exemple, si IXFLENG=00010, les données sont au format suivant :</p> <p>00005JONESxxxx</p> <p>Dans ce format, chaque x représente un caractère blanc (X'40').</p>	<p>IXFLENG est la longueur maximale de la chaîne de caractères. La longueur de données se compose de n octets indiqués par IXFLENG et précédés d'une zone spécifiant le nombre de caractères binaires à 2 octets. (La plage autorisée pour n est comprise entre 1 et 32 704 et 0 et n pour la zone de nombre.) Le nombre de caractères indiqué par la zone de nombre est valide ; le reste est sans importance. Par exemple, si IXFLENG=00010, le format de données est le suivant :</p> <p>nnJONESxxxx</p> <p>Dans ce format, $nm=X'0005'$ et chaque x représente un caractère blanc (X'40').</p>
452	CHAR	<p>IXFLENG est la longueur de la chaîne de caractères. La longueur de données est indiquée par n octets de IXFLENG. (La plage autorisée pour n est comprise entre 1 et 254). Par exemple, si IXFLENG=00005, les données sont au format suivant :</p> <p>JONES</p> <p>Dans ce format, JONES est la chaîne de caractères à 5 octets vers laquelle pointent les octets 44 à 49 ou 56 à 61 de l'enregistrement C.</p>	Identique au format caractère

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
464	VARGRAPHIC LONG VARGRAPHIC	<p>IXFLENG est le nombre maximal de caractères à double octet ($2 \times n$ octets). La longueur de données est composée d'une zone de nombre de caractères à 5 octets, plus deux fois le nombre d'octets indiqué par IXFLENG, plus 2 (pour les caractères de changement de mode). Le nombre de caractères à double octet dans la zone de nombre est le nombre de caractères valide plus un caractère hors code (X'0E') précédant immédiatement les données et un caractère en code (X'0F') suivant immédiatement les données. Le reste est sans importance. (La plage autorisée pour n est comprise entre 1 et 16 352 et 0 et n pour la zone de nombre.) Par exemple, si IXFLENG=00006, les données sont au format suivant :</p> <p>00003oZZYXXixxxxxx</p> <p>Dans ce format, la lettre <i>o</i> représente le caractère hors code, <i>i</i> le caractère en code et chaque <i>x</i> un caractère blanc (X'40').</p>	<p>La longueur de données est composée d'une zone de nombre de caractères binaires à double octet suivie de deux fois le nombre d'octets indiqué par IXFLENG. (La plage autorisée pour IXFLENG est comprise entre 1 et 16 352 et 0 et IXFLENG pour la zone de nombre.) Le nombre de caractères à double octet dans la zone de nombre est valide. Ils ne sont pas entourés de caractères hors code et en code. Le reste est sans importance. Par exemple, si IXFLENG=00008, les données sont au format suivant :</p> <p>nnZZYXXWwxxxxxxxxx</p> <p>Dans ce format, $nn=X'0004'$ et chaque <i>x</i> représente un caractère blanc (X'40').</p>
468	GRAPHIC	<p>IXFLENG est le nombre de caractères à double octet ($2 \times n$ octets). La longueur de données est $2 \times n$ octets plus un caractère hors code (X'0E') précédant immédiatement les données et un caractère en code (X'0F') suivant immédiatement les données. Par exemple, si IXFLENG=00005, les données sont au format suivant :</p> <p>oZZYXXWwVVi</p> <p>Dans ce format, la lettre <i>o</i> représente le caractère hors code et <i>i</i> le caractère en code.</p>	<p>Identique au format caractère à l'exception du fait qu'il n'existe pas de caractères en code et hors code entourant la chaîne de données.</p> <p>Par exemple, si IXFLENG=00005, le format de données est le suivant :</p> <p>ZZYXXWwVVi</p>

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
480	FLOAT	<p>La valeur figurant dans IXFLENG est 4 pour les valeurs simple précision et 8 pour les valeurs double précision. La longueur et le format des données sont déterminés par la précision de la valeur.</p> <p>Les valeurs simple précision se composent de 14 caractères au maximum, au format suivant :</p> <p><i>sn.nnnnnnnEsee</i></p> <p>Le format est le suivant :</p> <ul style="list-style-type: none"> • <i>s</i> est un signe facultatif (plus, moins ou, dans le cas de la première position, un blanc si le nombre est positif). • <i>n</i> représente les chiffres du significande, qui peut en comprendre 8. Un séparateur décimal doit être placé en deuxième position dans le significande. • <i>E</i> correspond au début de l'exposant. • <i>e</i> représente les chiffres de l'exposant, qui peut en comprendre un ou deux. <p>La valeur doit se trouver dans la plage de +/-5.4E-79 à +/-7.2E+75.</p> <p>Exemples :</p> <p>-1.2345679E+07 6.2345679E-01 0.0000000E+00</p> <p>Les valeurs double précision se composent de 23 caractères au maximum, au format suivant :</p> <p><i>sn.nnnnnnnnnnnnnnnEsee</i></p> <p>Le format est le suivant :</p> <ul style="list-style-type: none"> • <i>s</i> est un signe facultatif (plus, moins ou, dans le cas de la première position, un blanc si le nombre est positif). • <i>n</i> représente les chiffres du significande, qui peut en comprendre 18. Un séparateur décimal doit être placé en deuxième position dans le significande. • <i>E</i> correspond au début de l'exposant. • <i>e</i> représente les chiffres de l'exposant, qui peut en comprendre un ou deux. <p>La valeur doit se trouver dans la plage de +/-5.4E-79 à +/-7.2E+75.</p> <p>Exemples :</p> <p>-1.2345678901234568E+14 6.23456789012345678E-01 0.0000000000000000E+00</p>	<p>La valeur figurant dans IXFLENG est 4 pour les valeurs simple précision et 8 pour les valeurs double précision. Les données sont composées d'une valeur en virgule flottante à 4 octets pour la simple précision et d'une valeur en virgule flottante à 8 octets pour la double précision.</p>

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
484	DECIMAL	<p>Les octets 39 à 43 ou 51 à 55 de l'enregistrement C représentent la précision ou P (3 premiers octets), et l'échelle ou S (2 octets suivants), du nombre. La plage autorisée pour P est comprise entre 0 et 15. S peut être toute valeur inférieure ou égale à P.</p> <p>Les données sont formatées en tant que valeur P+caractère à double octet (ou P+1 octet si S=0), alignées à droite, le premier octet étant réservé pour un signe, et un séparateur décimal (dont la position est déduite par S) présent uniquement si S est différent de zéro. Par exemple, si P=005 et S=00, les données sont au format suivant :</p> <p>12345</p> <p>Si P=006 et S=02, les données sont au format suivant :</p> <p>+2345.10</p> <p>Si P=004 et S=03, les données sont au format suivant :</p> <p>-8.515</p>	<p>Les octets 39 à 43 ou 51 à 55 de l'enregistrement C représentent la précision ou P (3 premiers octets), et l'échelle ou S (2 octets suivants), du nombre. La plage autorisée pour P est comprise entre 0 et 15. S peut être toute valeur inférieure ou égale à P.</p> <p>Les données sont composées d'une valeur (P+2)/décimale à double octet au format en décimal condensé. Le dernier octet indique le signe de la valeur. Par exemple, si P=005 et S=00, le format de données est le suivant :</p> <p>X'12345C'</p> <p>Si P=006 et S=02, le format de données est le suivant :</p> <p>X'0234510D'</p>
492	BIGINT	<p>La valeur figurant dans IXFLENG n'est pas significative pour ce type de données. La longueur et le format des données sont inhérents au type de données. Les données consistent en une valeur de caractère à 20 octets, alignée à droite, le premier caractère étant réservé pour un signe.</p> <p>Exemples :</p> <p>00000000000000000033 +9223372036854775807 -9223372036854775808</p>	<p>La valeur figurant dans IXFLENG n'est pas significative. La longueur et le format des données sont inhérents au type de données. Les données consistent en une valeur binaire à 8 octets.</p>

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
496	Entier	<p>La valeur figurant dans IXFLENG n'est pas significative pour ce type de données. La longueur et le format des données sont inhérents au type de données.</p> <p>Les données consistent en une valeur de caractère à 11 octets, alignée à droite, le premier caractère étant réservé pour un signe. Exemples :</p> <pre>0000000013 +1187642200 -0033588727</pre>	<p>La valeur figurant dans IXFLENG n'est pas significative. La longueur et le format des données sont inhérents au type de données.</p> <p>Les données consistent en une valeur binaire à 4 octets.</p>
500	SMALLINT	<p>La valeur figurant dans IXFLENG n'est pas significative pour ce type de données. La longueur et le format des données sont inhérents au type de données.</p> <p>Les données consistent en une valeur de caractère à 6 octets, alignée à droite, le premier caractère étant réservé pour un signe. Exemples :</p> <pre>00023 +00763 -21311</pre>	<p>La valeur figurant dans IXFLENG n'est pas significative. La longueur et le format des données sont inhérents au type de données.</p> <p>Les données consistent en une valeur binaire à double octet.</p>
908	VARBINARY	Sans objet	<p>Identique à VARCHAR, à l'exception du fait que :</p> <ul style="list-style-type: none"> IXFLENG représente la longueur maximale (nombre d'octets) de la chaîne binaire. La plage autorisée pour <i>n</i> est comprise entre 0 et 32 704.

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
912	BINARY	Sans objet	Identique à CHAR, à l'exception du fait que : <ul style="list-style-type: none"> • IXFCLENG représente la longueur (nombre d'octets) de la séquence de chaînes binaires. • La plage autorisée pour <i>n</i> est comprise entre 1 et 255.

Tableau 20. Format des données de colonne IXF par type de données (suite)

Code de type de données	Type de données	Informations relatives à la longueur des données (lorsque OUTPUTMODE = CHARACTER)	Informations relatives à la longueur des données (lorsque OUTPUTMODE = BINARY)
2448	TIMESTAMP WITH TIME ZONE	<p>La longueur est 147 pour TIMESTAMP(0) WITH TIME ZONE et 148 + <i>n</i> pour TIMESTAMP(<i>n</i>) WITH TIME ZONE, où <i>n</i> = 1-12.</p> <p>Le format est le suivant :</p> <p><i>aaaa-mo-jj-hh-mm-ss-nnnnnnnnnnnzth:tm</i></p> <p>où <i>aaaa-mo-jj-hh-mm-ss</i> spécifie l'horodatage de la même façon que pour les données TIMESTAMP et :</p> <ul style="list-style-type: none"> • <i>nnnnnnnnnnnn</i> indique un nombre de 0 à 12 chiffres qui représente le nombre de fractions de seconde. • <i>z</i> est un signe plus (+) ou moins (-) qui indique le décalage horaire par rapport au temps universel coordonné (précédemment appelé temps moyen de Greenwich ou GMT). • <i>th</i> représente une valeur à deux chiffres désignant les heures du fuseau horaire. • <i>tm</i> représente une valeur à deux chiffres désignant les minutes du fuseau horaire. <p>La plage admise pour le fuseau horaire est comprise entre -24:00 et +24:00. Pour spécifier le temps universel coordonné (UTC), vous pouvez spécifier le fuseau horaire -0:00 ou +0:00 ou remplacer le décalage horaire et le signe correspondant par la lettre majuscule Z.</p> <p>Par exemple, 2010-09-30-13.08.36.123456654321-08:00 correspond à 1:08 P.M. et 36.123456654321 secondes le 30 septembre 2010, à San Jose, en Californie, USA. L'horodatage 2010-09-30-13.08.36.123456654321Z indique 1:08 P.M. et 36.123456654321 secondes quel que soit l'UTC.</p>	Identique au format caractère

Interprétation d'un objet exporté au format IXF

L'exemple suivant vous aide à interpréter les données qui sont exportées au format IXF.

Supposons que la table affichée dans l'exemple d'un objet données au format QMF soit maintenant exportée au format IXF (avec le paramètre OUTPUTMODE=CHARACTER). La table à exporter se présente comme suit :

ID	NOM	COMM
10	SANDERS	-
20	PERNAL	612,45

L'ensemble de données ou file d'attente CICS exporté est composé de sept enregistrements au total ; un enregistrement H, un enregistrement T, trois enregistrements C et deux enregistrements D sont indiqués ici :

```

HIXF0000QMF  VAR1M0201012040956000005N
T18                               base de données      CCI00003
C02ID          NYNR50000000        000002            00
C04NAME        YYNR44800000        00009000008      00
C04COMM        YYNR48400000        00702000023    00
D 00010 00007SANDERSxx -
D 00020 00006PERNALxxx 00612,45

```

Les caractères binaires non imprimables s'affichent sous forme de caractères x. La figure ci-dessous donne des informations plus détaillées sur ces enregistrements.

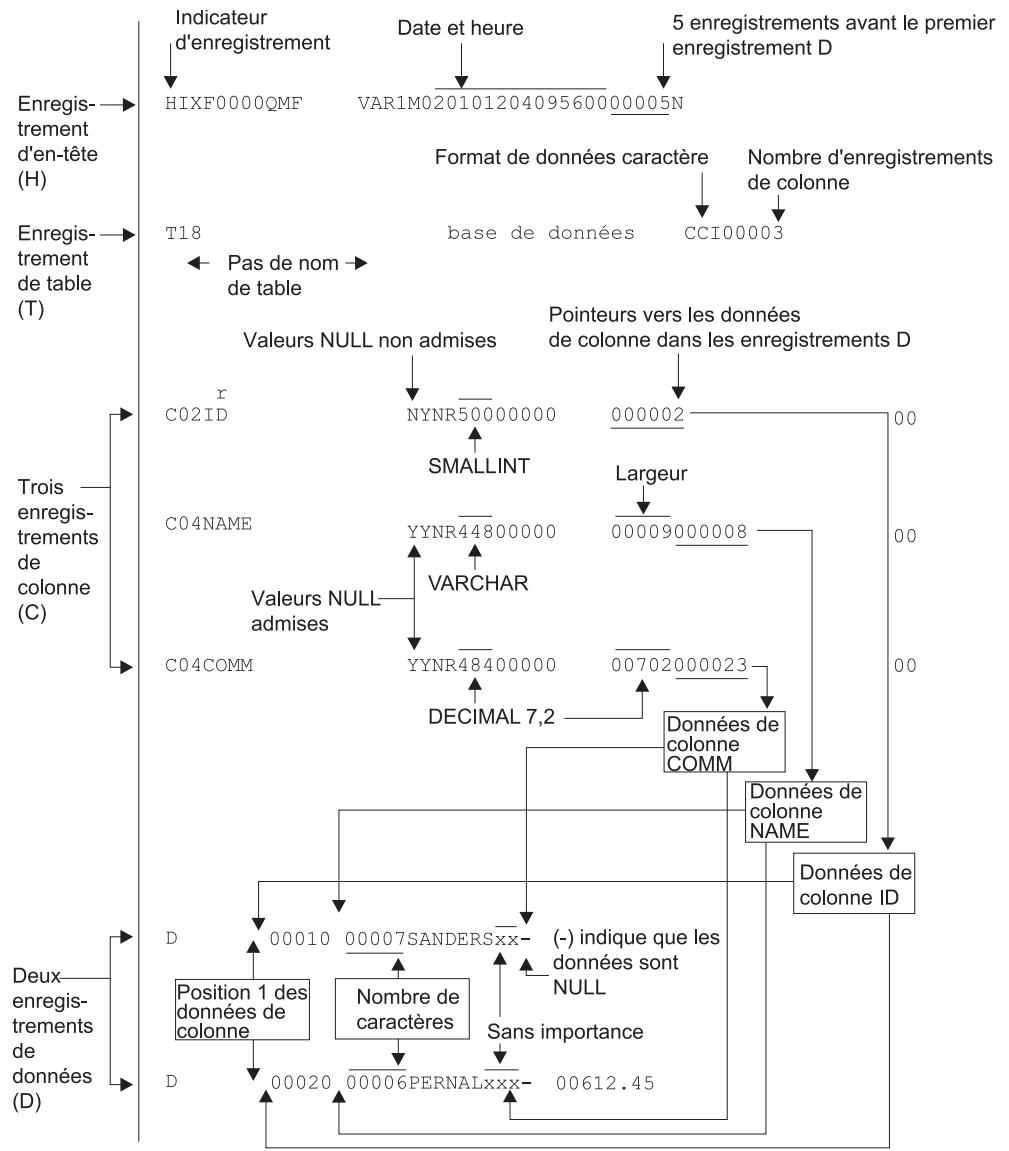


Figure 17. Format des exemples d'enregistrements ICF (OUTPUTMODE=CHARACTER)

Supposons maintenant que la même table soit exportée au format ICF, mais avec le paramètre OUTPUTMODE=BINARY. L'ensemble de données ou file d'attente CICS exporté est composé de sept enregistrements qui s'affichent dans la table :

```

HIXF0000QMF  VAR1M02010120409565000005N
T18                               database      CMI00003
C02ID          NYNR50000000        000003            00

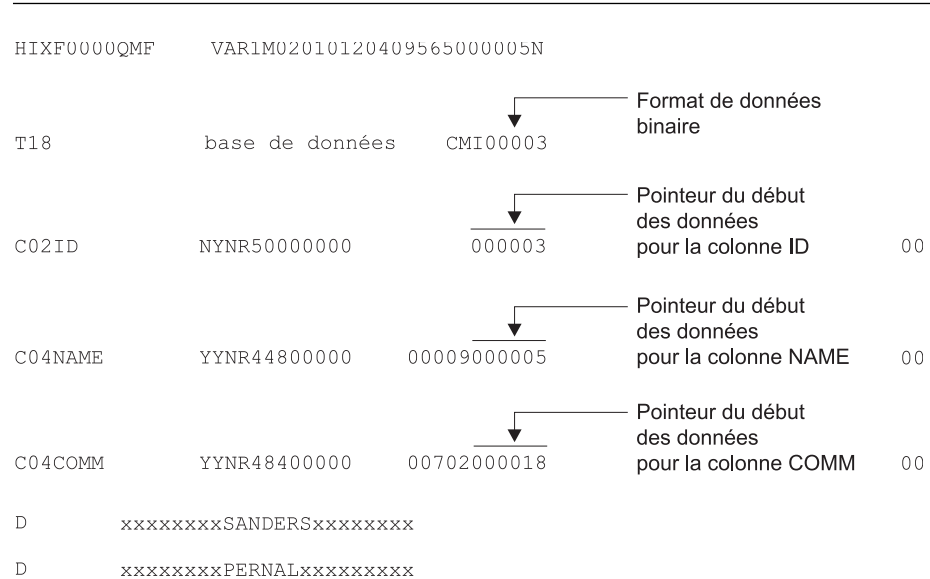
```

```

C04NOM          YYNR44800000      000090000005      00
C04COMM         YYNR48400000      007020000018      00
D      xxxxxxxxSANDERSxxxxxxxx
D      xxxxxxxxPERNALxxxxxxxx

```

Excepté pour les octets 44 à 49 ou 56 à 61 (position de départ des données de colonne), les informations figurant dans les enregistrements H, T et C sont essentiellement identiques. Toutefois, les données présentes dans les enregistrements D diffèrent de façon significative. La figure suivante contient plus d'informations sur les enregistrements de l'ensemble de données ou file d'attente CICS exporté.



Les deux enregistrements de données (D) sont affichés ci-dessous en notation hexadécimale avec une explication des diverses zones :

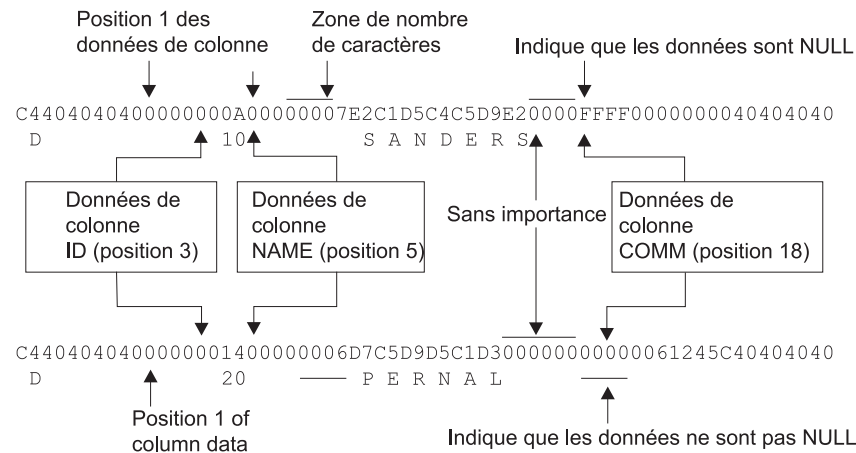


Figure 18. Format des exemples d'enregistrements IXF (OUTPUTMODE=BINARY)

Exportation de données ou de tables au format XML

Si votre table ou vos données contiennent une colonne XML ou des données LOB, vous devez utiliser la clause DATAFORMAT=XML dans la commande EXPORT

DATA ou EXPORT TABLE. Ce format peut également être utilisé lorsque les données ou la table à exporter ne contiennent pas de colonne XML ou de données LOB.

Restriction : QMF prend en charge les opérations avec des données XML uniquement lorsque vous êtes connecté à une édition de base de données prenant en charge le type de données XML.

Lorsque vous exportez des données ou des tables au format XML, les données sont exportées vers le fichier UNIX, l'ensemble de données TSO ou la file d'attente de données CICS que vous indiquez dans la commande. QMF utilise la spécification XML 1.0 (quatrième édition) lors de l'importation et de l'exportation des données.

QMF utilisant les services d'analyse XML de z/OS et les services de conversion Unicode de z/OS lors du traitement des données XML pour l'exportation ou l'importation, ces services doivent être configurés et actifs.

Toutes les balises figurant dans les exemples XML doivent être présentes avant l'importation des données de colonne XML. Les balises doivent se trouver dans le contenu du fichier, de l'ensemble de données ou de la file d'attente de données CICS car QMF les utilise pour analyser le fichier. Lorsque QMF détecte la balise <extensions> à la fin du fichier, le curseur est fermé et l'importation prend fin. La modification ou la suppression de cette balise provoque une lecture infinie des données.

Les données sont exportées sous la forme d'un document XML au format Unicode UTF-8 avec un CCSID de 1208. L'ensemble de données ou file d'attente CICS XML exporté se compose d'enregistrements d'en-tête, d'enregistrements définissant l'ensemble de résultats, d'enregistrements de métadonnées pour chaque colonne figurant dans les données ou la table, ainsi que d'enregistrements de données pour chaque ligne figurant dans la table ou les données exportées.

Enregistrements d'en-tête

Les enregistrements d'en-tête dans le fichier XML exporté contiennent la version de XML utilisée, le schéma de codage et la feuille de style utilisés pour mettre en forme le document XML exporté.

L'exemple suivant décrit le type d'informations incluses dans les enregistrements d'en-tête d'un fichier exporté :

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

QMF fournit une feuille de style que vous pouvez utiliser pour mettre en forme vos données XML exportées. La feuille de style par défaut (portant le nom par défaut qmf.xslt) est fournie en tant que membre DSQ1STSH de l'ensemble de données des exemples QMF, QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale). Copiez cette feuille de style par défaut à l'emplacement du fichier exporté, puis ouvrez le document XML pour qu'il soit mis en forme conformément à ces spécifications. Si vous utilisez un autre nom pour la feuille de style, modifiez l'en-tête figurant dans le fichier exporté pour qu'il se réfère au nouveau nom de la feuille de style.

Enregistrements définissant l'ensemble de résultats

La définition de l'ensemble de résultats contient une définition d'espace-noms et une définition de schéma pour le fichier schéma QMF utilisé avec le fichier XML.

L'exemple suivant présente les enregistrements pour la définition d'ensemble de résultats dans un exemple de fichier XML exporté contenant sept colonnes :

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

.....Les définitions pour chaque colonne se placent ici.

</Metadata>
<Data>

.....Les données pour chaque ligne se placent ici.

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

Enregistrements de métadonnées

Les métadonnées de colonne dans le fichier XML exporté se composent du nombre de colonnes, des noms de colonne, des libellés de colonne (le cas échéant), des types de données, des longueurs de données, du format et indiquent si les données sont de type NULL.

Un exemple de métadonnées pour une colonne appelée "ID" est présenté ici. Le fichier XML exporté contient un bloc de description d'une colonne pour chaque colonne.

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

Enregistrements de données

Le fichier XML exporté contient un bloc de définition de ligne pour chaque ligne de données exportées. Les enregistrements de données figurent au format de bloc variable sur plusieurs lignes (VBS).

Une balise <cell> identifie chaque colonne de la ligne par un numéro, comme indiqué ici pour la première ligne de l'exemple de table Q.PERS :

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

Lorsque vous utilisez la clause DATAFORMAT=XML dans la commande EXPORT DATA ou EXPORT TABLE et que les données contiennent une colonne définie avec le type de données XML, QMF renvoie les données XML à la ligne dans des balises CDATA. Les balises CDATA empêchent l'analyseur syntaxique de tenter de traiter les données XML. Voici un exemple d'affichage des données XML dans un fichier exporté :

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

Validation du fichier XML par QMF

Un document de schéma XML décrit la structure d'un document XML et définit des paramètres pour la validité des éléments et des attributs au sein du document XML.

Un fichier schéma par défaut est fourni avec QMF en tant que membre DSQ1SCEM de l'ensemble de données des exemples QMF, QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale). Copiez ce membre dans le répertoire où se trouve le fichier contenant le document XML. Nommez le document de schéma qmf_data.xsd, qui est le nom du document de schéma par défaut sous QMF for Workstation et QMF for WebSphere. Vous pouvez modifier le fichier schéma par défaut en fonction de vos besoins métier pour la mise en forme des données XML. Si vous utilisez un autre nom que qmf_data.xsd ou un autre fichier schéma, modifiez le nom dans les fichiers qui référencent le document de schéma.

Exportation de données ou de tables au format CSV

Vous pouvez exporter des données ou des tables au format CSV. Vous pouvez ensuite télécharger les données exportées vers votre poste de travail sur lequel les données au format CSV peuvent être utilisées avec des applications comme Microsoft Excel.

Lorsque vous exportez des données ou des tables au format CSV, vous pouvez utiliser l'option HEADER pour indiquer si les en-têtes de colonne doivent être exportés ou non avec les données. La valeur par défaut inclut les en-têtes de colonne. Si vous exportez les en-têtes de colonne, la valeur de la variable globale DSQDC_COL_LABELS détermine si les libellés ou les noms de colonne sont exportés ou non. La valeur par défaut de DSQDC_COL_LABELS est 1, ce qui signifie que les libellés de colonnes sont exportés.

Les données exportées sont formatées en tant que lignes de valeurs de colonne séparées à l'aide du séparateur de colonnes. Si l'option HEADER=YES est spécifiée, une ligne de noms de colonne séparés par un séparateur de colonnes précède les lignes de valeurs de colonne. La valeur du séparateur de colonnes est déterminée par la valeur Q.PROFILES.DECIMAL de l'utilisateur.

Les données qui sont exportées au format CSV sont codées dans la base de données DB2 locale pour le schéma de codage SBCS z/OS.

Règles et informations relatives à l'exportation et à l'importation d'objets données et de tables

QMF exporte des objets données et table vers la mémoire temporaire et définit des règles concernant l'allocation de cette mémoire. QMF dispose également de méthodes spécifiques de traitement des erreurs d'importation.

Allocation de fichiers UNIX, d'ensembles de données TSO ou de files d'attente de données CICS

La commande QMF IMPORT DATA permet de stocker les données dans la zone de stockage temporaire QMF et d'afficher le rapport à l'écran. En fait, seule une partie des données est stockée et affichée. Le fichier UNIX, l'ensemble de données TSO ou la file d'attente de données CICS reste ouvert et affecté à QMF. QMF lit les enregistrements lorsque l'utilisateur fait défiler les données.

Cette connexion est conservée tant que l'objet données n'est pas remplacé ou réinitialisé ou que QMF n'a pas lu tous les enregistrements. Ensuite, le fichier UNIX, l'ensemble de données TSO ou la file d'attente de données CICS est fermé et n'est plus considéré comme affecté à QMF. Une application ne doit pas tenter de supprimer ou de modifier un fichier UNIX, un ensemble de données TSO ou une file d'attente de données CICS alloué à QMF à l'aide d'une commande IMPORT DATA. L'application doit soit utiliser une autre source de données, soit vider l'espace de stockage temporaire QMF pour l'objet données (à l'aide d'une commande RESET DATA). Elle peut ensuite tenter de modifier ou supprimer le fichier, l'ensemble de données ou la file d'attente qu'elle lit.

Lors de l'exécution de la commande IMPORT, QMF ne verrouille pas le fichier UNIX, l'ensemble de données TSO ou la file d'attente de données CICS pendant sa lecture. Il ne prend pas de mesures pour empêcher la modification du fichier, de l'ensemble de données ou de la file d'attente de données pendant sa lecture. Si le fichier, l'ensemble de données ou la file d'attente est effacé ou modifié avant que QMF ait fini de le lire, vous risquez d'obtenir des résultats imprévisibles et de provoquer une erreur système.

Une invite de données incomplète peut s'afficher lorsque la mémoire temporaire est insuffisante pour extraire la totalité de l'objet à exporter. Si vous avez besoin de mémoire supplémentaire, vous pouvez utiliser le paramètre DSQSPILL pour spécifier l'utilisation d'une mémoire auxiliaire. Si vous utilisez QMF sous TSO, outre le paramètre DSQSPILL, vous pouvez indiquer le paramètre DSQSPTYP pour utiliser la mémoire étendue pour l'envoi de données.

Erreurs d'exportation

Une fois que QMF a importé des données à partir d'un fichier UNIX, d'un ensemble de données TSO ou de la file d'attente de données CICS, QMF affiche l'écran REPORT et un message de confirmation.

Si le fichier, l'ensemble de données ou la file d'attente de données contient des erreurs de format, QMF n'affiche pas l'écran REPORT. En cas d'erreurs de format, QMF affiche un message d'erreur sur l'écran objet qui était en cours avant le traitement de la commande IMPORT par QMF. Toutefois, si l'écran objet en cours était l'écran REPORT et que QMF détecte des erreurs dans les données importées, QMF affiche l'écran d'accueil et un message d'erreur.

Exportation de formats, de rapports et de requêtes créées de façon interactive

Les objets format et requête créée de façon interactive sont exportés et importés dans un format codé qui représente l'objet sous forme de tableau. Les rapports sont également exportés dans un format codé ; toutefois, ils ne peuvent pas être importés.

Le format codé vous aide à manipuler plus facilement des parties d'un objet. Les conditions suivantes s'appliquent lorsque vous exportez un objet au format codé :

- Tous les numéros de table et de colonne sont écrits sous forme de nombre à quatre chiffres.
- Les colonnes de table sont écrites dans l'ordre dans lequel elles apparaissent habituellement dans l'objet, à ceci près que la colonne avec la longueur maximale est déplacée vers l'extrémité droite de l'enregistrement de table et des enregistrements de ligne associés.
- Les longueurs numériques comportent trois chiffres (dont des zéros en tête, si nécessaire).
- Un caractère blanc est utilisé comme délimiteur dans tous les enregistrements.
- Le délimiteur n'est pas placé après le dernier caractère de chaque enregistrement.
- Des blancs sont placés dans toutes les zones réservées.
- Un enregistrement E est le dernier enregistrement écrit dans le fichier de sortie.

Référence associée:

«Spécifications de taille pour les objets externalisés», à la page 142

Les tables et les objets externes comportent des spécifications de taille et de format d'enregistrement qui varient en fonction du type d'objet.

Format général du fichier exporté

Le format codé d'un format, d'un rapport ou d'une requête créée de façon interactive se compose d'enregistrements d'en-tête de format fixe et d'enregistrements de format variable qui constituent l'objet.

Enregistrements d'en-tête

La plupart des enregistrements des rapports, requêtes créées de façon interactive ou formats exportés ont un format variable. Toutefois, les enregistrements d'en-tête ont un format fixe, même si l'ensemble de données ou la file d'attente de données qui contient les enregistrements peut être de format variable.

Ces enregistrements permettent d'identifier le contenu du rapport, de la requête créée de façon interactive ou du format exporté. Un enregistrement d'en-tête est le premier enregistrement de la file d'attente ou de l'ensemble de données exporté. Il décrit les caractéristiques de l'objet.

Un enregistrement d'en-tête contient les informations décrites dans le tableau ci-dessous. Un astérisque indique que la zone est requise pour l'importation.

Tableau 21. Informations relatives à l'enregistrement d'en-tête

Position de l'octet	Informations et type
01*	Indicateur d'enregistrement d'en-tête (H)
02	Blanc

Tableau 21. Informations relatives à l'enregistrement d'en-tête (suite)

Position de l'octet	Informations et type
03-05*	Identificateur de produit (QMF)
06	Blanc
07-08	Niveau d'édition de QMF dans lequel le rapport, la requête créée de façon interactive ou le format a été exporté ; ce numéro est 19 pour QMF Version 12.1
09	Blanc
10*	Type d'objet : <ul style="list-style-type: none"> • F pour un format • R pour un rapport • T pour une requête créée de façon interactive
11	Blanc
12-13*	Niveau d'objet QMF : <ul style="list-style-type: none"> • 01 pour un rapport • 04 pour un format • 01 pour les requêtes créées de façon interactive qui ne contiennent pas de spécification de période • 02 pour les requêtes créées de façon interactive dans lesquelles au moins une des tables de la requête comporte une spécification de période <p>Le niveau d'objet reflète une modification dans le format d'un objet. Chaque fois que le format est modifié dans une édition de QMF, son niveau d'objet l'est également. Le niveau d'objet augmente uniquement lorsque le changement de format risque de provoquer une erreur dans votre application.</p>
14	Blanc
15*	Format de données de l'objet ("E" pour le format codé utilisé pour exporter des objets format, rapport et requête créée de façon interactive)
16	Blanc
17	Etat de l'objet : E - Contient des erreurs (pour un format uniquement) ; W - Contient des avertissements ; V - Valide
18	Blanc
19	Tout ou partie de l'indicateur d'objet (W pour la totalité de l'objet)
20	Blanc
21	Langue nationale en cours d'utilisation lors de l'exportation de l'objet (E pour l'anglais)
22	Blanc
23*	Vous pouvez générer un rapport, une requête créée de façon interactive ou un format en dehors de QMF au format approprié et l'importer dans QMF. Codez un R dans cette position d'octet si vous voulez que QMF remplace l'objet dans la mémoire temporaire par l'objet que vous importez.
24	Blanc

Tableau 21. Informations relatives à l'enregistrement d'en-tête (suite)

Position de l'octet	Informations et type
25-26	Longueur de la zone de contrôle au début de chaque enregistrement : <ul style="list-style-type: none"> • 01 pour un format • 02 pour un rapport • 01 pour une requête créée de façon interactive
27	Blanc
28-29	Longueur des zones de longueur entière spécifiée dans les enregistrements V et T (03)
30	Blanc
31-38	Date au format <i>aa/mm/jj</i>
39	Blanc
40-44	Heure au format <i>hh:mmmm</i>
45	Blanc
51	Blanc

Référence associée:

«Exportation d'un format», à la page 113

L'objet format contient toutes les informations spécifiées dans tous les écrans de format QMF. Lorsque vous exportez un format, QMF convertit en format codé tous les écrans de format dont les valeurs sont différentes des valeurs par défaut.

«Exportation d'une requête créée de façon interactive», à la page 131

Un objet requête créée de façon interactive exporté contient les informations affichées dans la zone d'écho de l'écran principal Requête créée de façon interactive.

«Exportation d'un rapport standard», à la page 124

Lorsque QMF affiche un rapport, vous voyez le résultat de l'interaction entre le format et l'objet données dans la mémoire temporaire. Il n'existe pas d'objet rapport dans la mémoire temporaire. Lorsque vous exportez un rapport, QMF exporte en fait l'interaction d'un format et d'un objet données.

«Conventions pour les informations NLF (National Language Feature)», à la page 5
 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Enregistrements de l'objet exporté

A l'exception des enregistrements d'en-tête, qui sont des enregistrements de format fixe, tous les enregistrements des requêtes créées de façon interactive, des rapports et des formats exportés sont des enregistrements de format variable. Ces derniers sont acceptés en entrée.

Les enregistrements de format variable se présentent généralement comme dans la figure ci-dessous.



Figure 19. Format général des enregistrements de format variable

Le contenu de la zone de contrôle est présenté ici :

Tableau 22. Format général des enregistrements de format variable

Position de l'octet	Description
01	Identificateur d'enregistrement (H, V, T, R, E, *, L, C)
02	Blanc (parfois omis ; voir le type spécifique d'enregistrement de format variable)

La zone de données d'enregistrement est une zone de longueur variable qui contient des informations sur cet enregistrement spécifique. Les champs de cette zone sont séparés par un délimiteur (un caractère blanc est utilisé dans ces exemples).

Enregistrements de valeur de données (V) :

Les enregistrements de valeur des rapports, des requêtes créées de façon interactive ou des formats exportés sont utilisés pour fournir une valeur pour une zone unique d'un objet, telle que des lignes vierges avant l'en-tête du format.

Les enregistrements V contiennent :

- Un numéro de zone spécifique à l'objet
- La longueur de la zone
- La valeur de cette zone

La zone de contrôle pour les enregistrements V s'affiche dans le tableau suivant :

Tableau 23. Zone de contrôle pour les enregistrements V

Position de l'octet	Description
01	Identificateur d'enregistrement de valeur (V)
02	Blanc (utilisé uniquement pour les rapports ; omis pour les formats et les requêtes créées de façon interactive)

La zone de données d'enregistrement pour les enregistrements V s'affiche dans le tableau ci-dessous.

Tableau 24. Zone de données d'enregistrement pour les enregistrements V

Position de l'octet	Description
01	Blanc
02-05	Numéro de zone (1001-9999)
06	Blanc
07-09	Longueur de la valeur de données (000-999) Il peut également s'agir d'un astérisque (*) suivi de deux blancs. Un astérisque indique que la valeur de données est délimitée par la fin de l'enregistrement.
10	Blanc
De 11 à la fin	Données

Dans la zone de données d'enregistrement pour les enregistrements V :

- Les positions d'octet de la zone de données d'enregistrement sont décalées par rapport à la fin de la zone de contrôle, dont la longueur est indiquée dans l'enregistrement d'en-tête.
- L'omission d'une valeur de données (une fin d'enregistrement ou uniquement des blancs suivant la zone de longueur) indique que la zone contient une valeur NULL.
- Si la zone de longueur est égale à zéro, la valeur par défaut pour la zone est appliquée et un message d'avertissement est envoyé.
- Si la longueur spécifiée est différente des données réelles qui suivent, QMF émet un avertissement.

Enregistrements de description de table de données (T) :

Dans un format codé, la plupart des données d'un objet apparaissent dans des tables. Ces tables ne sont pas des tables relationnelles dans la base de données, mais plutôt un moyen de regrouper des informations au sein du format codé. Chaque enregistrement T définit une table et chaque table correspond à une partie particulière d'un objet, tels que des calculs récapitulatifs dans le format. Un fichier exporté peut donc contenir un grand nombre de ces tables codées.

Un enregistrement T est toujours suivi d'enregistrements R. L'enregistrement T décrit les enregistrements R qui le suivent. S'il n'y a pas d'enregistrements R qui suivent un enregistrement T, la table est omise.

Vérifiez que votre programme d'application se rapporte au contenu des tables d'un rapport, d'une requête créée de façon interactive ou d'un format exporté à l'aide du codage figurant dans l'enregistrement T pour localiser correctement les valeurs dans les enregistrements R. Votre programme d'application ne doit pas utiliser de décalages fixes pour localiser des informations dans les enregistrements R.

La zone de contrôle pour les enregistrements T s'affiche dans le tableau suivant :

Tableau 25. Zone de contrôle pour les enregistrements T

Position de l'octet	Description
01	Identificateur d'enregistrement de table (T)
02	Blanc (utilisé uniquement pour les rapports ; omis pour les formats et les requêtes créées de façon interactive)

La zone de données d'enregistrement pour les enregistrements T s'affiche dans le tableau ci-dessous. Les positions d'octet dans le tableau sont des décalages qui suivent la fin de la zone de contrôle, dont la longueur est indiquée dans l'enregistrement d'en-tête.

Tableau 26. Zone de données d'enregistrement pour les enregistrements T

Position de l'octet	Description
01	Blanc
02-05	Numéro de table (1001-9999)
06	Blanc

Tableau 26. Zone de données d'enregistrement pour les enregistrements T (suite)

Position de l'octet	Description
07-09	Nombre de lignes (enregistrements R) dans cette table Un astérisque (*) utilisé à la place d'une valeur numérique signifie que la table est composée de tous les enregistrements R qui suivent.
10	Blanc
11-13	Nombre de colonnes dans l'enregistrement (000-999)
14	Blanc
15-18, 24-27, ...	Numéro de zone pour cette colonne (zone répétée)
19, 28, ...	Blanc (zone répétée)
20-22, 29-31, ...	Longueur des valeurs de données dans cette colonne (zone répétée)

Les octets 11 à 13 (nombre de colonnes) indiquent le nombre de paires numéro de zone/longueur de valeur de données qui suivent. Les informations situées à partir de l'octet 15 sont répétées pour chaque colonne.

Gardez à l'esprit les informations suivantes relatives aux enregistrements T lors de l'exportation et de l'importation d'objets :

- Lors de l'importation d'un format ou d'une requête créée de façon interactive, le nombre d'enregistrements R doit correspondre au nombre de lignes spécifié dans les octets 07 à 09 de la zone de données de l'enregistrement T. Sinon, QMF envoie un avertissement.
- Lors de l'importation d'un format ou d'une requête créée de façon interactive, le nombre de colonnes indiqué dans les octets 11 à 13 doit correspondre aux paires numéro de zone/longueur dans les octets qui suivent. Sinon, QMF envoie un avertissement.
- Le nombre de paires numéro de zone/longueur est limité au nombre de colonnes dans la table et leur ordre est arbitraire.
- Les valeurs par défaut sont restaurées pour les colonnes avec une longueur égale à zéro lorsque l'objet situé dans la zone de stockage temporaire est mis à jour et un avertissement est émis. Les colonnes qui ne sont pas incluses dans cette table sont également traitées de cette façon. Toutefois, une valeur par défaut est fournie avec les requêtes créées de façon interactive, si possible. Sinon, une erreur survient.
- Pour définir une zone de colonne à blanc, la colonne doit avoir une longueur positive dans l'enregistrement T et une valeur à blanc dans l'enregistrement R.

Enregistrements de ligne de table (R) :

Les enregistrements R des rapports, requêtes créées de façon interactive ou formats exportés fournissent un ensemble de valeurs pour une ligne unique dans une table codée. Les enregistrements R contiennent une liste de valeurs disposées selon un ordre décrit par l'enregistrement T associé. Un enregistrement R correspond à la description des positions et longueurs des valeurs de données spécifiées dans l'enregistrement T.

La zone de contrôle pour les enregistrements R s'affiche dans le tableau suivant :

Tableau 27. Zone de contrôle pour les enregistrements R

Position de l'octet	Description
01	Identificateur d'enregistrement de ligne (R)
02	Blanc (utilisé uniquement pour les rapports ; omis pour les formats et les requêtes créées de façon interactive)

La zone de données pour les enregistrements R, qui suit la zone de contrôle, est composée d'une série de valeurs séparées par un délimiteur (caractère blanc). Le format est le suivant :

`_value.._value..._value..`

Dans ce format, `value...` représente la valeur de données pour cette ligne et cette colonne, et `_` le délimiteur.

Gardez à l'esprit les informations suivantes lorsque vous utilisez des enregistrements R :

- Un enregistrement R doit suivre immédiatement un autre enregistrement R ou un enregistrement T.
- Le nombre de valeurs de données doit correspondre à la description figurant dans l'enregistrement T associé.
- Une longueur de valeur de données égale à zéro dans l'enregistrement T associé indique qu'aucune valeur ne sera appliquée à cette ligne et à cette colonne de l'objet. En d'autres termes, la ligne et la colonne prennent leurs valeurs par défaut. Toutefois, la présence de la zone dans l'enregistrement T requiert un blanc supplémentaire dans l'enregistrement R pour cette zone. Une valeur de longueur égale à zéro provoque la présence de deux blancs à la suite dans l'enregistrement R.

Enregistrement de fin d'objet (E) :

L'enregistrement E d'un rapport, d'une requête créée de façon interactive ou d'un format exporté indique la fin d'un objet exporté. Il s'agit du dernier enregistrement d'un fichier exporté, qui apparaît sous la forme d'un caractère E. Pour un rapport exporté, un enregistrement E est suivi d'un blanc pour fermer sa zone de contrôle. Pour un format, le blanc est omis.

Les enregistrements qui suivent l'enregistrement E sont ignorés. Si aucun enregistrement E n'est inclus dans le fichier importé, QMF suppose qu'une fin de fichier implique la fin de l'objet.

Enregistrement de données d'application (*) :

Les enregistrements de données d'application des rapports, requêtes créées de façon interactive ou formats exportés permettent aux programmes d'application d'associer leurs propres données à un objet dans le fichier externe. Les programmes d'application utilisent fréquemment ces enregistrements comme enregistrements de commentaire pour décrire plus précisément l'objet dans le fichier.

Les informations qui suivent l'astérisque sont ignorées et n'ont aucun impact sur le processus de saisie.

Les enregistrements de données d'application peuvent apparaître à n'importe quel endroit du fichier externe, sauf avant l'enregistrement d'en-tête (H). QMF n'écrit pas les enregistrements de données d'application lors de l'exportation. Toutefois, vous pouvez utiliser ces enregistrements dans l'ensemble de données ou file d'attente CICS que vous créez. Le tableau suivant présente le contenu d'un enregistrement de données d'application :

Tableau 28. Contenu d'un enregistrement de données d'application

Position de l'octet	Description
01	Identificateur d'enregistrement de données d'application (*)
De 02 à la fin de l'enregistrement	Données

Voici un exemple d'enregistrement de données d'application qui s'affiche dans un format exporté :

*Voici le format qui effectue un regroupement par DEPT.

Enregistrements de ligne de rapport (L) :

Chaque ligne formatée dans un rapport est décrite par un enregistrement L. Il existe un enregistrement L pour chaque ligne du rapport.

Comme les autres enregistrements de format variable (V, T et R), les enregistrements L sont composés d'une zone de contrôle suivie d'une zone de données d'enregistrement. Le format de la zone de contrôle est similaire aux autres enregistrements. La zone de données d'enregistrement se compose d'une zone fixe qui précède la ligne de rapport formatée. La zone fixe fournit des informations sur la ligne de rapport qui la suit. Le format d'un enregistrement L s'affiche dans la figure ci-dessous.

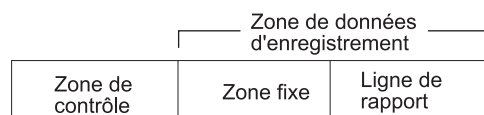


Figure 20. Format d'un enregistrement L

La zone de contrôle pour un enregistrement L s'affiche dans le tableau suivant :

Tableau 29. Zone de contrôle pour un enregistrement L

Position de l'octet	Description
01	Identificateur d'enregistrement de valeur (L)

Tableau 29. Zone de contrôle pour un enregistrement L (suite)

Position de l'octet	Description
02	<p>Indicateur de continuation</p> <p>Indique si l'enregistrement en cours se poursuit sur un enregistrement de continuation de données :</p> <ul style="list-style-type: none"> • C indique une continuation Un enregistrement C suit immédiatement un enregistrement L marqué à l'aide d'un caractère de continuation dans l'octet 2 de la zone de contrôle. • D indique une continuation avec insertion des délimiteurs DBCS S0 et SI à la fin de l'enregistrement en cours et au début de la portion de données de l'enregistrement suivant D est spécifié pour l'indicateur de continuation dans la zone de contrôle lorsque l'enregistrement en cours est trop long pour tenir dans un seul enregistrement physique. Lors du fractionnement de l'enregistrement, des caractères S0 (hors code) et SI (en code) sont ajoutés à l'enregistrement en cours et à l'enregistrement suivant pour préserver l'intégrité des données DBCS qui font l'objet d'une continuation. • Blanc en cas de non-continuation

La zone de données d'enregistrement pour un enregistrement L s'affiche dans le tableau ci-dessous. Les octets 6 à 13 sont des attributs de type de ligne. L'octet 06 est toujours 1. Chaque octet figurant dans les octets 7 à 13 indique la présence ou l'absence de l'attribut de type de ligne correspondant dans la ligne de rapport formatée (1 = attribut présent, 0 = attribut absent).

Tableau 30. Zone de données d'enregistrement pour un enregistrement L

Position de l'octet	Description
01	Blanc
02-04	<p>Indicateur de partie de rapport :</p> <p>110 = En-tête de page 120 = Bas de page 13<i>n</i> = En-tête de rupture (<i>n</i> est le numéro de rupture, compris entre 1 et 6) 15<i>n</i> = Bas de page de rupture (<i>n</i> est le numéro de rupture, compris entre 1 et 6) 170 = En-tête de colonne 171 = En-tête de détail 180 = Ligne de détail 181 = Ligne de résumé de groupe 190 = Bas de page final</p>
05	Blanc

Tableau 30. Zone de données d'enregistrement pour un enregistrement L (suite)

Position de l'octet	Description
06	1
07	Données
08	Texte
09	Séparateur
10	Renvoi à la ligne dans les colonnes Les attributs de retour à la ligne dans les colonnes (octet 10) et de renvoi à la ligne (octet 11) sont utilisés pour indiquer la continuation d'une ligne de rapport logique sur plusieurs lignes de rapport physiques. La présence de l'un de ces attributs dans un enregistrement de format L signifie que les données de colonne ou la ligne avec retour à la ligne continuent sur un enregistrement de format L suivant.
11	Renvoi à la ligne Les attributs de retour à la ligne dans les colonnes (octet 10) et de renvoi à la ligne (octet 11) sont utilisés pour indiquer la continuation d'une ligne de rapport logique sur plusieurs lignes de rapport physiques. La présence de l'un de ces attributs dans un enregistrement de format L signifie que les données de colonne ou la ligne avec retour à la ligne continuent sur un enregistrement de format L suivant.
12	Seconde ligne de données (rapports transversaux uniquement) Les rapports transversaux qui contiennent des colonnes de pourcentage ou de somme cumulée peuvent comporter deux lignes de données pour chaque résumé de groupe (rupture et final également). La première ligne de données récapitulatives contient les valeurs de pourcentage cumulé ou de somme cumulée de la colonne, qui ont été calculées sur chaque valeur «transversale» unique. La deuxième ligne de données récapitulatives contient les valeurs de pourcentage cumulé ou de somme cumulée de la colonne, qui ont été calculées pour chaque groupe (dans le rapport ou au sein d'une rupture de séquence). Le type de la deuxième ligne de données (octet 12) identifie la seconde ligne de données dans les rapports exportés de cette nature.
13	Réservé
14	Blanc

L'exemple suivant montre un enregistrement L pour une ligne de bas de page de rupture dans un rapport qui contient du texte et des données :

Enregistrements de continuation de données (C) :

Un enregistrement C dans une requête créée de façon interactive, un rapport ou un format exporté permet de continuer une valeur ou un ensemble de valeurs sur plusieurs enregistrements. Il suit immédiatement l'enregistrement auquel s'applique la continuation. Le format d'un enregistrement C correspond au format de l'enregistrement à l'origine de la continuation. QMF utilise les enregistrements C uniquement pour continuer des enregistrements L.

La zone de contrôle pour un enregistrement C s'affiche dans le tableau suivant :

Tableau 31. Zone de contrôle pour un enregistrement C

Position de l'octet	Description
01	Identificateur d'enregistrement de valeur (C)
02	<p>Indicateur de continuation. Indique si l'enregistrement en cours se poursuit sur un autre enregistrement C :</p> <ul style="list-style-type: none"> • C indique une continuation Un enregistrement C suit immédiatement un enregistrement L marqué à l'aide d'un caractère de continuation dans l'octet 2 de la zone de contrôle. • D indique une continuation avec insertion des délimiteurs DBCS S0 et SI à la fin de l'enregistrement en cours et au début de la portion de données de l'enregistrement suivant D est spécifié pour l'indicateur de continuation dans la zone de contrôle lorsque l'enregistrement en cours est trop long pour tenir dans un seul enregistrement physique. Lors du fractionnement de l'enregistrement, des caractères S0 (hors code) et SI (en code) sont ajoutés à l'enregistrement en cours et à l'enregistrement suivant pour préserver l'intégrité des données DBCS qui font l'objet d'une continuation. • Blanc en cas de non-continuation

La zone de données d'enregistrement pour un enregistrement C s'affiche dans le tableau ci-dessous. Les positions d'octet affichées sont décalées par rapport à la fin de la zone de contrôle, dont la longueur est indiquée dans l'enregistrement d'en-tête.

Tableau 32. Zone de données d'enregistrement pour un enregistrement C

Position de l'octet	Description
01	Blanc
De 02 à la fin	Valeur ou ensemble de valeurs qui font l'objet d'une continuation

Exportation d'un format

L'objet format contient toutes les informations spécifiées dans tous les écrans de format QMF. Lorsque vous exportez un format, QMF convertit en format codé tous les écrans de format dont les valeurs sont différentes des valeurs par défaut.

Les écrans suivants sont en format codé, uniquement si vous avez modifié l'écran :

- FORM.BREAK n , où n = un nombre de 1 à 6
- FORM.CALC
- FORM.CONDITIONS
- Tous les écrans de variante supérieurs à 1 pour FORM.DETAIL

L'élimination des écrans inutilisés du format externalisé vous permet d'économiser de l'espace sur votre système.

Création d'un format par défaut pour afficher des exemples de résultats d'exportation

Pour créer un format par défaut, exécutez une requête qui crée un rapport vide, telle que la requête présentée ici :

```
SELECT POSTE
FROM Q.PERS
WHERE NOM='NO_NAME'
```

Lorsque QMF affiche le rapport, saisissez EXPORT FORM TO DEFAULT (y compris le paramètre QUEUETYPE=xx dans CICS).

Présentation du format exporté

Votre ensemble de données ou file d'attente CICS, qui porte le nom DEFAULT, contient les informations présentées dans cet exemple de format exporté :

```
H QMF 19 F 04 E V W E R 01 03 13/01/15 16:20

T 1110 001 011 1112 007 1113 040 1114 007 1115 006 1116 005 1117 005 1118 003 1119 008 1120 008
  1122 006 1121 050
R CHAR JOB 2 5 C 1 DEFAULT
  DEFAULT NO
V 1201 001 0
V 1202 001 2
T 1210 001 003 1212 004 1213 006 1214 055
R 1 CENTER
V 1301 001 2
V 1302 001 0
T 1310 001 003 1312 004 1313 006 1314 055
R 1 CENTER
V 1401 002 NO
V 1402 001 1
V 1403 001 0
T 1410 001 003 1412 004 1413 006 1414 055
R 1 RIGHT
V 1501 001 1
V 1502 003 YES
V 1503 003 YES
V 1504 003 YES
V 1505 003 YES
V 1506 003 YES
V 1507 003 YES
V 1508 003 YES
V 1509 003 YES
V 1510 003 YES
V 1511 004 NONE
V 1512 002 NO
V 1513 007 DEFAULT
V 1514 002 NO
V 1515 004 NONE
V 2790 001 1
```

```

V 2791 003 YES
V 2805 003 YES
T 2810 001 003 2812 004 2813 006 2814 055
R 1 LEFT
V 2901 002 NO
V 2902 001 1
V 2904 001 0
V 2906 002 NO
V 2907 002 NO
T 2910 001 003 2912 004 2913 006 2914 055
R 1 LEFT
V 3080 001 1
V 3101 002 NO
V 3102 002 NO
V 3103 001 0
V 3104 001 0
T 3110 001 003 3112 004 3113 006 3114 055
R 1 LEFT
V 3201 002 NO
V 3202 001 1
V 3203 001 0
V 3204 001 1T 3210 001 003 3212 004 3213 006 3214 055
R 1 RIGHT
V 3080 001 2
V 3101 002 NO
V 3102 002 NO
V 3103 001 0
V 3104 001 0
T 3110 001 003 3112 004 3113 006 3114 055
R 1 LEFT
V 3201 002 NO
V 3202 001 1
V 3203 001 0
V 3204 001 1
T 3210 001 003 3212 004 3213 006 3214 055
R 1 RIGHT
F

```

Vous pouvez importer votre ensemble de données ou file d'attente CICS par défaut chaque fois que vous vous connectez à l'aide de la commande IMPORT FORM FROM DEFAULT (y compris le paramètre QUEUETYPE=xx dans CICS) dans votre procédure de lancement.

Interprétation de l'enregistrement d'en-tête dans la file d'attente ou l'ensemble de données exporté

L'exemple suivant présente un enregistrement d'en-tête pour un format QMF :

```
H QMF 19 F 04 E V W E R 01 03 13/01/15 16:20
```

Le tableau ci-dessous explique l'exemple.

Tableau 33. Exemple d'enregistrement d'en-tête de format

Valeur provenant de l'exemple	Description
H QMF 19 F	Cet enregistrement est un enregistrement d'en-tête de format QMF pour Version 12.1.
04	La structure du format est au niveau d'objet 4.
F	Le format est codé (format utilisé pour les rapports, les requêtes créées de façon interactive et les formats exportés).
V	Le format exporté ne contient pas d'erreurs ni d'avertissements.
T	Le fichier contient la totalité du format.
F	La langue nationale en cours d'utilisation lors de l'exportation de l'objet est l'anglais.

Tableau 33. Exemple d'enregistrement d'en-tête de format (suite)

Valeur provenant de l'exemple	Description
R	L'objet figurant dans la mémoire temporaire est remplacé lors de l'importation.
01	La longueur de la zone de contrôle est égale à 1 octet.
03	La longueur des zones de longueur entière est égale à 3 octets.
13/01/15	L'horodatage indique le 15 janvier 2013.
16:20	L'horodatage indique 4h20 de l'après-midi.

Lorsque vous exportez un format à partir d'une session qui n'est pas en anglais, vous pouvez exporter le format dans la langue de la session en cours ou en anglais. L'identificateur de langue nationale dans l'enregistrement H risque donc de ne pas refléter la langue de la session à partir de laquelle vous avez exporté le format.

Interprétation des enregistrements du format exporté

La section «Présentation du format exporté», à la page 113 montre un exemple de format exporté. Le format exporté contient des enregistrements V, T et R dont les codes associés ont des significations spéciales pour vous aider à interpréter le résultat exporté. Ce tableau explique chaque zone et code dans le format exporté.

La zone 080, qui est un enregistrement V, agit en tant que «déclencheur» pour les écrans de rupture qui la suivent. Cet enregistrement apparaît une fois pour chaque écran de rupture dans votre format. La valeur de la zone correspond au numéro de l'écran de rupture que décrivent les zones qui suivent la zone 3080.

Tableau 34. Numéros de table et de zone pour un objet FORM exporté

Numéro de table ou de zone	Type d'enregistrement	Description	Ecran de format
1110	T	Table d'en-tête de colonne	FORM.COLUMNS

Tableau 34. Numéros de table et de zone pour un objet FORM exporté (suite)

Numéro de table ou de zone	Type d'enregistrement	Description	Ecran de format
1112	R	<p>Type de données de colonne</p> <p>Le type de données de colonne ne s'affiche pas sur les écrans de format mais est associé au format dans son format externe.</p> <p>Le type de données de colonne n'est pas requis lors de l'importation d'un format. S'il manque pendant l'importation, CICS fournit les informations de type de données par défaut à partir des codes d'édition. (Pour plus d'informations, voir «Importation d'un objet format», à la page 120.)</p> <p>Au cours de l'exportation, le type de données de colonne fourni par QMF se base sur le code d'édition spécifié. Pour les codes d'édition U, V, M ou non valides, QMF indique le mot clé de type de données UNKNOWN. Le tableau 35, à la page 119 affiche les mots clés de type de données utilisés par QMF.</p>	FORM.COLUMNS
1113	R	Titre de la colonne	FORM.COLUMNS
1114	R	Code usage de la colonne	FORM.COLUMNS
1115	R	Retrait de colonne	FORM.COLUMNS
1116	R	Largeur des colonnes	FORM.COLUMNS
1117	R	Code d'édition de colonne	FORM.COLUMNS
1118	R	Séquence de colonnes	FORM.COLUMNS
1119	R	Alignement d'en-tête de colonne	FORM.COLUMNS
1120	R	Alignement de données de colonne	FORM.COLUMNS
1121	R	Définition de colonne	FORM.COLUMNS
1122	R	Passer les valeurs NULL dans la définition de colonne	FORM.COLUMNS
1180	T	Table des calculs récapitulatifs	FORM.CALC
1182	R	Numéro d'identification du calcul	FORM.CALC
1183	R	Expression du calcul récapitulatif	FORM.CALC
1184	R	Largeur du calcul récapitulatif	FORM.CALC
1185	R	Code d'édition du calcul récapitulatif	FORM.CALC
1186	R	Passer les valeurs NULL lors du calcul	FORM.CALC

Tableau 34. Numéros de table et de zone pour un objet FORM exporté (suite)

Numéro de table ou de zone	Type d'enregistrement	Description	Ecran de format
1201	V	Lignes vides avant l'en-tête	FORM.PAGE
1202	V	Lignes vides après l'en-tête	FORM.PAGE
1210	T	Table d'en-tête de page	FORM.PAGE
1212	R	Numéro de ligne d'en-tête de page	FORM.PAGE
1213	R	Alignement d'en-tête de page	FORM.PAGE
1214	R	Texte d'en-tête de page	FORM.PAGE
1301	V	Lignes vides avant le bas de page	FORM.PAGE
1302	V	Lignes vides après le bas de page	FORM.PAGE
1310	T	Table de bas de page	FORM.PAGE
1312	R	Numéro de ligne de bas de page	FORM.PAGE
1313	R	Alignement de bas de page	FORM.PAGE
1314	R	Texte de bas de page	FORM.PAGE
1401	V	Texte final sur une nouvelle page	FORM.FINAL
1402	V	Numéro de ligne du résumé final	FORM.FINAL
1403	V	Lignes vides avant le texte final	FORM.FINAL
1410	T	Table de texte final	FORM.FINAL
1412	R	Numéro de ligne du texte final	FORM.FINAL
1413	R	Alignement du texte final	FORM.FINAL
1414	R	Texte final	FORM.FINAL
1501	V	Espacement entre les lignes de détail	FORM.OPTIONS
1502	V	Mise en évidence des colonnes de rupture	FORM.OPTIONS
1503	V	Texte de rupture par défaut	FORM.OPTIONS
1504	V	Nom de fonction dans l'en-tête de colonne pour le regroupement	FORM.OPTIONS
1505	V	Conservation sur une même page des lignes d'une colonne	FORM.OPTIONS
1506	V	Colonne de récapitulatif transversal	FORM.OPTIONS
1507	V	Séparateurs d'en-tête de colonne	FORM.OPTIONS
1508	V	Séparateurs de récapitulatif de rupture	FORM.OPTIONS
1509	V	Séparateurs d'en-tête transversal	FORM.OPTIONS
1510	V	Séparateurs de résumé final	FORM.OPTIONS
1511	V	Largeur des lignes de rapport renvoyées à la ligne	FORM.OPTIONS
1512	V	Renumérotation de page aux ruptures	FORM.OPTIONS

Tableau 34. Numéros de table et de zone pour un objet FORM exporté (suite)

Numéro de table ou de zone	Type d'enregistrement	Description	Ecran de format
1513	V	Largeur de la rupture ou du texte final	FORM.OPTIONS
1514	V	Réorganisation des colonnes	FORM.OPTION
1515	V	Colonnes fixes	FORM.OPTIONS
2790	V	Numéro de variante de détail	FORM.DETAIL
2791	V	Sélection de variante de détail	FORM.DETAIL
2805	V	Inclure en-têtes de colonnes	FORM.DETAIL
2810	T	Table d'en-tête de détail	FORM.DETAIL
2812	R	Ligne de texte d'en-tête de détail	FORM.DETAIL
2813	R	Alignement d'en-tête de détail	FORM.DETAIL
2814	R	Texte d'en-tête de détail	FORM.DETAIL
2901	V	Nouvelle page pour le texte de détail	FORM.DETAIL
2902	V	Numéro de ligne des données de colonne	FORM.DETAIL
2904	V	Nombre de lignes à ignorer après le texte de détail	FORM.DETAIL
2906	V	Répéter l'en-tête de détail	FORM.DETAIL
2907	V	Nombre de lignes de texte de détail à conserver ensemble	FORM.DETAIL
2910	T	Table de texte de détail	FORM.DETAIL
2912	R	Numéro de ligne du texte de détail	FORM.DETAIL
2913	R	Alignement du texte de détail	FORM.DETAIL
2914	R	Texte de détail	FORM.DETAIL
3080	V	Numéro de l'écran de rupture	FORM.BREAKn
3101	V	Nouvelle page pour l'en-tête de rupture	FORM.BREAKn
3102	V	Répéter l'en-tête de rupture	FORM.BREAKn
3103	V	Nombre de lignes à ignorer avant l'en-tête de rupture	FORM.BREAKn
3104	V	Nombre de lignes à ignorer après l'en-tête de rupture	FORM.BREAKn
3110	T	Table de texte d'en-tête de rupture	FORM.BREAKn
3112	R	Numéro de ligne d'en-tête de rupture	FORM.BREAKn
3113	R	Alignement d'en-tête de rupture	FORM.BREAKn
3114	R	Texte d'en-tête de rupture	FORM.BREAKn
3201	V	Nouvelle page pour le texte de rupture	FORM.BREAKn
3202	V	Ligne récapitulative de texte de rupture	FORM.BREAKn

Tableau 34. Numéros de table et de zone pour un objet FORM exporté (suite)

Numéro de table ou de zone	Type d'enregistrement	Description	Ecran de format
3203	V	Nombre de lignes à ignorer avant le texte de rupture	FORM.BREAKn
3204	V	Nombre de lignes à ignorer après le texte de rupture	FORM.BREAKn
3210	T	Table de texte de rupture	FORM.BREAKn
3212	R	Ligne de texte de rupture	FORM.BREAKn
3213	R	Alignement de texte de rupture	FORM.BREAKn
3214	R	Texte de rupture	FORM.BREAKn
3310	T	Table de conditions	FORM.CONDITIONS
3312	R	Numéro d'identification de la condition	FORM.CONDITIONS
3313	R	Expression conditionnelle	FORM.CONDITIONS
3314	R	Passer les valeurs NULL sur l'écran des conditions	FORM.CONDITIONS

Le tableau suivant présente les mots clés de type de données générés par QMF pour les codes d'édition spécifiés sur le format. Dans ce tableau, *x* représente le nombre de décimales à afficher, où *x* correspond à un entier compris entre 0 et 99.

Tableau 35. Mots clés de type de données générés pour les codes d'édition spécifiés sur les écrans de format QMF

Code d'édition spécifié	Mot clé de type de données
C, CW, CT, CD _x	CHAR
B, BW, X, XW	BINARY
G, GW	GRAPHIC
E, D, I, J, K, L, P, EZ, DZ, IZ, JZ, KZ, LZ, PZ, DZC, D _x , I _x , J _x , K _x , L _x , P _x	NUMERIC
Tout code d'édition commençant par les caractères TD	DATE
Tout code d'édition commençant par les caractères TT	TIME
TSI	TIMEST
TSZ	TSTMPTZ
M	UNKNOWN
U, V	UNKNOWN
Codes d'édition saisis non valides	UNKNOWN

Lors de l'exportation d'un format, QMF exporte uniquement les écrans de variante dont les valeurs sont différentes des valeurs par défaut. Le nombre total de variantes dans le format externe peut donc être inférieur au nombre affiché dans l'indicateur du nombre de variantes à l'écran. QMF peut modifier les numéros de variante individuels pour remplacer les variantes dans une séquence continue.

Importation d'un objet format

Lors de l'importation d'un format, ces zones doivent être en majuscules :

- Identificateur d'enregistrement pour tous les enregistrements
- Zones suivantes dans l'enregistrement d'en-tête :
 - Identificateur de produit (QMF)
 - Type d'objet (F)
 - Format de l'objet (E)
 - Action sur l'objet (R)
- Valeurs de type de données (type de données numérique, caractère, graphique ou inconnu) dans les enregistrements R pour la table COLUMNS
Si votre site prend en charge les types de données date/heure, les valeurs de type de données DATE, TIME, TIMEST et TSTMPTZ doivent également être en majuscules.
- Tous les mots clés de format et variables de substitution utilisés dans les écrans de format

Lors de l'importation d'un format, toutes les entrées du format restent intactes. Si un mot clé de format est en minuscules, l'indicateur d'erreur dans l'écran de format est activé. Pour corriger l'erreur, vous devez entrer d'autres données dans la zone. Si la valeur de type de données n'est pas en majuscules, une erreur se produit et l'importation prend fin.

L'enregistrement T de la table COLUMNS (numéro de zone 1110) doit suivre immédiatement l'enregistrement d'en-tête. L'enregistrement T doit également inclure un comptage numérique du nombre de lignes au format codé (un comptage de ligne * n'est pas autorisé).

Si la totalité de la table COLUMNS est lue, les valeurs par défaut sont affectées aux zones non spécifiées et le format est affiché.

- Ecrans de variante
La zone du numéro de variante (numéro de zone 2790) détermine quel écran de variante est mis à jour par toutes les informations d'écran de variante qui suivent la zone. Cet enregistrement V doit précéder tous les autres enregistrements V, T et R pour un écran de variante.
Si une valeur pour une variante donnée apparaît plusieurs fois au format codé, les dernières valeurs remplacent celles d'origine. Le nombre de variantes dans le format est égal au nombre de variantes le plus élevé dans le format. Aucun ordre n'est requis pour les numéros de variantes lors de l'importation.
- Formats traduits
Lorsque vous importez un format en anglais dans une session dans une autre langue et que l'identificateur de langue nationale dans l'enregistrement H est E, QMF traduit les mots réservés. QMF traduit les mots réservés dans la langue de votre session en cours. Les exemples de mots réservés sont les valeurs qui figurent dans la colonne USAGE de FORM.COLUMNS
- Omission du type de données, du code d'édition et de la largeur dans un format importé
Dans la table COLUMNS, le type de données (numéro de zone 1112), le code d'édition (numéro de zone 1117) et la largeur (numéro de zone 1116) peuvent éventuellement être omis lorsque les règles suivantes sont respectées :

- Le code d'édition doit être inclus si le type de données et la largeur sont omis. En fonction du code d'édition spécifié, QMF insère les valeurs par défaut appropriées pour le type de données et la largeur.
- Le type de données doit être inclus si le code d'édition et la largeur sont omis. QMF fournit les valeurs par défaut pour le code d'édition et la largeur.
- La largeur doit être accompagnée du type de données ou du code d'édition.

Le tableau ci-dessous contient des informations sur les valeurs pour la zone contenant le type de données de la colonne. Outre les valeurs de type de données affichées ici, le mot clé de type de données UNKNOWN est utilisé par QMF en réponse à un code d'édition U, V ou non valide.

Tableau 36. Valeurs pour la zone contenant le type de données de la colonne

Type de données tel qu'il apparaît au format externalisé	Code en notation décimale	Type de données	Signification
DATE	384	DATE	Date
TIME	388	TIME	Heure
TIMEST	392	TIMESTAMP	Horodatage
NUMERIC	496 500 492 484 480 996	INTEGER SMALLINT BIGINT DECIMAL FLOAT DECFLOAT	Entier Petit entier Grand entier Décimal Virgule flottante Virgule décimale flottante
CHAR	448 452 456 904	VARCHAR CHAR LONG VARCHAR ROWID	Caractère de longueur variable Caractère de longueur fixe Caractère long de longueur variable Identificateur de ligne
GRAPHIC	464 468 472	VARGRAPHIC GRAPHIC LONG VARGRAPHIC	Graphique de longueur variable Graphique de longueur fixe Graphique long de longueur variable
BINARY	908 912	VARBINARY BINARY	Binaire de longueur variable Binaire de longueur fixe
TSTMPTZ	2448	TIMESTAMP WITH TIME ZONE	Horodatage avec fuseau horaire

- Détection d'erreurs lors de l'importation

Si QMF détecte une erreur de mise en forme du fichier de format lors de l'importation, la fonction d'importation prend fin. QMF envoie un message décrivant l'erreur et son emplacement dans le fichier.

Si une erreur est détectée dans l'enregistrement d'en-tête et qu'un format existe dans la zone de stockage temporaire, il s'affiche. Si le format a été importé correctement, QMF affiche l'écran de format.

Si une erreur est détectée après la lecture de l'enregistrement d'en-tête, le format existant dans la zone de stockage temporaire est effacé et l'écran d'accueil s'affiche. Toutefois, si l'objet données existe, QMF génère un format par défaut pour les données, mais ne l'affiche pas.

Certaines erreurs mineures détectées par QMF n'interrompent pas l'importation. Dans certains cas, QMF émet un message d'avertissement et applique les valeurs par défaut, si nécessaire. Voici quelques exemples :

- Enregistrements V
 - Zones de longueur égale à zéro
 - La longueur spécifiée pour la zone ne correspond pas à la longueur des données fournies.
- Enregistrements T
 - Longueur de colonne égale à zéro
 - Le nombre de colonnes spécifié ne correspond pas aux paires numéro de zone/longueur suivantes.

Vous pouvez répondre aux erreurs et avertissements comme suit :

- Corrigez un problème à la fois.
- Définissez l'option TRACE du profil sur L2 (à l'aide de la commande SET PROFILE (TRACE=L2) et exécutez la commande IMPORT FORM. L'option de traçage L2 trace les messages et les commandes au niveau de détail le plus élevé. Cette option vous permet de visualiser l'ensemble du texte de message relatif à la commande IMPORT.

La commande suivante affiche le texte du message associé à un code message donné :

```
HELP DSQnnnnn
```

où *nnnnn* est un code message unique à 5 caractères.

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Considérations relatives aux objets format QMF dans des applications

Certaines astuces peuvent vous aider à créer et utiliser un format QMF dans une application.

Lorsque vous utilisez un format QMF dans une application, tenez compte des points suivants :

Création d'un ensemble de données ou file d'attente CICS de format en dehors de QMF

Si vous créez un format en dehors de QMF (sans utiliser EXPORT FORM), il n'est pas nécessaire de disposer d'un objet format complet pour l'importer dans QMF. Vous n'avez besoin que de l'enregistrement d'en-tête (H) suivi des enregistrements T et R de la table COLUMNS. Les valeurs par défaut sont appliquées au reste du format lors de son importation.

Lorsque vous créez votre propre ensemble de données ou file d'attente CICS de format, il ne doit pas être exactement identique à l'ensemble de données ou à la file d'attente que vous obtenez si vous utilisez EXPORT FORM. Par exemple, lorsque QMF exporte un format, toutes les valeurs de données dans un enregistrement de valeur (V) sont précédées d'une

longueur. Toutefois, vous pouvez utiliser un astérisque (*) pour indiquer que la valeur de données est délimitée par la fin de l'enregistrement lors de l'importation d'un format.

QMF conserve les lignes superflues si le nombre d'enregistrements R dans un format importé est inférieur au nombre de lignes par défaut qu'il a alloué à la zone associée dans le format par défaut.

Vérification de niveau d'objet dans l'enregistrement d'en-tête

Le niveau d'objet dans l'enregistrement d'en-tête d'un ensemble de données ou d'une file d'attente contenant un format indique le niveau de la structure de format au moment de la génération du format. (Le niveau d'objet est indiqué aux octets 12 et 13 de l'enregistrement d'en-tête.) Vérifiez que votre application interprète correctement le contenu de l'ensemble de données ou de la file d'attente de données contenant le format. Assurez-vous que le niveau d'objet représente le format sur lequel se base votre application.

Utilisation des enregistrements de données d'application

Les enregistrements de données d'application peuvent s'avérer utiles dans votre programme d'application. Utilisez-les pour inclure vos propres commentaires dans un ensemble de données ou file d'attente CICS pour un objet format. Vous pouvez placer les enregistrements à n'importe quel endroit de l'ensemble de données ou file d'attente CICS à la suite de l'enregistrement d'en-tête. Lorsque QMF lit un enregistrement de ce type, il ignore toutes les données de l'enregistrement qui suivent le caractère *. Par conséquent, l'enregistrement n'a aucune incidence sur le processus d'importation.

Restrictions relatives à l'utilisation de formats dans CICS

REXX n'est pas disponible sous QMF for CICS. Les zones du format QMF qui s'appuient sur REXX ne fonctionnent pas si vous tentez d'exécuter le format dans l'environnement CICS. Ces zones incluent tout élément saisi sur les écrans FORM.CALC, FORM.CONDITIONS et dans la fenêtre de spécification de définition. Les utilisateurs de QMF for CICS n'ont pas accès aux calculs REXX, au formatage de lignes conditionnel et aux définitions de colonne.

Concepts associés:

«Importation de formats et de requêtes créées de façon interactive», à la page 140
Prenez connaissance des règles relatives à l'importation d'un format ou d'une requête créée de façon interactive.

Référence associée:

«Exportation d'un format», à la page 113

L'objet format contient toutes les informations spécifiées dans tous les écrans de format QMF. Lorsque vous exportez un format, QMF convertit en format codé tous les écrans de format dont les valeurs sont différentes des valeurs par défaut.

«Enregistrement de données d'application (*)», à la page 108

Les enregistrements de données d'application des rapports, requêtes créées de façon interactive ou formats exportés permettent aux programmes d'application d'associer leurs propres données à un objet dans le fichier externe. Les programmes d'application utilisent fréquemment ces enregistrements comme enregistrements de commentaire pour décrire plus précisément l'objet dans le fichier.

«Enregistrements d'en-tête», à la page 102

La plupart des enregistrements des rapports, requêtes créées de façon interactive ou formats exportés ont un format variable. Toutefois, les enregistrements d'en-tête ont un format fixe, même si l'ensemble de données ou la file d'attente de données

qui contient les enregistrements peut être de format variable.

Exportation d'un rapport standard

Lorsque QMF affiche un rapport, vous voyez le résultat de l'interaction entre le format et l'objet données dans la mémoire temporaire. Il n'existe pas d'objet rapport dans la mémoire temporaire. Lorsque vous exportez un rapport, QMF exporte en fait l'interaction d'un format et d'un objet données.

Un rapport ne peut pas être sauvegardé dans la base de données et un rapport exporté ne peut pas être réimporté dans QMF. Toutefois, vous pouvez utiliser les rapports exportés pour :

- Extraire des données du rapport et les utiliser dans une application
- Modifier l'apparence du rapport en vue de son impression ou son réaffichage par l'application

Exemple de rapport (avant l'exportation)

Cet exemple affiche un rapport en mode tableau avec une rupture de niveau 1.

Pour obtenir la liste des numéros de zone, voir «Interprétation de l'enregistrement d'en-tête de rapport dans la file d'attente de données ou l'ensemble de données exporté», à la page 126.

REPORT	LINE 1	POS 1	79
J & H SUPPLY COMPANY SALAIRE MOYEN (DEPTS 10, 15, 20) REPORT 17			
DEPT	POSTE	SALAIRE MOYEN	
-----	-----	-----	
10	DIR	20865.86	
	*	20865.86	
15	EMPL	12383.35	
	DIR	20659.80	
	VENTES	16502.83	
	*	15482.33	
20	EMPL	13878.68	
	DIR	18357.50	
	VENTES	18171.25	
	*	16071.53	
		=====	
		17473.24	
NOM DE SOCIETE REPORT 17			

Figure 21. Rapport QMF en mode tableau avant l'exportation

Format du rapport exporté

Voici le format du rapport exporté à partir de l'exemple de rapport en mode tableau.

```

H QMF 19 R 01 E V W E R 02 03 13/01/15 16:20
V 1001 006 PERIOD
V 1002 003 016
T 1010 003 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000008 000001 BREAK1
R C 000009 000011 000015 000001 GROUP
R L2 000016 000018 000027 000001 AVERAGE
L 110 10100000
L 110 10100000
L 110 10100000
L 110 10000000
L 110 10000000
L 170 10000000
L 170 10000000 DEPT POSTE SALAIRE
L 170 10010000 -----
L 181 11000000 10 DIR 20865.86
L 151 10010000 -----
L 151 11100000 * 20865.86
L 151 10000000
L 181 11000000 15 EMPL 12383.35
L 181 11000000 DIR 20659.80
L 181 11000000 VENTES 16502.83
L 151 10010000 -----
L 151 11100000 * 15482.33
L 151 10000000
L 181 11000000 20 EMPL 13878.67
L 181 11000000 DIR 18357.50
L 181 11000000 VENTES 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
L 120 10100000 NOM DE SOCIETE
F REPORT 17

```

Lors de l'exportation d'un rapport, QMF écrit la totalité du texte du rapport formaté ainsi que des informations supplémentaires pour interpréter le contenu du rapport.

L'enregistrement d'en-tête est le premier enregistrement du fichier exporté. Il est suivi des enregistrements V, T et R appropriés. S'il s'agit d'un rapport de style transversal, il comporte un autre groupe d'enregistrements V, T et R qui suit le premier.

Outre les enregistrements H, V, T, R et E, les rapports exportés requièrent également deux types d'enregistrements supplémentaires :

- Enregistrements de ligne de rapport ou L
- Enregistrements de continuation de données ou C

Ces deux enregistrements suivent le dernier groupe d'enregistrements V, T et R.

Si vous ne voulez utiliser que les données formatées du rapport dans votre application, QMF peut envoyer une sortie imprimée vers un ensemble de données

ou file d'attente CICS. Cet ensemble de données ou file d'attente CICS contient uniquement les données formatées sans aucune information de présentation.

Interprétation de l'enregistrement d'en-tête de rapport dans la file d'attente de données ou l'ensemble de données exporté

L'exemple suivant présente un enregistrement d'en-tête pour un rapport QMF :

H QMF 19 R 01 E V W E R 02 03 13/01/15 16:20

Le tableau suivant explique cet exemple.

Tableau 37. Exemple d'enregistrement d'en-tête pour un rapport

Valeur provenant de l'exemple	Description
H QMF 19 R	Cet enregistrement est un enregistrement d'en-tête de rapport QMF pour Version 12.1.
01	La structure du rapport est au niveau d'objet 1.
F	Le format est codé (format utilisé pour les rapports, les requêtes créées de façon interactive et les formats exportés).
V	Le rapport exporté ne contient pas d'erreurs ni d'avertissements.
T	Le fichier contient la totalité du rapport.
F	L'anglais était la langue nationale en cours d'utilisation lors de l'exportation de l'objet.
R	Cet indicateur est ignoré.
02	La longueur de la zone de contrôle est égale à 2 octets.
03	La longueur des zones de longueur entière est égale à 3 octets.
13/01/15	L'horodatage indique le 15 janvier 2013.
16:20	L'horodatage indique 4h20 de l'après-midi.

Interprétation des enregistrements du rapport exporté

Ce tableau affiche les numéros de table pour les enregistrements T et les numéros de zone pour les enregistrements V du rapport exporté affiché dans la section «Format du rapport exporté», à la page 125.

Tableau 38. Numéros de table et de zone pour un rapport exporté

Numéro de table ou de zone	Type d'enregistrement	Description
1001	V	Option de profil DECIMAL
1002	V	Longueur de la zone de contrôle d'enregistrement L + zone fixe
1010	T	Table de rapport formatée
		Pour chaque colonne de données formatées dans le rapport :
1012	T	Pour tous les codes usage à l'exception d'OMIT
1013	T	Code d'édition à l'aide duquel les données sont formatées

Tableau 38. Numéros de table et de zone pour un rapport exporté (suite)

Numéro de table ou de zone	Type d'enregistrement	Description
1014	T	Position de départ pour la zone contenant les données formatées (y compris la zone Interv)
1015	T	Position de départ pour la zone contenant les données formatées (sauf la zone Interv)
1016	T	Position de fin pour la zone contenant les données formatées (y compris la zone Interv)
1017	T	Numéro de ligne de rapport physique relatif au sein de la ligne de rapport logique dans laquelle la valeur de colonne formatée apparaît

Lorsque vous utilisez des numéros de table et de zone dans un rapport exporté, notez les points suivants :

- La position 1 de la ligne de rapport suit immédiatement la zone fixe d'enregistrement L.
- Les enregistrements R pour les lignes de texte dans chaque en-tête (PAGE ou BREAK) ou bas de page (PAGE, BREAK ou FINAL) de rapport sont uniquement écrits jusqu'à la dernière ligne (comprise) qui contient les modifications apportées aux valeurs par défaut du format.
Au moins un enregistrement R est écrit pour chaque en-tête ou bas de page, même lorsque les zones pour un en-tête ou un bas de page donné contiennent toutes leurs valeurs d'origine.
- Des enregistrements de continuation sont écrits pour l'objet rapport pour éviter que la longueur d'enregistrement maximale soit dépassée.

Référence associée:

«Exportation d'un rapport de style transversal», à la page 129

Les rapports de style transversaux exportés incluent des zones introuvables dans les rapports exportés standard.

«Enregistrements de continuation de données (C)», à la page 112

Un enregistrement C dans une requête créée de façon interactive, un rapport ou un format exporté permet de continuer une valeur ou un ensemble de valeurs sur plusieurs enregistrements. Il suit immédiatement l'enregistrement auquel s'applique la continuation. Le format d'un enregistrement C correspond au format de l'enregistrement à l'origine de la continuation. QMF utilise les enregistrements C uniquement pour continuer des enregistrements L.

«Enregistrements de ligne de rapport (L)», à la page 109

Chaque ligne formatée dans un rapport est décrite par un enregistrement L. Il existe un enregistrement L pour chaque ligne du rapport.

Exportation d'un rapport au format HTML

Lorsque vous exportez un rapport au format HTML, QMF place les balises HTML nécessaires avant et après le corps du rapport. Vous pouvez ensuite placer le rapport sur un serveur Web et l'afficher dans un navigateur Web compatible HTML.

Cet exemple illustre le codage HTML que QMF place autour du rapport. Chacun de ces ensembles de balises est composé d'une balise de début et d'une balise de

fin. Les balises de fin commencent par une barre oblique (/) et toutes les balises sont insérées entre des signes supérieur et inférieur.

Pour obtenir une description complète de ces balises, consultez votre documentation HTML.

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

```

                J & H SUPPLY COMPANY
SALAIRE MOYEN (DEPTS 10, 15, 20)
REPORT 17
```

```

                SALAIRE
DEPT  POSTE      MOYEN
-----
10  DIR          20865.86
                * 20865.86

15  EMPL          12383.53
    DIR          20659.80
    VENTES       16052.83
                * 15482.33

20  EMPL          13878.67
    DIR          18357.50
    VENTES       18171.25
                * 16071.52
                =====
                17473.52
```

```

                NOM DE SOCIETE
                REPORT 17
```

```
</PRE>
</BODY>
</HTML>
```

Le tableau ci-dessous explique brièvement ce codage HTML :

Tableau 39. Balises HTML utilisées dans les rapports HTML exportés

Ensemble de balises	Description
<HTML></HTML>	Ces balises définissent le fichier en tant que document HTML.
<HEAD></HEAD>	Ces balises marquent les limites de l'en-tête du document.
<TITLE></TITLE>	QMF insère le mot "Rapport" entre ces balises. Le contenu situé entre ces balises est inclus dans le titre du document HTML. Le placement du titre dépend du navigateur et de la plateforme. Ces balises sont placées à l'intérieur de l'en-tête.
<BODY></BODY>	Ces balises suivent l'en-tête et contiennent le corps du document. La sortie de rapport est placée dans le corps du document.

Tableau 39. Balises HTML utilisées dans les rapports HTML exportés (suite)

Ensemble de balises	Description
<PRE></PRE>	Le contenu placé entre ces balises s'affiche en l'état. Aucun formatage HTML n'est effectué entre les balises. QMF place la sortie de rapport entre ces balises dans le corps du document HTML.

Exportation d'un rapport sans informations de contrôle

Lorsque vous importez un rapport, QMF place les informations de contrôle autour du rapport par défaut.

Par exemple, un rapport exporté peut être au format suivant :

```
H QMF 17 R 01 E V W E R 02 03 14/03/05 11:07
V 1001 006 PERIOD
V 1002 003 016
T 1010 005 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000010 000001
R C 000011 000013 000026 000001
R L 000027 000029 000035 000001
R C 000036 000038 000047 000001
R C 000048 000050 000062 000001
L 110 10000000
L 110 10000000
L 170 10000000 DEPT_NO NOMDEPT RESPONSABLE DIVISION LOCATION
L 170 10010000 -----
L 180 11000000 10 HEAD OFFICE 160 CORPORATE NEW YORK
L 180 11000000 15 NEW ENGLAND 50 EASTERN BOSTON
L 180 11000000 20 MID ATLANTIC 10 EASTERN WASHINGTON
L 180 11000000 38 SOUTH ATLANTIC 30 EASTERN ATLANTA
L 180 11000000 42 GREAT LAKES 100 MIDWEST CHICAGO
L 180 11000000 51 PLAINS 140 MIDWEST DALLAS
L 180 11000000 66 PACIFIC 270 WESTERN SAN FRANCISCO
L 180 11000000 84 MOUNTAIN 290 WESTERN DENVER
L 120 10000000
L 120 10000000
E
```

Figure 22. Exemple de rapport exporté avec informations de contrôle.

Si vous spécifiez l'option DATAFORMAT=TEXT dans la commande EXPORT REPORT, vous pouvez exporter les rapports sans informations de contrôle, comme dans l'exemple suivant.

```
DEPT_NO NOMDEPT RESPONSABLE DIVISION LOCATION
-----
10 HEAD OFFICE 160 CORPORATE NEW YORK
15 NEW ENGLAND 50 EASTERN BOSTON
20 MID ATLANTIC 10 EASTERN WASHINGTON
38 SOUTH ATLANTIC 30 EASTERN ATLANTA
42 GREAT LAKES 100 MIDWEST CHICAGO
51 PLAINS 140 MIDWEST DALLAS
66 PACIFIC 270 WESTERN SAN FRANCISCO
84 MOUNTAIN 290 WESTERN DENVER
```

Figure 23. Exemple de rapport exporté sans informations de contrôle

Exportation d'un rapport de style transversal

Les rapports de style transversaux exportés incluent des zones introuvables dans les rapports exportés standard.

Cet exemple présente un rapport de style transversal exporté.

```

REPORT                                     LINE 1   POS 1   79
                                     J & H SUPPLY COMPANY
                                     DEPT AVERAGE SALARIES
                                     REPORT 18 (ACROSS REPORT)

      <----- POSTE ----->
      <- EMPL --> <- DIR ---> <- VENTES --> <- TOTAL -->
      SALAIRE      SALAIRE      SALAIRE      SALAIRE
      MOYEN        MOYEN        MOYEN        MOYEN
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

                                     NOM DE SOCIETE
                                     REPORT 18
                                     PAGE 1
  
```

Le format codé suivant est le résultat de l'exportation de l'exemple de rapport de style transversal.

```

H QMF 19 R 01 E V W E R 02 03 13/01/15 16:20

V 1001 006 PERIOD
V 1002 003 016
T 1010 002 006 1013 005 1014 006 1015 006 1016 006 1017 006 1012 008
R L 000001 000003 000008 000001 GROUP
R L2 000003 000005 000014 000001 AVERAGE
V 2001 005 C
V 2002 003 001
V 2003 003 YES
T 2010 004 003 2012 006 2013 006 2014 006
R 000014 000018 000009
R 000029 000031 000023
R 000042 000046 000037
R 000056 000060 000051
L 110 10100000
L 110 10100000
L 110 10100000
L 110 10000000
L 110 10000000
L 170 10000000
L 170 11000000
L 170 10000000
L 170 10000000
L 170 10010000
L 181 11000000
L 181 11000000
L 181 11000000
L 181 11000000
L 190 10010000
L 190 11000000
ddL 120 10000000
L 120 10000000
L 120 10100000
L 120 10100000
L 120 10100000
F

                                     J & H SUPPLY COMPANY
                                     SALAIRE MOYEN DE DEPT
                                     REPORT 18 (ACROSS REPORT)

      <----- POSTE ----->
      <- EMPL --> <- DIR ---> <- VENTES --> <- TOTAL -->
      SALAIRE      SALAIRE      SALAIRE      SALAIRE
      MOYEN        MOYEN        MOYEN        MOYEN
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

                                     NOM DE SOCIETE
                                     REPORT 18
                                     PAGE 1
  
```


Le tableau 38, à la page 126 décrit les numéros de zone qui sont communs aux rapports standard et aux rapports de style transversaux. Le tableau ci-dessous présente les zones supplémentaires qui s'affichent dans les rapports de style transversaux exportés.

Tableau 40. Numéros de zone pour un rapport de style transversal exporté

Numéro de zone	Type d'enregistrement	Description
2001	V	Code d'édition à l'aide duquel la valeur transversale est formatée
2002	V	Nombre de lignes de données par groupe transversal
2003	V	Indique si la colonne récapitulative transversale existe
2010	T	Table de rapport transversal
		Pour chaque valeur transversale :
2012	T	Position de départ pour la valeur transversale formatée (la valeur transversale apparaît dans les lignes d'en-tête de colonne)
2013	T	Position de fin pour la valeur transversale formatée
2014	T	Position de départ pour l'ensemble de colonnes du rapport associé à cette valeur transversale, y compris la zone Interv précédente

Pour les colonnes agrégées dans un rapport transversal, les zones 1014, 1015 et 1016 décrivent les positions de départ et de fin relatives de la zone au sein de l'ensemble de colonnes agrégées d'une valeur transversale. (Voir la zone 2014 du tableau.)

Exportation d'une requête créée de façon interactive

Un objet requête créée de façon interactive exporté contient les informations affichées dans la zone d'écho de l'écran principal Requête créée de façon interactive.

Exemple de requête (avant l'exportation)

Un ensemble de données ou une file d'attente de données qui contient une requête créée de façon interactive exportée peut être importé dans QMF de deux façons. L'ensemble de données ou la file d'attente de données peut être soit importé dans la zone de stockage temporaire QMF, soit directement dans la base de données. Lorsque vous importez une requête créée de façon interactive, QMF vérifie si la requête entrante est cohérente avec les données figurant dans la base de données. Par exemple, si la requête créée de façon interactive qui est importée comporte les colonnes A, B et C dans la table XYZ, QMF vérifie que la table XYZ avec les colonnes A, B et C existe dans la base de données.

Cet exemple présente un échantillon de texte d'écho qui apparaît sur l'écran principal Requête créée de façon interactive avant l'exportation.

```

Tables :
  Q.PERS(A)
  Q.DEPT(B)
  Q.PERS(C)

Jointure de tables :
  A.DEPT And B.DEPT_NO
  And A.ID And C.ID

Colonnes :
  A.ID
  A.DEPT
  A.POSTE
  A.SALAIRE
  DEPT_NO
  C.SALAIRE
  C.SALAIRE+A.COMM

Conditions de ligne :
  If A.SALAIRE Is Greater Than 10000
  And A.DEPT Is Equal To 84 or 96

Tri :
  Décroissant par C.SALAIRE+A.COMM

Lignes en double :
  Conserver les lignes en double

```

Format de la requête exportée

Cet exemple présente le format de la requête créée de façon interactive exportée.

```

H QMF 19 T 01 E V W E R 01 03 13/01/15 16:20
T 1110 003 002 1112 001 1113 050
R A Q.PERS
R B Q.DEPT
R C Q.PERS
T 1150 002 002 1152 020 1153 020
R A.DEPT          B.DEPT_NO
R A.ID            C.ID
T 1210 007 002 1212 001 1213 255
R C A.ID
R C A.DEPT
R C A.POSTE
R C A.SALAIRE
R C B.DEPT_NO
R C C.SALAIRE
R C C.SALAIRE+A.COMM
T 1310 009 003 1312 001 1313 008 1314 255
R 1 C          A.SALAIRE
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.SALAIRE+A.COMM
V 1501 001 K
F

```

Interprétation de l'enregistrement d'en-tête dans la file d'attente ou l'ensemble de données exporté

Le tableau suivant indique la signification de cet enregistrement d'en-tête dans la requête créée de façon interactive exportée qui s'affiche dans la section «Format de la requête exportée», à la page 132.

H QMF 19 T 01 E V W E R 01 03 13/01/15 16:20

Tableau 41. Exemple d'enregistrement d'en-tête de requête créée de façon interactive

Valeur provenant de l'exemple	Description
H QMF 19 T	Cet enregistrement d'en-tête de requête créée de façon interactive spécifie QMF Version 12.1.
01	La structure de la requête créée de façon interactive est au niveau d'objet 1. Si l'objet requête exporté contient une spécification de période, le niveau d'objet 2 est spécifié à la place.
F	Le format est codé (format utilisé pour les rapports, les requêtes créées de façon interactive et les formats exportés).
V	La requête créée de façon interactive exportée ne contient pas d'erreurs ni d'avertissements.
T	Le fichier contient la totalité de la requête créée de façon interactive.
F	L'anglais était la langue nationale en cours d'utilisation lors de l'exportation de l'objet.
R	L'objet figurant dans la zone de stockage temporaire est remplacé lors de l'importation.
01	La longueur de la zone de contrôle est égale à 1 octet.
03	La longueur des zones de longueur entière est égale à 3 octets.
13/01/15	L'horodatage indique le 15 janvier 2013.
16:20	L'horodatage indique 4h20 de l'après-midi.

Pour consulter un exemple complet de format codé de requête créée de façon interactive, voir «Format de la requête exportée», à la page 132.

Interprétation des enregistrements de la requête créée de façon interactive exportée

Les définitions de table (numéro de zone 1110) sont toujours exportées. Les conditions de jointure (numéro de zone 1510) sont toujours exportées si plusieurs tables sont sélectionnées.

Pour qu'un fichier de requête créée de façon interactive soit importé, il doit contenir un enregistrement H suivi de l'enregistrement T de la table codée. Si aucune table n'est spécifiée, une requête vide est importée. Aucune condition de jointure n'est requise si plusieurs tables ne sont pas sélectionnées.

Tableau 42. Numéros de table et de zone pour un objet requête créée de façon interactive exporté

Type d'enregistrement	Numéro de table	Numéro de zone	Description de la zone
T	1110	-	<p>Table des définitions de table</p> <p>L'enregistrement T figurant dans cette section de la requête créée de façon interactive exportée dans «Format de la requête exportée», à la page 132 identifie cette section comme la partie contenant les noms de table impliqués dans la requête :</p> <p>T 1110 003 002 1112 001 1113 050</p> <p>'003' fait référence à 3 tables, tandis que '002' fait référence à 2 numéros de zone (1112 et 1113). Si l'objet requête créée de façon interactive exporté contient une spécification de période, la valeur '003' est utilisée pour indiquer 3 numéros de zone (1112, 1113 et 1114) à la place.</p> <p>Chaque enregistrement T est suivi d'enregistrements R. Dans cet exemple, les enregistrements R identifient les tables qui entrent en jeu dans la jointure des requêtes créées de façon interactive :</p> <p>R A Q.PERS R B Q.DEPT R C Q.PERS</p> <p>Cette portion du fichier exporté correspond à la partie suivante de la requête créée de façon interactive présentée dans «Exemple de requête (avant l'exportation)», à la page 131 :</p> <p>Tables :</p> <p>Q.PERS(A) Q.DEPT(B) Q.PERS(C)</p>
		1112	ID de table (les ID de table valides sont A à Z et #,\$,@)
		1113	Nom de table (280 caractères au maximum)
		1114	Spécification de période (560 caractères au maximum). Ce numéro de zone est inclus uniquement si l'objet requête exporté contient une spécification de période.

Tableau 42. Numéros de table et de zone pour un objet requête créée de façon interactive exporté (suite)

Type d'enregistrement	Numéro de table	Numéro de zone	Description de la zone
T	1150	-	<p>Table des conditions de jointure</p> <p>L'enregistrement T figurant dans cette section de la requête créée de façon interactive exportée dans «Format de la requête exportée», à la page 132 identifie cette section comme la partie contenant les conditions de jointure impliquées dans la requête. Chaque enregistrement T est suivi d'enregistrements R qui identifient les tables qui seront jointes :</p> <pre>T 1150 002 002 1152 020 1153 020 R A.DEPT B.DEPT_NO R A.ID C.ID</pre> <p>Cette portion de l'exemple de requête exportée correspond à la partie suivante de l'exemple de requête créée de façon interactive présentée dans «Exemple de requête (avant l'exportation)», à la page 131 :</p> <p>Jointure de tables :</p> <pre>A.DEPT And B.DEPT_NO And A.ID And C.ID</pre>
		1152	Nom de la colonne 1 : Longueur abrégée (22) Longueur développée (34)
		1153	Nom de la colonne 2 : Longueur abrégée (22) Longueur développée (34)

Tableau 42. Numéros de table et de zone pour un objet requête créée de façon interactive exporté (suite)

Type d'enregistrement	Numéro de table	Numéro de zone	Description de la zone
T	1210	-	<p>Table de colonnes</p> <p>L'enregistrement T figurant dans cette section de la requête créée de façon interactive exportée dans «Format de la requête exportée», à la page 132 identifie cette section comme la partie contenant les noms de colonne impliqués dans la requête. Chaque enregistrement T est suivi d'enregistrements R qui identifient les noms de colonne. La section se présente comme suit dans la requête exportée :</p> <pre>T 1210 007 002 1212 001 1213 255 R C A.ID R C A.DEPT R C A.POSTE R C A.SALAIRE R C B.DEPT_NO R C C.SALAIRE R C C.SALAIRE+A.COMM</pre> <p>Cette section de la requête exportée correspond à la section suivante de l'exemple de requête présenté dans «Exemple de requête (avant l'exportation)», à la page 131 :</p> <p>Colonnes :</p> <pre>A.ID A.DEPT A.POSTE A.SALAIRE DEPT_NO C.SALAIRE C.SALAIRE+A.COMM</pre>
		1212	<p>Type de colonne :</p> <ul style="list-style-type: none"> • C=colonne • E=expression • S=fonction récapitulative avec expression • F=fonction récapitulative avec uniquement une colonne
		1213	<p>Nom de colonne, expression ou fonction récapitulative : Longueur abrégée (255) Longueur développée (560)</p>

Tableau 42. Numéros de table et de zone pour un objet requête créée de façon interactive exporté (suite)

Type d'enregistrement	Numéro de table	Numéro de zone	Description de la zone
T	1310	-	<p>Conditions de sélection de lignes</p> <p>L'enregistrement T figurant dans cette section de la requête créée de façon interactive exportée dans «Format de la requête exportée», à la page 132 identifie cette section de la requête exportée comme la partie contenant les conditions de requête. Chaque enregistrement T est suivi d'enregistrements R qui caractérisent chaque condition. La section se présente comme suit dans la requête créée de façon interactive exportée :</p> <pre>T 1310 009 003 1312 001 1313 008 1314 255 R 1 C A.SALAIRE R 2 IS GT R 3 10000 R 4 I R 1 C A.DEPT R 2 IS EQ R 3 84 R 3 96 R 4 A</pre> <p>Cette section de la requête exportée correspond à la section suivante de la requête présentée dans «Exemple de requête (avant l'exportation)», à la page 131 :</p> <p>Conditions de ligne :</p> <pre>If A.SALAIRE Is Greater Than 10000 And A.DEPT Is Equal To 84 or 96</pre>
		1312	<p>Type de poste :</p> <ul style="list-style-type: none"> • 1 - partie gauche de l'opérateur • 2 - opérateur • 3 - partie droite de l'opérateur • 4 - connecteur
		1313	<p>Pour le type d'entrée '1', identifie le type de colonne :</p> <ul style="list-style-type: none"> • C=colonne • E=expression • S=fonction récapitulative • F=fonction récapitulative (seul le nom de colonne est spécifié)
			<p>Pour le type d'entrée '2', identifie l'instruction :</p> <ul style="list-style-type: none"> • IS pour 'est' (valeur par défaut) • ISN pour 'n'est pas'
			Pour le type d'entrée '3' (non utilisé)

Tableau 42. Numéros de table et de zone pour un objet requête créée de façon interactive exporté (suite)

Type d'enregistrement	Numéro de table	Numéro de zone	Description de la zone
			Pour le type d'entrée '4', identifie un connecteur : <ul style="list-style-type: none"> • O pour 'ou' • A pour 'et' (valeur par défaut)
		1314	Pour le type d'entrée '1', cette zone est un nom de colonne, une expression ou une fonction récapitulative : Longueur abrégée (255) Longueur développée (560)
			Pour le type d'entrée '2', identifie l'opérateur : <ul style="list-style-type: none"> • EQ pour 'égal à' • LT pour 'inférieur à' • LE pour 'inférieur ou égal à' • GT pour 'supérieur à' • GE pour 'supérieur ou égal à' • BT pour 'entre' • SW pour 'commençant par' • EW pour 'finissant par' • CT pour 'contenant' • NL pour NULL
			Pour le type d'entrée '3', identifie une valeur
			Pour le type d'entrée '4' (non utilisé)
T	1410	-	Table des conditions de tri L'enregistrement T figurant dans cette section de la requête créée de façon interactive exportée dans «Format de la requête exportée», à la page 132 identifie cette section comme la partie contenant les conditions de tri pour la requête. Chaque enregistrement T est suivi d'enregistrements R qui caractérisent chaque condition de tri. Cette section se présente comme suit dans la requête créée de façon interactive exportée : T 1410 001 002 1412 001 1413 255 R D C.SALAIRE+A.COMM Cette section de la requête exportée correspond à la section suivante de l'exemple de requête présenté dans «Exemple de requête (avant l'exportation)», à la page 131 : Tri : Décroissant par C.SALAIRE+A.COMM
		1412	Sens de tri : <ul style="list-style-type: none"> • A pour 'croissant' • D pour 'décroissant'
		1413	Colonne : Longueur abrégée (255) Longueur développée (560)

Tableau 42. Numéros de table et de zone pour un objet requête créée de façon interactive exporté (suite)

Type d'enregistrement	Numéro de table	Numéro de zone	Description de la zone
V		1501	<p>Traitement des lignes en double :</p> <ul style="list-style-type: none"> • K pour 'conserver' • D pour 'effacer' <p>Par exemple, la ligne suivante dans l'exemple de requête créée de façon interactive exportée dans la section «Format de la requête exportée», à la page 132 montre que la longueur d'enregistrement de la valeur K est 1 ("001"). La ligne montre également que l'utilisateur qui a créé la requête a demandé de conserver les lignes en double ("K") :</p> <p>V 1501 001 K</p> <p>Cette section de la requête exportée correspond à la section suivante de l'exemple de requête présenté dans «Exemple de requête (avant l'exportation)», à la page 131 :</p> <p>Lignes en double : Conserver les lignes en double</p>

La signification des valeurs pour les zones 1313 et 1314 dépend du numéro de séquence indiqué dans le numéro de zone 1312 dans le numéro de table 1310.

Référence associée:

«Enregistrements d'en-tête», à la page 102

La plupart des enregistrements des rapports, requêtes créées de façon interactive ou formats exportés ont un format variable. Toutefois, les enregistrements d'en-tête ont un format fixe, même si l'ensemble de données ou la file d'attente de données qui contient les enregistrements peut être de format variable.

«Enregistrements de ligne de table (R)», à la page 107

Les enregistrements R des rapports, requêtes créées de façon interactive ou formats exportés fournissent un ensemble de valeurs pour une ligne unique dans une table codée. Les enregistrements R contiennent une liste de valeurs disposées selon un ordre décrit par l'enregistrement T associé. Un enregistrement R correspond à la description des positions et longueurs des valeurs de données spécifiées dans l'enregistrement T.

Vérification de la validité du format de la requête créée de façon interactive exportée

L'importation d'un objet requête créée de façon interactive qui a été modifié par votre application est soumise à certaines règles.

Si vous voulez importer un objet requête créée de façon interactive qui a été modifié par votre application, tenez compte des points suivants :

- Lorsqu'un fichier de requête créée de façon interactive est importé, les enregistrements entrants doivent se trouver dans cet ordre spécifique après l'enregistrement d'en-tête (H) :
 1. Enregistrements T pour les définitions de table

2. Enregistrements R pour les noms de table
3. Enregistrements T pour les définitions de colonne
4. Enregistrements R pour les colonnes
5. Les enregistrements de condition de ligne (numéro de table 1310) doivent être classés au sein de chaque condition en fonction du numéro de séquence du type d'entrée (numéro de zone 1312). En d'autres termes, les enregistrements doivent figurer dans le même ordre que celui dans lequel apparaissent les données de ligne dans la zone d'écho Requête créée de façon interactive.

Les enregistrements restants peuvent se trouver dans n'importe quel ordre.

- La table des définitions de table (enregistrement T 1110) doit figurer avant les autres tables ou enregistrements V.
- La valeur du nombre de lignes dans l'enregistrement T Tables doit être * ou un entier compris entre 0 et 15. Si une valeur égale à zéro figure dans le nombre de lignes, l'ensemble des données de la requête est ignoré, ce qui signifie qu'une requête vide est importée.
- QMF n'émet pas d'avertissements pour les importations de requête vide.
- Si une deuxième table Tables (table 1110) est spécifiée, QMF renvoie une erreur et le contenu de la table est ignoré. La requête créée de façon interactive ne fournit pas de valeurs par défaut lors de l'importation.
- S'il existe une table Tri, une table Colonnes doit la précéder.
- QMF accepte les enregistrements en double dans le fichier d'importation. La valeur la plus récente pour l'enregistrement est utilisée.
- Tous les noms de colonne doivent être qualifiés par l'identificateur de table lors de l'importation.
- Lorsqu'une requête créée de façon interactive est exportée vers un ensemble de données préalloué, la longueur d'enregistrement logique (LRECL) minimale autorisée est 259 octets.
- Le format exporté d'une requête créée de façon interactive est identique quelle que soit la langue nationale utilisée ; le format est indépendant de la langue. L'octet de langue dans l'enregistrement d'en-tête est ignoré lors de l'importation. Les codes utilisés lors de l'exportation d'une requête créée de façon interactive sont décrits dans la liste des numéros de table et de zone pour un objet requête créée de façon interactive exporté.

Les fonctions récapitulatives et les expressions ne sont pas traduites ; les fonctions récapitulatives COUNT, AVG, SUM, MIN et MAX demeurent donc inchangées.

Importation de formats et de requêtes créées de façon interactive

Prenez connaissance des règles relatives à l'importation d'un format ou d'une requête créée de façon interactive.

Lorsque vous importez un format ou une requête créée de façon interactive :

- Le fichier peut être composé d'enregistrements de longueur variable ou fixe.
- L'identificateur d'enregistrement (H, V, T, R, E, *, L ou C) doit être en première position dans chaque enregistrement.
- Les deux premiers octets sont réservés aux informations de contrôle (zone de contrôle).
- Chaque zone de données (notamment les numéros de zone, longueurs et valeurs) doit être précédée et suivie d'un délimiteur, à l'exception suivante : la

dernière zone de données d'un enregistrement n'a pas besoin d'être suivie d'un délimiteur, car la fin d'enregistrement fait office de délimiteur. (Les exemples figurant dans ce document utilisent le caractère blanc comme délimiteur.)

- Si QMF détecte une table ou une valeur de données en double lors de l'importation, il remplace la valeur ou la table précédente. Toutefois, les doublons ne sont pas autorisés dans le cas où ils enfreindraient les règles relatives à un objet donné. Par exemple, le nombre de colonnes fournies pour un format ne peut pas être modifié une fois que la première table COLUMNS a été traitée.
- Les numéros de table, de zone et les longueurs numériques peuvent contenir des zéros en tête ou des blancs de début. Cependant, les blancs de fin (sauf le délimiteur blanc) ne sont pas autorisés ; les zones doivent être alignées à droite.
- Lorsqu'un astérisque (*) est utilisé à la place d'une longueur ou d'un nombre, il doit être aligné à gauche et rempli à l'aide de blancs de fin.
- Si la valeur fournie pour une zone de saisie est inférieure à la zone, elle est remplie à l'aide de blancs de fin. Si la valeur fournie est supérieure à la zone, elle est tronquée.
- Si l'enregistrement est plus court que sa longueur de format fixe, ces zones non spécifiées sont censées être vides.

Référence associée:

«Exportation d'un format», à la page 113

L'objet format contient toutes les informations spécifiées dans tous les écrans de format QMF. Lorsque vous exportez un format, QMF convertit en format codé tous les écrans de format dont les valeurs sont différentes des valeurs par défaut.

«Spécifications de taille pour les objets externalisés», à la page 142

Les tables et les objets externes comportent des spécifications de taille et de format d'enregistrement qui varient en fonction du type d'objet.

Procédures et requêtes SQL

Le format de l'ensemble de données TSO ou de la file d'attente de données CICS qui représente ces objets est le plus simple de tous. Chaque enregistrement figurant dans l'ensemble de données ou la file d'attente de données est avant tout une image d'une ligne telle qu'elle s'affiche à l'écran (enregistrement de longueur fixe de 79 octets).

Bien que chaque ligne de ces objets comporte 79 octets, la longueur d'enregistrement logique (LRECL) pour les ensembles de données nouveaux et existants peut être comprise entre 79 et 32 760 octets. Si vous effectuez l'exportation vers un nouvel ensemble de données, la valeur LRECL est spécifiée par la variable globale DSQEC_DSLRECL1. Si la valeur LRECL est supérieure à 79, QMF remplit chaque enregistrement d'objet à l'aide de blancs lors de l'exportation.

Voici un exemple de requête SQL simple :

Requête SQL

```
SELECT *  
FROM Q.PERS
```

Figure 24. Requête SQL simple

Cet exemple présente la requête dans son format externalisé :

```
SELECT *  
FROM Q.PERS
```

En raison de la simplicité du format d'enregistrement, la création ou l'édition d'une procédure ou d'une requête SQL en dehors de QMF est simple. Une procédure ou requête SQL est composée de données de longueur fixe dans les colonnes 1 à 79. Les données situées dans les colonnes 80 à 32 760 sont ignorées lors de l'importation. Lorsque vous importez la file d'attente de données ou l'ensemble de données généré, votre requête ou procédure se trouve dans la zone de stockage temporaire QMF, prête à être exécutée.

Exportation de diagrammes basés sur des formats et de requêtes QBE

Vous pouvez exporter des diagrammes basés sur des formats et des objets requête QBE (Query-by-Example) en vue d'un traitement en dehors de l'environnement QMF.

Exportation de diagrammes basés sur des formats

Un diagramme basé sur un format ne peut pas être sauvegardé sous forme d'objet QMF dans la base de données, ni extrait de cette dernière. Vous ne pouvez pas importer de diagrammes basés sur des formats dans QMF.

Lorsque vous exportez un diagramme dans QMF, il convertit les données du rapport au format GDF (Graphics Data Format). GDF, qui est un format GDDM, est une norme pour l'échange de données. Vous pouvez imprimer les données du diagramme exporté à l'aide des utilitaires GDDM ou les inclure dans des documents.

Vous pouvez utiliser un objet diagramme exporté comme n'importe quel ensemble de données au format GDF. Par exemple, grâce au programme Document Composition Facility (DCF), une application peut combiner un rapport QMF (qui utilise un rapport imprimé ou exporté) avec un diagramme QMF (qui utilise un diagramme exporté) et envoyer les informations formatées vers une imprimante.

Exportation de requêtes QBE

Les objets requête QBE sont exportés à l'aide d'un format interne à QMF. Ce format ne peut en aucun cas être modifié.

Spécifications de taille pour les objets externalisés

Les tables et les objets externes comportent des spécifications de taille et de format d'enregistrement qui varient en fonction du type d'objet.

Le tableau ci-dessous contient les spécifications pour les fichiers d'importation et d'exportation TSO et CICS. Pour CICS, les tailles d'enregistrement ne sont pas imposées. Par exemple, si vous importez une requête SQL à partir d'une file d'attente de mémoire temporaire de 32 ko, QMF la tronque à 79 octets.

Le format d'enregistrement n'est pas un facteur pour les files d'attente de données transitoires ou les files d'attente de mémoire temporaire CICS. Une file d'attente de mémoire temporaire accueille les enregistrements quel que soit leur format. Une file d'attente de données transitoires est définie sur une table de contrôle des destinations (DCT) et ignore le format d'enregistrement.

Vous devez spécifier un nom pour votre ensemble de données ou file d'attente de données CICS dans la commande EXPORT ou IMPORT. Les noms de file d'attente ne comportent pas de préfixe ou suffixe par défaut. Les noms de file d'attente de mémoire temporaire CICS contiennent 8 octets, tandis que les noms de file d'attente de données transitoires en comportent 4.

Les abréviations suivantes sont utilisées pour les formats d'enregistrement dans le tableau :

- FB - format de bloc fixe
- VB - format de bloc variable
- VBS - format de bloc variable sur plusieurs lignes

Tableau 43. Attributs de fichier et d'ensemble de données

Objet	Taille d'enregistrement	Format d'enregistrement
Données ou table (format QMF)	Taille maximale : 7 000 octets	Longueur fixe (FB)
Données ou table (format IXF)	<p>Taille maximale : 32 756</p> <p>La valeur LRECL minimale pour un format exporté incluant des colonnes définies est 161 octets.</p> <p>La valeur LRECL minimale acceptée par QMF pour un ensemble de données ou file d'attente CICS IXF lors de l'importation est 49 octets.</p> <p>La taille d'enregistrement est généralement la longueur d'une ligne de données dans la table qui est en cours d'exportation (y compris l'espace pour les indicateurs de type NULL et les délimiteurs DBCS) plus la longueur de la zone du nombre d'enregistrements de type D IXF (5 octets).</p>	Longueur variable (VB)
Données ou table (format XML)	Taille maximale : 2 Go	Longueur variable (VBS)

Tableau 43. Attributs de fichier et d'ensemble de données (suite)

Objet	Taille d'enregistrement	Format d'enregistrement
Données ou table (format CSV)	<p>La valeur LRECL maximale pour l'exportation vers de nouveaux ensembles de données est calculée selon que la valeur YES ou NO est spécifiée pour l'option HEADER de la commande EXPORT.</p> <ul style="list-style-type: none"> • Si HEADER=YES est spécifié, la formule suivante est utilisée : $\max(\text{Column_Names_Total_Length}, \text{Column_Data_Value_Total_Length}) + (\text{nombre de colonnes} * 3 - 1) + 4$ • Si HEADER=NO est spécifié, la formule suivante est utilisée : $(\text{longueur des valeurs de données de colonne}) + (\text{nombre de colonnes} - 1) + (\text{nombre de colonnes} * 2) + 4$ <p>Pour une exportation vers des ensembles de données existants, la valeur LRECL de l'ensemble de données existant est la valeur LRECL maximale pouvant être exportée.</p>	Longueur variable (VB)
Requête créée de façon interactive	<p>Maximum : 7 290 octets</p> <p>Minimum : 266 octets lors de l'exportation ; 41 octets lors de l'importation</p>	<p>Longueur variable (VB) lors de l'exportation</p> <p>Longueur fixe (FB) ou longueur variable (VB) lors de l'importation</p>
Requête SQL	Doit être comprise entre 79 et 32 760 octets lors de l'exportation vers des ensembles de données nouveaux et existants ; la taille peut avoir n'importe quelle valeur lors de l'importation, mais elle est tronquée à 79 octets.	<p>Longueur fixe (FB) ou longueur variable (VB) lors de l'exportation vers des ensembles de données existants ; longueur fixe (FB) lors de l'exportation vers de nouveaux ensembles de données</p> <p>Longueur fixe (FB) ou longueur variable (VB) lors de l'importation</p>
Requête QBE	<p>Doit être de 1 024 octets</p> <p>Une requête QBE vide est de 828 octets.</p>	Longueur variable (VB)
Formulaire	<p>Maximum : 7 290 octets</p> <p>Minimum : 161 octets lors de l'exportation ; 23 octets lors de l'importation</p>	<p>Longueur fixe (FB) lors de l'exportation</p> <p>Longueur fixe (FB) ou longueur variable (VB) lors de l'importation</p>
Proc	Doit être comprise entre 79 et 32 760 octets lors de l'exportation vers des ensembles de données nouveaux et existants ; la taille peut avoir n'importe quelle valeur lors de l'importation, mais elle est tronquée à 79 octets.	<p>Longueur fixe (FB) ou longueur variable (VB) lors de l'exportation vers des ensembles de données existants ; longueur fixe (FB) lors de l'exportation vers de nouveaux ensembles de données</p> <p>Longueur fixe (FB) ou longueur variable (VB) lors de l'importation</p>

Tableau 43. Attributs de fichier et d'ensemble de données (suite)

Objet	Taille d'enregistrement	Format d'enregistrement
Rapport	Maximum : 7 290 octets Minimum : 65 octets	Longueur variable (VB)
Rapport HTML	Maximum : 32 000 octets	Longueur variable (VB)

Considérations relatives au stockage

Lorsque vous importez et exportez des objets vers des files d'attente de données CICS et des ensembles de données TSO, prenez connaissance de la façon dont QMF traite le stockage.

Files d'attente de données CICS

Pour les objets exportés vers une file d'attente de données CICS, vous devez comprendre la façon dont QMF traite les files d'attente.

Lorsque vous exportez un objet vers une file d'attente de données CICS, tenez compte des points suivants :

- Dans CICS, les commandes IMPORT et EXPORT requièrent l'option QUEUETYPE. Il n'existe pas de paramètre par défaut.
- Lors de l'importation d'un objet dans CICS à partir d'une file d'attente de données transitoires (TD), vous devez spécifier le type d'objet correct. La file d'attente est vidée une fois que QMF a extrait son contenu. Par exemple, si vous indiquez «Format» lorsque le type d'objet dans la file d'attente de données transitoires est une procédure, QMF envoie un message d'erreur. Toutefois, vous ne pouvez pas émettre à nouveau la commande IMPORT (même avec le type d'objet correct) en utilisant la même file d'attente, car cette dernière est désormais vide.
- Dans CICS, la file d'attente de données transitoires ou de mémoire temporaire (TS) doit contenir un objet QMF terminé unique pour que vous puissiez émettre la commande IMPORT.
- Si vous effectuez l'exportation vers une file d'attente de données transitoires, cette file d'attente doit être ouverte, activée et vide avant l'émission de la commande EXPORT.

QMF traite les files d'attente de données transitoires CICS différemment des files d'attente de mémoire temporaire.

- Files d'attente de données transitoires
QMF importe la totalité de la file d'attente de données transitoires avant d'afficher l'objet à l'écran. Cela signifie que le contenu de la totalité de la file d'attente doit tenir dans votre mémoire ou zone de sauvegarde. Vous pouvez utiliser le paramètre DSQSPILL pour spécifier l'utilisation d'une mémoire auxiliaire. L'objet peut s'afficher avec un certain retard s'il est volumineux et que vous utilisez un fichier pour les données temporaires.

Une file d'attente de données transitoires CICS intrapartition peut contenir jusqu'à 32 ko de lignes de données. Une file d'attente de données transitoires extrapartition peut prendre la taille nécessaire pour accueillir l'objet.

- Files d'attente de mémoire temporaire

Par défaut, QMF lit environ 100 lignes de mémoire temporaire avant de les afficher à destination de l'utilisateur. Une file d'attente de mémoire temporaire peut contenir jusqu'à 32 ko de lignes de données.

QMF utilise le paramètre SUSPEND dans les commandes IMPORT et EXPORT pour permettre une régulation par CICS lors de l'exécution de la commande.

Le paramètre SUSPEND dans les commandes IMPORT et EXPORT détermine l'action à effectuer lorsqu'une file d'attente est occupée. Lorsque le paramètre SUSPEND est défini sur YES, QMF émet une commande CICS ENQ (mise en file d'attente) pour le nom de file d'attente de données CICS. Ce paramètre demande à CICS d'attendre que la file d'attente soit disponible avant d'y placer l'objet QMF. L'attente garantit que la transaction QMF n'interfère pas avec d'autres travaux traités par la file d'attente.

Lorsque le paramètre SUSPEND est défini sur NO, la commande EXPORT est annulée et un message est renvoyé. La valeur par défaut de l'option SUSPEND est NO.

Ensembles de données TSO

Pour les objets exportés vers un ensemble de données TSO, vérifiez que vous configurez correctement votre système de gestion du stockage.

Si vous utilisez des unités de stockage à accès direct (DASD) standard, vérifiez que votre logiciel de gestion du stockage est configuré pour traiter l'allocation dynamique des jeux de données étendus. Lorsque vous configurez ces jeux de données, indiquez les classes de stockage par défaut. Lorsque votre système de gestion du stockage est configuré ainsi et que vous exportez un objet, QMF alloue dynamiquement un jeu de données. QMF utilise le nom spécifié dans la commande EXPORT si le jeu de données n'existe pas. Si vous exportez des données au format XML, il se peut que vous receviez des erreurs d'allocation dynamique si les jeux de données ne sont pas correctement configurés. Reportez-vous aux informations fournies avec votre logiciel de gestion du stockage pour en savoir plus sur la configuration de l'allocation dynamique des jeux de données étendus.

Toutefois, si vous n'utilisez pas l'unité de stockage standard, vous devez pré-allouer vos jeux de données avant de faire appel à la commande EXPORT. Vous pouvez utiliser des variables globales pour spécifier le type et la taille des nouveaux ensembles de données qui vont contenir les objets exportés :

- Utilisez la variable globale DSQEC_PO pour spécifier le type d'ensemble de données partitionné à créer lors de l'exportation d'un objet vers un membre d'un nouvel ensemble de données. Il peut s'agir du type par défaut pour votre site, d'un ensemble de données PDS ou PDSE.
- Utilisez la variable globale DSQEC_DSALLOC_DIR pour indiquer le nombre de blocs de répertoire lors de l'exportation d'un membre vers un nouvel ensemble de données PDS. La valeur par défaut est 20.
- Utilisez la variable globale DSQEC_DSALLOC_PRI pour spécifier l'allocation d'espace principal dans les pistes. La valeur par défaut est de 15 pistes.
- Utilisez la variable globale DSQEC_DSALLOC_SEC pour spécifier l'allocation d'espace secondaire dans les pistes. La valeur par défaut est de 105 pistes.

Référence associée:

«Variables globales contrôlant le mode d'exécution des commandes et des procédures», à la page 237

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

Chapitre 10. Débogage des applications QMF

Outre le traitement des erreurs, QMF offre des fonctions de débogage pour vos applications d'interface callable.

Vous pouvez utiliser la fonction de trace REXX via l'instruction REXX TRACE.

Concepts associés:

Chapitre 6, «Ecriture d'applications QMF utilisant des services ISPF», à la page 33

Vous pouvez ignorer les écrans QMF en écrivant des applications qui possèdent leurs propres interfaces utilisateur. Vous pouvez utiliser soit l'interface callable, soit l'interface de commande, pour écrire des applications utilisant ISPF.

Débogage de vos applications d'interface callable

QMF fournit deux options de trace, L et A, ainsi que divers niveaux de détail de trace pour le débogage de vos applications.

Option L pour le traçage

L'option L permet d'écrire des messages et des commandes dans un ensemble de données TSO ou une file d'attente de données CICS externe.

Vous avez le choix entre deux options L :

- L1 Tous les messages QMF sont écrits dans la sortie de données de trace QMF.
- L2 Tous les messages et toutes les commandes QMF sont écrits dans la sortie de données de trace QMF.

Vous pouvez définir l'option L de l'une des deux façons suivantes :

- Emettez la commande DISPLAY PROFILE puis, quand le PROFIL est affiché, modifiez l'option TRACE pour lui affecter le paramètre L1 ou L2.
- Exécutez la commande suivante :
SET PROFILE (TRACE=x

Dans cette instruction, *x* représente L1 ou L2.

Concepts associés:

«Allocation de la sortie de données de trace QMF», à la page 150

Vous devez allouer la sortie de données de trace QMF avant de démarrer QMF si le traçage doit être utilisé.

Option A pour le traçage

Vous pouvez utiliser l'option A pour spécifier un niveau de traçage pour les services de support d'application QMF.

L'option A peut être A0, A1 ou A2. A0 est la valeur par défaut et aucun traçage n'est effectué. A1 et A2 fournissent des résultats de plus en plus détaillés. Ce modèle est également utilisé pour les autres options de trace QMF.

Spécifiez l'option A de la même façon que l'option L : via une commande QMF SET PROFILE ou en la saisissant à l'écran après avoir émis la commande DISPLAY

PROFILE. Par exemple, vous pouvez entrer l'instruction suivante immédiatement avant d'appeler l'application que vous êtes en train de déboguer :

```
SET PROFILE (TRACE=L2A1)
```

Lorsque vous démarrez votre application, les traçages L2 et A1 sont tous deux en vigueur.

Pour déterminer le paramètre d'option A en cours, consultez la variable DSQAO_APPL_TRACE. Sa valeur est 0, 1 ou 2 pour les paramètres A0, A1 ou A2. Vous pouvez utiliser la valeur de DSQAO_APPL_TRACE pour sélectionner le traçage souhaité dans votre application, comme dans l'application REXX présentée ici :

```
/* programme REXX permettant de définir le traçage                               */
call dsqcix "GET GLOBAL(A_TRACE=DSQAO_APPL_TRACE"
if a_trace > 0 then
do
  /* code de trace pour A1 et A2 */
  :
  if a_trace = 2 then
  do
    /* code de trace pour A2 uniquement */
    :
  end
end
```

Figure 25. Structure d'un exemple de programme REXX que vous pouvez utiliser pour définir le traçage pour les services de support d'application

Désactivation du traçage

Pour désactiver le traçage, utilisez la commande SET PROFILE.

Si vous avez besoin de désactiver le traçage pour une raison quelconque, émettez la commande suivante : SET PROFILE (TRACE=NONE)

Cette commande interrompt le traçage pendant le reste de la session QMF, mais n'affecte pas le profil QMF permanent.

Allocation de la sortie de données de trace QMF

Vous devez allouer la sortie de données de trace QMF avant de démarrer QMF si le traçage doit être utilisé.

Vous voudrez peut-être réallouer l'ensemble de données ou la file d'attente de données si l'allocation d'origine ne répond pas à vos besoins.

Pour obtenir des exemples d'allocation de sortie de données de trace QMF pour TSO, consultez la spécification du langage de programmation que vous utilisez.

Les commandes figurant dans les exemples allouent un ensemble de données ou une file d'attente de données de trace séquentielles que vous pouvez examiner à la fin de votre session QMF. La sortie est constituée d'enregistrements de 80 caractères de longueur fixe.

Pour CICS, vous pouvez utiliser les paramètres de programme DSQSDBQT et DSQSDBQN pour indiquer l'endroit où QMF place vos données de trace. Faites

attention lors de l'utilisation de la mémoire temporaire CICS, car QMF peut générer une grande quantité de données de trace. Comme les données de trace qui dépassent la taille de la file d'attente sont effacées, utilisez la mémoire temporaire CICS uniquement pour les données de trace provenant de messages ou d'applications de petite taille.

Référence associée:

Chapitre 11, «Spécifications de langage de programmation pour l'utilisation de l'interface callable», à la page 153

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

«Mots clés de la commande START», à la page 56

Vous pouvez spécifier des mots clés dans la commande START.

Commande QMF MESSAGE pour le traçage

Vous pouvez utiliser la commande QMF MESSAGE pour effectuer d'autres opérations que le simple affichage d'un message à la fin d'une application. Vous pouvez également vous en servir pour enregistrer des messages dans la sortie de données de trace QMF.

Pour enregistrer des messages, exécutez l'application avec l'option de traçage L définie sur L1 ou L2. Chaque message traité via la commande MESSAGE est enregistré, avec d'autres messages QMF, dans la sortie de données de trace QMF. Si l'option de traçage L est définie sur L2, les commandes sont également enregistrées.

En plaçant des commandes MESSAGE à des emplacements stratégiques de votre programme, vous pouvez consigner des informations utiles dans le fichier de trace QMF. Vous pouvez consulter les informations, soit à l'écran, soit sur une sortie imprimée.

Les lignes suivantes présentent un exemple d'activation du traçage et d'émission de messages pertinents qui s'affichent dans la sortie de trace :

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

Dans cet exemple, les enregistrements contenant les messages «QUERYA COMPLETED SUCCESSFULLY» et «EXECB ENTERED WITH A VALUE OF 7» sont écrits dans la sortie de données de trace QMF.

Les messages QMF pouvant changer d'une édition à l'autre, n'utilisez pas la sortie de données de trace QMF comme données d'entrée pour une application.

Concepts associés:

«Allocation de la sortie de données de trace QMF», à la page 150

Vous devez allouer la sortie de données de trace QMF avant de démarrer QMF si le traçage doit être utilisé.

«Option L pour le traçage», à la page 149

L'option L permet d'écrire des messages et des commandes dans un ensemble de données TSO ou une file d'attente de données CICS externe.

Erreurs relatives à la commande **START** ou à d'autres commandes **QMF**

En fonction de son niveau, DSQCOMM peut contenir un texte de message. Si la commande **START** (ou toute commande **QMF**) échoue, vous pouvez utiliser ce texte pour identifier et résoudre les problèmes.

Chapitre 11. Spécifications de langage de programmation pour l'utilisation de l'interface callable

L'interface de programmation d'application QMF est disponible pour plusieurs langages de programmation.

IBM fournit des informations sur l'assemblage (ou la compilation) des programmes et l'édition de leurs liens, ainsi que sur l'exécution de ces programmes à l'aide de l'interface callable. IBM ne fournit pas les commandes EXEC REXX, JCL ou CLIST dans les exemples, mais vous pouvez les copier et les modifier en fonction de vos besoins.

Interface en langage Assembler

Vous pouvez utiliser le langage Assembler avec l'interface callable dans QMF.

Mappage de la zone de communication d'interface pour le langage Assembler (DSQCOMMA)

DSQCOMMA offre un mappage DSQCOMM pour le langage Assembler ; il est fourni avec le produit.

Le tableau ci-dessous affiche les valeurs pour DSQCOMMA.

Tableau 44. Contenu de la zone de communication d'interface DSQCOMMA

Nom de structure	Type de données	Description
DSQ_RETURN_CODE	DS F	Indique l'état d'une commande QMF après son exécution Les valeurs possibles sont les suivantes : DSQ_SUCCESS Exécution réussie de la demande DSQ_WARNING Achèvement normal avec des avertissements DSQ_FAILURE La commande ne s'est pas exécutée correctement DSQ_SEVERE Erreur grave ; la session QMF a pris fin
DSQ_INSTANCE_ID	DS F	Identificateur défini par QMF lors de l'exécution de la commande START
DSQ_COMM_LEVEL	DS CL12	Identifie le niveau de DSQCOMM Dans votre application, incluez des instructions qui initialisent cette variable sur la valeur affectée à DSQ_CURRENT_COMM_LEVEL avant d'émettre la commande QMF START.

Tableau 44. Contenu de la zone de communication d'interface DSQCOMMA (suite)

Nom de structure	Type de données	Description
DSQ_PRODUCT	DS CL2	Identifie le produit de requête IBM en cours d'utilisation Les variables commençant par DSQ_QMF indiquent les versions de QMF for TSO et QMF for CICS.
DSQ_PRODUCT_RELEASE	DS CL2	Niveau d'édition de QMF en cours d'utilisation La variable DSQ_QMF_V11R1 indique QMF Version 12.1.
DSQ_RESERVE1	DS XL28	Réservé à une utilisation ultérieure
DSQ_MESSAGE_ID	DS CL8	ID de message d'achèvement
DSQ_Q_MESSAGE_ID	DS CL8	ID de message de requête
DSQ_START_PARM_ERROR	DS CL8	Nom du paramètre erroné lorsque la commande START a échoué en raison d'une erreur de paramètre
DSQ_CANCEL_IND	DS C	Contient l'une des deux valeurs ci-dessous, selon que l'utilisateur a annulé ou non la session QMF pendant l'exécution d'une commande QMF : <ul style="list-style-type: none"> • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	DS XL23	Réservé à une utilisation ultérieure
DSQ_RESERVE3	DS XL156	Réservé à une utilisation ultérieure
DSQ_MESSAGE_TEXT	DS CL128	Texte du message d'achèvement
DSQ_Q_MESSAGE_TEXT	DS CL128	Texte du message de requête

Appels de fonction pour le langage Assembler

QMF fournit un appel de fonction (DSQCIA) pour les programmes en langage Assembler. L'appel de fonction comporte deux formats : la syntaxe classique et la syntaxe étendue.

DSQCIA, syntaxe classique

Cet appel est destiné aux commandes QMF qui ne requièrent pas d'accès à des variables de programme d'application. Utilisez cet appel pour la plupart des commandes QMF.

```
CALL DSQCIA, (DSQCOMM,CMDLTH,CMDSTR),VL
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre FULLWORD

CMDSTR

Commande QMF émise lors de l'appel de fonction ; chaîne de caractères majuscules de la longueur spécifiée par CMDLTH

VL est l'instruction Assembler VARIABLE LIST.

DSQCIA, syntaxe étendue

Ce format de syntaxe étendue de l'appel de fonction DSQCIA est destiné aux commandes QMF qui requièrent un accès à des variables de programme d'application : START, TRACE et les formats étendus de GET GLOBAL et SET GLOBAL.

```
CALL DSQCIA, (DSQCOMM, CMDLTH, CMDSTR,  
             PNUM, KLTH, KWORD, VLTH, VALUE, VTYPE), VL
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre FULLWORD

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par CMDLTH

PNUM

Nombre de mots clés de commande ou de zones de trace ; paramètre FULLWORD

KLTH

Longueur de chaque mot clé ou titre de trace spécifié ; paramètre FULLWORD ou tableau de paramètres FULLWORD.

KWORD

Mot(s) clé(s) QMF ou adresse des titres de trace ; caractère, tableau de caractères ou d'adresses vers des titres de trace dont les longueurs sont spécifiées par KLTH.

VLTH

Longueur de chaque valeur qui est associée au mot clé ou au titre de trace ; paramètre FULLWORD ou tableau de paramètres FULLWORD.

VALUE

Valeur associée à chaque mot clé ou adresse d'une valeur qui est associée à un titre de trace.

Son type est spécifié dans le paramètre VTYPE et peut être un caractère, un tableau de caractères, un paramètre FULLWORD ou un tableau de paramètres FULLWORD. Pour les données de trace, FINT doit être affecté à VTYPE.

VTYPE

Type de données du contenu du paramètre VALUE.

Ce paramètre comporte l'une des deux valeurs qui sont fournies dans la zone de communication d'interface DSQCOMMMA :

- DSQ_VARIABLE_CHAR pour les valeurs de type caractère
Si DSQ_VARIABLE_CHAR est affecté à VTYPE, VALUE n'est pas validé.
- DSQ_VARIABLE_FINT pour les valeurs entières

Si DSQ_VARIABLE_FINT est affecté à VTYPE, VALUE est validé et doit être un nombre entier.

Toutes les valeurs spécifiées dans la zone VALUE doivent avoir le type de données qui est indiqué dans VTYPE.

VL est l'instruction Assembler VARIABLE LIST.

Exemple de programmation Assembler

IBM fournit un exemple de programme Assembler pour CICS et TSO. Cet exemple de programme fait partie de la bibliothèque QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale).

Exemple de programme Assembler pour CICS

IBM fournit un exemple de programme Assembler pour CICS appelé DSQABFAC.

Cet exemple de programme Assembler pour CICS se trouve dans la bibliothèque QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale).

Cet exemple de programme pour l'interface callable d'Assembler exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

```
TITLE 'Sample HLASM Query Callable Interface'
*****
*
* Sample Program: DSQABFAC
* Assembler Version of the QMF Callable Interface for CICS
*
*****
DSQABFAC DFHEIENT CODEREG=(12),DATAREG=(13),EIBREG=(11)
DSQABFAC AMODE 31
DSQABFAC RMODE ANY
        SPACE 1
*****
* Start a query interface session
*****
        LA    R4,CICOMM           ESTABLISH ACCESS TO DSQCOMM
        USING DSQCOMM,R4
        SPACE 1
        MVC   DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL
        ST    R4,QMFP1           Address of DSQCOMM
        LA    R1,STARTQIL        Address of START command length
        ST    R1,QMFP2
        LA    R1,STARTQI        Address of START command
        ST    R1,QMFP3
        LA    R1,1               One Start command parameter
        ST    R1,NUMPARMS
        LA    R1,NUMPARMS        Address of number of parameters
        ST    R1,QMFP4
        LA    R1,STARTKYL        Address of keyword lengths
        ST    R1,QMFP5
        LA    R1,STARTKY        Address of keywords
        ST    R1,QMFP6
```

```

LA R1,STARTVL          Address of value lengths
ST R1,QMFP7
LA R1,STARTV           Address of values
ST R1,QMFP8
LA R1,DSQ_VARIABLE_CHAR Address of value data type
ST R1,QMFP9
OI QMFP9,X'80'         Set end of parameter list
LA R1,QMFPLIST         Address of parameter list
CALL DSQCIA
SPACE 1
*****
* Set numeric values into query using SET command *
*****
SPACE 1
LA R1,20               Set values for SET GLOBAL command
ST R1,VVAL1
LA R1,40
ST R1,VVAL2
LA R1,84
ST R1,VVAL3
LA R1,SETGL            Addr of SET GLOBAL command length
ST R1,QMFP2
LA R1,SETG             Address of SET GLOBAL command
ST R1,QMFP3
LA R1,3                Three SET GLOBAL variables
ST R1,NUMPARMS
LA R1,NUMPARMS         Address of number of parameters
ST R1,QMFP4
LA R1,VNAME1L          Address of variable name lengths
ST R1,QMFP5
LA R1,VNAME1           Address of variable names
ST R1,QMFP6
LA R1,VVAL1L           Address of value lengths
ST R1,QMFP7
LA R1,VVAL1            Address of values
ST R1,QMFP8
LA R1,DSQ_VARIABLE_FINT Address of value data type
ST R1,QMFP9
OI QMFP9,X'80'         Set end of parameter list
LA R1,QMFPLIST         Address of parameter list
CALL DSQCIA
SPACE 1
*****
* Run a query *
*****
LA R1,QUERYL           Addr of RUN QUERY command length
ST R1,QMFP2
LA R1,QUERY            Address of RUN QUERY command
ST R1,QMFP3
OI QMFP3,X'80'         Set end of parameter list
LA R1,QMFPLIST         Address of parameter list
CALL DSQCIA
SPACE 1
*****
* Print the result of the query *
*****
LA R1,REPTL           Addr of PRINT Report command lth
ST R1,QMFP2
LA R1,REPT            Address of PRINT Report command
ST R1,QMFP3
OI QMFP3,X'80'         Set end of parameter list
LA R1,QMFPLIST         Address of parameter list
CALL DSQCIA
SPACE 1
*****
* End the query interface session *
*****

```

```

        LA R1,ENDQIL           Address of EXIT command length
        ST R1,QMFP2
        LA R1,ENDQI           Address of EXIT command
        ST R1,QMFP3
        OI QMFP3,X'80'        Set end of parameter list
        LA R1,QMFLIST         Address of parameter list
        CALL DSQCIA
        SPACE 1
*****
* Free Keyboard *
*****
        EXEC CICS SEND CONTROL FREEKB
        SPACE 1
*****
* Return *
*****
        SPACE 1
        XR R15,R15           ZERO RETURN CODE
        DFHEIRET RCREG=15
*****
* Data Areas *
*****
        SPACE 1
* Query Interface commands
        SPACE 1
STARTQI DC C'START'          START FUNCTION
SETG    DC C'SET GLOBAL'    SET GLOBAL FUNCTION
QUERY   DC C'RUN QUERY Q1'  RUN QUERY
REPT    DC C'PRINT REPORT (FORM=F1,QUEUEN=DSQP,QUEUET=TS)'
ENDQI   DC C'EXIT'          END INTERFACE
        SPACE 1
        DS 0F
STARTQIL DC AL4(L'STARTQI)   LENGTH OF START FUNCTION
SETGL   DC AL4(L'SETG)      LENGTH OF SET GLOBAL FUNCTION
QUERYL  DC AL4(L'QUERY)     LENGTH OF RUN QUERY COMMAND
REPTL   DC AL4(L'REPT)     LENGTH OF PRINT REPORT COMMAND
ENDQIL  DC AL4(L'ENDQI)     LENGTH OF END INTERFACE COMMAND
        SPACE 1
* START command keyword
        SPACE 1
STARTKY DC C'DSQSMODE'
STARTV  DC C'INTERACTIVE'
        DS 0F
STARTKYL DC AL4(L'STARTKY)
STARTVL  DC AL4(L'STARTV)
        SPACE 1
* SET GLOBAL command variable names
        SPACE 1
VNAME1  DC C'MYVAR01'
VNAME2  DC C'SHORT'
VNAME3  DC C'MYVAR03'
        DS 0F
VNAME1L DC AL4(L'VNAME1)
VNAME2L DC AL4(L'VNAME2)
VNAME3L DC AL4(L'VNAME3)
        SPACE 1
* SET GLOBAL command values
        SPACE 1
VVAL1L  DC AL4(L'VVAL1)
VVAL2L  DC AL4(L'VVAL2)
VVAL3L  DC AL4(L'VVAL3)
* Callable interface communications definition
        DSQCOMMA
* Equates for registers 0-15
R0      EQU 00
R1      EQU 01
R2      EQU 02

```

```

R3      EQU  03
R4      EQU  04
R5      EQU  05
R6      EQU  06
R7      EQU  07
R8      EQU  08
R9      EQU  09
R10     EQU  10
R11     EQU  11
R12     EQU  12
R13     EQU  13
R14     EQU  14
R15     EQU  15
* Local variables located in CICS working storage
DFHEISTG DSECT
        ORG  DFHEIUSR
NUMPARMS DS  F          NUMBER OF KEYWORDS
* QMF SET GLOBAL command values
VVAL1   DS  F
VVAL2   DS  F
VVAL3   DS  F
* QMF Callable interface parameter list
QMFLIST DS  0D
QMFP1   DS  F
QMFP2   DS  F
QMFP3   DS  F
QMFP4   DS  F
QMFP5   DS  F
QMFP6   DS  F
QMFP7   DS  F
QMFP8   DS  F
QMFP9   DS  F
* Callable interface communications area
CICOMM  DS  CL(DSQCOMM_LEN)
        CSECT
        SPACE 1
        END  DSQABFAC

```

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Exemple de programme Assembler pour TSO

IBM fournit un exemple de programme Assembler pour TSO appelé DSQABFA.

Cet exemple de programme Assembler pour TSO se trouve dans la bibliothèque QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale).

Cet exemple de programme pour l'interface callable d'Assembler exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

```

DSQABFA TITLE 'SAMPLE QMF CALLABLE INTERFACE'
DSQABFA CSECT
DSQABFA AMODE 31

```

```

DSQABFA RMODE ANY
*****
*
* Sample Program: DSQABFA
* Assembler Version of the QMF Callable Interface
*
*****
SPACE 1
STM R14,R12,12(R13)      SAVE ENTRY REGISTERS
BALR R12,0              INITIALIZE BASE REGISTER
USING *,R12
LA R2,SAVEAREA          CHAIN SAVE AREAS
ST R2,8(R13)
ST R13,SAVEAREA+4
LR R13,R2               ESTABLISH SAVE AREA
SPACE 1
*****
* Start a query interface session
*****
LA R4,CICOMM            ESTABLISH ACCESS TO DSQCOMM
USING DSQCOMM,R4
SPACE 1
MVC DSQ_COMM_LEVEL,DSQ_CURRENT_COMM_LEVEL
LA R1,1                1 PARAMETER
ST R1,NUMPARMS
CALL DSQCIA,
(CICOMM,                QI COMMON AREA
STARTQIL,              START COMMAND LENGTH
STARTQI,               START COMMAND
NUMPARMS,              NUMBER OF KEYWORDS
STARTKYL,              KEYWORD LENGTHS
STARTKY,               KEYWORDS
STARTVL,               VALUE LENGTHS
STARTV,                VALUES
DSQ_VARIABLE_CHAR),VL VALUES ARE CHARACTERS
SPACE 1
*****
* Set numeric values into query using SET command
*****
SPACE 1
LA R1,20               SET VALUES TO BE MODIFIED
ST R1,VVAL1
LA R1,40
ST R1,VVAL2
LA R1,84
ST R1,VVAL3
LA R1,3                3 PARAMETERS
ST R1,NUMPARMS
SPACE 1
CALL DSQCIA,
(CICOMM,
SETGL,                 SET GLOBAL COMMAND LENGTH
SETG,                  SET GLOBAL COMMAND
NUMPARMS,              NUM OF VARIABLES TO BE SET
VNAME1L,               VARIABLE NAME LENGTHS
VNAME1,                VARIABLE NAMES
VVAL1L,                VALUE LENGTHS
VVAL1,                 VALUES
DSQ_VARIABLE_FINT),VL VALUES ARE INTEGERS
SPACE 1
*****
* Run a query
*****
SPACE 1
CALL DSQCIA,
(CICOMM,
QUERYL,                QUERY COMMAND LENGTH

```

```

                QUERY),VL          TEXT OF QUERY COMMAND
        SPACE 1
*****
* Print the result of the query                                     *
*****
        SPACE 1
        CALL DSQCIA,(CICOMM,REPTL,REPT),VL
        SPACE 1
*****
* End the query interface session                                 *
*****
        SPACE 1
        CALL DSQCIA,(CICOMM,ENDQIL,ENDQI),VL
        SPACE 1
*****
* Return                                                         *
*****
        SPACE 1
        SR   R15,R15          SET RETURN CODE
        L    R13,4(R13)
        L    R14,12(R13)     RESTORE CALLER REGISTERS
        LM   R0,R12,20(R13)
        BR   R14
        EJECT
*****
* Data Areas                                                    *
*****
        SPACE 1
* Query Interface commands
        SPACE 1
STARTQI DC C'START'          START FUNCTION
SETG   DC C'SET GLOBAL'     SET GLOBAL FUNCTION
QUERY  DC C'RUN QUERY Q1'   RUN QUERY
REPT   DC C'PRINT REPORT (FORM=F1)' PRINT REPORT
ENDQI  DC C'EXIT'           END INTERFACE
        SPACE 1
        DS   0F
STARTQIL DC AL4(L'STARTQI)   LENGTH OF START FUNCTION
SETGL   DC AL4(L'SETG)      LENGTH OF SET GLOBAL FUNCTION
QUERYL  DC AL4(L'QUERY)     LENGTH OF RUN QUERY COMMAND
REPTL   DC AL4(L'REPT)     LENGTH OF PRINT REPORT COMMAND
ENDQIL  DC AL4(L'ENDQI)     LENGTH OF END INTERFACE COMMAND
        SPACE 1
* START command keyword
        SPACE 1
STARTKY DC C'DSQSMODE'
STARTV  DC C'INTERACTIVE'
        DS   0F
STARTKYL DC AL4(L'STARTKY)
STARTVL  DC AL4(L'STARTV)
        SPACE 1
* SET GLOBAL command variable names
        SPACE 1
VNAME1  DC C'MYVAR01'
VNAME2  DC C'SHORT'
VNAME3  DC C'MYVAR03'
        DS   0F
VNAME1L DC AL4(L'VNAME1)
VNAME2L DC AL4(L'VNAME2)
VNAME3L DC AL4(L'VNAME3)
        SPACE 1
* SET GLOBAL command values
        SPACE 1
VVAL1   DS   F
VVAL2   DS   F
VVAL3   DS   F
VVAL1L  DC   AL4(L'VVAL1)

```

```

VVAL2L  DC    AL4(L'VVAL2)
VVAL3L  DC    AL4(L'VVAL3)
        SPACE 1
NUMPARMS DS    F                NUMBER OF KEYWORDS
        SPACE 1
* Callable interface communications area
        SPACE 1
CICOMM  DS    CL(DSQCOMM_LEN)
        SPACE 1
SAVEAREA DS   18F
        EJECT
        DSQCOMMA
        SPACE 1
R0      EQU   00                EQUATES FOR REGISTERS 0-15
R1      EQU   01
R2      EQU   02
R3      EQU   03
R4      EQU   04
R5      EQU   05
R6      EQU   06
R7      EQU   07
R8      EQU   08
R9      EQU   09
R10     EQU   10
R11     EQU   11
R12     EQU   12
R13     EQU   13
R14     EQU   14
R15     EQU   15
        SPACE 1
        END    DSQABFA

```

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

DSQCOMM pour Assembler

Le fichier de la zone de communication d'interface Assembler se nomme DSQCOMMA.

Le fichier DSQCOMMA est fourni dans la bibliothèque QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale). Une copie du fichier est présentée ici :

```

        MACRO
        DSQCOMMA
*****
* Callable interface - variable constants *
*****
*
* Communications Level ID
*
DSQ_CURRENT_COMM_LEVEL  DC    CL12'DSQL>001002<'
*
* Query Product IDs
*
DSQ_QRW                  DC    C'01'
DSQ_QMF                  DC    C'02'
DSQ_QM4                  DC    C'03'
*
* Query Product Release IDs
*
DSQ_QRW_V1R2            DC    C'01'
DSQ_QRW_V1R3            DC    C'02'

```



```

DSQ_QMF_V2R4      DC    C'01'
DSQ_QMF_V3R1      DC    C'02'
DSQ_QMF_V3R1M1    DC    C'03'
DSQ_QMF_V3R2      DC    C'04'
DSQ_QMF_V3R3      DC    C'05'
DSQ_QMF_V6R1      DC    C'06'
DSQ_QMF_V7R1      DC    C'07'
DSQ_QM4_V1R1      DC    C'01'
DSQ_QMF_V7R2      DC    C'08'
DSQ_QMF_V8R1      DC    C'09'
DSQ_QMF_V9R1      DC    C'10'
DSQ_QMF_V10R1     DC    C'11'
*
* Extended parameter data types
*
DSQ_VARIABLE_CHAR DC    C'CHAR'
DSQ_VARIABLE_FINT DC    C'FINT'
*
* Return codes
*
DSQ_SUCCESS       EQU    0
DSQ_WARNING       EQU    4
DSQ_FAILURE       EQU    8
DSQ_SEVERE        EQU    16
*
* Instance ID values
*
DSQ_CONTINUE      EQU    0
*
* Cancel indicator
*
DSQ_CANCEL_YES    EQU    C'1'
DSQ_CANCEL_NO     EQU    C'0'
*
*
DSQ_INTERACTIVE   EQU    C'1'
DSQ_BATCH         EQU    C'2'
*
DSQ_YES           EQU    C'1'
DSQ_NO           EQU    C'2'
*
*****
* Callable interface communications area
*****
DSQCOMM          DSECT
DSQ_RETURN_CODE  DS    F          FUNCTION RETURN CODE
DSQ_INSTANCE_ID  DS    F          ID ESTABLISHED IN START CMD
DSQ_COMM_LEVEL   DS    CL12       COMMUNICATIONS LEVEL ID
DSQ_PRODUCT      DS    CL2        QUERY PRODUCT ID
DSQ_PRODUCT_RELEASE DS    CL2     QUERY PRODUCT RELEASE ID
DSQ_RESERVE1     DS    CL28       RESERVED
DSQ_MESSAGE_ID   DS    CL8        COMPLETION MESSAGE ID
DSQ_Q_MESSAGE_ID DS    CL8        QUERY MESSAGE ID
DSQ_START_PARM_ERROR DS    CL8    START PARAMETER IN ERROR
DSQ_CANCEL_IND   DS    C          CMD CANCEL INDICATOR
DSQ_RESERVE2     DS    CL23       RESERVED
DSQ_RESERVE3     DS    CL156      RESERVED
DSQ_MESSAGE_TEXT DS    CL128      COMPLETION MESSAGE
DSQ_Q_MESSAGE_TEXT DS    CL128    QUERY MESSAGE
          SPACE 1
DSQCOMM_LEN      EQU    *-DSQCOMM  LENGTH OF DSQCOMM AREA
          MEND

```

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5
 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie
 par une fonction NLF (National Language Feature).

Exécution de vos programmes Assembler dans CICS

Une fois que vous avez écrit votre programme, vous devez le convertir, l'assembler et procéder à l'édition de ses liens avant de pouvoir l'exécuter.

Les commandes REXX JCL et CLIST présentées dans ces exemples ne sont pas fournies avec QMF, mais vous pouvez les copier à partir de cet emplacement, en les modifiant en fonction de vos besoins.

Lorsque vous convertissez un programme qui utilise l'interface appelable QMF, que vous l'assemblez et que vous procédez à l'édition de ses liens, tenez compte des points suivants :

- La zone de communication d'interface, DSQCOMMA, doit être accessible lors de l'étape d'assemblage ou copiée dans votre programme en tant que DSECT.
- Le module d'interface QMF, DSQCIA, doit être accessible lors de l'étape d'édition des liens de votre programme.

La commande JCL présentée ici constitue un exemple d'utilisation de la procédure DFHEBTAL, qui est fournie avec CICS.

```
//sampasm JOB
// EXEC PROC=DFHEBTAL
//TRN.SYSIN DD *
*ASM XOPTS(options du convertisseur CICS.....)
.
.
.
.
.
.
.
.
.
.
/*
/* Permet d'accéder à la macro de communications QMF DSQCOMM
//ASM.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Permet d'accéder au module d'interface QMF
/* Allocation pour votre bibliothèque CICS
//LKED.CICSLOAD DD
/* Allocation pour votre bibliothèque cible
//LKED.SYSLMOD DD
/* Allocation pour la bibliothèque de modules de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE CICSLOAD(DFHEAI)
INCLUDE CICSLOAD(DFHEAI0)
INCLUDE QMFLOAD(DSQCIA)
ORDER DFHEAI,DFHEAI0
ENTRY sampasm
MODE AMODE(31) RMODE(31)
NAME sampasm(R)
/*
```

Exécution de vos programmes Assembler dans TSO

Vous devez assembler votre programme et procéder à l'édition de ses liens avant de pouvoir l'exécuter dans TSO.

Assemblage et édition de liens dans TSO

Cette liste présente un exemple de travail qui assemble votre programme et procède à l'édition de ses liens. Certains paramètres peuvent varier d'une installation QMF à l'autre.

```

//sampasm JOB
//STEP1 EXEC PROC=ASMACL
/* Permet d'accéder à la macro de communications QMF DSQCOMM
//C.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//C.SYSIN DD *
.
.
.
    Votre programme ou copie de l'exemple QMF DSQABFA
.
.
.
/*
/* Permet d'accéder au module d'interface QMF
/* Allocation pour votre bibliothèque cible
//L.SYSLMOD DD
/* Allocation pour la bibliothèque de chargement QMF
//L.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//L.SYSIN DD *
        INCLUDE QMFLOAD(DSQCIA)
        ENTRY sampasm
        MODE AMODE(31) RMODE(31)
        NAME sampasm(R)
/*

```

Exécution dans TSO avec ISPF

Une fois que votre programme est assemblé correctement, vous pouvez l'exécuter sous ISPF.

Exécutez votre programme dans TSO sous ISPF en écrivant un programme similaire à la commande CLIST présentée ici :

```

PROC 0
CONTROL ASIS
/*****
/* Spécifiez une liste d'attributs pour les allocations d'ensembles de données */
/*****
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)
/*****
/* Ensembles de données utilisés par TSO
/*****
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****
/* Ensembles de données utilisés par ISPF
/*****
ALLOC FI(ISPLLIB) SHR REUSE +

        DA('QMF1210.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
        DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
        DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
        DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
        DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* Ensembles de données QMF/GDDM
/*****
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE

```

```

ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme comme la boîte de dialogue ISPF initiale */
/*****/
ISPSTART PGM(sampasm) NEWAPPL(DSQE)
EXIT CODE(4)

```

L'instruction EXIT CODE(4) supprime l'écran des dispositions ISPF.

Exécution dans TSO sans ISPF

Une fois que votre programme est assemblé correctement, vous pouvez l'exécuter sans ISPF.

Exécutez votre programme dans TSO sans ISPF en écrivant un programme similaire à la commande CLIST présentée ici :

```

PROC 0
CONTROL ASIS
/*****/
/* Remarque : Les bibliothèques de chargement QMF, */
/* DB2 et GDDM doivent être allouées */
/* avant l'exécution de cette commande CLIST. */
/* Le nom de la bibliothèque de chargement est "QMF1210.SDSQLOAD". */
/*****/
/* Spécifiez une liste d'attributs pour les allocations */
/* d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/

```

```

/* Démarrez votre programme à l'aide de la commande TSO CALL*/
/*****
CALL sampasm
EXIT CODE(0)

```

Interface en langage C

Vous pouvez utiliser le langage C avec l'interface callable dans QMF.

Mappage de la zone de communication d'interface pour le langage C (DSQCOMM)

DSQCOMM, qui est fourni avec QMF, offre un mappage DSQCOMM pour les programmes en langage C.

Le tableau ci-dessous affiche les valeurs pour DSQCOMM.

Tableau 45. Zone de communication d'interface pour DSQCOMM

Nom de structure	Type de données	Description
DSQ_RETURN_CODE	entier long signé	Indique l'état d'une commande QMF après son exécution Les valeurs possibles sont les suivantes : DSQ_SUCCESS Exécution réussie de la demande DSQ_WARNING Achèvement normal avec des avertissements DSQ_FAILURE La commande ne s'est pas exécutée correctement DSQ_SEVERE Erreur grave ; la session QMF a pris fin
DSQ_INSTANCE_ID	entier long signé	Identificateur défini par QMF lors de l'exécution de la commande START
DSQ_COMM_LEVEL	caractère, de longueur 12	Identifie le niveau de DSQCOMM Dans votre application, incluez des instructions qui initialisent cette variable sur la valeur affectée à DSQ_CURRENT_COMM_LEVEL avant d'émettre la commande QMF START.
DSQ_PRODUCT	caractère, de longueur 2	Identifie le produit de requête IBM en cours d'utilisation Les variables commençant par DSQ_QMF indiquent les versions de QMF for TSO et QMF for CICS.
DSQ_PRODUCT_RELEASE	caractère, de longueur 2	Niveau d'édition de QMF en cours d'utilisation La variable DSQ_QMF_V11R1 indique QMF Version 12.1.
DSQ_RESERVE1	caractère, de longueur 28	Réservé à une utilisation ultérieure
DSQ_MESSAGE_ID	caractère, de longueur 8	ID de message d'achèvement
DSQ_Q_MESSAGE_ID	caractère, de longueur 8	ID de message de requête
DSQ_START_PARM_ERROR	caractère, de longueur 8	Paramètre erroné lorsque la commande START a échoué en raison d'une erreur de paramètre

Tableau 45. Zone de communication d'interface pour DSQCOMM (suite)

Nom de structure	Type de données	Description
DSQ_CANCEL_IND	caractère, de longueur 1	Contient l'une des deux valeurs ci-dessous, selon que l'utilisateur a effectué une annulation pendant l'exécution d'une commande QMF : <ul style="list-style-type: none"> • DSQ_CANCEL_YES • DSQ_CANCEL_NO
DSQ_RESERVE2	caractère, de longueur 23	Réservé à une utilisation ultérieure
DSQ_RESERVE3	caractère, de longueur 156	Réservé à une utilisation ultérieure
DSQ_MESSAGE_TEXT	caractère, de longueur 128	Texte du message d'achèvement
DSQ_Q_MESSAGE_TEXT	caractère, de longueur 128	Texte du message de requête

Appels de fonction pour le langage C

QMF fournit deux appels de fonction pour le langage C : DSQCIC et DSQCICE.

DSQCIC

Cet appel est destiné aux commandes QMF qui ne requièrent pas d'accès à des variables de programme d'application. Utilisez cet appel pour la plupart des commandes QMF ; sa syntaxe est la suivante :

```
DSQCIC (&DSQCOMM,&CMDLTH,&CMDSTR)
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre de type long

CMDSTR

Commande QMF à exécuter, spécifiée sous forme de tableau de type de caractères non signés de la longueur indiquée par CMDLTH

La commande QMF doit être en majuscules.

DSQCICE

Cet appel comporte une syntaxe étendue pour les commandes QMF qui requièrent un accès à des variables de programme d'application : START, TRACE et les formats étendus de GET GLOBAL et SET GLOBAL.

```
DSQCICE (&DSQCOMM,&CMDLTH,&CMDSTR,
         &PNUM,&KLTH,&KWORD,
         &VLTH,&VALUE,&VTYPE);
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre entier long

CMDSTR

Commande QMF à exécuter ; tableau de type de caractères non signés. La commande QMF doit être en majuscules.

PNUM

Nombre de mots clés de commande ou de zones de trace ; paramètre entier long

KLTH

Longueur de chaque mot clé (KWORD) ou titre de trace spécifié ; paramètre entier long ou tableau de paramètres entiers longs

KWORD

Mot(s) clé(s) QMF ou adresse des titres de trace ; chacun représente un caractère, un tableau de caractères ou d'adresses vers des titres de trace

VLTH

Longueur de chaque valeur qui est associée au mot clé ou au titre de trace ; paramètre entier long ou tableau de paramètres entiers longs

VALUE

Valeur associée à chaque mot clé ou adresse d'une valeur qui est associée à un titre de trace.

Son type est spécifié dans le paramètre VTYPE et peut être un tableau de caractères non signés, un paramètre entier long ou un tableau de paramètres entiers longs. Pour les données de trace, FINT doit être affecté à VTYPE.

VTYPE

Type de données du contenu du paramètre VALUE.

Ce paramètre comporte l'une des deux valeurs qui sont fournies dans la zone de communication d'interface DSQCOMM :

- DSQ_VARIABLE_CHAR pour le type de caractères non signés
- DSQ_VARIABLE_FINT pour les entiers longs

Toutes les valeurs spécifiées dans la zone VALUE doivent avoir le type de données qui est indiqué par VTYPE.

L'interface en langage C tient compte des considérations suivantes en matière de paramètres :

- Les chaînes de commande et les paramètres de commande START, GET et SET sont tous des chaînes de caractères en entrée. Pour ces chaînes, le langage C nécessite que vous transmettiez une zone de stockage qui se termine par une valeur NULL, qui doit être incluse dans la longueur du paramètre. Utilisez la fonction de longueur chargée au moment de la compilation pour obtenir la longueur du paramètre qui est transmise à l'interface QMF.
- Si la chaîne ne se termine pas par une valeur NULL avant la fin, une erreur est renvoyée par QMF. La valeur NULL (X'00') indique la fin d'une chaîne de caractères.
- Pour les paramètres en langage C qui sont des chaînes de caractères en sortie, notamment les valeurs obtenues à l'aide de la commande GET, QMF déplace les données de la mémoire de stockage QMF vers la zone de stockage de l'application. QMF définit également l'indicateur de type NULL à la fin de la chaîne. Si la chaîne de caractères ne tient pas dans la zone de stockage de l'utilisateur, un message d'avertissement est émis et les données sont tronquées à droite. Un indicateur de type NULL est toujours placé à la fin de la chaîne de données.

Exemple de programmation en langage C

L'exemple de programme en langage C, DSQABFC, est fourni avec QMF. Cet exemple de programme fait partie de la bibliothèque QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale).

L'exemple de programme pour l'interface callable en langage C d'IBM exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

```
/*
*****
/* 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.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
    */

```



```

/*****
/* 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
*/
/*****
    strcpy (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);
    dsqice(&zone_communication,;

        &longueur_cmd,;

        &démarrer_interface_requête[0],

        &nombre_paramètres,;

        &keyword_lengths[0],
        &start_keywords[0],
        &data_lengths[0],
        &start_keyword_values[0],
        &char_data_type[0]);

/*****
/* Set numeric values into query using SET command
*/
/*****
    number_of_parameters = 3;
    command_length = sizeof(set_global_variables);
    strcpy(set_keywords[0], "MYVAR01");
    strcpy(set_keywords[1], "SHORT");
    strcpy(set_keywords[2], "MYVAR03");
    keyword_lengths[0] = SIZE_VAL;
    keyword_lengths[1] = SIZE_VAL;
    keyword_lengths[2] = SIZE_VAL;
    data_lengths[0] = sizeof(long);

```

```

data_lengths[1] = sizeof(long);
data_lengths[2] = sizeof(long);
set_values[0] = 20;
set_values[1] = 40;
set_values[2] = 84;
dsqcice(&zone_communication,;

        &longueur_cmd,;

        &set_global_variables[0],
        &nombre_paramètres,;

        &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);
        dsqcic(&zone_communication,&longueur_cmd,;

                &run_query[0]);

/*****
/* Print the results of the query
*/
/*****
        command_length = sizeof(print_report);
        dsqcic(&zone_communication,&longueur_cmd,;

                &print_report[0]);

/*****
/* End the query interface session
*/
/*****
        command_length = sizeof(end_query_interface);
        dsqcic(&zone_communication,&longueur_cmd,;

                &end_query_interface[0]);
        exit(0);
}

```

DSQCOMM pour le langage C

Le fichier de la zone de communication d'interface pour le langage C se nomme DSQCOMM.C.

Le fichier d'inclusion DSQCOMM.C présenté ici est fourni avec QMF.

```

/*****
/* C include for query callable interface
*/
/*****

/* Structure declare for communications area
*/

struct dsqcomm {
        long int dsq_return_code; /* Function return code */
        long int dsq_instance_id; /* ID established in START cmd*/
        char dsq_comm_level[12]; /* Communications level id */
        char dsq_product[2]; /* Query product id */
        char dsq_product_release[2]; /* Query product release */
        char dsq_reserve1[28]; /* Reserved */
}

```

```

        char dsq_message_id[8];      /* Completion message ID      */
        char dsq_q_message_id[8];   /* Query message ID           */
        char dsq_start_parm_error[8]; /* Start parameter in error */
        char dsq_cancel_ind[1];     /* Cmd cancelled indicator   */
                                   /* 1 = cancelled, 0 = not cancelled*/
        char dsq_reserve2[23];      /* RESERVED AREAS           */
        char dsq_reserve3[156];
        char dsq_message_text[128]; /* Message text              */
        char dsq_q_message_text[128]; /* Query message text       */
    };

/* RETURN CODES */

#define DSQ_SUCCESS          0
#define DSQ_WARNING         4
#define DSQ_FAILURE         8
#define DSQ_SEVERE          16

/* Communications Level */

#define DSQ_CURRENT_COMM_LEVEL "DSQL>001002<"

/* Query Product Codes */

#define DSQ_QRW              "01"
#define DSQ_QMF              "02"
#define DSQ_QM3              "03"

/* Query Product Release Levels */

#define DSQ_QRW_V1R2         "01"
#define DSQ_QRW_V1R3         "02"
#define DSQ_QMF_V2R4         "01"
#define DSQ_QMF_V3R1         "02"
#define DSQ_QMF_V3R1M1       "03"
#define DSQ_QMF_V3R2         "04"
#define DSQ_QMF_V3R3         "05"
#define DSQ_QMF_V6R1         "06"
#define DSQ_QMF_V7R1         "07"
#define DSQ_QM4_V1R1         "01"
#define DSQ_QMF_V7R2         "08"
#define DSQ_QMF_V8R1         "09"
#define DSQ_QMF_V9R1         "10"
#define DSQ_QMF_V10R1        "11"

/* INSTANCE CODES */

#define DSQ_CONTINUE         0

/* CANCELLED INDICATOR */

#define DSQ_CANCEL_YES       "1"
#define DSQ_CANCEL_NO        "0"

/* VARIABLE TYPES */

#define DSQ_VARIABLE_CHAR    "CHAR"
#define DSQ_VARIABLE_FINT    "FINT"

#define DSQ_INTERACTIVE      "1"
#define DSQ_BATCH             "2"

#define DSQ_YES               "1"
#define DSQ_NO                 "2"

/* Call interface structure */

```

```

/* Calling format for normal call with 3 parameters          */
#define dsqcic(parm1, parm2, parm3 )\
    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)

```

Exécution de vos programmes en langage C dans CICS

Une fois que vous avez écrit votre programme en langage C, vous devez le convertir, le compiler et procéder à l'édition de ses liens avant de pouvoir l'exécuter.

Les exemples ci-dessous montrent les étapes nécessaires à la conversion du programme, sa compilation et l'édition de ses liens. Les commandes REXX JCL et CLIST présentées dans ces exemples ne sont pas fournies avec QMF, mais vous pouvez les copier à partir de cet emplacement, en les modifiant en fonction de vos besoins.

Lorsque vous convertissez un programme qui utilise l'interface callable QMF, que vous le compilez et que vous procédez à l'édition de ses liens sous CICS, tenez compte des points suivants :

- La zone de communication d'interface DSQCOMM doit être accessible lors de l'étape de compilation ou copiée dans votre programme.
- Le module d'interface QMF DSQCICX doit être accessible lors de l'étape d'édition des liens de votre programme.
- L'édition des liens des programmes écrits en langage C doit être réalisée avec AMODE=31.

L'exemple présenté ici utilise la procédure DFHYITDL, qui est fournie avec CICS.

```

//sampleC JOB
// EXEC PROC=DFHYITDL
//TRN.SYSIN DD *
.
.
.
Votre programme ou copie de l'exemple QMF DSQABFC
.
.
.
/*
/* Permet d'accéder à la macro de communications QMF DSQCOMM
/*C.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Allocation pour la bibliothèque cible
//LKED.SYSLMOD DD
/* Allocation pour la bibliothèque de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCICX)
NAME sampleC(R)
/*

```

Exécution de vos programmes en langage C dans TSO

Pour exécuter votre programme en langage C dans TSO, compilez-le et procédez à l'édition de ses liens, puis lancez-le avec ou sans ISPF.

Compilation et édition de liens dans TSO

Vous devez compiler votre programme C et procéder à l'édition de ses liens avant de pouvoir l'exécuter dans TSO.

Ce travail compile votre application d'interface callable et procède à l'édition de ses liens à l'aide du compilateur C IBM pour z/OS. Certains paramètres peuvent varier d'une installation QMF à l'autre.

```
//sampleC JOB
//STEP1 EXEC PROC=EDCCB,
// INFILE='nom du fichier contenant le code source',
// OUTFILE='nom du fichier contenant l'exécutable'
/* Permet d'accéder à la macro de communications QMF DSQCOMM
//COMPILE.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//BIND.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//BIND.SYSIN DD *
INCLUDE QMFLOAD(DSQCICX)
/*
```

Exécution de vos programmes dans TSO sans ISPF

Une fois que votre programme en langage C est compilé correctement, vous pouvez l'exécuter sans ISPF.

Exécutez votre programme dans TSO sans ISPF en écrivant un programme similaire à la commande CLIST présentée :

```
PROC 0
CONTROL ASIS
/*****
/* Remarque : Les bibliothèques de chargement QMF, */
/* DB2, GDDM et C doivent être */
/* allouées avant l'exécution de cette commande CLIST. */
/* Le nom de la bibliothèque de chargement est "QMF1210.SDSQLOAD". */
/*****
/* Spécifiez une liste d'attributs pour les allocations */
/* d'ensembles de données */
/*****
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)
/*****
/* Ensembles de données utilisés par TSO */
/*****
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****
/* Ensembles de données QMF/GDDM */
/*****
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****
/* Ensembles de données utilisés par QMF */
/*****
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
```

```

ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****
/* Démarrez votre programme à l'aide de la commande TSO CALL*/
/*****
CALL sampleC
EXIT CODE(0)

```

Exécution de vos programmes dans TSO sous ISPF

Une fois que votre programme en langage C est compilé correctement, vous pouvez l'exécuter sous ISPF.

Exécutez votre programme dans TSO sous ISPF en écrivant un programme similaire à la commande CLIST d'exécution de DSQABFC présentée ici :

```

PROC 0
CONTROL ASIS
/*****
/* Spécifiez une liste d'attributs pour les allocations d'ensembles de données */
/*****
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)
/*****
/* Ensembles de données utilisés par TSO */
/*****
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****
/* Ensembles de données utilisés par ISPF */
/*****
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
        'EDC.SEDCLINK','PLI.SIBMLINK')
ALLOC FI(ISPLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****
/* Ensembles de données QMF/GDDM */
/*****
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****
/* Ensembles de données utilisés par QMF */
/*****
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****

```

```

/* Démarrez votre programme comme la boîte de dialogue ISPF initiale      */
/*****                                                                    */
ISPSTART PGM(samp1eC) NEWAPPL(DSQE)
EXIT CODE(4)

```

L'instruction EXIT CODE(4) supprime l'écran des dispositions ISPF.

Interface en langage COBOL

Vous pouvez utiliser le langage COBOL avec l'interface appellable dans QMF.

Mappage de la zone de communication d'interface pour le langage COBOL (DSQCOMMB)

DSQCOMMB, qui est fourni avec QMF, offre un mappage DSQCOMM pour les programmes en langage COBOL.

Le tableau ci-dessous affiche les valeurs pour DSQCOMMB.

Tableau 46. Zone de communication d'interface pour le langage COBOL (DSQCOMMB)

Nom de structure	Type de données	Description
DSQ-RETURN-CODE	PIC 9(8)	Indique l'état d'une commande QMF après son exécution Les valeurs possibles sont les suivantes : DSQ-SUCCESS Exécution réussie de la demande DSQ-WARNING Achèvement normal avec des avertissements DSQ-FAILURE La commande ne s'est pas exécutée correctement DSQ-SEVERE Erreur grave ; la session QMF a pris fin
DSQ-INSTANCE-ID	PIC 9(8)	Identificateur défini par QMF lors de l'exécution de la commande START
DSQ-COMM-LEVEL	PIC X(12)	Identifie le niveau de DSQCOMM Dans votre application, incluez des instructions qui initialisent cette variable sur la valeur affectée à DSQ_CURRENT_COMM_LEVEL avant d'émettre la commande QMF START.
DSQ-PRODUCT	PIC X(2)	Identifie le produit de requête IBM en cours d'utilisation Les variables commençant par DSQ-QMF indiquent les versions de QMF for TSO et QMF for CICS.
DSQ-PRODUCT-RELEASE	PIC X(2)	Niveau d'édition de QMF en cours d'utilisation La variable DSQ_QMF_V11R1 indique QMF Version 12.1.
DSQ-RESERVE1	PIC X(28)	Réservé à une utilisation ultérieure
DSQ-MESSAGE-ID	PIC X(8)	ID de message d'achèvement
DSQ-Q-MESSAGE-ID	PIC X(8)	ID de message de requête

Tableau 46. Zone de communication d'interface pour le langage COBOL (DSQCOMMB) (suite)

Nom de structure	Type de données	Description
DSQ-START-PARM-ERROR	PIC X(8)	Paramètre erroné lorsque la commande START a échoué en raison d'une erreur de paramètre
DSQ-CANCEL-IND	PIC X(1)	Contient l'une des deux valeurs ci-dessous, selon que l'utilisateur a effectué une annulation pendant l'exécution d'une commande QMF : <ul style="list-style-type: none"> • DSQ-CANCEL-YES • DSQ-CANCEL-NO
DSQ-RESERVE2	PIC X(23)	Réservé à une utilisation ultérieure
DSQ-RESERVE3	PIC X(156)	Réservé à une utilisation ultérieure
DSQ-MESSAGE-TEXT	PIC X(128)	Texte du message d'achèvement
DSQ-Q-MESSAGE-TEXT	PIC X(128)	Texte du message de requête

Appels de fonction pour le langage COBOL

QMF fournit un appel de fonction (DSQCIB) pour le langage COBOL. L'appel de fonction comporte deux formats : DSQCIB et DSQCIB (format étendu).

DSQCIB

Cet appel est destiné aux commandes QMF qui ne requièrent pas d'accès à des variables de programme d'application. Utilisez cet appel pour la plupart des commandes QMF.

```
CALL DSQCIB USING DSQCOMM CMDLTH CMDSTR
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre entier

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par CMDLTH

DSQCIB (format étendu)

Cet appel comporte une syntaxe étendue pour les commandes QMF qui requièrent un accès à des variables de programme d'application : START, TRACE et les formats étendus de GET GLOBAL et SET GLOBAL.

```
DSQCIB USING
    DSQCOMM CMDLTH CMDSTR
    PNUM KLTH KWORD VLTH VALUE VTYPE
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre entier

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par CMDLTH

PNUM

Nombre de mots clés de commande ou de zones de trace ; paramètre entier

KLTH

Longueur de chaque mot clé ou titre de trace spécifié ; paramètre entier ou tableau de paramètres entiers

KWORD

Mot(s) clé(s) QMF ou adresse des titres de trace.

Chacun représente un caractère, un tableau de caractères ou d'adresses vers des titres de trace dont les longueurs sont spécifiées par KLTH. Si tous les mots clés sont de même longueur, vous pouvez utiliser un tableau de caractères.

VLTH

Longueur de chaque valeur qui est associée au mot clé ou au titre de trace ; paramètre entier ou tableau de paramètres entiers

VALUE

Valeur associée à chaque mot clé ou adresse d'une valeur qui est associée à un titre de trace.

Son type est spécifié dans le paramètre VTYPE et peut être un caractère, un tableau de caractères, un paramètre entier ou un tableau de paramètres entiers. Pour les données de trace, FINT doit être affecté à VTYPE.

VTYPE

Type de données du contenu du paramètre VALUE.

Ce paramètre comporte l'une des deux valeurs qui sont fournies dans la zone de communication DSQCOMMB :

- DSQ-VARIABLE-CHAR pour les valeurs de type caractère
- DSQ-VARIABLE-FINT pour les valeurs entières

Toutes les valeurs spécifiées dans la zone VALUE doivent avoir le type de données qui est indiqué par VTYPE.

Service ISPF LIBDEF avec le langage COBOL

Modifiez les appels dynamiques vers l'interface QMF (DSQCIB) en appels statiques pour utiliser la fonction LIBDEF dans votre application QMF.

Par exemple, examinons l'instruction d'identificateur d'appel suivante :

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

Vous pouvez remplacer cette instruction par son format littéral d'appel comme suit :

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

Exemple de programmation COBOL

L'exemple de programme en COBOL, DSQABFCO, est fourni avec QMF. Cet exemple de programme fait partie de la bibliothèque QMF1210.SDSQSAP n (où n correspond à l'identificateur de langue nationale).

L'exemple de programme pour l'interface callable en COBOL exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

Pour CICS, l'instruction STOP RUN doit être modifiée en instruction GOBACK.

```

*****
*   The following is a COBOL version of the query           *
*   callable interface *** DSQABFCO **.                    *
*****
IDENTIFICATION DIVISION.
PROGRAM-ID. DSQABFCO.
DATE-COMPILED.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*****
* Copy DSQCOMMB definition - contains query interface variables *
*****
COPY DSQCOMMB.

* Query interface commands
01 STARTQI      PIC X(5)  VALUE "START".
01 SETG         PIC X(10) VALUE "SET GLOBAL".
01 QUERY        PIC X(12) VALUE "RUN QUERY Q1".
01 REPT         PIC X(22) VALUE "PRINT REPORT (FORM=F1 ".
01 ENDQI        PIC X(4)  VALUE "EXIT".

* Query command length
01 QICLTH       PIC 9(8)  USAGE IS COMP-4.
* Number of variables
01 QIPNUM       PIC 9(8)  USAGE IS COMP-4.
* Keyword variable lengths
01 QIKLTHS.
03 KLTHS        PIC 9(8)  OCCURS 10 USAGE IS COMP-4.
* Value Lengths
01 QIVLTHS.
03 VLTHS        PIC 9(8)  OCCURS 10 USAGE IS COMP-4.
* Start command keyword
01 SNAMES.
03 SNAME1       PIC X(8)  VALUE "DSQSMODE".
* Start command keyword value
01 SVALUES.
03 SVALUE1      PIC X(11) VALUE "INTERACTIVE".
* Set GLOBAL command variable names to set
01 VNAMES.
03 VNAME1       PIC X(7)  VALUE "MYVAR01".
03 VNAME2       PIC X(5)  VALUE "SHORT".
03 VNAME3       PIC X(7)  VALUE "MYVAR03".
* Variable value parameters
01 VVALUES.
03 VVALS        PIC 9(8)  OCCURS 10 USAGE IS COMP-4.

01 TEMP         PIC 9(8)          USAGE IS COMP-4.
PROCEDURE DIVISION.
*
* Start a query interface session

```

```

MOVE DSQ-CURRENT-COMM-LEVEL TO DSQ-COMM-LEVEL.
MOVE 5 TO QICLTH.
MOVE 8 TO KLTHS(1).
MOVE 11 TO VLTHS(1).
MOVE 1 TO QIPNUM.
CALL DSQCIB USING DSQCOMM, QICLTH, STARTQI,
               QIPNUM, QIKLTHS, SNAMES,
*
* Set numeric values into query variables using SET GLOBAL command
MOVE 10 TO QICLTH.
MOVE 7 TO KLTHS(1).
MOVE 5 TO KLTHS(2).
MOVE 7 TO KLTHS(3).
MOVE 4 TO VLTHS(1).
MOVE 4 TO VLTHS(2).
MOVE 4 TO VLTHS(3).
MOVE 20 TO VVALS(1).
MOVE 40 TO VVALS(2).
MOVE 84 TO VVALS(3).
MOVE 3 TO QIPNUM.
CALL DSQCIB USING DSQCOMM, QICLTH, SETG,
               QIPNUM, QIKLTHS, VNAMES,
               QIVLTHS, VVALUES, DSQ-VARIABLE-FINT.
*
* Run a query
MOVE 12 TO QICLTH.
CALL DSQCIB USING DSQCOMM, QICLTH, QUERY.
*
* Print the results of the query
MOVE 22 TO QICLTH.
CALL DSQCIB USING DSQCOMM, QICLTH, REPT.
*
* End the query interface session
MOVE 4 TO QICLTH.
CALL DSQCIB USING DSQCOMM, QICLTH, ENDQI.

STOP RUN.

```

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

DSQCOMM pour COBOL

Le fichier de la zone de communication d'interface pour le langage COBOL se nomme DSQCOMMB.

Le fichier d'inclusion DSQCOMMB présenté ici est fourni avec QMF :

```

*****
* COBOL INCLUDE FOR QUERY CALLABLE INTERFACE
*****

* STRUCTURE DECLARE FOR COMMUNICATIONS AREA

01 DSQCOMM.

    03 DSQ-RETURN-CODE PIC 9(8) USAGE IS COMP.
*                               FUNCTION RETURN CODE           *
    03 DSQ-INSTANCE-ID PIC 9(8) USAGE IS COMP.
*                               IDENTIFIER FROM START CMD       *
    03 DSQ-COMM-LEVEL  PIC X(12).
*                               COMMUNICATIONS LEVEL            *
    03 DSQ-PRODUCT     PIC X(2).
*                               QUERY PRODUCT ID                *
    03 DSQ-PRODUCT-RELEASE PIC X(2).

```

```

*          03 DSQ-RESERVE1 PIC X(28).          QUERY PRODUCT RELEASE      *
*          03 DSQ-MESSAGE-ID PIC X(8).        RESERVED AREA                  *
*          03 DSQ-Q-MESSAGE-ID PIC X(8).      COMPLETION MESSAGE ID        *
*          03 DSQ-START-PARM-ERROR PIC X(8).  QUERY MESSAGE ID             *
*          03 DSQ-CANCEL-IND PIC X(1).        START PARAMETER IN ERROR     *
*          1 = COMMAND CANCELLED              *
*          0 = COMMAND NOT CANCELLED         *
*          03 DSQ-RESERVE2 PIC X(23).        RESERVED AREA                  *
*          03 DSQ-RESERVE3 PIC X(156).       RESERVED AREA                  *
*          03 DSQ-MESSAGE-TEXT PIC X(128).   QMF MESSAGE TEXT             *
*          03 DSQ-Q-MESSAGE-TEXT PIC X(128). QMF QUERY MESSAGE TEXT      *
*          512 BYTES TOTAL                    *

* VALUES FOR DSQ-RETURN-CODE

01 DSQ-SUCCESS PIC 9(8) USAGE IS COMP VALUE 0.
01 DSQ-WARNING PIC 9(8) USAGE IS COMP VALUE 4.
01 DSQ-FAILURE PIC 9(8) USAGE IS COMP VALUE 8.
01 DSQ-SEVERE PIC 9(8) USAGE IS COMP VALUE 16.

* VALUES FOR DSQ-INSTANCE-ID

01 DSQ-CONTINUE PIC 9(8) USAGE IS COMP VALUE 0.

* VALUES FOR DSQ-COMM-LEVEL

01 DSQ-CURRENT-COMM-LEVEL PIC X(12) VALUE "DSQL>001002<".

* VALUES FOR DSQ-PRODUCT

01 DSQ-QRW PIC X(2) VALUE "01".
01 DSQ-QMF PIC X(2) VALUE "02".
01 DSQ-QM4 PIC X(2) VALUE "03".

* VALUES FOR DSQ-PRODUCT-RELEASE

01 DSQ-QRW-V1R2 PIC X(2) VALUE "01".
01 DSQ-QRW-V1R3 PIC X(2) VALUE "02".
01 DSQ-QMF-V2R4 PIC X(2) VALUE "01".
01 DSQ-QMF-V3R1 PIC X(2) VALUE "02".
01 DSQ-QMF-V3R1M1 PIC X(2) VALUE "03".
01 DSQ-QMF-V3R2 PIC X(2) VALUE "04".
01 DSQ-QMF-V3R3 PIC X(2) VALUE "05".
01 DSQ-QMF-V6R1 PIC X(2) VALUE "06".
01 DSQ-QMF-V7R1 PIC X(2) VALUE "07".
01 DSQ-QM4-V1R1 PIC X(2) VALUE "01".
01 DSQ-QMF-V7R2 PIC X(2) VALUE "08".
01 DSQ-QMF-V8R1 PIC X(2) VALUE "09".
01 DSQ-QMF-V9R1 PIC X(2) VALUE "10".
01 DSQ-QMF-V10R1 PIC X(2) VALUE "11".

* VALUES FOR DSQ-CANCEL-IND

01 DSQ-CANCEL-YES PIC X(1) VALUE "1".
01 DSQ-CANCEL-NO PIC X(1) VALUE "0".

* VALUES FOR MODE

```

```

01 DSQ-INTERACTIVE    PIC X(1) VALUE "1".
01 DSQ-BATCH          PIC X(1) VALUE "2".

*  VALUES YES AND NO

01 DSQ-YES            PIC X(1) VALUE "1".
01 DSQ-NO             PIC X(1) VALUE "2".

*  CALLABLE INTERFACE PROGRAM NAME

01 DSQCIB             PIC X(6) VALUE "DSQCIB".

*  VALUES FOR VARIABLE TYPE ON CALL PARAMETER

01 DSQ-VARIABLE-CHAR PIC X(4) VALUE "CHAR".
01 DSQ-VARIABLE-FINT PIC X(4) VALUE "FINT".

```

Considérations relatives à l'exécution du programme d'interface callable COBOL

Prêtez attention aux détails relatifs à l'exécution d'un programme COBOL utilisant l'interface callable QMF.

Lorsque vous convertissez un programme qui utilise l'interface callable QMF, que vous le compilez et que vous procédez à l'édition de ses liens, tenez compte des points suivants :

- Environnement d'exécution

QMF s'exécute en tant que programme Assembler dans l'environnement COBOL. Votre programme COBOL doit appeler le programme d'interface QMF, DSQCIB, à l'aide d'un appel dynamique en langage COBOL.
- Utilisation ou non de guillemets ou d'apostrophes

Vous devez utiliser des guillemets (") ou des apostrophes (') pour délimiter des littéraux dans un programme COBOL. Vous pouvez affecter le délimiteur de votre choix au processus de conversion CICS et au compilateur COBOL en indiquant QUOTE ou APOST. Vérifiez que l'option APOST ou QUOTE en vigueur pour le compilateur COBOL correspond à celle du convertisseur CICS.

La zone de communication (DSQCOMMB) et l'exemple de programme COBOL (DSQABFCO) fournis par QMF utilisent des guillemets pour délimiter les littéraux. Si votre site ou programme utilise des apostrophes au lieu de guillemets, modifiez DSQCOMMB ou copiez la structure dans votre programme, en remplaçant les guillemets par des apostrophes.
- Disponibilité de la zone de communication (DSQCOMMB)

La zone de communication DSQCOMMB doit être accessible lors de l'étape de compilation COBOL ou copiée dans votre programme en tant que structure de contrôle.
- Disponibilité du module d'interface (DSQCIB)

Le module d'interface QMF doit être accessible lors de l'étape d'édition des liens de votre programme.

Exécution de vos programmes en COBOL dans CICS

Une fois que vous avez écrit votre programme en COBOL, vous devez le convertir, le compiler et procéder à l'édition de ses liens avant de pouvoir l'exécuter.

La commande JCL présentée dans ces exemples n'est pas fournie avec QMF, mais vous pouvez la copier à partir de cet emplacement, en la modifiant en fonction de vos besoins.

L'exemple montre la procédure DFHEBTVL, qui est fournie avec CICS et prend en charge le langage COBOL.

```
//samCOBOL JOB
//          EXEC PROC=DFHEBTVL
//TRN.SYSIN DD *
*CBL      XOPTS(options du convertisseur CICS ...QUOTE COBOL2)
.
.
.
      Votre programme ou copie de l'exemple QMF DSQABFCO
.
.
.
/*
/** Permet d'accéder à la macro de communications QMF DSQCOMMB
//COB.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/** Allocation pour la bibliothèque cible
//LKED.SYSLMOD DD
/** Allocation pour la bibliothèque de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE CICSLOAD(DFHECI)
          INCLUDE QMFLOAD(DSQCIB)
          ORDER DFHECI
          ENTRY samCOBOL
          MODE AMODE(31) RMODE(31)
          NAME samCOBOL(R)
/*
```

Exécution de vos programmes en COBOL dans TSO

Pour exécuter votre programme en COBOL dans TSO, compilez-le et procédez à l'édition de ses liens, puis lancez-le avec ou sans ISPF.

Compilation et édition de liens dans TSO

Vous devez compiler votre programme en COBOL et procéder à l'édition de ses liens avant de pouvoir l'exécuter dans TSO.

Ce travail utilise le compilateur COBOL pour compiler votre application d'interface callable. Il procède ensuite à l'édition des liens de votre application. Certains paramètres peuvent varier d'une installation QMF à l'autre.

```
//samCOBOL JOB
//STEP1     EXEC PROC=IGYWCL
/** Permet d'accéder à la macro de communications QMF DSQCOMM
//COBOL.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//COBOL.SYSIN DD *
.
.
.
      Votre programme ou copie de l'exemple QMF DSQABFCO
.
.
.
/** Permet d'accéder au module d'interface QMF
/** Allocation pour la bibliothèque cible
//LKED.SYSLMOD DD
/** Allocation pour la bibliothèque de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
          INCLUDE QMFLOAD(DSQCIB)
          ENTRY samCOBOL
          MODE AMODE(31) RMODE(31)
          NAME samCOBOL(R)
/*
```

Exécution de vos programmes dans TSO sans ISPF

Une fois que votre programme en COBOL est compilé correctement, vous pouvez l'exécuter avec JCL sans ISPF.

Exécutez le compilateur COBOL et l'éditeur de liens dans TSO sans ISPF en écrivant un programme similaire à la commande CLIST présentée ici :

```
PROC 0
CONTROL ASIS
/*****/
/* Remarque : Les bibliothèques de chargement QMF, */
/* DB2, GDDM et COBOL doivent être */
/* allouées avant l'exécution de cette commande CLIST. */
/* Le nom de la bibliothèque de chargement est "QMF1210.SDSQLOAD". */
/*****/
/* Spécifiez une liste d'attributs pour les allocations */
/* d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme à l'aide de la commande TSO CALL */
/*****/
CALL samCOBOL
EXIT CODE(0)
```

Exécution de vos programmes dans TSO sous ISPF

Une fois que votre programme en COBOL est compilé correctement, vous pouvez l'exécuter sous ISPF.

Exécutez votre programme dans TSO sous ISPF en écrivant un programme similaire à la commande CLIST présentée ici :

```
PROC 0
CONTROL ASIS
/*****/
/* Spécifiez une liste d'attributs pour les allocations */
/* d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Ensembles de données utilisés par ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLOAD','DSN.DSNEXIT','DSN.DSNLOAD', +
      'PRODUCT.COB2LIB')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme comme la boîte de dialogue */
/* ISPF initiale */
/*****/
ISPSTART PGM(samCOBOL) NEWAPPL(DSQE)
EXIT CODE(4)

```

L'instruction EXIT CODE(4) supprime l'affichage de l'écran des dispositions ISPF.

Interface en langage FORTRAN

Vous pouvez utiliser le langage FORTRAN avec l'interface appellable dans QMF for TSO.

Restriction : Le langage FORTRAN n'étant pas disponible sous CICS, l'interface appellable de QMF pour FORTRAN ne fonctionne pas sous CICS.

Mappage de la zone de communication d'interface pour le langage FORTRAN (DSQCOMMF)

DSQCOMMF, qui est fourni avec QMF, offre un mappage DSQCOMM pour les programmes en langage FORTRAN.

Le tableau ci-dessous affiche les informations pour DSQCOMMF. Vous ne devez pas les modifier.

Tableau 47. Zone de communication d'interface pour le langage FORTRAN (DSQCOMMF)

Nom de structure	Type de données	Description
DSQ_RETURN_CODE	Entier	Indique l'état d'une commande QMF après son exécution Les valeurs possibles sont les suivantes : DSQ_SUCCESS Exécution réussie de la demande DSQ_WARNING Achèvement normal avec des avertissements DSQ_FAILURE La commande ne s'est pas exécutée correctement DSQ_SEVERE Erreur grave ; la session QMF a pris fin
DSQ_INSTANCE_ID	Entier	Identificateur défini par QMF lors de l'exécution de la commande START
DSQ_COMM_LEVEL	CHARACTER(12)	Identifie le niveau de DSQCOMM Dans votre application, incluez des instructions qui initialisent cette variable sur la valeur affectée à DSQ_CURRENT_COMM_LEVEL avant d'émettre la commande QMF START.
DSQ_PRODUCT	CHARACTER(2)	Identifie le produit de requête IBM en cours d'utilisation Les variables commençant par DSQ_QMF indiquent les versions de QMF for TSO.
DSQ_PRODUCT_RELEASE	CHARACTER(2)	Niveau d'édition de QMF en cours d'utilisation La variable DSQ_QMF_V11R1 indique QMF Version 12.1.
DSQ_RESERVE1	CHARACTER(28)	Réservé à une utilisation ultérieure
DSQ_MESSAGE_ID	CHARACTER(8)	ID de message d'achèvement
DSQ_Q_MESSAGE_ID	CHARACTER(8)	ID de message de requête
DSQ_START_PARM_ERROR	CHARACTER(8)	Paramètre erroné lorsque la commande START a échoué en raison d'une erreur de paramètre
DSQ_CANCEL_IND	CHARACTER(1)	Contient l'une des deux valeurs ci-dessous, selon que l'utilisateur a effectué une annulation pendant l'exécution d'une commande QMF : <ul style="list-style-type: none">• DSQ_CANCEL_YES• DSQ_CANCEL_NO
DSQ_RESERVE2	CHARACTER(23)	Réservé à une utilisation ultérieure
DSQ_RESERVE3	CHARACTER(156)	Réservé à une utilisation ultérieure
DSQ_MESSAGE_TEXT	CHARACTER(128)	Texte du message d'achèvement
DSQ_Q_MESSAGE_TEXT	CHARACTER(128)	Texte du message de requête

Appels de fonction pour le langage FORTRAN

QMF fournit deux appels de fonction pour le langage FORTRAN : DSQCIF et DSQCIFE. Les deux appels sont décrits dans la zone de communication (DSQCOMM).

DSQCIF

Cet appel est destiné aux commandes QMF qui ne requièrent pas d'accès à des variables de programme d'application. Utilisez cet appel pour la plupart des commandes QMF.

```
RC = DSQCIF(DSQCOMM,  
+   CMDLTH,  
+   CMDSTR)
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication

CMDLTH

Longueur de la chaîne de commande (**CMDSTR**) ; paramètre entier

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par **CMDLTH**

DSQCIFE

Cet appel comporte une syntaxe étendue pour les commandes qui requièrent un accès à des variables de programme d'application : START, TRACE et les formats étendus de GET GLOBAL et SET GLOBAL.

La syntaxe de cet appel est la suivante :

```
RC = DSQCIFE(DSQCOMM,  
+   CMDLTH,  
+   CMDSTR,  
+   PNUM,  
+   KLTH,  
+   KWORD,  
+   VLTH,  
+   VALUE,  
+   VTYPE)
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (**CMDSTR**) ; paramètre entier

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par **CMDLTH**

PNUM

Nombre de mots clés de commande ou de zones de trace ; paramètre entier

KLTH

Longueur de chaque mot clé ou titre de trace spécifié ; paramètre entier ou tableau de paramètres

KEYWORD

Mot(s) clé(s) QMF ou adresse des titres de trace ; caractère, tableau de caractères ou d'adresses vers des titres de trace dont les longueurs sont spécifiées par **KLTH**.

Vous pouvez utiliser un tableau de caractères si tous les mots clés sont de même longueur. Les mots clés doivent se trouver dans une mémoire contiguë et ne doivent pas être séparés par des délimiteurs spéciaux.

VLTH

Longueur de chaque valeur qui est associée au mot clé ou au titre de trace ; paramètre entier ou tableau de paramètres

VALUE

Valeur associée à chaque mot clé ou adresse d'une valeur qui est associée à un titre de trace.

Son type est spécifié dans le paramètre **VTTYPE** et peut être un caractère, un tableau de caractères, un paramètre entier ou un tableau de paramètres. Pour les données de trace, **FINT** doit être affecté à **VTTYPE**. Dans le cas de valeurs de type caractère, ces dernières doivent se trouver dans une mémoire contiguë et ne doivent pas être séparées par des délimiteurs spéciaux.

VTTYPE

Type de données du contenu du paramètre **VALUE**.

Ce paramètre comporte l'une des deux valeurs qui sont fournies dans la zone de communication **DSQCOMM** :

- **DSQ_VARIABLE_CHAR** pour les valeurs de type caractère
- **DSQ_VARIABLE_FINT** pour les valeurs entières

Toutes les valeurs spécifiées dans la zone **VALUE** doivent avoir le type de données qui est indiqué par **VTTYPE**.

Exemple de programmation en langage FORTRAN

L'exemple de programme, **DSQABFF**, est fourni avec QMF. Cet exemple de programme fait partie de la bibliothèque **QMF1210.SDSQSAPn** (où *n* correspond à l'identificateur de langue nationale).

L'exemple de programme pour l'interface callable en FORTRAN exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

```
C*****
C Sample program: DSQABFF
C FORTRAN version of QMF manager callable interface
C
C Creation Date: 11/21/89
C
C ENVIRONMENT:   API IN FORTRAN
C*****
C
C Processing:
```

- C a. Start a query manager session using the callable interface.
- C
- C b. Set global query manager numeric variables.
- C
- C d. Run a query manager query using the callable interface.
- C
- C e. Print a report using the callable interface.
- C
- C f. Exit the query manager session.
- C
- C Prerequisites:1. Create the SAMPLE database.
- C
- C 2. Create a prompted query, Q1, which has a SELECT state
- C
- C 3. Create a form, F1, that displays data for query Q1.
- C

```

C*****
PROGRAM DSQABFF
C*****
C Include and declare query interface communications area
C*****
INCLUDE (DSQCOMM)

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)

C*****
C Keyword parameter and values for SET command
C*****
CHARACTER SET_KEYWORDS(19)
CHARACTER SET_KEYWORD_1*7,
+ SET_KEYWORD_2*5,
+ SET_KEYWORD_3*7

EQUIVALENCE (SET_KEYWORDS( 1), SET_KEYWORD_1),
+ (SET_KEYWORDS( 8), SET_KEYWORD_2),
+ (SET_KEYWORDS(13), SET_KEYWORD_3)

CHARACTER SET_VALUES(12)
INTEGER*4 SET_VALUE_1,
+ SET_VALUE_2,

```

```

+           SET_VALUE_3

EQUIVALENCE (SET_VALUES(1), SET_VALUE_1),
+           (SET_VALUES(5), SET_VALUE_2),
+           (SET_VALUES(9), SET_VALUE_3)

C*****
C Declare command length and return code variables
C*****
      INTEGER   LEN,
+           RC

C*****
C Initialization
C*****

      DATA START_QUERY_INTERFACE /'START' /
      DATA SET_GLOBAL_VARIABLES /'SET GLOBAL' /
      DATA RUN_QUERY /'RUN QUERY Q1' /
      DATA PRINT_REPORT /'PRINT REPORT (FORM=F1)' /
      DATA END_QUERY_INTERFACE /'EXIT' /

      DATA CHAR_DATA_TYPE /DSQ_VARIABLE_CHAR /
      DATA INT_DATA_TYPE /DSQ_VARIABLE_FINT /

C*****
C Start query session
C*****
      DSQ_COMM_LEVEL = DSQ_CURRENT_COMM_LEVEL
      NUMBER_OF_PARAMETERS = 1
      COMMAND_LENGTH = LEN(START_QUERY_INTERFACE)
      KEYWORD_LENGTHS(1) = LEN(START_KEYWORDS(1))
      DATA_LENGTHS(1) = LEN(START_KEYWORD_VALUES(1))
      START_KEYWORDS(1) = 'DSQSMODE'
      START_KEYWORD_VALUES(1) = 'INTERACTIVE'

      RC = DSQCIFE(DSQCOMM,
+           COMMAND_LENGTH,
+           START_QUERY_INTERFACE,
+           NUMBER_OF_PARAMETERS,
+           KEYWORD_LENGTHS,
+           START_KEYWORDS,
+           DATA_LENGTHS,
+           START_KEYWORD_VALUES,
+           CHAR_DATA_TYPE)

C*****
C Set numeric values into query using SET command
C*****
      NUMBER_OF_PARAMETERS = 3
      COMMAND_LENGTH = LEN(SET_GLOBAL_VARIABLES)
      SET_KEYWORD_1 = 'MYVAR01'
      SET_KEYWORD_2 = 'SHORT'
      SET_KEYWORD_3 = 'MYVAR03'
      KEYWORD_LENGTHS(1) = LEN(SET_KEYWORD_1)
      KEYWORD_LENGTHS(2) = LEN(SET_KEYWORD_2)
      KEYWORD_LENGTHS(3) = LEN(SET_KEYWORD_3)
      DATA_LENGTHS(1) = 4
      DATA_LENGTHS(2) = 4
      DATA_LENGTHS(3) = 4
      SET_VALUE_1 = 20
      SET_VALUE_2 = 40
      SET_VALUE_3 = 84

      RC = DSQCIFE(DSQCOMM,
+           COMMAND_LENGTH,
+           SET_GLOBAL_VARIABLES,

```

```

+          NUMBER_OF_PARAMETERS,
+          KEYWORD_LENGTHS,
+          SET_KEYWORDS,
+          DATA_LENGTHS,
+          SET_VALUES,
+          INT_DATA_TYPE)

C*****
C   Run a query
C*****
      COMMAND_LENGTH = LEN(RUN_QUERY)
      RC = DSQCIF(DSQCOMM,
+             COMMAND_LENGTH,
+             RUN_QUERY)

C*****
C   Print the results of the query
C*****
      COMMAND_LENGTH = LEN(PRINT_REPORT)
      RC = DSQCIF(DSQCOMM,
+             COMMAND_LENGTH,
+             PRINT_REPORT)

C*****
C   End the query interface session
C*****
      COMMAND_LENGTH = LEN(END_QUERY_INTERFACE)
      RC = DSQCIF(DSQCOMM,
+             COMMAND_LENGTH,
+             END_QUERY_INTERFACE)

      END

```

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

DSQCOMM pour le langage FORTRAN

Le fichier de la zone de communication d'interface pour le langage FORTRAN se nomme DSQCOMM.F.

Le fichier d'inclusion DSQCOMM.F présenté ici est fourni avec QMF :

```

C*****
C   FORTRAN include file for callable interface
C*****
C   Return codes
      INTEGER DSQ_SUCCESS, DSQ_WARNING, DSQ_FAILURE, DSQ_SEVERE
      PARAMETER(
+         DSQ_SUCCESS = 0,
+         DSQ_WARNING = 4,
+         DSQ_FAILURE = 8,
+         DSQ_SEVERE = 16)

C   Communications level
      CHARACTER DSQ_CURRENT_COMM_LEVEL*12
      PARAMETER(
+         DSQ_CURRENT_COMM_LEVEL = 'DSQL>001002<')

C   Query product IDs
      CHARACTER DSQ_QRW*2, DSQ_QMF*2, DSQ_QM4*2
      PARAMETER(
+         DSQ_QRW = '01',
+         DSQ_QMF = '02',
+         DSQ_QM4 = '03')

```

```

C      Query product release levels
CHARACTER DSQ_QRW_V1R2*2, DSQ_QRW_V1R3*2,
+        DSQ_QMF_V2R4*2, DSQ_QMF_V3R1*2,
+        DSQ_QMF_V3R1M1*2, DSQ_QMF_V3R2*2,
+        DSQ_QMF_V3R3*2, DSQ_QMF_V6R1*2,
+        DSQ_QMF_V7R1*2, DSQ_QM4_V1R1*2,
+        DSQ_QMF_V7R2*2, DSQ_QMF_V8R1*2,
+        DSQ_QMF_V9R1*2, DSQ_QMF_V10R1*2
PARAMETER(
+        DSQ_QRW_V1R2 = '01',
+        DSQ_QRW_V1R3 = '02',
+        DSQ_QMF_V2R4 = '01',
+        DSQ_QMF_V3R1 = '02',
+        DSQ_QMF_V3R1M1 = '03',
+        DSQ_QMF_V3R2 = '04',
+        DSQ_QMF_V3R3 = '05',
+        DSQ_QMF_V6R1 = '06',
+        DSQ_QMF_V7R1 = '07',
+        DSQ_QM4_V1R1 = '01',
+        DSQ_QMF_V7R2 = '08',
+        DSQ_QMF_V8R1 = '09',
+        DSQ_QMF_V9R1 = '10',
+        DSQ_QMF_V10R1 = '11')

C      Host variable types
CHARACTER DSQ_VARIABLE_CHAR*4, DSQ_VARIABLE_FINT*4
PARAMETER(
+        DSQ_VARIABLE_CHAR = 'CHAR',
+        DSQ_VARIABLE_FINT = 'FINT')

C      Cancel indicator
CHARACTER DSQ_CANCEL_YES, DSQ_CANCEL_NO
PARAMETER(
+        DSQ_CANCEL_YES = '1',
+        DSQ_CANCEL_NO = '0')
CHARACTER DSQCOMM(512)
INTEGER DSQ_RETURN_CODE, DSQ_INSTANCE_ID
CHARACTER DSQ_COMM_LEVEL*12,
+        DSQ_PRODUCT*2,
+        DSQ_PRODUCT_RELEASE*2,
+        DSQ_RESERVE1*28,
+        DSQ_MESSAGE_ID*8,
+        DSQ_Q_MESSAGE_ID*8,
+        DSQ_START_PARM_ERROR*8,
+        DSQ_CANCEL_IND*1,
+        DSQ_RESERVE2*23,
+        DSQ_RESERVE3*156,
+        DSQ_MESSAGE_TEXT*128,
+        DSQ_Q_MESSAGE_TEXT*128

EQUIVALENCE (DSQCOMM( 1), DSQ_RETURN_CODE ),
+ (DSQCOMM( 5), DSQ_INSTANCE_ID ),
+ (DSQCOMM( 9), DSQ_COMM_LEVEL ),
+ (DSQCOMM(21), DSQ_PRODUCT ),
+ (DSQCOMM(23), DSQ_PRODUCT_RELEASE ),
+ (DSQCOMM(25), DSQ_RESERVE1 ),
+ (DSQCOMM(53), DSQ_MESSAGE_ID ),
+ (DSQCOMM(61), DSQ_Q_MESSAGE_ID ),
+ (DSQCOMM(69), DSQ_START_PARM_ERROR ),
+ (DSQCOMM(77), DSQ_CANCEL_IND ),
+ (DSQCOMM(78), DSQ_RESERVE2 ),
+ (DSQCOMM(101), DSQ_RESERVE3 ),
+ (DSQCOMM(257), DSQ_MESSAGE_TEXT ),
+ (DSQCOMM(385), DSQ_Q_MESSAGE_TEXT )

```

```

C          Callable interface normal and extended calls
          EXTERNAL DSQCIF
          EXTERNAL DSQCIFE

```

Exécution de vos programmes en langage FORTRAN

Pour exécuter votre programme en langage FORTRAN dans TSO, compilez-le et procédez à l'édition de ses liens, puis lancez-le avec ou sans ISPF.

Compilation de votre programme et édition de ses liens

Vous devez compiler votre programme FORTRAN et procéder à l'édition de ses liens avant de pouvoir l'exécuter dans TSO.

Le langage JCL pour l'exécution du compilateur FORTRAN et de l'éditeur de liens utilise le compilateur FORTRAN pour z/OS. Certains paramètres peuvent varier d'une installation QMF à l'autre.

La commande JCL présentée dans cet exemple n'est pas fournie avec QMF, mais vous pouvez la copier à partir de cet emplacement, en la modifiant en fonction de vos besoins.

```

//samFORT    JOB
//STEP1      EXEC PROC=VSF2CL
/* Permet d'accéder à la macro de communications QMF DSQCOMM
//FORT.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//FORT.SYSIN  DD *
            .
            .
            .
            Votre programme ou copie de l'exemple QMF DSQABFF
            .
            .
            .
/*
/* Permet d'accéder au module d'interface QMF
/* Allocation pour la bibliothèque cible
//LKED.SYSLMOD DD
/* Allocation pour la bibliothèque de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
            INCLUDE QMFLOAD(DSQCIF)
            INCLUDE QMFLOAD(DSQCIFE)
            ENTRY samFORT
            MODE  AMODE(31) RMODE(31)
            NAME  samFORT(R)
/*

```

Exécution de vos programmes dans TSO sans ISPF

Une fois que votre programme en FORTRAN est compilé correctement, vous pouvez l'exécuter avec JCL sans ISPF.

La commande JCL présentée dans cet exemple n'est pas fournie avec QMF, mais vous pouvez la copier à partir de cet emplacement, en la modifiant en fonction de vos besoins.

Le programme présenté ici exécute votre application d'interface appelable à l'aide du compilateur FORTRAN. Certains paramètres peuvent varier d'une installation QMF à l'autre :

```

PROC 0
CONTROL ASIS
/*****/

```



```

/* Remarque : Les bibliothèques de chargement QMF, */
/* DB2, GDDM et FORTRAN doivent être */
/* allouées avant l'exécution de cette commande CLIST. */
/* Le nom de la bibliothèque de chargement est "QMF1210.SDSQLOAD". */
/*****/
/* Spécifiez une liste d'attributs pour les allocations */
/* d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGMGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme à l'aide de la commande TSO CALL*/
/*****/
CALL samFORT
EXIT CODE(0)

```

Exécution dans TSO sous ISPF

Une fois que votre programme en FORTRAN est compilé correctement, vous pouvez l'exécuter sous ISPF.

La commande CLIST présentée dans cet exemple n'est pas fournie avec QMF, mais vous pouvez la copier à partir de cet emplacement, en la modifiant en fonction de vos besoins.

La commande CLIST présentée ici exécute votre application d'interface appellable à l'aide du compilateur FORTRAN. Certains paramètres peuvent varier d'une installation QMF à l'autre :

```

PROC 0
CONTROL ASIS
/*****/
/* Spécifiez une liste d'attributs pour les allocations d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/

```

```

ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Ensembles de données utilisés par ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD', +
        'PRODUCT.VSF2LOAD')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQUCFRM) DA('QMF1210.DSQUCFRM') 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
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme comme la boîte de dialogue ISPF initiale */
/*****/
ISPSTART PGM(samFORT) NEWAPPL(DSQE)
EXIT CODE(4)

```

L'instruction EXIT CODE(4) supprime l'affichage de l'écran des dispositions ISPF.

Interface en langage PL/I

Vous pouvez utiliser le langage PL/I avec l'interface appelable dans QMF.

Mappage de la zone de communication d'interface pour le langage PL/I (DSQCOMML)

DSQCOMML, qui est fourni avec QMF, offre un mappage DSQCOMM pour le langage PL/I.

Le tableau ci-dessous affiche les valeurs pour DSQCOMML.

Tableau 48. Zone de communication d'interface pour le langage PL/I (DSQCOMML)

Nom de structure	Type de données	Description
DSQ_RETURN_CODE	FIXED BIN(31)	Indique l'état d'une commande QMF après son exécution Les valeurs possibles sont les suivantes : DSQ_SUCCESS Exécution réussie de la demande DSQ_WARNING Achèvement normal avec des avertissements DSQ_FAILURE La commande ne s'est pas exécutée correctement DSQ_SEVERE Erreur grave ; la session QMF a pris fin
DSQ_INSTANCE_ID	FIXED BIN(31)	Identificateur défini par QMF lors de l'exécution de la commande START
DSQ_COMM_LEVEL	CHAR(12)	Identifie le niveau de la zone de communication DSQCOMM Dans votre application, incluez des instructions qui initialisent cette variable sur la valeur affectée à DSQ_CURRENT_COMM_LEVEL avant d'émettre la commande QMF START.
DSQ_PRODUCT	CHAR(2)	Identifie le produit de requête IBM en cours d'utilisation. Les variables commençant par DSQ_QMF indiquent les versions de QMF for TSO et CICS.
DSQ_PRODUCT_RELEASE	CHAR(2)	Niveau d'édition de QMF en cours d'utilisation. La variable DSQ_QMF_V11R1 indique QMF Version 12.1.
DSQ_RESERVE1	CHAR(28)	Réservé à une utilisation ultérieure
DSQ_MESSAGE_ID	CHAR(8)	ID de message d'achèvement
DSQ_Q_MESSAGE_ID	CHAR(8)	ID de message de requête
DSQ_START_PARM_ERROR	CHAR(8)	Paramètre erroné lorsque la commande START a échoué en raison d'une erreur de paramètre
DSQ_CANCEL_IND	CHAR(1)	Contient l'une des deux valeurs ci-dessous, selon que l'utilisateur a effectué une annulation pendant l'exécution d'une commande QMF : <ul style="list-style-type: none">• DSQ_CANCEL_YES• DSQ_CANCEL_NO
DSQ_RESERVE2	CHAR(23)	Réservé à une utilisation ultérieure
DSQ_RESERVE3	CHAR(156)	Réservé à une utilisation ultérieure
DSQ_MESSAGE_TEXT	CHAR(128)	Texte du message d'achèvement
DSQ_Q_MESSAGE_TEXT	CHAR(128)	Texte du message de requête

Appels de fonction pour le langage PL/I

QMF fournit deux appels de fonction pour le langage PL/I : DSQCIPL et DSQCIPX. Les deux appels sont décrits dans la zone de communication (DSQCOMML).

Syntaxe de DSQCIPL

Cet appel est destiné aux commandes QMF qui ne requièrent pas d'accès à des variables de programme d'application. Utilisez cet appel pour la plupart des commandes QMF.

```
CALL DSQCIPL(DSQCOMM,  
            CMDLTH,  
            CMDSTR)
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR)

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par CMDLTH

Syntaxe de DSQCIPX

Cet appel est destiné aux commandes qui requièrent un accès à des variables de programme d'application : START, TRACE et les formats étendus de GET GLOBAL et SET GLOBAL.

La syntaxe de cet appel est la suivante :

```
CALL DSQCIPX(DSQCOMM,  
            CMDLTH,  
            CMDSTR,  
            PNUM,  
            KLTH,  
            KWORD,  
            VLTH,  
            VALUE,  
            VTYPE)
```

Les paramètres prennent les valeurs suivantes :

DSQCOMM

Zone de communication d'interface

CMDLTH

Longueur de la chaîne de commande (CMDSTR) ; paramètre entier FIXED BIN(31)

CMDSTR

Commande QMF à exécuter ; chaîne de caractères majuscules de la longueur spécifiée par CMDLTH

PNUM

Nombre de mots clés de commande ou de zones de trace ; paramètre entier FIXED BIN(31)

KLTH

Longueur de chaque mot clé ou titre de trace spécifié ; paramètre entier FIXED BIN(31) ou tableau de paramètres

KWORD

Mot(s) clé(s) QMF ou adresse des titres de trace.

Chacun représente un caractère, un tableau de caractères ou d'adresses vers des titres de trace dont les longueurs sont spécifiées par KLTH. Vous pouvez utiliser un tableau de caractères si tous les mots clés sont de même longueur. Les mots clés doivent se trouver dans une mémoire contiguë et ne doivent pas être séparés par des délimiteurs spéciaux.

VLTH

Longueur de chaque valeur qui est associée au mot clé ou au titre de trace ; paramètre entier FIXED BIN(31) ou tableau de paramètres

VALUE

Valeur associée à chaque mot clé ou adresse d'une valeur qui est associée à un titre de trace.

Son type est spécifié dans le paramètre VTYPE et peut être un caractère, un tableau de caractères, un paramètre entier FIXED BIN(31) ou un tableau de paramètres. Dans le cas de valeurs de type caractère, ces dernières doivent se trouver dans une mémoire contiguë et ne doivent pas être séparées par des délimiteurs spéciaux.

VTYPE

Type de données du contenu du paramètre VALUE.

Ce paramètre comporte l'une des deux valeurs qui sont fournies dans la zone de communication DSQCOMML :

- DSQ_VARIABLE_CHAR pour les valeurs de type caractère
- DSQ_VARIABLE_FINT pour les valeurs entières FIXED BIN(31)

Toutes les valeurs spécifiées dans la zone VALUE doivent avoir le type de données qui est indiqué dans VTYPE.

Exemple de programmation en langage PL/I

L'exemple de programme, DSQABFP, est fourni avec QMF et utilise le langage PL/I. Il fait partie de la bibliothèque QMF1210.SDSQSAPn (où *n* correspond à l'identificateur de langue nationale).

L'exemple de programme pour l'interface callable en langage PL/I exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

```
DSQABFP: PROCEDURE OPTIONS(MAIN REENTRANT) REORDER;
/*****
/* Sample program: DSQABFP
/* PL/I version of the QMF callable interface
*****/

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

/*****
/* Built in function
*****/
```

```

/*****
DCL LENGTH BUILTIN;

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

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

/*****
/* Keyword parameter and value for START command */
/*****
DCL START_KEYWORDS        CHAR(8)   INIT('DSQSMODE');
DCL START_KEYWORD_VALUES  CHAR(11)  INIT('INTERACTIVE');

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

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

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

/*****
/* Start a query interface session */
/*****
NUMBER_OF_PARAMETERS = 1;
COMMAND_LENGTH = LENGTH(START_QUERY_INTERFACE);
KEYWORD_LENGTHS(1) = LENGTH(START_KEYWORDS);
DATA_LENGTHS(1) = LENGTH(START_KEYWORD_VALUES);

CALL DSQCIPX(DSQCOMM,
            COMMAND_LENGTH,
            START_QUERY_INTERFACE,
            NUMBER_OF_PARAMETERS,
            KEYWORD_LENGTHS,
            START_KEYWORDS,
            DATA_LENGTHS,
            START_KEYWORD_VALUES,
            DSQ_VARIABLE_CHAR);

/*****
/* Set numeric values into query using SET command */
/*****
NUMBER_OF_PARAMETERS = 3;

```

```

COMMAND_LENGTH = LENGTH(SET_GLOBAL_VARIABLES);
KEYWORD_LENGTHS(1) = LENGTH(SET_KEYWORDS_1);
KEYWORD_LENGTHS(2) = LENGTH(SET_KEYWORDS_2);
KEYWORD_LENGTHS(3) = LENGTH(SET_KEYWORDS_3);
DATA_LENGTHS(1) = 4;
DATA_LENGTHS(2) = 4;
DATA_LENGTHS(3) = 4;
SET_VALUES_1 = 20;
SET_VALUES_2 = 40;
SET_VALUES_3 = 84;

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

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

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             RUN_QUERY);

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

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             PRINT_REPORT);

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

CALL DSQCIPL(DSQCOMM,
             COMMAND_LENGTH,
             END_QUERY_INTERFACE);

END      DSQABFP;

```

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

DSQCOMM pour le langage PL/I

Le fichier de la zone de communication d'interface pour le langage PL/I se nomme DSQCOMML.

```

/*****
/* PL/I include for query callable interface */
*****/

/* Structure declare for communications area */
DCL
1 DSQCOMM,

```

```

3 DSQ_RETURN_CODE      FIXED BIN(31), /* Function return code */
3 DSQ_INSTANCE_ID     FIXED BIN(31), /* Start ID */
3 DSQ_COMM_LEVEL      CHAR(12), /* Communications level */
3 DSQ_PRODUCT         CHAR(2), /* Query product ID */
3 DSQ_PRODUCT_RELEASE CHAR(2), /* Query product release */
3 DSQ_RESERVE1        CHAR(28), /* Reserved */
3 DSQ_MESSAGE_ID      CHAR(8), /* Completion message ID */
3 DSQ_Q_MESSAGE_ID    CHAR(8), /* Query message ID */
3 DSQ_START_PARM_ERROR CHAR(8), /* Start parms in error */
3 DSQ_CANCEL_IND      CHAR(1), /* Cmd cancel indicator */
                        /* 1 = cancelled, 0 = not cancelled*/
3 DSQ_RESERVE2        CHAR(23), /* Reserved */
3 DSQ_RESERVE3        CHAR(156), /* Reserved */
3 DSQ_MESSAGE_TEXT    CHAR(128), /* QMF command message */
3 DSQ_Q_MESSAGE_TEXT  CHAR(128); /* QMF query message */

/* Return codes */
DCL
DSQ_SUCCESS          FIXED BIN(31) INIT(0) STATIC,
DSQ_WARNING          FIXED BIN(31) INIT(4) STATIC,
DSQ_FAILURE          FIXED BIN(31) INIT(8) STATIC,
DSQ_SEVERE           FIXED BIN(31) INIT(16) STATIC;

/* Communications level */
DCL
DSQ_CURRENT_COMM_LEVEL CHAR(12) INIT('DSQL>001002<') STATIC;

/* Query product ID */
DCL
DSQ_QRW              CHAR(2) INIT('01') STATIC,
DSQ_QMF              CHAR(2) INIT('02') STATIC,
DSQ_QM4              CHAR(2) INIT('03') STATIC;

/* Query product release ID */
DCL
DSQ_QRW_V1R2         CHAR(2) INIT('01') STATIC,
DSQ_QRW_V1R3         CHAR(2) INIT('02') STATIC,
DSQ_QMF_V2R4         CHAR(2) INIT('01') STATIC,
DSQ_QMF_V3R1         CHAR(2) INIT('02') STATIC,
DSQ_QMF_V3R1M1       CHAR(2) INIT('03') STATIC,
DSQ_QMF_V3R2         CHAR(2) INIT('04') STATIC,
DSQ_QMF_V3R3         CHAR(2) INIT('05') STATIC,
DSQ_QMF_V6R1         CHAR(2) INIT('06') STATIC,
DSQ_QMF_V7R1         CHAR(2) INIT('07') STATIC,
DSQ_QM4_V1R1         CHAR(2) INIT('01') STATIC,
DSQ_QMF_V7R2         CHAR(2) INIT('08') STATIC,
DSQ_QMF_V8R1         CHAR(2) INIT('09') STATIC,
DSQ_QMF_V9R1         CHAR(2) INIT('10') STATIC,
DSQ_QMF_V10R1        CHAR(2) INIT('11') STATIC;

/* Cancelled indicator */
DCL
DSQ_CANCEL_YES       CHAR(1) INIT('1') STATIC,
DSQ_CANCEL_NO        CHAR(1) INIT('0') STATIC;

/* Variable types */
DCL
DSQ_VARIABLE_CHAR    CHAR(4) INIT('CHAR') STATIC,
DSQ_VARIABLE_FINT    CHAR(4) INIT('FINT') STATIC;

/* Mode */
DCL
DSQ_INTERACTIVE      CHAR(1) INIT('1') STATIC,
DSQ_BATCH            CHAR(1) INIT('2') STATIC;

/* Yes or no */
DCL

```



```

DSQ_YES          CHAR(1) INIT('1') STATIC,
DSQ_NO           CHAR(1) INIT('2') STATIC;

/* Query interface entry point */
DCL
DSQCIPL ENTRY (*, /* Interface block */
              FIXED BIN(31), /* Length of command */
              CHAR(*)) /* Command string */
              EXTERNAL OPTIONS(ASSEMBLER);

DCL
DSQCIPX ENTRY (*, /* Interface block */
              FIXED BIN(31), /* Length of command */
              CHAR(*), /* Command string */
              FIXED BIN(31), /* # of command keywords */
              *, /* Length of keyword */
              *, /* Keyword string */
              *, /* Length of value */
              *, /* Value of keyword */
              CHAR(4)) /* Data type of value */
              EXTERNAL OPTIONS(ASSEMBLER);

```

Exécution de vos programmes sous CICS

Une fois que vous avez écrit votre programme en langage PL/I, vous devez le compiler et l'exécuter.

Lorsque vous convertissez un programme qui utilise l'interface appellable QMF, que vous le compilez et que vous procédez à l'édition de ses liens, tenez compte des points suivants :

- La zone de communication (DSQCOMML) doit être accessible lors de l'étape de compilation ou copiée dans votre programme.
- Les modules d'interface QMF DSQCIPL et DSQCIPX doivent être accessibles lors de l'étape d'édition des liens de votre programme.

Cet exemple utilise la procédure DFHVITVL, qui est fournie avec CICS. Cette commande JCL n'est pas fournie avec QMF, mais vous pouvez la copier et la modifier en fonction de vos besoins.

```

//samPLI JOB
// EXEC PROC=DFHVITVL
//PLI.SYSIN DD *
.
.
.
Votre programme ou copie de l'exemple QMF DSQABFP
.
.
.
/*
/* Permet d'accéder à la macro de communications QMF DSQCOMML
//PLI.SYSLIB DD DSN=QMF1210.SDSQSAPE,DISP=SHR
/* Permet d'accéder au module d'interface QMF
/* Allocation pour la bibliothèque cible
//LKED.SYSLMOD DD
/* Allocation pour la bibliothèque de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCIPL)
INCLUDE QMFLOAD(DSQCIPX)
MODE AMODE(31) RMODE(31)
NAME sampPLI(R)
/*

```

Exécution de vos programmes sous TSO

Pour exécuter votre programme en langage PL/I dans TSO, compilez-le et procédez à l'édition de ses liens, puis lancez-le avec ou sans ISPF.

Compilation et édition de liens dans TSO

Vous devez compiler votre programme PL/I et procéder à l'édition de ses liens avant de pouvoir l'exécuter dans TSO.

Ce langage JCL utilise le compilateur PL/I pour compiler votre application d'interface appellable, puis procéder à l'édition des liens de l'application. Certains paramètres peuvent varier d'une installation QMF à l'autre.

```
//samPLI JOB
//STEP1 EXEC PROC=IBMZCPL
//* Permet d'accéder à la macro de communications QMF DSQCOMML
//PLI.SYSLIB DD DSN=QMF1210.SAMPLIB,DISP=SHR
//PLI.SYSIN DD *
.
.
.
Votre programme ou copie de l'exemple QMF DSQABFP
.
.
.
//* Allocation pour la bibliothèque cible
//LKED.SYSLMOD DD
//* Allocation pour la bibliothèque de chargement QMF
//LKED.QMFLOAD DD DSN=QMF1210.SDSQLOAD,DISP=SHR
//LKED.SYSIN DD *
INCLUDE QMFLOAD(DSQCIPL)
INCLUDE QMFLOAD(DSQCIPX)
ENTRY CEESTART
MODE AMODE(31) RMODE(ANY)
NAME sampPLI(R)
/*
```

Exécution dans TSO sans ISPF

Une fois que votre programme en langage PL/I est assemblé correctement, vous pouvez l'exécuter sans ISPF.

Exécutez votre programme dans TSO sans ISPF en écrivant un programme similaire à la commande CLIST présentée ici :

```
PROC 0
CONTROL ASIS
/*****/
/* Remarque : Les bibliothèques de chargement QMF, */
/* DB2, GDDM et PL/I doivent être */
/* allouées avant l'exécution de cette commande CLIST. */
/* Le nom de la bibliothèque de chargement est "QMF1210.SDSQLOAD". */
/*****/
/* Spécifiez une liste d'attributs pour les allocations */
/* d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
```

```

/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE
ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme à l'aide de la commande TSO CALL*/
/*****/
CALL sampPLI
EXIT CODE(0)

```

Exécution dans TSO sous ISPF

Une fois que votre programme en langage PL/I est assemblé correctement, vous pouvez l'exécuter sous ISPF.

Exécutez votre programme dans TSO sous ISPF en écrivant un programme similaire à la commande CLIST présentée ici :

```

PROC 0
CONTROL ASIS
/*****/
/* Spécifiez une liste d'attributs pour les allocations d'ensembles de données */
/*****/
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)
/*****/
/* Ensembles de données utilisés par TSO */
/*****/
ALLOC FI(SYSPROC) DA('QMF1210.SDSQCLTE','ISR.ISRCLIB')
ALLOC FI(SYSEXEC) DA('QMF1210.SDSQEXCE')
/*****/
/* Ensembles de données utilisés par ISPF */
/*****/
ALLOC FI(ISPLLIB) SHR REUSE +
    DA('QMF1210.SDSQLOAD','ADM.GDDMLoad','DSN.DSNEXIT','DSN.DSNLOAD', +
    'PLI.PLILINK','PLI.SIBMLINK')
ALLOC FI(ISPMLIB) SHR REUSE +
    DA('QMF1210.SDSQMLBE','ISR.ISRMLIB','ISP.ISPMLIB')
ALLOC FI(ISPPLIB) SHR REUSE +
    DA('QMF1210.SDSQPLBE','ISR.ISRPLIB','ISP.ISPPLIB')
ALLOC FI(ISPSLIB) SHR REUSE +
    DA('QMF1210.SDSQSLBE','ISR.ISRSLIB','ISP.ISPSLIB')
ALLOC FI(ISPTLIB) SHR REUSE +
    DA('ISR.ISRTLIB','ISP.ISPTLIB')
/*****/
/* Ensembles de données QMF/GDDM */
/*****/
ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE
ALLOC FI(ADMCFORM) DA('QMF1210.DSQCFORM') SHR REUSE
ALLOC FI(DSQCFRM) DA('QMF1210.DSQCFRM') SHR REUSE
ALLOC FI(ADMSYMBL) DA('ADM.GDDMSYM') SHR REUSE
ALLOC FI(ADMGDF) DA('ADM.GDDM.CHARTLIB') SHR REUSE

```

```

ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE
/*****/
/* Ensembles de données utilisés par QMF */
/*****/
ALLOC FI(DSQPRINT) SYSOUT(X) USING(PRINTDCB)
ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)
ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)
ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS
ALLOC FI(DSQEDIT) NEW UNIT(SYSDA) USING(EDITDCB)
ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE
/*****/
/* Démarrez votre programme comme la boîte de dialogue ISPF initiale */
/*****/
ISPSTART PGM(sampPLI) NEWAPPL(DSQE)
EXIT CODE(4)

```

L'instruction EXIT CODE(4) supprime l'écran des dispositions ISPF.

Interface en langage REXX

REXX est un langage interprétatif ; il n'a pas besoin d'être compilé.

Toutefois, les performances des programmes qui utilisent un langage REXX compilé ou d'autres langages compilés sont meilleures que celles des mêmes programmes écrits qui utilisent le langage REXX interprétatif. Un compilateur REXX est disponible pour les programmes REXX, mais pas pour les procédures navigationnelles.

Sous TSO, démarrez QMF avec l'interface appelable REXX lorsque vous utilisez des procédures navigationnelles et certaines fonctions de format QMF (calculs, colonnes définies et conditions). L'interface appelable REXX peut réduire le nombre de ressources requises pour l'utilisation des services REXX.

Par exemple, un nombre inférieur de ressources est requis pour émettre les commandes PRINT REPORT ou BOTTOM sur l'écran REPORT si QMF est démarré avec l'interface appelable REXX. La réduction de la consommation des ressources peut être importante et est plus visible lors de l'exécution de QMF sous TSO.

Le langage REXX fonctionne toujours dans un environnement de commande qui détermine le mode et l'emplacement de traitement de la commande. Si vous écrivez un programme REXX qui émet des commandes QMF, vous pouvez utiliser l'environnement de commande QMF via la commande ADDRESS QRW.

Restriction : Le langage REXX n'étant pas disponible sous QMF for CICS, l'interface appelable de QMF pour REXX ne fonctionne pas sous CICS.

Concepts associés:

Chapitre 5, «ADDRESS QRW et environnement de commande QMF», à la page 31
 Lorsque QMF est démarré dans TSO, ISPF ou z/OS natif, QMF crée un environnement de commande REXX appelé QRW. Lorsque vous exécutez un programme REXX, vous pouvez définir l'environnement de commande par défaut sur QRW en émettant la commande REXX ADDRESS appelée ADDRESS QRW. Grâce à ADDRESS QRW, QMF reste l'environnement de commande par défaut jusqu'à ce que vous émettiez une autre commande ADDRESS.

Variables de communication d'interface pour REXX

Les variables de communication d'interface sont composées de plusieurs variables REXX. Elles sont définies à la fin de chaque appel et ne doivent pas être modifiées par le programme appelant.

Les variables de communication d'interface pour les variables REXX sont présentées dans le tableau ci-dessous.

Tableau 49. Variables de communication d'interface pour REXX

Nom de structure	Description
dsq_return_code	Entier indiquant les résultats de l'exécution d'une commande QMF Les valeurs admises sont les suivantes : dsq_success Traitement réussi de la demande dsq_warning Achèvement normal avec des avertissements dsq_failure La commande n'a pas été traitée correctement dsq_severe Erreur grave provoquant la fin de la session QMF Aucun appel supplémentaire vers QMF ne peut être effectué avec cet ID instance. La valeur de dsq_return_code est également placée dans la variable REXX <i>rc</i> .
dsq_instance_id	Identificateur défini par QMF lors du traitement de la commande START
dsq_product	Identifie le produit de requête IBM en cours d'utilisation Les variables commençant par dsq_qmf indiquent les versions de QMF for TSO.
dsq_product_release	Niveau d'édition de QMF en cours d'utilisation La variable dsq_qmf_v12r1 indique QMF Version 12.1.
dsq_message_id	ID de message d'achèvement
dsq_q_message_id	ID de message de requête
dsq_start_parm_error	Paramètre erroné lorsque la commande START a échoué en raison d'une erreur de paramètre
dsq_cancel_ind	Indicateur d'annulation de commande qui indique si l'utilisateur a annulé le traitement de la commande pendant que QMF exécutait une commande ; les valeurs possibles sont les suivantes : dsq_cancel_yes L'utilisateur a annulé la commande dsq_cancel_no L'utilisateur n'a pas annulé la commande
dsq_message_text	Texte du message d'achèvement
dsq_q_message_text	Texte du message de requête

Appel de fonction pour REXX

Vous pouvez accéder à l'interface callable à l'aide d'appels de fonction REXX classiques. QMF fournit une sous-routine externe appelée DSQCIX, qui est utilisée pour exécuter toutes les commandes QMF émises via l'interface callable.

La syntaxe de l'appel de fonction DSQCIX est la suivante :

```
call DSQCIX cmd parmlist
```

Dans cette syntaxe, *cmd* représente une commande QMF écrite sous forme de chaîne de caractères majuscules et *parmlist* une liste de paires paramètre/valeur.

Syntaxe de la liste de paramètres pour l'appel de fonction DSQCIX



Transmettez la totalité de la commande, y compris le paramètre *parmlist*, à QMF en tant que variable REXX unique écrite sous la forme d'une chaîne de caractères. Cette chaîne doit être placée entre guillemets simples (' ') ou doubles (" "). Lorsque vous utilisez des variables REXX dans le cadre de la chaîne de commande, ne placez pas l'argument entre guillemets. Par exemple :

```
CALL DSQCIX "RUN QUERY NAME (&ECN="REXAUG",CONFIRM=YES)"
```

parmname

Nom de paramètre

value

Valeur à associer au nom de paramètre spécifié par *parmname*

Voici des exemples d'appels de fonction :

```
call DSQCIX "RUN QUERY Q1"  
call DSQCIX "PRINT REPORT (FORM=F1"  
call DSQCIX "EXIT"
```

Dans *parmlist*, les mêmes résultats se produisent que les éléments suivants soient présents ou non :

- Virgule (,) entre des paramètres (un espace produit le même résultat)
- Parenthèse fermante (qui n'est pas obligatoire)
- Signe égal (=) entre *parmname* et *value* (un espace produit le même résultat)

Chacun des exemples suivants produit le même résultat :

```
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)"
```

Exemple de programmation en langage REXX

L'exemple de programme en langage REXX, DSQABFX, est fourni avec QMF.

Vous pouvez examiner l'exemple d'affichage de code source ici ou y accéder en ligne. Cet exemple de programme fait partie de la bibliothèque QMF1210.SDSQEXCE.

L'exemple de programme pour l'interface callable en langage REXX exécute les fonctions suivantes :

- Démarre QMF
- Définit trois variables globales
- Exécute une requête appelée Q1
- Imprime le rapport généré à l'aide du format F1
- Met fin à la session QMF

QMF ne fournit pas la requête Q1 ou le format F1, mais l'exemple de programme utilise ces objets.

```

/*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

```

Exécution de vos programmes REXX

Une fois que vous avez écrit votre programme en langage REXX, vous devez l'exécuter.

Vous pouvez exécuter votre programme REXX dans TSO en écrivant un programme similaire à celui présenté ici :

```

/*****
/* Emettez des allocations TSO pour le produit QMF
/*****
Address TSO

"ATTR PRINTDCB LRECL(133) RECFM(F B A) BLKSIZE(1330)"
"ATTR DEBUGDCB LRECL(80) RECFM(F B) BLKSIZE(3120)"
"ATTR UDUMPDCB LRECL(125) RECFM(V B A) BLKSIZE(1632)"
"ATTR EDITDCB LRECL(79) RECFM(F B A) BLKSIZE(4029)"
"ALLOC FI(SYSPROC) SHR REUSE ",
"DA('QMF1210.DSQCLSTE,'"",
" 'DSN.DSNCLIST')"
"ALLOC FI(SYSEXEC) SHR REUSE ",
"DA('QMF1210.SDSQEXCE')"
"ALLOC FI(ISPLLIB) SHR REUSE ",
"DA('QMF1210.SDSQLOAD,'"",
" 'ADM.GDDM.GDDMLOAD,'"",
" 'DSN.DSNLOAD')"
"ALLOC FI(DSQPNLE) DA('QMF1210.DSQPNLE') SHR REUSE"
"ALLOC FI(DSQPRINT) SYSOUT USING(PRINTDCB)"
"ALLOC FI(SYSPRT) SYSOUT(X) LRECL(132) RECFM(FBA) BLKSIZE(132)"
"ALLOC FI(DSQDEBUG) SYSOUT(X) USING(DEBUGDCB)"
"ALLOC FI(DSQDUMP) SYSOUT(X) USING(UDUMPDCB)"
"ALLOC FI(DSQSPILL) NEW UNIT(SYSDA) SPACE(1,1) TRACKS"
"ALLOC DDNAME(DSQEDIT) UNIT(SYSDA) NEW USING(EDITDCB)"
"ALLOC FI(ADMDEFS) DA('ADM.GDDM.NICKNAME') SHR REUSE"
"ALLOC FI(ADMGGMAP) DA('QMF1210.SDSQMAPE') SHR REUSE"
"ALLOC FI(ADMCFORM) DA('QMF1210.DSQCHART') SHR REUSE"
"ALLOC FI(DSQUCFRM) DA('QMF1210.DSQUCFRM') SHR REUSE"
"ALLOC FI(ADMGDF) DA('GDDM.ADMGDF') SHR REUSE"
"ALLOC FI(ADMSYMBL) DA('ADM.GDDM.GDDMSYM') SHR REUSE"

/* Début de votre programme REXX ..... */
.
.
.
/* Fin de votre programme REXX ..... */

```

Exemple REXX d'utilisation d'une boucle INTERACT

Vous pouvez faire en sorte que la commande END dans une session interactive se comporte de la même façon que dans une session interactive QMF.

Normalement, lorsque votre programme d'interface appellable émet une commande INTERACT et que l'utilisateur émet la commande END, QMF renvoie immédiatement à votre programme. Toutefois, une session interactive QMF permet à l'utilisateur d'émettre la commande END pour renvoyer à l'écran d'aide QMF. L'émission d'une deuxième commande END met fin à la session QMF.

Ajoutez la logique figurant dans l'exemple suivant à votre programme pour que la commande END se comporte de la même façon qu'une session interactive QMF.

Ce programme utilise *dsq_message_id* pour déterminer la façon de procéder. Ces valeurs peuvent varier d'une édition à l'autre.

Ce programme n'est pas fourni avec QMF.

```

/*REXX*****
/* Sample program: Using INTERACT loop
/*****
/*****
/* Start an interactive QMF session
*/

```



```

/*****/
trace error

parms = "START (DSQSMODE=INTERACTIVE"
call dsqcix parms
if dsq_return_code = dsq_severe then exit dsq_return_code
/*****/
/* SET GLOBAL to show panel IDs */
/*****/
call dsqcix "SET GLOBAL (DSQDC_SHOW_PANID=1"
if dsq_return_code = dsq_severe then exit dsq_return_code
/*****/
/* Issue message */
/*****/
call dsqcix "MESSAGE (TEXT='OK, you may enter a command.')"
if dsq_return_code = dsq_severe then exit dsq_return_code
/*****/
/* INTERACT loop */
/*****/
Continue = "yes"
Do while continue = "yes"
  call DSQCIX "INTERACT"
  Select
    When (dsq_return_code = dsq_severe) Then /* Severe error */
      Continue = "no"
    When (dsq_message_id = "DSQ21869") Then /* END from HOME panel */
      Continue = "no"
    When (dsq_message_id = "DSQ90557") Then /* User issued EXIT */
      Continue = "no"
    Otherwise nop /* OK continue session */
  End
End
/*****/
/* End the session */
/*****/
if dsq_message_id <> "DSQ90557" then /* EXIT not issued */
  call dsqcix "EXIT" /* Issue EXIT */
endif

exit dsq_return_code

```

Annexe A. Macros d'interface de produit

Le tableau ci-dessous répertorie les macros qui sont fournies aux clients avec QMF en tant qu'interfaces de programmation générique.

Important : N'utilisez pas d'autres macros QMF que celles identifiées ici en tant qu'interfaces de programmation.

Tableau 50. Macros fournissant des interfaces aux fonctions QMF

Fonction	Noms des macros
Macro d'interface de produit	DSQQMF n Dans ce nom de programme, n est un identificateur de langue nationale. Pour l'anglais, cet identificateur est E.
Macros d'interface callable	<ul style="list-style-type: none">• Assembler<ul style="list-style-type: none">- DSQCIA- DSQCOMMA• COBOL<ul style="list-style-type: none">- DSQCIB- DSQCOMMB• C/C++<ul style="list-style-type: none">- DSQCIC- DSQCICE- DSQCOMMC• FORTRAN<ul style="list-style-type: none">- DSQCIF- DSQCIFE- DSQCOMMF• PL/I<ul style="list-style-type: none">- DSQCIPL- DSQCIPX- DSQCOMML• REXX<ul style="list-style-type: none">- DSQCIX
Macro d'interface de commande	DSQCCI
Macros d'interface de routine d'exit du gestionnaire QMF	<ul style="list-style-type: none">• DXEGOVA• DXEXCBA
Macro de routine d'exit d'édition utilisateur QMF	DXECS

Référence associée:

«Conventions pour les informations NLF (National Language Feature)», à la page 5 DB2 QMF est disponible dans plusieurs langues différentes, chacune étant fournie par une fonction NLF (National Language Feature).

Annexe B. Variables globales QMF

QMF fournit de nombreuses variables globales qui vous aident à contrôler divers aspects de votre session QMF, de vos commandes QMF et de l'affichage d'écran. Les variables globales vous aident également à gérer le comportement des fonctions QMF dans des procédures et des applications.

Convention de dénomination des variables globales QMF

La convention de dénomination de la plupart des variables globales fournies avec QMF est `DSQcc_xxxxxxxxxxxx`. `cc` identifie la catégorie de la variable et `xxxxxxxxxxx` est un nom descriptif pouvant contenir jusqu'à 12 caractères. Un caractère de soulignement (`_`) est inclus après `cc`.

`cc` peut être l'un des identificateurs suivants :

- AP** Variables relatives aux informations d'état associées au profil
- AO** Variables relatives aux autres informations d'état (non associées au profil)
- CM** Variables relatives aux informations concernant le message généré par la commande précédente
- CP** Variables relatives aux informations concernant l'éditeur de table
- DC** Variables contrôlant la façon dont QMF affiche les informations à l'écran
- EC** Variables contrôlant la façon dont QMF exécute les commandes et les procédures
- QC** Variables dont les valeurs sont générées par une option CONVERT QUERY
- QM** Variables contenant des informations sur le message d'erreur RUN QUERY
- QW** Variables uniques à QMF for Workstation.

Variables de session

Les variables de session suivent une convention de dénomination différente. Ce sont des variables globales qui stockent les valeurs entrées par les utilisateurs dans certaines zones de certains panneaux si la variable globale `DSQEC_SESSGLV_SAV` a la valeur 1 ou 2. La convention de dénomination des variables de session est la suivante :

`DXYnpppp_ln_dd`

où :

- `n` désigne l'identificateur de la langue nationale
- `pppp` désigne les quatre dernières lettres de l'identificateur du panneau
- `ln` désigne un ID associé à la zone
- `dd` désigne un ID associé à la zone (utilisé uniquement si la zone est liée à une autre zone)

Définition et affichage des valeurs des variables globales

Si la valeur que vous voulez affecter à une variable globale est égale ou inférieure à 55 octets, utilisez la commande SET GLOBAL. Si la valeur est supérieure à 55 octets, utilisez la commande SHOW GLOBALS.

Pourquoi et quand exécuter cette tâche

Par défaut, une valeur de variable globale est conservée tant que vous ne la réinitialisez pas ou que vous ne mettez pas fin à la session QMF. Vous pouvez cependant définir la variable globale DSQEC_USERGLV_SAV de sorte à sauvegarder les valeurs de variable globale d'une session à l'autre.

Pour personnaliser les variables globales lors de l'initialisation, consultez les informations concernant l'initialisation des variables globales et le comportement d'une session QMF au démarrage de QMF dans Installation et gestion de DB2 QMF for TSO and CICS.

Procédure

Pour affecter une valeur supérieure à 55 octets à une variable globale :

1. Utilisez la commande SHOW GLOBALS pour afficher le panneau GLOBALS.
2. Appuyez sur la touche **Show Field** pour afficher la totalité de la zone de saisie. La longueur maximale d'une variable globale dans le panneau Show Global Variable est de 32 768 octets.
3. Entrez la valeur de la variable sur les lignes fournies.

Variables globales relatives aux informations d'état non liées au profil

Les variables globales DSQAO contiennent des informations d'état, des paramètres ou des indicateurs. La commande SET GLOBAL ne permet pas de les modifier.

Tableau 51. Variables globales relatives aux informations d'état non liées au profil

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_APPL_TRACE	DSQATRAC	01	0 pour le niveau A0 1 pour le niveau A1 2 pour le niveau A2
DSQAO_ATTENTION	DSQCATTN	01	Indicateur d'attention pour les utilisateurs.
DSQAO_BATCH	DSQABATC	01	Mode de traitement par lots ou interactif ; les valeurs admises sont : 1 pour une session interactive 2 pour une session en mode de traitement par lots

Tableau 51. Variables globales relatives aux informations d'état non liées au profil (suite)

Nom de variable de l'interface callable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_CONNECT_ID	DSQAAUTH	128	ID utilisateur utilisé pour se connecter à la base de données et sous lequel le travail est effectué. La valeur de cette variable change lorsque vous exécutez la commande ou l'instruction suivante : <ul style="list-style-type: none"> • Exécutez une commande QMF CONNECT pour vous reconnecter à la base de données sous un autre ID autorisation • Exécutez une instruction SET CURRENT SQLID sur une base de données DB2 for z/OS.
DSQAO_CONNECT_LOC	Aucun	18	Nom d'emplacement de la base de données à laquelle vous êtes actuellement connecté ; ce nom comprend 16 caractères (il est rempli à droite par des espaces, le cas échéant).
DSQAO_CURSOR_OPEN	DSQACRSR	01`	Etat du curseur de base de données ; les valeurs admises sont : 1 si le curseur est ouvert 2 si le curseur est fermé
DSQAO_DATE_FORMAT	Aucun	05	Contient la valeur spécifiée dans SYSIBM.DATE_FORMAT. Les valeurs peuvent être ISO, USA, EUR, JIS ou LOCAL.
DSQAO_DB_MANAGER	DSQADBMG	01	Gestionnaire de base de données, indiqué par l'une des valeurs suivantes : 1 DB2 for VSE et VM 2 DB2 for z/OS 3 DB2 for Linux, UNIX and Windows 4 DB2 for iSeries
DSQAO_DBCS	DSQADBCS	01	Etat de la prise en charge DBCS ; les valeurs admises sont : 1 pour la prise en charge DBCS 2 pour la non prise en charge DBCS
DSQAO_DSQSBSTG	Aucun	10	Contient la valeur spécifiée par le paramètre DSQSBSTG ou la valeur par défaut si le paramètre n'a pas été spécifié.
DSQAO_DSQSFISO	Aucun	01	Contient la valeur spécifiée par le paramètre DSQSFISO ou la valeur par défaut si le paramètre n'a pas été spécifié. Les valeurs suivantes sont utilisées : 0 QMF n'est pas précompilé avec DATE(ISO) et TIME(ISO). 1 QMF est précompilé avec DATE(ISO) et TIME(ISO). Il s'agit de la valeur par défaut.

Tableau 51. Variables globales relatives aux informations d'état non liées au profil (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_DSQSMRFI	Aucun	01	<p>Cette zone reflète la valeur spécifiée pour le paramètre du programme DSQSMRFI lors du démarrage de QMF.</p> <p>0 NO a été spécifié pour le paramètre du programme DSQSMRFI, ce qui signifie que l'extraction et l'insertion d'une ligne unique DB2 est utilisée.</p> <p>1 YES a été spécifié pour le paramètre du programme DSQSMRFI, ce qui signifie que l'extraction et l'insertion multilignes DB2 est utilisée. L'extraction multiligne utilise un curseur d'ensemble de lignes.</p>
DSQAO_DSQSMTHD	Aucun	01	<p>Contient la valeur spécifiée par le paramètre DSQSMTHD ou la valeur par défaut si le paramètre n'a pas été spécifié.</p> <p>Les valeurs suivantes sont utilisées :</p> <p>0 NO a été spécifié ; QMF s'exécute avec une seule unité d'exécution. Il s'agit de la valeur par défaut.</p> <p>1 YES a été spécifié ; QMF s'exécutera avec une deuxième unité d'exécution qui sera utilisée pour les commandes (RUN QUERY, DISPLAY TABLE) et le défilement consécutif (BOTTOM, TOP, FORWARD, BACKWARD, RIGHT et LEFT) des rapports avec des curseurs ouverts.</p>
DSQAO_DSQSPILL	Aucun	01	<p>Contient la valeur spécifiée par le paramètre DSQSPILL ou la valeur par défaut si le paramètre n'a pas été spécifié.</p> <p>Les valeurs suivantes sont utilisées :</p> <p>0 pour le non recours à la mémoire auxiliaire. Cette valeur correspond à la valeur de paramètre DSQSPILL NO.</p> <p>1 pour le recours à la mémoire auxiliaire. Cette valeur correspond à la valeur de paramètre DSQSPILL YES.</p>
DSQAO_DSQSPTYP	Aucun	5	<p>Contient la valeur spécifiée par le paramètre DSQSPTYP ou la valeur par défaut si le paramètre n'a pas été spécifié.</p> <p>Les valeurs suivantes sont utilisées :</p> <p>FILE pour l'envoi de données dans un fichier.</p> <p>64BIT pour l'envoi de données dans la mémoire virtuelle étendue.</p>
DSQAO_DSQSRSTG	Aucun	8	<p>Contient la valeur spécifiée par le paramètre DSQSRSTG ou la valeur par défaut si le paramètre n'a pas été spécifié.</p>

Tableau 51. Variables globales relatives aux informations d'état non liées au profil (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_FORM_PANEL	DSQASUBP	02	<p>Panneau de format en cours ; les valeurs admises sont :</p> <p>1 pour FORM.MAIN</p> <p>2 pour FORM.COLUMNS</p> <p>3 pour FORM.PAGE</p> <p>4 pour FORM.FINAL</p> <p>5 pour FORM.BREAK1</p> <p>6 pour FORM.BREAK2</p> <p>7 pour FORM.BREAK3</p> <p>8 pour FORM.BREAK4</p> <p>9 pour FORM.BREAK5</p> <p>10 pour FORM.BREAK6</p> <p>11 pour FORM.OPTIONS</p> <p>12 pour FORM.CALC</p> <p>13 pour FORM.DETAIL</p> <p>14 pour FORM.CONDITIONS</p> <p>Une valeur vide signifie que le format n'existe pas dans la mémoire temporaire de QMF.</p>
DSQAO_INTERACT	DSQAIACT	01	<p>Paramètre de l'indicateur d'interaction ; les valeurs admises sont :</p> <p>0 pour l'exécution non interactive</p> <p>1 lorsque l'exécution interactive est autorisée</p>
DSQAO_LOCAL_DB2	Aucun	18	<p>Nom d'emplacement de la base de données DB2 for z/OS locale.</p> <p>Cette valeur correspond au nom d'emplacement du sous-système nommé dans la variable DSQAO_SUBSYS_ID. Dans un environnement d'unité d'oeuvre éloignée, DSQ_LOCAL_DB2 correspond au nom du demandeur d'application. Ce nom comprend 16 caractères (rempli à droite par des espaces, le cas échéant).</p>
DSQAO_LOCATION	DSQAITLO	18	<p>Nom d'emplacement de l'objet en cours (le cas échéant).</p> <p>Cette valeur ne s'applique que si un nom en trois parties a été utilisé. Ce nom comprend 16 caractères (rempli à droite par des espaces, le cas échéant).</p>
DSQAO_NLF_LANG	DSQALANG	01	<p>Langue nationale de l'utilisateur ; pour l'environnement régional en anglais, cette valeur est 'E'.</p>
DSQAO_NUM_FETCHED	DSQAROWS	16	<p>Lignes de données extraites ; contient la valeur '0' lorsque l'objet DATA est vide.</p>

Tableau 51. Variables globales relatives aux informations d'état non liées au profil (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_OBJ_NAME	DSQAITMN	128	Nom de la table (contenue dans un rapport), de la requête, de la procédure ou du format présenté dans le panneau actuellement affiché. Si le panneau en cours n'affiche pas d'objet ou si l'objet affiché n'a pas de nom, cette variable contient des blancs.
DSQAO_OBJ_OWNER	DSQAITMO	128	Propriétaire de la table (contenue dans un rapport), de la requête, de la procédure ou du format présenté dans le panneau actuellement affiché. Si le panneau en cours n'affiche pas d'objet ou si l'objet affiché n'a pas de propriétaire, cette variable contient des blancs.
DSQAO_OTC_LICENSE	Aucun	01	Indique si le produit autonome OTC DB2 QMF for z/OS est installé. 0 Indique que l'identificateur du produit autonome OTC DB2 QMF for z/OS, 5697-QMF, est introuvable. 1 Indique que l'identificateur du produit autonome DB2 QMF for z/OS, 5697-QMF, a été trouvé.
DSQAO_PANEL_TYPE	DSQAITEM	01	Type du panneau en cours ; les valeurs admises sont : 1 pour HOME 2 pour QUERY 3 pour REPORT 4 pour FORM 5 pour PROC 6 pour PROFILE 7 pour CHART 8 pour LIST 9 pour l'éditeur de table A pour GLOBALS
DSQAO_QMF_RELEASE	DSQAREVN	02	Numéro d'édition numérique de QMF, qui s'affiche dans les enregistrements d'en-tête des formats, rapports et requêtes assistées exportés. Pour QMF Version 12.1, cette valeur est '19'.
DSQAO_QMF_VER_RLS	DSQAQMF	10	Version et édition de QMF. Pour QMF Version 12.1, cette valeur est 'QMFV12R1.0'.
DSQAO_QMFADM	Aucun	01	Droits d'accès d'administrateur QMF : 0 L'ID autorisation en cours ne dispose pas des droits d'accès d'administrateur QMF. 1 L'ID autorisation en cours dispose des droits d'accès d'administrateur.

Tableau 51. Variables globales relatives aux informations d'état non liées au profil (suite)

Nom de variable de l'interface callable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_QRY_SUBTYPE	DSQASUBI	01	Sous-type de requête ; les valeurs admises sont : 1 pour un sous-type de SQL 2 pour un sous-type de QBE 3 pour un sous-type de PROMPTED Une valeur vide signifie que le panneau en cours n'est pas QUERY.
DSQAO_QUERY_MODEL	DSQAMODL	01	Modèle de la requête en cours ; la valeur doit être '1' (pour le modèle de données relationnelles).
DSQAO_SAME_CMD	DSQACMDM	01	Les valeurs possibles sont : 0 si les deux commandes ne sont pas identiques 1 si les deux commandes sont identiques
DSQAO_STO_PROC_INT	Aucun	01	Indique si QMF for TSO a été démarré en tant que procédure mémorisée DB2 for z/OS. Les valeurs admises sont : 0 QMF n'a pas été démarré en tant que procédure mémorisée. 1 QMF a été démarré en tant que procédure mémorisée.
DSQAO_SUBSYS_ID	Aucun	04	Si QMF s'exécute dans TSO, cette valeur correspond à l'ID du sous-système DB2 local auquel QMF est attaché. Si vous spécifiez une valeur pour le paramètre du programme DSQSSUBS dans CICS, cette variable globale contient cette valeur. Le paramètre est toléré et la valeur n'est pas traitée. La valeur est placée dans la zone de variable globale mais rien ne se passe. Cette logique permet au même programme d'être utilisé dans plusieurs environnements.
DSQAO_SYSTEM_ID	DSQASYST	01	Système d'exploitation en cours ; les valeurs admises sont : 2 TSO sous z/OS 3 TSO ou z/OS natif 5 CICS
DSQAO_TERMINATE	DSQCSESC	01	Indicateur d'arrêt QMF ; les valeurs admises sont : 0 si la session n'a pas été marquée pour l'arrêt 1 si la session a été marquée pour l'arrêt
DSQAO_TIME_FORMAT	Aucun	05	Contient la valeur spécifiée dans SYSIBM.TIME_FORMAT. Les valeurs peuvent être ISO, USA, EUR, JIS ou LOCAL.
DSQAO_VARIATION	DSQAVARN	02	Numéro du panneau Format ; une valeur vide signifie que FORM.DETAIL n'est pas le panneau en cours.

Variables globales stockant les informations d'état associées aux profils

Les variables globales DSQAP stockent des informations associées aux paramètres de profil QMF. Aucune d'elles n'est modifiable à l'aide de la commande SET GLOBAL.

Tableau 52. Variables globales stockant les informations d'état associées aux profils

Nom de variable de l'interface appellable	Nom de variable de l'interface de commande	Longueur	Description
DSQAP_CASE	DSQAPCAS	01	Paramètre CASE ; les valeurs admises sont : 1 pour les majuscules (UPPER) 2 pour la casse mixte (MIXED) 3 pour la chaîne (STRING) Si votre site utilise la prise en charge RACF pour les mots de passe à casse mixte sous TSO, affectez à ce paramètre la valeur 2. Sans ce paramètre, toutes les entrées (y compris les mots de passe) sont converties en majuscule, provoquant alors l'échec de la commande CONNECT. Lorsque vous affectez au paramètre CASE la valeur MIXED, utilisez uniquement des majuscules pour vos entrées car QMF ne reconnaît que les commandes en majuscule.
DSQAP_CONFIRM	DSQAPRMP	01	Paramètre CONFIRM ; les valeurs admises sont : 0 pour NO 1 pour YES
DSQAP_DECIMAL	DSQAPDEC	01	Paramètre DECIMAL ; les valeurs admises sont : 1 pour le point (PERIOD) 2 pour la virgule (COMMA) 3 pour le français (FRENCH)
DSQAP_LENGTH	DSQAPLEN	18	Paramètre LENGTH ; sa valeur est celle du paramètre (de '1' à '999' ou 'CONT').
DSQAP_PFKKEY_TABLE	DSQAPPFK	31	Nom de la table des touches de fonction.
DSQAP_PRINTER	DSQAPPRT	08	Paramètre PRINTER ; les valeurs admises sont : <ul style="list-style-type: none"> • Pseudonyme pour une imprimante GDDM. • Blancs pour l'imprimante associée à DSQPRINT.
DSQAP_QUERY_LANG	DSQAPLNG	01	Paramètre LANGUAGE ; les valeurs admises sont : 1 pour SQL 2 pour QBE 3 pour PROMPTED (création assistée de requêtes)
DSQAP_QUERY_MODEL	DSQAMODP	01	Paramètre MODEL (la valeur est '1' pour relationnel).

Tableau 52. Variables globales stockant les informations d'état associées aux profils (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAP_RESOURCE_GRP	DSQAPGRP	16	Paramètre RESOURCE GROUP.
DSQAP_SPACE	DSQAPSPC	50	Paramètre SPACE ; sa valeur est celle du paramètre.
DSQAP_SYNONYM_TBL	DSQAPSYN	31	Nom de la table de synonymes utilisée pour la session QMF en cours. Lorsqu'un utilisateur entre un synonyme de commande, la définition du synonyme doit être stockée dans la table nommée dans cette zone, sinon la commande échoue.
DSQAP_TRACE	DSQAPTRC	18	Paramètre TRACE ; les valeurs admises sont : TOUT (traçage maximum) NONE (traçage minimum) Vous pouvez également spécifier une série de lettres et de chiffres qui définit les composants à contrôler ainsi que les niveaux de détail (par exemple, A2L2C1).
DSQAP_WIDTH	DSQAPWID	18	Paramètre WIDTH ; sa valeur est celle du paramètre (de '22' à '999').

Variables globales associées à CICS

Les variables globales DSQAP sont associées aux environnements CICS. La commande SET GLOBAL ne permet de modifier que les variables DSQAP_CICS_PQNAME et DSQAP_CICS_PQTYPE.

Lorsque le type de file d'attente est "données transitoires" (TD), la longueur maximum du nom de file d'attente correspondant est 4. Par exemple, si DSQAO_CICS_SQTYPE est TD, la longueur maximum de DSQAO_CICS_SQNAME est 4.

Tableau 53. Variables globales associées à l'environnement CICS

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAP_CICS_PQNAME	Aucun	08	Désigne la file d'attente de données CICS qui contiendra la sortie d'impression QMF.
DSQAP_CICS_PQTYPE			Type de stockage CICS utilisé pour contenir la sortie d'impression QMF : TS Ecrit l'impression QMF dans une file de stockage temporaire CICS sur un périphérique de stockage auxiliaire. Il s'agit de la valeur par défaut. TD Ecrit l'impression QMF dans une file d'attente de données transitoires CICS.
DSQAO_CICS_SQNAME	Aucun	08	Désigne la file d'attente de données CICS qui sera utilisée comme fichier auxiliaire.

Tableau 53. Variables globales associées à l'environnement CICS (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQAO_CICS_SQTYPE	Aucun	02	Type de stockage CICS utilisé pour contenir le fichier auxiliaire QMF : TS Ecrit les données auxiliaires QMF dans une file de stockage temporaire CICS sur une unité de mémoire secondaire. Il s'agit de la valeur par défaut. TD Ecrit les données auxiliaires QMF dans une file d'attente de données transitoires CICS.
DSQAO_CICS_TQNAME	Aucun	08	Désigne la file d'attente de données CICS qui contiendra les données de trace QMF.
DSQAO_CICS_TQTYPE	Aucun	02	Type de stockage CICS utilisé pour contenir les données de trace QMF : TS Ecrit la trace QMF dans une file de stockage temporaire CICS sur un périphérique de stockage auxiliaire. TD Ecrit la trace QMF dans une file d'attente de données transitoires CICS. Il s'agit de la valeur par défaut.

Variables globales associées à un message généré par la commande la plus récente

Les variables globales DSQCM contiennent des informations sur la dernière commande QMF exécutée. Aucune de ces variables n'est modifiable à l'aide de la commande SET GLOBAL.

Tableau 54. Variables globales qui capturent des informations sur la dernière commande exécutée

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQCM_MESSAGE	DSQCM_MESSAGE	80	Texte du message.
DSQCM_MESSAGE_ALL	DSQCIMSA	360	Texte du message complet.
DSQCM_MSG_HELP	DSQCIMID	08	ID du panneau d'aide du message.
DSQCM_MSG_NUMBER	DSQCIMNO	08	Numéro du message.
DSQCM_SUB_TXT _{nn}	DSQCIM _{nn}	20	Valeur de substitution <i>nn</i> .

Variables globales associées à l'éditeur de table

Les variables globales DSQCP sont associées aux opérations de l'éditeur de table. Toutes ces variables sont modifiables à l'aide de la commande SET GLOBAL.

Le tableau suivant affiche les variables globales qui sont associées aux opérations de l'éditeur de table. Toutes ces variables sont modifiables à l'aide de la commande SET GLOBAL.

Si l'option CONFIRM de la commande EDIT TABLE est NO, l'éditeur de table supprime l'affichage de tous les panneaux de confirmation. Si l'option CONFIRM est YES, l'éditeur de table détermine quelles catégories de confirmation sont activées en vérifiant les valeurs des variables globales affichées dans cette table.

Les valeurs par défaut de l'éditeur de table sont liées aux valeurs du mot clé SAVE indiquées dans la commande EDIT TABLE :

- Lorsque SAVE=IMMEDIATE, la valeur par défaut de chaque catégorie est l'activation.
- Lorsque SAVE=END, la valeur par défaut des catégories DELETE, MODIFY et END/CANCEL est l'activation ; celle des catégories ADD et CHANGE est la désactivation.

Tableau 55. Variables globales associées à l'éditeur de table

Nom de variable de l'interface appellable	Nom de variable de l'interface de commande	Longueur	Description
DSQCP_RMV_BLANKS	Aucun	01	Conserve ou supprime les blancs à droite dans les colonnes VARCHAR. Cette variable affecte uniquement l'éditeur de table en mode Modifier. Les valeurs possibles sont : 0 Les blancs à droite dans les colonnes VARCHAR ne sont pas supprimés. 1 Les blancs à droite dans les colonnes VARCHAR sont supprimés. Il s'agit de la valeur par défaut.
DSQCP_TEADD	Aucun	01	Affiche un panneau de confirmation après une sous-commande ADD ; les valeurs admises sont : 0 Le panneau est désactivé. 1 Le panneau est activé. 2 Le panneau est activé ou désactivé, selon les valeurs par défaut de l'éditeur de table. Il s'agit de la valeur par défaut.
DSQCP_TECHG	Aucun	01	Affiche un panneau de confirmation après une sous-commande CHANGE ; les valeurs admises sont : 0 Le panneau est désactivé. 1 Le panneau est activé. 2 Le panneau est activé ou désactivé, selon les valeurs par défaut de l'éditeur de table. Il s'agit de la valeur par défaut.
DSQCP_TEDEL	Aucun	01	Affiche un panneau de confirmation après une sous-commande DELETE ; les valeurs admises sont : 0 Le panneau est désactivé. 1 Le panneau est activé. 2 Le panneau est activé ou désactivé, selon les valeurs par défaut de l'éditeur de table. Il s'agit de la valeur par défaut.
DSQCP_TEDFLT	Aucun	01	Caractère réservé indiquant la valeur par défaut d'une colonne dans l'éditeur de table ; il s'agit initialement du signe plus (+).

Tableau 55. Variables globales associées à l'éditeur de table (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQCP_TEDFLT_DBCS	Aucun	04	<p>Caractère DBCS réservé indiquant la valeur par défaut d'une colonne de chaîne graphique dans l'éditeur de table.</p> <p>La valeur doit être un chaîne mixte de 4 octets constituée d'un caractère DBCS, précédée du caractère de code spécial et suivie du caractère de code normal. Ce caractère est initialement un signe plus (+) DBCS. Cette variable globale n'est utilisée que dans un environnement DBCS.</p>
DSQCP_TEEND	Aucun	01	<p>Affiche un panneau de confirmation lorsque vous exécutez une sous-commande END ou CANCEL pour terminer une sous-session de l'éditeur de table.</p> <p>Ce panneau peut s'afficher dans diverses situations :</p> <ul style="list-style-type: none"> • Exécution de la commande END ou CANCEL • Modifications apportées à la base de données • Présence dans l'écran de données modifiées au moment de l'exécution de la commande END ou CANCEL <p>Les valeurs admises sont :</p> <p>0 Le panneau est désactivé.</p> <p>1 Le panneau est activé.</p> <p>2 Le panneau est activé ou désactivé, selon les valeurs par défaut de l'éditeur de table. Il s'agit de la valeur par défaut.</p>
DSQCP_TEMOD	Aucun	01	<p>Affiche un panneau de confirmation lorsque les données affichées sont modifiées et qu'une sous-commande PREVIOUS, CLEAR, SHOW CHANGE, SHOW SEARCH, REFRESH ou NEXT est exécutée. Le panneau résultant inclut le nom de la sous-commande dans son nom. Les valeurs admises sont :</p> <p>0 Le panneau est désactivé.</p> <p>1 Le panneau est activé.</p> <p>2 Le panneau est activé ou désactivé, selon les valeurs par défaut de l'éditeur de table.</p>
DSQCP_TENULL	Aucun	01	<p>Le caractère réservé indiquant la valeur NULL d'une colonne dans l'éditeur de table est initialement un tiret (-).</p>

Tableau 55. Variables globales associées à l'éditeur de table (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQCP_TENULL_DBCS	Aucun	04	<p>Caractère DBCS réservé indiquant la valeur NULL d'une colonne de chaîne graphique dans l'éditeur de table. Ce caractère indique également 'ignorer' dans le contexte des critères de recherche.</p> <p>Cette valeur doit être une chaîne mixte de 4 octets constituée d'un caractère DBCS, précédée du caractère de code spécial et suivie du caractère de code normal. Il s'agit initialement d'un tiret (-) DBCS. Cette variable globale n'est utilisée que dans un environnement DBCS.</p>

Variables globales qui contrôlent les différents affichages

Les variables globales DSQDC contrôlent l'affichage de certains types d'informations. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_COL_LABELS	Aucun	01	<p>Contrôle si l'en-tête de colonne affiché dans les panneaux FORM.MAIN et FORM.COLUMNS correspond par défaut au libellé de base de données affecté à la colonne ou au nom de la colonne dans la table à partir de laquelle il a été sélectionné.</p> <p>0 Les noms de colonne sont utilisés comme en-têtes de colonne dans les formats QMF par défaut.</p> <p>1 Les libellés de base de données sont utilisés comme en-têtes de colonne dans les formats QMF par défaut. Il s'agit de la valeur par défaut.</p>
DSQDC_COST_EST	Aucun	01	<p>Contrôle l'affichage de l'estimation du coût de base de données ; les valeurs admises sont :</p> <p>0 N'affiche pas l'estimation du coût.</p> <p>1 Affiche l'estimation du coût. Il s'agit de la valeur par défaut.</p> <p>2 N'affiche pas les panneaux de statut de la base de données et d'estimation du coût.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_CURRENCY	Aucun	18	<p>Symbole monétaire utilisé lorsque le code d'édition DC est spécifié.</p> <p>La valeur peut être une chaîne de longueur comprise entre 1 et 18. Pour l'anglais, la valeur par défaut est le symbole monétaire euro. La valeur par défaut varie pour les autres langues. Dans un environnement DBCS, cette valeur peut être une chaîne composée à la fois de caractères SBCS et de caractères DBCS. La longueur totale de cette chaîne mixte, incluant le caractère de code spécial et le caractère de code normal, ne peut pas dépasser 18 octets.</p> <p>Si vous avez besoin d'un symbole monétaire qui n'est pas représenté sur le clavier, vous avez la possibilité de le spécifier. Dans une procédure navigationnelle, affectez à la variable DSQDC_CURRENCY la valeur hexadécimale qui est équivalente au symbole correct. Par exemple, la procédure suivante définit le symbole monétaire à HEX '9F', qui spécifie le symbole monétaire euro dans QMF en anglais :</p> <pre data-bbox="976 926 1421 978">/* */ "SET GLOBAL (DSQDC_CURRENCY =" '9F'X</pre> <p>Si des blancs à droite sont nécessaires pour le symbole monétaire, mettez le symbole monétaire entre guillemets. Cet exemple illustre l'insertion de blancs pour la version de QMF en français :</p> <pre data-bbox="976 1136 1421 1167">SET GLOBAL (DSQDC_CURRENCY = 'FR '</pre> <p>Vous pouvez exécuter cette commande à partir de la ligne de commande ou dans une procédure linéaire.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_DISPLAY_RPT	DSQADPAN	01	<p>Affiche un rapport après la commande RUN QUERY ; les valeurs admises sont :</p> <p>0 QMF n'affiche pas le rapport d'une commande RUN QUERY.</p> <p>Il s'agit de la valeur par défaut si QMF est démarré de manière interactive avec DSQQMFn, où n est a lorsque n identifie une fonction NLF (National Language Feature). Cette valeur est également la valeur par défaut si QMF est démarré en mode de traitement par lots. Le changement de cette variable lorsque QMF est démarré en mode de traitement par lots ne provoque pas l'affichage d'écrans QMF.</p> <p>1 QMF affiche automatiquement le rapport.</p> <p>Il s'agit de la valeur par défaut si QMF est démarré à l'aide de l'interface appelable. Cette valeur peut être substituée à l'aide du paramètre de programme DSQADPAN associé à la commande START.</p> <p>Lorsque vous associez cette variable globale à la valeur 1, vous pouvez réviser le rapport affiché et choisir de valider ou d'annuler les modifications. Pour ce faire, appuyez sur F3 (FIN) une fois que vous avez terminé de réviser vos modifications. Vous êtes alors invité à valider ou annuler les modifications. Sélectionnez 1 pour valider vos modifications ou 2 pour les annuler, puis appuyez sur Entrée.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_EC_CHAR	Aucun	05	<p>Code d'édition par défaut défini par l'utilisateur pour les données de type caractère (caractère de longueur fixe, caractère de longueur variable et caractère de très grande longueur.</p> <p>Langage C Ne modifie pas l'affichage des données. Il s'agit de la valeur par défaut.</p> <p>CW Renvoie les données à la ligne dans la colonne en fonction de la largeur.</p> <p>CT Renvoie les données à la ligne à l'extrémité de la colonne, en créant une rupture de ligne à l'espace le plus proche.</p> <p>CDx Renvoie les données à la ligne dans la colonne en fonction du délimiteur que vous spécifiez si elles sont trop longues pour tenir sur une seule ligne. Ce délimiteur peut être n'importe quel caractère, y compris un espace, et il n'apparaît pas dans la sortie.</p> <p>Uxxxx Formatage défini par l'utilisateur. Remplacez xxxx par des caractères entre 0 et 4 (lettres, chiffres ou caractères spéciaux).</p> <p>Vxxxx Formatage défini par l'utilisateur. Remplacez xxxx par des caractères entre 0 et 4 (lettres, chiffres ou caractères spéciaux).</p> <p>B Formatage binaire.</p> <p>BW Formatage binaire avec renvoi des données à la ligne dans la colonne en fonction de la largeur.</p> <p>X Formatage hexadécimal.</p> <p>XW Formatage hexadécimal avec renvoi des données à la ligne dans la colonne en fonction de la largeur.</p> <p>M Affiche les métadonnées (type de données et longueur) et non les données réelles.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_EC_DATE	Aucun	05	<p>Code d'édition par défaut pour les données de type DATE. Les valeurs admises sont :</p> <p>TDYx Année composée de quatre chiffres avec l'année en premier.</p> <p>TDMx Année composée de quatre chiffres avec le mois en premier.</p> <p>TDDx Année composée de quatre chiffres avec le jour en premier.</p> <p>TDYAx Année abrégée composée de deux chiffres avec l'année en premier.</p> <p>TDYmx Année abrégée composée de deux chiffres avec le mois en premier.</p> <p>TDDAx Année abrégée composée de deux chiffres avec le jour en premier.</p> <p>TDL Format de date défini localement.</p> <p>TD Format de date par défaut du système de base de données. Il s'agit de la valeur par défaut de cette variable globale.</p> <p><i>x</i> représente le caractère que vous spécifiez comme délimiteur entre les parties de la date.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_EC_DEC	Aucun	05	<p>Code d'édition défini par l'utilisateur pour les données décimales.</p> <p>E ou EZ Notation scientifique. Un Z situé en seconde position du code d'édition supprime les valeurs nulles.</p> <p>D, DC, DZ, DZC, I, IZ, J, JZ, K, KZ, L, LZ, P, PZ Notation décimale avec des combinaisons différentes de zéros non significatifs, des signes moins pour les nombres négatifs, des séparateurs de milliers, des symboles monétaires et des signes pourcentage.</p> <p>Chaque code peut être suivi d'un nombre (0 à 99) qui indique le nombre de chiffres autorisé après le séparateur décimal.</p> <p>Un C, placé en deuxième ou troisième position du code d'édition D, affiche un symbole monétaire défini par l'utilisateur à la place du symbole monétaire standard.</p> <p>Un Z situé en seconde position du code d'édition supprime les valeurs nulles.</p> <p>La valeur par défaut est L. Lorsque L* est spécifié, QMF formate les données décimales en fonction de la définition des colonnes de la base de données. Ce comportement est cohérent avec les éditions précédentes de QMF.</p> <p>Uxxxx Formatage défini par l'utilisateur. Remplacez xxxx par des caractères entre 0 et 4 (lettres, chiffres ou caractères spéciaux).</p> <p>Vxxxx Formatage défini par l'utilisateur. Remplacez xxxx par des caractères entre 0 et 4 (lettres, chiffres ou caractères spéciaux).</p> <p>M Affiche les métadonnées (type de données et longueur) et non les données réelles.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_EC_NUM	Aucun	05	<p>Code d'édition par défaut défini par l'utilisateur pour les données numériques (entier, chiffre entier et entier complexe.)</p> <p>E ou EZ Notation scientifique. Un Z situé en seconde position du code d'édition supprime les valeurs nulles.</p> <p>D, DC, DZ, DZC, I, IZ, J, JZ, K, KZ, L, LZ, P, PZ Notation décimale avec des combinaisons différentes de zéros non significatifs, des signes moins pour les nombres négatifs, des séparateurs de milliers, des symboles monétaires et des signes pourcentage.</p> <p>Un C, placé en deuxième ou troisième position du code d'édition D, affiche un symbole monétaire défini par l'utilisateur à la place du symbole monétaire standard.</p> <p>Un Z situé en seconde position du code d'édition supprime les valeurs nulles.</p> <p>La valeur par défaut est L.</p> <p>Uxxxx Formatage défini par l'utilisateur. Remplacez xxxx par des caractères entre 0 et 4 (lettres, chiffres ou caractères spéciaux).</p> <p>Vxxxx Formatage défini par l'utilisateur. Remplacez xxxx par des caractères entre 0 et 4 (lettres, chiffres ou caractères spéciaux).</p> <p>M Affiche les métadonnées (type de données et longueur) et non les données réelles.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_EC_TIME	Aucun	05	<p>Code d'édition par défaut pour les données de type TIME. Les valeurs admises sont :</p> <p>TTSx Horloge au format 24 heures (avec secondes).</p> <p>TTCx Horloge au format 12 heures (avec secondes).</p> <p>TTAx Horloge au format abrégé (sans secondes).</p> <p>TTAN Horloge au format abrégé (sans secondes, sans délimiteur).</p> <p>TTUx Format USA.</p> <p>TTL Format d'heure défini localement.</p> <p>TT Format d'heure par défaut du système de base de données. Il s'agit de la valeur par défaut de cette variable globale.</p> <p><i>x</i> représente le caractère que vous spécifiez comme délimiteur entre les parties de l'heure.</p>
DSQDC_LIST_ORDER	Aucun	02	<p>Définit l'ordre de tri par défaut des objets dans une liste d'objets de base de données. Les valeurs possibles du premier caractère sont :</p> <p>1 La liste utilise l'ordre de tri par défaut.</p> <p>2 La liste est triée par propriétaire d'objet.</p> <p>3 La liste est triée par nom d'objet.</p> <p>4 La liste est triée par type d'objet.</p> <p>5 La liste est triée par date de modification.</p> <p>6 La liste est triée par date de dernière utilisation. La liste des commandes qui permettent de mettre à jour cette date est définie à l'aide de la variable globale DSQEC_LAST_RUN.</p> <p>Les valeurs possibles du second caractère sont :</p> <p>A La liste est triée par ordre croissant.</p> <p>D La liste est triée par ordre décroissant.</p> <p>Cette variable ne s'applique qu'aux objets sélectionnés par la commande LIST. Elle ne s'applique ni aux listes générées dans d'autres contextes, comme un panneau d'affichage d'invite, ni aux listes de tables.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_POS_SQLCODE	Aucun	01	<p>Définit l'action prise par QMF lorsqu'un code SQL positif est renvoyé de la base de données. Les valeurs admises sont :</p> <p>0 Ne consigne pas le message dans le fichier de données de trace (DSQDEBUB) et aucun texte d'aide n'est fourni.</p> <p>1 Consigne le message QMF associé au code SQL dans le fichier de données de trace (DSQDEBUB).</p> <p>2 L'aide des messages QMF est disponible pour le code SQL positif.</p> <p>Cette variable globale ne s'applique pas aux codes SQL +495 et +100.</p>
DSQDC_SCROLL_AMT	Aucun	04	<p>Définit la quantité de défilement des panneaux QMF ; les valeurs admises sont :</p> <p>Cur Définit la quantité de défilement sur cursor.</p> <p>QMF fait défiler l'écran vers l'arrière, en amenant la ligne ou la colonne sur laquelle se trouve le curseur en bas de la zone déroulante. De la même façon, QMF amène l'écran vers le haut lorsque vous faites défiler vers l'avant ou vers l'extrémité gauche ou droite lorsque vous faites défiler vers la gauche ou la droite.</p> <p>Half Fixe la quantité de défilement à la moitié de la zone déroulante.</p> <p>Page Fixe la quantité de défilement à une page complète. Il s'agit de la valeur par défaut.</p> <p>n Définit la quantité de défilement sur <i>n</i> lignes ou colonnes. Vous pouvez indiquer un nombre compris entre 1 et 9999 pour <i>n</i>.</p>

Tableau 56. Variables globales qui contrôlent l'affichage de certains types d'informations (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQDC_SHORT_EXPT	Aucun	01	<p>S'applique aux données ou aux tables exportées avec une valeur de QMF sur le paramètre DATAFORMAT de la commande EXPORT. Contrôle la longueur de toutes les zones de nom de colonne dans les enregistrements d'en-tête. Les valeurs admises sont :</p> <p>0 QMF définit la longueur des zones de colonne dans les enregistrements d'en-tête à 30 octets. Il s'agit de la longueur par défaut pour :</p> <ul style="list-style-type: none"> • DB2 for z/OS version 8.1.5 ou ultérieure • DB2 for iSeries version 5.2 ou ultérieure • DB2 for Linux, UNIX and Windows version 8.1 ou ultérieure <p>1 QMF définit la longueur des zones de colonne dans les enregistrements d'en-tête à 18 octets. Il s'agit de la longueur par défaut pour :</p> <ul style="list-style-type: none"> • DB2 for z/OS version 8.1.5 ou antérieure • DB2 for iSeries version 5.2 ou antérieure • DB2 for Linux, UNIX and Windows version 8.1 ou antérieure • Toutes les bases de données DB2 Server for VSE et VM
DSQDC_SHOW_PANID	DSQCPDSP	01	<p>Affiche les ID des panneaux QMF ; les valeurs admises sont :</p> <p>0 Supprime les identificateurs de panneaux. Il s'agit de la valeur par défaut.</p> <p>1 Affiche les identificateurs des panneaux.</p>

Référence associée:

«Variables globales contrôlant le mode d'exécution des commandes et des procédures», à la page 237

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

VARIABLES GLOBALES CONTRÔLANT LE MODE D'EXÉCUTION DES COMMANDES ET DES PROCÉDURES

Les variables globales DSQEC contrôlent l'exécution des commandes et des procédures. Toutes ces variables globales peuvent être modifiées par la commande SET GLOBAL.

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures

Nom de variable de l'interface appellable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_ALIASES	Aucun	31	Vue d'extraction des listes de tables et des alias lorsque vous demandez une liste de tables à partir d'un emplacement DB2 for z/OS. S'applique également si le serveur en cours est DB2 for z/OS ou DB2 for Linux, UNIX and Windows.
DSQEC_BUFFER_SIZE	Aucun	03	Définit la longueur du tampon de données utilisé pour extraire des données de la base de données. Les valeurs valides sont comprises entre 4 et 256 (chaque entier représente 1 Ko ; par exemple, 4 est égal à 4 Ko et 256 est égal à 256 Ko, etc.). La valeur par défaut est 4 (4 Ko).
DSQEC_CC	Aucun	01	Supprime les caractères de contrôle chariot dans le format de sortie du rapport ; les valeurs possibles sont : 0 Aucun caractère de contrôle chariot dans la colonne 1. 1 Le contrôle chariot est actif ; le rapport comporte un caractère de contrôle chariot dans la colonne 1.
DSQEC_COLS_LDB2	Aucun	31	Vue d'extraction des informations de colonne pour une table à l'emplacement en cours, si cet emplacement est DB2 for z/OS.
DSQEC_COLS_RDB2	Aucun	31	Vue d'extraction des informations de colonne pour une table à un emplacement DB2 for z/OS distant (s'il ne s'agit pas de l'emplacement distant).
DSQEC_COLS_SQL	Aucun	31	Vue d'extraction des informations de colonne pour une table d'une base de données DB2 for VSE et VM.
DSQEC_CON_ACC_RES	Aucun	01	S'applique aux requêtes SELECT exécutables que QMF soumet à DB2 for z/OS. Utilisez cette variable pour définir le comportement de la base de données lorsque les données à sélectionner sont verrouillées par une opération d'insertion, de mise à jour ou de suppression. Lorsque vous définissez cette variable, QMF spécifie la clause associée à la valeur de la variable sur l'attribut de résolution des accès simultanés de l'instruction PREPARE de la requête SELECT. Les requêtes SELECT exécutables peuvent résulter non seulement de requêtes QMF (telles que des requêtes SQL SELECT, des requêtes assistées ou des requêtes QBE P.), mais également d'autres opérations QMF telles que DISPLAY TABLE. Les valeurs admises sont : 0 QMF ne spécifie aucune option de résolution des accès simultanés sur l'instruction PREPARE associée à l'instruction SQL SELECT en attente. Il s'agit de la valeur par défaut. 1 SKIP LOCKED DATA Cette valeur peut être spécifiée pour des instructions SELECT exécutables dirigées vers DB2 for z/OS version 9 (NFM) ou ultérieure. 2 USE CURRENTLY COMMITTED Cette valeur peut être spécifiée pour des instructions SELECT exécutables dirigées vers DB2 for z/OS version 10 (NFM) ou ultérieure. 3 WAIT FOR OUTCOME Cette valeur peut être spécifiée pour des instructions SELECT exécutables dirigées vers DB2 for z/OS version 10 (NFM) ou ultérieure.

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_CURR_FOLDER	Aucun	128	<p>Indique le nom du dossier en cours à utiliser pour les commandes QMF qui autorisent le traitement de dossiers (SAVE, LIST et ERASE). La valeur par défaut est vide.</p> <p>Lorsqu'un nom de dossier est identifié dans cette variable globale, ce dossier est utilisé lors du traitement de toute commande QMF utilisant des objets de dossier QMF. Par exemple, lorsque DSQEC_CURR_FOLDER est défini et que la commande SAVE QUERY AS Q1 est exécutée, la requête est sauvegardée et l'objet de requête est inclus dans le dossier identifié dans la variable globale.</p> <p>Vous pouvez remplacer cette variable globale en spécifiant le nom de dossier avec le mot clé FOLDER associé à la commande QMF. Dans ce cas, le nom du dossier spécifié avec le mot clé FOLDER remplace le nom par défaut indiqué dans la variable globale DSQEC_CURR_FOLDER. Si cette dernière est vide et que le mot clé FOLDER n'est pas spécifié, le traitement du dossier n'est pas utilisé.</p> <p>Restriction : Cette variable globale n'est pas prise en charge lorsque QMF est connecté à DB2 Server for VSE et VM.</p>
DSQEC_DISABLEADM	Aucun	01	<p>Suppression des droits d'accès d'administrateur QMF. Lorsque la valeur de cette variable globale est modifiée, l'effet est immédiat. Les valeurs possibles sont :</p> <p>0 Les droits d'accès d'administrateur QMF sont disponibles (si l'ID autorisation est associé aux droits d'accès d'administrateur QMF).</p> <p>1 Les droits d'accès d'administrateur QMF sont supprimés (quels que soient les droits d'accès de l'ID autorisation).</p> <p>La valeur initiale par défaut de cette variable globale peut être substituée par la routine d'exit d'initialisation DSQUOPTS.</p>
DSQEC_DSALLOC_DIR	Aucun	03	<p>Indique le nombre de blocs de répertoire à utiliser lors de l'exportation d'un membre d'un nouvel ensemble de données PDS dans TSO. La valeur doit être supérieure à zéro pour les ensembles de données PDS.</p> <p>Si vous utilisez le type par défaut du site de l'ensemble de données ou les ensembles de données PDSE, QMF ignore la valeur de cette variable globale. Pour utiliser le type par défaut du site de l'ensemble de données, affectez à DSQEC_PO la valeur 0. Pour utiliser les ensembles de données PDSE, affectez à DSQEC_PO la valeur 2.</p> <p>Si votre site utilise des ensembles de données séquentiels, affectez à cette variable globale la valeur zéro.</p>
DSQEC_DSALLOC_PRI	Aucun	08	<p>QMF alloue des ensembles de données dans les pistes. Cette variable globale spécifie la quantité principale de pistes pour l'ensemble de données TSO destiné au stockage des résultats de la commande QMF EXPORT.</p> <p>Les valeurs peuvent être comprises entre 1 et la taille maximale autorisée par l'unité de stockage et le système d'exploitation. La valeur par défaut est 15. Une valeur de zéro n'est pas admise.</p> <p>Les ensembles de données PS, PDS et PDSE peuvent avoir une valeur maximum de 16 777 215 pistes.</p>
DSQEC_DSALLOC_SEC	Aucun	08	<p>QMF alloue des ensembles de données dans les pistes. Cette variable globale spécifie la quantité secondaire de pistes pour l'ensemble de données TSO destiné au stockage des résultats de la commande QMF EXPORT.</p> <p>Les valeurs peuvent être comprises entre zéro et la taille maximale autorisée par l'unité de stockage et le système d'exploitation. La valeur par défaut est de 105 pistes.</p> <p>Les ensembles de données PS et PDS peuvent avoir une valeur maximum de 65 535 ; les ensembles de données PDSE peuvent avoir une valeur maximum de 16 777 215 pistes.</p>
DSQEC_DSRECL1	Aucun	05	<p>Indique la longueur des enregistrements logiques (LRECL) qui doit être utilisée lorsqu'une requête SQL ou une procédure QMF est exportée vers un nouvel ensemble de données. Les valeurs valides sont comprises entre 79 et 32 760.</p> <p>La valeur par défaut est 79.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_DS_SUPPORT	Aucun	01	<p>Fournit le support pour QMF Data Service (QDS)</p> <p>0 Ne pas autoriser l'accès à QMF Data Service (par défaut).</p> <p>1 Autoriser l'accès à QMF Data Service.</p> <p>La requête SQL est analysée par le composant QMF Data Service. Si un objet qui est référencé dans la requête est défini dans le composant QMF Data Service, l'intégralité de la requête est exécutée par QDS. Si aucun des objets référencés dans la requête n'accède à un objet défini dans QDS, la requête est exécutée par la connexion DB2 en cours.</p> <p>Si le service QDS n'a pas pu être chargé ou n'est pas disponible, cette valeur est ignorée et toutes les demandes sont acheminées vers DB2.</p>
DSQEC_EDITOR	Aucun	18	<p>Indique la valeur à utiliser pour le mot clé EDITOR dans la commande EDIT lorsque le mot clé EDITOR n'est pas spécifié.</p> <p>Les valeurs valides pour cette variable globale sont :</p> <p>PDF L'éditeur ISPF/PDF est utilisé pour éditer une procédure ou une requête. Pour utiliser l'éditeur ISPF/PDF, lancez QMF comme dialogue ISPF.</p> <p>EE L'éditeur étendu SQL QUERY ou PROC est utilisé pour éditer une procédure ou une requête.</p> <p>nom_éditeur Peut désigner tout autre éditeur autorisé, ou bien le nom d'une liste de commandes (CLIST) destinée à lancer un éditeur. Pour plus d'informations sur les éditeurs disponibles, prenez contact avec votre administrateur QMF.</p> <p>La valeur par défaut est vide.</p>
DSQEC_EXPL_MODE	Aucun	07	<p>Indique le paramètre à utiliser pour le registre spécial DB2 CURRENT EXPLAIN MODE lorsque la commande RUN QUERY est exécutée. Le registre spécial contrôle le comportement de la fonction EXPLAIN pour les instructions SQL dynamiques éligibles. Avant l'exécution d'une requête, QMF affecte au registre spécial CURRENT EXPLAIN MODE la valeur spécifiée par cette variable globale.</p> <p>Les valeurs valides pour cette variable globale sont :</p> <p>NON La fonction EXPLAIN est désactivée et aucune information EXPLAIN n'est capturée lors de l'exécution des instructions dynamiques explicables. Il s'agit de la valeur par défaut.</p> <p>OUI La fonction EXPLAIN est activée et les informations EXPLAIN sont insérées dans les tables EXPLAIN pour les instructions SQL dynamiques éligibles une fois l'instruction préparée et exécutée. Toutes les instructions SQL dynamiques sont compilées et exécutées.</p> <p>EXPLAIN La fonction EXPLAIN est activée et les informations EXPLAIN sont insérées dans les tables EXPLAIN pour les instructions SQL dynamiques éligibles une fois l'instruction préparée. Les instructions dynamiques, à l'exception de SET, ne sont pas exécutées.</p> <p>Pour les serveurs autres que DB2 for Linux, UNIX and Windows ou DB2 10 for z/OS (NFM) ou version ultérieure, la seule valeur valide est NO.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appellable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_EXTND_STG	Aucun	31	<p>Indique le nombre de mégaoctets de la zone de stockage étendu que QMF acquiert à chaque demande au gestionnaire de stockage étendu lorsque le paramètre du programme DSQSPTYP est défini sur 64BIT. Ce paramètre de programme est disponible dans QMF pour TSO uniquement.</p> <p>Lorsqu'une opération requiert une zone de stockage étendu, QMF demande le volume spécifié jusqu'à ce que l'opération soit terminée ou que la zone de stockage étendu soit épuisée.</p> <p>Lorsque vous définissez cette variable globale, tenez compte de la taille moyenne des objets DATA avec lesquels travaillent vos utilisateurs QMF. Si cette taille moyenne est importante et que vous définissez une valeur trop faible, QMF lance de nombreux appels au gestionnaire de stockage étendu pour exécuter l'objet DATA, ce qui peut avoir un impact sur la performance globale.</p> <p>Les valeurs peuvent être comprises entre 1 et 1000. La valeur par défaut est 25, ce qui signifie que QMF demande 25 Mo d'espace de stockage pour chaque requête.</p>
DSQEC_FORM_LANG	Aucun	01	<p>Etablit la langue NLF par défaut dans un format sauvegardé, exporté ou importé ; les valeurs possibles sont :</p> <p>0 Le format utilise la langue NLF principale.</p> <p>1 Le format utilise l'anglais. Il s'agit de la valeur par défaut.</p>
DSQEC_ISOLATION	Aucun	01	<p>Niveau d'isolement de la requête par défaut.</p> <p>Les valeurs admises sont :</p> <p>0 Niveau d'isolement UR (lecture non validée)</p> <p>La lecture non validée peut être utile dans un environnement réparti. Si vous utilisez cette fonction, sachez toutefois que tous les rapports que les utilisateurs verront pourront contenir des données qui ont été supprimées de la base de données après l'affichage des rapports.</p> <p>1 Niveau d'isolement CS (lecture non reproductible)</p> <p>Il s'agit de la valeur par défaut. Si vous utilisez cette fonction, QMF n'affiche pas le rapport tant que toutes les commandes de base de données affectant les données du rapport n'ont pas été exécutées.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_KEEP_THREAD	Aucun	01	<p>Indique si une unité d'exécution est libérée ou laissée active à la fin d'une requête.</p> <p>Cette variable globale n'affecte pas les unités d'exécution qui sont créées pour des procédures exécutées en mode de traitement par lots ou les unités d'exécution créées lorsque QMF est connecté à une base de données éloignée via la commande CONNECT. Lorsque des procédures sont exécutées en mode de traitement par lots, les unités d'exécution sont conservées jusqu'à la fin de la procédure. Lorsque QMF est connecté à une base de données éloignée, les unités d'exécution sont conservées jusqu'à ce que la connexion se termine.</p> <p>Les valeurs valides pour cette variable globale sont :</p> <p>0 L'unité d'exécution est libérée à la fin de la requête. Il s'agit de la valeur par défaut.</p> <p>Si ce paramètre est utilisé, l'instruction SET <i>variable globale</i> DB2 échoue sauf si elle est exécutée dans l'une des situations suivantes :</p> <ul style="list-style-type: none"> • L'instruction est incluse dans une procédure exécutée en mode de traitement par lots. La valeur par défaut de la variable globale DB2 est rétablie à la fin de l'exécution de la procédure. • La commande QMF CONNECT est exécutée pour se connecter à une base de données éloignée et l'instruction SET <i>variable globale</i> DB2 est exécutée sur la base de données éloignée. • L'instruction SET <i>variable globale</i> DB2 est incluse dans une requête à plusieurs instructions et la variable globale QMF DSQEC_RUN_MQ est définie sur 1. La valeur par défaut de la variable globale DB2 est rétablie à la fin de l'exécution de la requête. <p>1 L'unité d'exécution reste active jusqu'à la fin de la session QMF ou si la variable globale DSQEC_KEEP_THREAD est définie à 0. Ce paramètre permet aux utilisateurs d'exécuter l'instruction SET <i>variable globale</i> DB2 afin de définir les variables globales DB2.</p> <p>Si vous définissez des variables globales DB2 alors que la variable DSQEC_KEEP_THREAD est définie à 1 et que vous changez la valeur de la variable en 0, les variables globales DB2 reprennent leurs valeurs par défaut.</p>
DSQEC_LAST_RUN	Aucun	01	<p>Indique l'ensemble de commandes qui déclenche la mise à jour de la zone LAST_USED sur les listes d'objets QMF. Cette zone s'appuie sur la colonne LAST_USED de la table de contrôle Q.OBJECT_DIRECTORY. La valeur de la colonne LAST_USED est mise à jour que la commande émise aboutisse ou pas. Il arrive cependant que la colonne LAST_USED ne soit pas mise à jour immédiatement et qu'elle ne le soit pas en cas d'arrêt anormal de QMF.</p> <p>Les valeurs admises sont :</p> <p>0 QMF met à jour l'horodatage de LAST_USED chaque fois que l'une des commandes suivantes est exécutée :</p> <ul style="list-style-type: none"> • CONVERT • DISPLAY • EXPORT • IMPORT • LAYOUT • PRINT • RUN • SAVE <p>Il s'agit de la valeur par défaut.</p> <p>1 QMF limite les mises à jour de l'horodatage LAST_USED aux commandes RUN, SAVE et IMPORT uniquement.</p> <p>2 QMF limite les mises à jour de l'horodatage LAST_USED à la commande RUN uniquement.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_LIST_OWNER	Aucun	128	<p>Fournit la valeur par défaut du paramètre OWNER de la commande LIST. Indiquez un ID autorisation de 128 caractères au plus. Cette variable est vide par défaut, avec une liste d'objets appartenant à l'ID autorisation en cours.</p> <p>Vous pouvez utiliser des symboles de sélection dans la valeur de la variable. Utilisez un trait de soulignement (_) à la place d'un caractère unique et un signe de pourcentage (%) à la place de caractères (zéro ou plus). Par exemple, la commande suivante, suivie d'une commande LIST, demande à QMF de répertorier uniquement les objets appartenant à des ID utilisateur commençant par les caractères RO :</p> <pre>SET GLOBAL (DSQEC_LIST_OWNER=RO%</pre> <p>La commande suivante définit le propriétaire par défaut sur n'importe quel ID utilisateur commençant par I, avec n'importe quel caractère en seconde position et n'importe quel caractère pour les positions suivantes :</p> <pre>SET GLOBAL (DSQEC_LIST_OWNER=I_%</pre> <p>La valeur que vous définissez à l'aide de cette variable globale ne s'applique pas aux listes affichées lorsque vous appuyez sur la touche List dans les panneaux QMF autres que le panneau principal.</p>
DSQEC_LOB_COLMAX	Aucun	10	<p>Indique la taille maximale des données d'une colonne LOB pouvant être extraites, en octets, jusqu'à la taille maximale LOB de 2147483637 ou 2 Go.</p> <p>Par défaut, les métadonnées LOB sont extraites à la place des données LOB. Cependant, si un code d'édition autre que M est spécifié ou que la variable globale DSQEC_LOB_RETRV est définie sur 3, les données LOB sont extraites à la place des métadonnées. Dans ce cas, si un utilisateur interroge une table qui contient des données LOB dont la taille dépasse la limite maximale, une erreur est générée et aucune donnée du rapport ne s'affiche. Si un utilisateur exécute une commande EXPORT TABLE, PRINT TABLE, SAVE DATA ou EXPORT DATA sur une table ou un objet de données contenant des données LOB dont la taille dépasse la limite, une erreur est générée et la commande s'arrête.</p> <p>La valeur par défaut est 0, ce qui signifie pas de limite.</p>
DSQEC_LOB_RETRV	Aucun	01	<p>Indique le mode d'extraction des données ou des métadonnées LOB. Les valeurs correctes sont :</p> <ol style="list-style-type: none"> 1 Affiche les métadonnées LOB dans les résultats. Pour afficher les données LOB réelles, vous pouvez remplacer le code d'édition M par un autre code d'édition. Lorsque cette valeur est spécifiée, QMF utilise les releveurs de coordonnées LOB pour accéder aux données LOB. Il s'agit du paramètre par défaut. 2 Affiche les métadonnées LOB uniquement dans les résultats. Le code d'édition M est le seul code d'édition valide pour les données LOB. Lorsque cette valeur est spécifiée, QMF n'utilise pas les releveurs de coordonnées LOB. 3 Extrait et affiche les données LOB réelles dans les résultats. Lorsque cette valeur est spécifiée, QMF n'utilise pas les releveurs de coordonnées LOB pour accéder aux données LOB.
DSQEC_LOB_SAVE	Aucun	01	<p>Indique si les utilisateurs peuvent sauvegarder les données LOB dans une table de la base de données à l'aide de la commande QMF SAVE DATA ou IMPORT TABLE. Les valeurs correctes sont :</p> <p>0 - Disable LOB Save Indique que les utilisateurs ne peuvent pas exécuter les commandes QMF SAVE DATA ou IMPORT TABLE pour sauvegarder les données dans une table de la base de données si des colonnes contiennent des données LOB. Un message d'erreur s'affiche et aucune donnée n'est sauvegardée s'il existe une colonne LOB.</p> <p>1 - Enable LOB Save Indique si les utilisateurs peuvent sauvegarder les données LOB dans une table de la base de données à l'aide de la commande QMF SAVE DATA ou IMPORT TABLE. Il s'agit de la valeur par défaut.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_NLFCMD_LANG	Aucun	01	Définit la langue NLF attendue pour les commandes. Les valeurs possibles sont : 0 Les commandes doivent être dans la langue NLF principale. Il s'agit de la valeur par défaut. 1 Les commandes doivent être en anglais.
DSQEC_PO	Aucun	01	Spécifie le type d'ensemble de données partitionné (PO) à créer lors de l'exportation d'un objet QMF vers un nouvel ensemble de données TSO. Les valeurs possibles sont : 0 Alloue un ensemble de données du type répertorié comme valeur par défaut pour votre site. Ce type est spécifié dans le membre IGDSMSxx de SYS1.PARMLIB. Il s'agit de la valeur par défaut. 1 Alloue un ensemble de données PDS pour les données exportées. 2 Alloue un ensemble de données PDSE pour les données exportées.
DSQEC_PRO_ENABLE	Aucun	01	Contrôle si un panneau de confirmation doit s'afficher avant que QMF écrase ou annule le contenu des zones de stockage temporaire QUERY, FORM, PROC ou PROFILE. Les valeurs admises sont : 0 Aucun panneau de confirmation ne s'affiche avant l'écrasement du contenu des zones de stockage temporaire prises en charge. Il s'agit de la valeur par défaut. 1 Un panneau de confirmation s'affiche si la variable globale qui correspond à la zone de stockage temporaire en question est également définie sur 1. Les variables globales suivantes contrôlent de façon individuelle les écrasements dans chacune des zones de stockage temporaire prises en charge : <ul style="list-style-type: none"> • DSQEC_PRO_FORM contrôle les écrasements de la zone de stockage temporaire FORM, qui stocke les spécifications de formatage de rapport QMF en cours. • DSQEC_PRO_PROC contrôle les écrasements de la zone de stockage temporaire PROC, qui stocke les procédures QMF en cours. • DSQEC_PRO_PROF contrôle les écrasements de la zone de stockage temporaire PROFILE, qui stocke les paramètres de profil QMF en cours. • DSQEC_PRO_QUERY contrôle les écrasements de la zone de stockage temporaire QUERY, qui stocke la requête QMF en cours.
DSQEC_PRO_FORM	Aucun	01	Cette variable contrôle si un panneau de confirmation doit s'afficher avant que QMF écrase ou annule le contenu de la zone de stockage temporaire FORM. La variable globale DSQEC_PRO_ENABLE doit être définie sur 1. Les valeurs admises sont : 0 Aucun panneau de confirmation ne s'affiche avant l'annulation du contenu de la zone de stockage temporaire. 1 Un panneau de confirmation s'affiche, donnant à l'utilisateur la possibilité de continuer ou d'annuler la commande ayant provoqué l'annulation en attente. Le contenu de la zone de stockage temporaire peut ensuite être sauvegardé à l'aide de la commande SAVE.
DSQEC_PRO_PROC	Aucun	01	Cette variable contrôle si un panneau de confirmation doit s'afficher avant que QMF écrase ou annule le contenu de la zone de stockage temporaire PROC. La variable globale DSQEC_PRO_ENABLE doit être définie sur 1. Les valeurs admises sont : 0 Aucun panneau de confirmation ne s'affiche avant l'annulation du contenu de la zone de stockage temporaire. 1 Un panneau de confirmation s'affiche avant l'annulation du contenu de la zone de stockage temporaire. L'utilisateur peut alors continuer ou annuler la commande ayant entraîné l'annulation en attente. Le contenu de la zone de stockage temporaire peut ensuite être sauvegardé à l'aide de la commande SAVE.

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_PRO_PROF	Aucun	01	Cette variable contrôle si un panneau de confirmation doit s'afficher avant que QMF écrase ou annule le contenu de la zone de stockage temporaire PROFILE. La variable globale DSQEC_PRO_ENABLE doit être définie sur 1. Les valeurs admises sont : 0 Aucun panneau de confirmation ne s'affiche avant l'annulation du contenu de la zone de stockage temporaire. 1 Un panneau de confirmation s'affiche avant l'annulation du contenu de la zone de stockage temporaire. L'utilisateur peut alors continuer ou annuler la commande ayant entraîné l'annulation en attente. Le contenu de la zone de stockage temporaire peut ensuite être sauvegardé à l'aide de la commande SAVE.
DSQEC_PRO_QUERY	Aucun	01	Cette variable contrôle si un panneau de confirmation doit s'afficher avant que QMF écrase ou annule le contenu de la zone de stockage temporaire QUERY. La variable globale DSQEC_PRO_ENABLE doit être définie sur 1. Les valeurs admises sont : 0 Aucun panneau de confirmation ne s'affiche avant l'annulation du contenu de la zone de stockage temporaire. 1 Un panneau de confirmation s'affiche avant l'annulation du contenu de la zone de stockage temporaire. L'utilisateur peut alors continuer ou annuler la commande ayant entraîné l'annulation en attente. Le contenu de la zone de stockage temporaire peut ensuite être sauvegardé à l'aide de la commande SAVE.
DSQEC_RERUN_IPROC	Aucun	01	Réexécute la procédure d'invocation après la commande END ; les valeurs possibles sont : 0 Supprime la réexécution de la procédure d'invocation après la commande END. 1 Réexécute la procédure d'invocation après la commande END. Il s'agit de la valeur par défaut. Si vous démarrez QMF avec une procédure d'invocation, affectez à cette variable la valeur '0' ; QMF met fin à la procédure au lieu de la réexécuter.
DSQEC_RESET_RPT	Aucun	31	Détermine si QMF vous envoie un message d'invite si un objet DATA incomplet dans la zone de stockage temporaire est susceptible d'affecter les performances : les valeurs possibles sont : 0 Le panneau d'invite Reset Report ne s'affiche pas et QMF termine le rapport en cours d'exécution. Il s'agit de la valeur par défaut. 1 Le panneau d'invite Reset Report s'affiche ; ce panneau vous invite à terminer ou à réinitialiser le rapport en cours d'exécution avant de lancer la nouvelle commande. 2 Le panneau d'invite Reset Report ne s'affiche pas et QMF réinitialise le rapport en cours d'exécution.
DSQEC_RUN_MQ	Aucun	01	Indique si la commande RUN QUERY prend en charge plusieurs instructions dans une requête SQL. Les valeurs admises sont : 0 Plusieurs instructions SQL ne sont pas prises en charge. Si vous affectez à cette variable la valeur 0 et que vous exécutez une requête SQL contenant plusieurs instructions, QMF ignore toutes les instructions après le point-virgule. Il s'agit de la valeur par défaut. 1 Plusieurs instructions SQL sont prises en charge. Un point-virgule doit être placé à la fin de chaque instruction, sauf la dernière. Restrictions : Bien qu'une instruction SELECT puisse être insérée dans une requête avec d'autres instructions, une seule instruction SELECT peut être insérée par requête. Les instructions CALL et CREATE PROCEDURE doivent être utilisées seules dans une requête SQL.
DSQEC_SAV_ACCELNM	Aucun	128	Indique le nom de l'accélérateur par défaut à utiliser lors de la création de tables uniquement d'accélérateur à partir de SAVE DATA, IMPORT TABLE et RUN QUERY dans les commandes TABLE. Cette variable est référencée uniquement si le mot clé ACCELERATOR est spécifié. La variable globale peut avoir une valeur vide, mais pas lorsque la variable globale DSQEC_SAV_ALLOWED a la valeur '4'.

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_SAV_ALLOWED	Aucun	01	<p>Détermine si les utilisateurs sauvegardent les données dans une nouvelle table de la base de données ou dans un accélérateur en utilisant QMF SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE. Sauf si l'option 0 est sélectionnée, cette zone n'a pas d'impact sur l'emplacement des tables existantes dans lesquelles se trouvent les données remplacées ou auxquelles les données sont ajoutées. Les tables existantes sont remplacées ou ajoutées dans la base de données ou l'accélérateur quelle que soit la valeur de cette variable.</p> <p>Les valeurs valides pour cette variable globale sont :</p> <p>0 - Disable Save Data Les utilisateurs ne peuvent pas utiliser QMF SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE pour sauvegarder les données dans une table de la base de données ou l'accélérateur. Un message d'erreur s'affiche et aucune donnée n'est sauvegardée.</p> <p>1 - Enable Save Data to database tables only Les utilisateurs peuvent sauvegarder des données dans une table de la base de données en utilisant QMF SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE. Ils ne peuvent pas sauvegarder les données dans des tables uniquement d'accélérateur. Il s'agit du paramètre par défaut.</p> <p>2 - Enable Save Data to accelerator only tables only Les utilisateurs peuvent sauvegarder des données dans une table uniquement d'accélérateur en utilisant QMF SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE. Ils ne peuvent pas sauvegarder les données dans des tables de base de données. La variable globale DSQEC_SAV_ACCELNM contient le nom par défaut de l'accélérateur, mais elle peut être remplacée par le mot clé ACCELERATOR.</p> <p>3 - Enable Save Data to either database or accelerator only tables (database default) Les utilisateurs peuvent sauvegarder les données dans une table de la base de données ou dans une table uniquement d'accélérateur en utilisant QMF SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE. S'il n'existe pas de remplacement de mot clé de commande, comme SPACE ou ACCEL, les tables sont sauvegardées dans la base de données.</p> <p>4 - Enable Save Data to either database or accelerator only tables (accelerator default) Les utilisateurs peuvent sauvegarder les données dans une table de la base de données ou dans une table uniquement d'accélérateur en utilisant QMF SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE. S'il n'existe pas de remplacement de mot de clé de commande, comme SPACE ou ACCELERATOR, les tables sont sauvegardées dans l'accélérateur. Lorsque cette option est choisie, la variable globale DSQEC_SAV_ACCELNM doit contenir le nom de l'accélérateur.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_SESSGLV_SAV	Aucun	01	<p>Contrôle si les entrées utilisateur effectuées dans certaines zones de saisie de données de certains panneaux sont sauvegardées dans les sessions QMF et d'une session à l'autre. Les entrées utilisateurs sont sauvegardées en tant que variables de session et stockées dans la table Q.GLOBAL_VARS sous forme de variables globales qui sont nommées avec un préfixe DXY. La variable globale DSQEC_SESSGLV_SAV est vérifiée tout au long de la session, ainsi que lors du démarrage et de l'arrêt de QMF. Les valeurs correctes sont :</p> <p>0 Si ce paramètre est spécifié au démarrage de QMF, toutes les variables de session sont supprimées de la table Q.GLOBAL_VARS.</p> <p>Si ce paramètre est spécifié au cours d'une session QMF, toutes les variables de session sont supprimées de la mémoire. Aucune variable de session n'est sauvegardée pour la suite de la session en cours, sauf si ce paramètre est remplacé par 1 ou 2.</p> <p>Si ce paramètre est spécifié au démarrage de QMF, toutes les variables de session sont supprimées de la table Q.GLOBAL_VARS, ce qui signifie qu'aucune entrée utilisateur n'est conservée jusqu'à la prochaine session QMF.</p> <p>Il s'agit de la valeur par défaut.</p> <p>1 Si ce paramètre est spécifié au démarrage de QMF, toutes les variables de session de l'utilisateur sont restaurées à partir de la table Q.GLOBAL_VARS.</p> <p>Si ce paramètre est spécifié au cours d'une session QMF, les variables de session sont sauvegardées pour la suite de la session en cours. Par exemple, si vous entrez des valeurs dans le panneau d'invite de la commande LIST, quittez le panneau LIST puis revenez-y au cours de la même session ; ces zones seront remplies avec les valeurs que vous avez précédemment saisies.</p> <p>Si ce paramètre est spécifié à l'arrêt de QMF, toutes les variables de session ayant été créées ou modifiées par l'utilisateur au cours de la session en cours sont annulées et ne sont pas enregistrées dans la table Q.GLOBAL_VARS. Toutes les valeurs de variable de session qui existaient dans la table Q.GLOBAL_VARS avant la session en cours restent inchangées. Vous pouvez utiliser cette option, par exemple, pour réinitialiser les mêmes valeurs de variable de session au démarrage de chaque session QMF.</p> <p>Lorsque la prochaine session QMF démarrera, la valeur reprendra la valeur 0 sauf si elle est écrasée par une variable globale initiale définie par un administrateur.</p> <p>2 Si ce paramètre est spécifié au démarrage de QMF, toutes les variables de session de l'utilisateur sont restaurées à partir de la table Q.GLOBAL_VARS.</p> <p>Si ce paramètre est spécifié au cours d'une session QMF, les variables de session sont sauvegardées pour la suite de la session en cours, sauf s'il prend la valeur 0. Par exemple, si vous entrez des valeurs dans le panneau d'invite de la commande LIST, quittez le panneau LIST puis revenez-y au cours de la même session ; ces zones seront remplies avec les valeurs que vous avez précédemment saisies.</p> <p>Si ce paramètre est spécifié au démarrage de QMF, toutes les variables de session sont sauvegardées dans la table Q.GLOBAL_VARS, ce qui signifie que les entrées utilisateur sauvegardées au cours de la session sont conservées jusqu'à la prochaine session QMF.</p> <p>Ce paramètre s'applique à la plupart des zones des panneaux d'invite de commande qui sont accessibles via les commandes suivantes : CONNECT, CONVERT, DISPLAY, DRAW, EDIT, ERASE, EXPORT, IMPORT, LIST, PRINT, RESET, RUN, SAVE, SET et SHOW.</p>
DSQEC_SHARE	Aucun	31	<p>Indique la valeur par défaut du paramètre SHARE ; les valeurs admises sont :</p> <p>0 Ne partage pas les données avec les autres utilisateurs.</p> <p>1 Partage les données avec les autres utilisateurs.</p>

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_SP_RS_NUM	Aucun	04	Indique quel ensemble de résultats renvoyé par une procédure mémorisée est utilisé pour créer le rapport. Les valeurs admises sont : 0 Ignore les ensembles de résultats. 1 Renvoie le premier ensemble de résultats. 2 Renvoie le second ensemble de résultats. n Renvoie le <i>n</i> ème ensemble de résultats. La valeur maximale pour <i>n</i> est 63. ALL Renvoie tous les ensembles de résultats.
DSQEC_SPAC_OVERRIDE	Aucun	01	Indique si les utilisateurs peuvent remplacer l'espace table indiqué dans le profil QMF. Les valeurs valides pour cette variable globale sont : 0 - Disable Space Keyword Option Les utilisateurs ne peuvent pas utiliser SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE avec l'option de mot clé SPACE. 1 - Enable Space Keyword Option Les utilisateurs peuvent utiliser SAVE DATA, RUN QUERY dans les commandes TABLE ou IMPORT TABLE avec l'option de mot clé SPACE. Il s'agit du paramètre par défaut.
DSQEC_SQLQRYSZ_2M	Aucun	01	Contrôle si les requêtes SQL dont la longueur est supérieure à 32767 octets (32 Ko) sont prises en charge par la commande RUN QUERY. 0 La taille des requêtes SQL dirigées vers les bases de données DB2 for z/OS, DB2 for iSeries et DB2 for Linux, UNIX and Windows est limitée à 32767 octets (32 ko). Il s'agit de la valeur par défaut. 1 La taille des requêtes SQL peut être supérieure à 32 ko. La taille maximale de requête prise en charge varie en fonction du type de base de données à laquelle la requête est destinée : <ul style="list-style-type: none"> • La longueur des requêtes dirigées vers DB2 for z/OS peut aller jusqu'à 2 mo. • La longueur des requêtes dirigées vers DB2 for iSeries ou DB2 for Linux, UNIX and Windows peut aller jusqu'à 65 ko. Ces valeurs maximales supposent que la version de la base de données à laquelle la commande RUN QUERY est destinée prend en charge des requêtes de cette taille. Les requêtes SQL dirigées vers DB2 for VSE et VM sont limitées à 8 ko. Des personnalisations supplémentaires peuvent s'avérer nécessaires pour exécuter des requêtes de plus de 32 ko à partir de QMF for CICS.
DSQEC_TABS_LDB2	Aucun	31	Vue d'extraction des listes de tables et de vues sur le serveur actuel, s'il s'agit de DB2 for z/OS ou de DB2 for Linux, UNIX and Windows.
DSQEC_TABS_RDB2	Aucun	31	Vue d'extraction des listes de tables et de vues sur des sous-systèmes DB2 distants.
DSQEC_TABS_SQL	Aucun	31	Vue d'extraction des listes de tables et de vues pour une base de données DB2 for VSE et VM.
DSQEC_TRACE_LIMIT	Aucun	31	Limite la quantité de sortie de trace au nombre d'octets spécifié. La plage de valeurs valide est comprise entre 0 et 2147483647. Cette variable peut être utilisée pour réduire la taille de la sortie de trace QMF. Elle est généralement définie conformément aux instructions du service de support logiciel IBM.

Tableau 57. Variables globales contrôlant le mode d'exécution des commandes et des procédures (suite)

Nom de variable de l'interface appelable	Nom de variable de l'interface de commande	Longueur	Description
DSQEC_TRACE_MODULE	Aucun	54	<p>Contient les noms des modules QMF à tracer.</p> <p>Vous pouvez spécifier jusqu'à 6 modules en les séparant par une virgule.</p> <p>Une fois les noms de module spécifiés dans la variable globale, initiez la trace en émettant la commande SET PROFILE avec le mot clé TRACE afin de définir la valeur ALL. Exemple : SET PROFILE (TRACE=ALL</p> <p>Remarque : si des modules sont spécifiés via la commande SET GLOBAL depuis la ligne de commande, les noms de module doivent être placés entre apostrophes.</p>
DSQEC_TWO_GB_ROW	Aucun	01	<p>Contrôle la longueur des lignes renvoyées dans les rapports QMF. utilisez l'une des valeurs suivantes :</p> <p>0 Limite la longueur d'une ligne de données dans un rapport QMF à 32 ko, sauf si le rapport contient une colonne LOB ou XML.</p> <p>1 Augmente la longueur maximale d'une ligne de données, lui permettant ainsi de passer de 32 ko à 2 Go.</p> <p>Important :</p> <ul style="list-style-type: none"> • Quel que soit le paramètre DSQEC_TWO_GB_ROW défini, jusqu'à 2 Go de données XML, CLOB ou BLOB et jusqu'à 1 Go de données DBCLOB peuvent s'afficher par défaut. Toutefois, la longueur maximale d'une ligne LOB peut être restreinte par la variable globale DSQEC_LOB_COLMAX. • Quel que soit le paramètre de la variable DSQEC_TWO_GB_ROW défini, une table unique ne peut pas avoir une taille d'enregistrement maximale supérieure à la taille de la page. DB2 stocke les enregistrements dans des pages de 4 ko, 8 ko, 16 ko ou 32 ko. Par conséquent, la longueur maximale d'une ligne de données pouvant être affichée reste égale à 32 ko lorsque vous affichez une table unique. Si vous affichez ou sélectionnez des données d'une vue qui joint plusieurs tables, la longueur de ligne peut atteindre 2 Go. <p>Compte tenu de ces considérations sur la taille des pages, la longueur d'une ligne de données d'un rapport QMF pouvant être enregistrée à l'aide de la commande SAVE DATA est également limitée à 32 ko. La capacité de sauvegarde des données LOB est contrôlée par la variable globale DSQEC_LOB_SAVE.</p>
DSQEC_USERGLV_SAV	Aucun	01	<p>Détermine si les variables globales créées ou modifiées par l'utilisateur, notamment celles qui commencent par "DSQ", sont sauvegardées à la fin de la session QMF. Les valeurs à sauvegarder sont stockées dans la table Q.GLOBAL_VARS et associées à l'ID utilisateur de la session. Si les valeurs sont sauvegardées, elles sont restaurées au début de la session QMF suivante de l'utilisateur. Les valeurs correctes sont :</p> <p>0 Lorsque QMF s'arrête, toutes les variables globales sont supprimées de la table Q.GLOBAL_VARS, et aucune variable globale de la session en cours n'est sauvegardée dans la table Q.GLOBAL_VARS. Il s'agit de la valeur par défaut.</p> <p>1 Lorsque QMF s'arrête, toutes les variables globales ayant été créées ou modifiées par l'utilisateur au cours de la session en cours sont annulées et ne sont pas enregistrées dans la table Q.GLOBAL_VARS. Toutes les valeurs de variable globale déjà stockées dans la table Q.GLOBAL_VARS restent inchangées dans la session QMF en cours. Vous pouvez utiliser cette option, par exemple, pour réinitialiser les mêmes valeurs de variable globale au démarrage de chaque session QMF.</p> <p>Lorsque la prochaine session QMF démarrera, la valeur reprendra la valeur 0 sauf si elle est écrasée par une variable globale initiale définie par un administrateur.</p> <p>2 Lorsque QMF s'arrête, toutes les variables globales ayant été créées ou modifiées par l'utilisateur ne sont pas enregistrées dans la table Q.GLOBAL_VARS. Lorsque l'utilisateur redémarrera QMF, les variables globales sauvegardées de la session précédente de l'utilisateur seront restaurées. Toutes les valeurs ayant été définies par un administrateur dans la table Q.GLOBAL_VARS seront remplacées par les valeurs de l'utilisateur sauf si la variable a été définie en lecture seule.</p>

Référence associée:

➡ Instruction PREPARE pour DB2

Reportez-vous aux informations sur l'attribut de résolution des accès simultanés de l'instruction PREPARE.

Variables globales qui stockent les résultats de la commande CONVERT QUERY

Les variables globales DSQQC reflètent les résultats d'une commande CONVERT QUERY. La commande SET GLOBAL ne permet pas de les modifier.

Tableau 58. Variables globales qui reflètent les résultats d'une commande CONVERT QUERY

Nom de variable de l'interface callable	Nom de variable de l'interface de commande	Longueur	Description
DSQQC_LENGTH_ <i>mmm</i>	DSQCL <i>mmm</i>	05	Longueur du résultat converti <i>mmm</i> .
DSQQC_QRY_COUNT	DSQCQCNT	03	Nombre de requêtes dans le résultat converti ; la valeur doit toujours être '1' sauf si la requête initiale est de type QBE I. ou U.
DSQQC_QRY_LANG	DSQCQLNG	01	Langage de la requête convertie : les valeurs admises sont : 1 pour SQL 2 pour QBE 3 pour la création assistée de requêtes
DSQQC_QRY_TYPE	DSQCQTYP	Non précisé	Premier mot dans les résultats convertis.
DSQQC_RESULT_ <i>mmm</i>	DSQCQ <i>mmm</i>	Non précisé	<i>mmm</i>

Variables globales qui affichent des informations sur le message d'erreur RUN QUERY

Les variables globales DSQQM stockent les résultats d'une commande RUN QUERY. La commande SET GLOBAL ne permet pas de les modifier.

Tableau 59. Variables globales qui stockent les résultats d'une commande RUN QUERY

Nom de variable de l'interface callable	Nom de variable de l'interface de commande	Longueur	Description
DSQQM_MESSAGE	DSQCIQMG	80	Texte du message de la requête.
DSQQM_MESSAGE_ALL	DSQCIQMA	360	Texte complet du message de la requête.
DSQQM_MSG_HELP	DSQCIQID	08	ID du panneau d'aide du message.
DSQQM_MSG_NUMBER	DSQCIQNO	08	Numéro du message.
DSQQM_SQL_RC	DSQCISQL	16	SQLCODE de la dernière commande ou requête.
DSQQM_SQL_STATE	Aucun	05	SQLSTATE associé au SQLCODE dans DSQQM_SQL_RC, si SQLSTATE est renvoyé par le gestionnaire de base de données.
DSQQM_SUB_TXT_ <i>mm</i>	DSQCIQ <i>mm</i>	20	Valeur de substitution <i>mm</i> .
DSQQM_SUBST_VARS	DSQCIQ00	04	Nombre de variables de substitution.

Variables globales qui stockent les valeurs d'entrée d'écran

Les variables globales DXY stockent les valeurs entrées par les utilisateurs dans les zones de saisie des données si la variable globale DSQEC_SESSGLV_SAV est définie sur 1 ou 2. Les valeurs entrées dans certaines zones de certains écrans seulement sont sauvegardées. Les entrées utilisateur effectuées dans les zones qui ne sont pas indiquées dans le tableau suivant ne sont pas sauvegardées, quel que soit le paramétrage de la variable globale DSQEC_SESSGLV_SAV.

La commande SET GLOBAL permet de modifier l'ensemble de ces variables globales. Cependant, modifiez ou supprimez ces variables avec précaution car cela entraîne la modification des valeurs générées sur les panneaux d'invite de commande.

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PCO1_ <i>ln</i>	01 - 03	CONNECT	Utilisateur
DXY n PCO1_05	–	CONNECT	Emplacement
DXY n PC03_01	–	CONNECT (CICS)	Emplacement
DXY n PCNV_ <i>ln</i>	02 - 07	CONVERT	Nom
DXY n PDSP_ <i>ln</i>	02 - 07	DISPLAY	Nom
DXY n PDSP_ <i>ln</i> _01	02 - 07	DISPLAY QUERY	Nom
DXY n PDSP_ <i>ln</i> _02	02 - 07	DISPLAY PROC	Nom
DXY n PDSP_ <i>ln</i> _03	02 - 07	DISPLAY FORM	Nom
DXY n PDSP_ <i>ln</i> _05	02 - 07	DISPLAY REPORT	Nom
DXY n PDSP_ <i>ln</i> _07	02 - 07	DISPLAY CHART	Nom
DXY n PDSP_ <i>ln</i> _08	02 - 07	DISPLAY TABLE	Nom
DXY n PDRS_ <i>ln</i>	01 - 06	DRAW	Nom
DXY n PDRS_07	–	DRAW	Type
DXY n PDRS_08	–	DRAW	Identificateur
DXY n PEDT_01	–	EDIT	Type
DXY n PED2_ <i>ln</i>	01 - 06	EDIT TABLE	Nom
DXY n PED2_07	–	EDIT TABLE	Mode
DXY n PERA_ <i>ln</i>	02 - 07	ERASE	Nom
DXY n PERA_ <i>ln</i> _01	02 - 07	ERASE QUERY	Nom
DXY n PERA_ <i>ln</i> _02	02 - 07	ERASE PROC	Nom
DXY n PERA_ <i>ln</i> _03	02 - 07	ERASE FORM	Nom
DXY n PERA_ <i>ln</i> _08	02 - 07	ERASE TABLE	Nom
DXY n PEXM_ <i>ln</i>	02 - 07	EXPORT	Nom
DXY n PEXM_ <i>ln</i> _01	02 - 07	EXPORT QUERY	Nom
DXY n PEXM_ <i>ln</i> _02	02 - 07	EXPORT PROC	Nom

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PEXM_ <i>ln</i> _03	02 - 07	EXPORT FORM	Nom
DXY n PEXM_ <i>ln</i> _05	02 - 07	EXPORT REPORT	Nom
DXY n PEXM_ <i>ln</i> _06	02 - 07	EXPORT DATA	Nom
DXY n PEXM_ <i>ln</i> _07	02 - 07	EXPORT CHART	Nom
DXY n PEXM_ <i>ln</i> _08	02 - 07	EXPORT TABLE	Nom
DXY n PXM1_ <i>ln</i>	01 - 05	EXPORT, puis Entrée (dans TSO)	Vers
DXY n PXM1_ <i>ln</i> _01	01 - 05	EXPORT QUERY, puis Entrée (dans TSO)	Vers
DXY n PXM1_ <i>ln</i> _02	01 - 05	EXPORT PROC, puis Entrée (dans TSO)	Vers
DXY n PXM1_06	–	EXPORT, puis Entrée (dans TSO)	Membre
DXY n PXM1_06_01	–	EXPORT QUERY, puis Entrée (dans TSO)	Membre
DXY n PXM1_06_02	–	EXPORT PROC, puis Entrée (dans TSO)	Membre
DXY n PXM2_01_07	–	EXPORT CHART, puis Entrée (dans TSO)	Membre
DXY n PXM3_ <i>ln</i> _05	01 - 05	EXPORT REPORT, puis Entrée (dans TSO)	Vers
DXY n PXM3_06_05	–	EXPORT REPORT, puis Entrée (dans TSO)	Membre
DXY n PXM3_08_05	–	EXPORT REPORT, puis Entrée (dans TSO)	Form. données
DXY n PXM4_ <i>ln</i> _06	01 - 05	EXPORT DATA, puis Entrée (dans TSO)	Vers
DXY n PXM4_06_06	–	EXPORT DATA, puis Entrée (dans TSO)	Membre
DXY n PXM4_08_06	–	EXPORT DATA, puis Entrée (dans TSO)	Form. données
DXY n PXM4_09_06	–	EXPORT DATA, puis Entrée (dans TSO)	Mode sortie
DXY n PXM4_10_06	–	EXPORT DATA, puis Entrée (dans TSO)	En-tête

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PXM5_03	01 - 05	EXPORT FORM, puis Entrée (dans TSO)	Vers
DXY n PXM5_06_03	–	EXPORT FORM, puis Entrée (dans TSO)	Membre
DXY n PXM5_08_03	–	EXPORT FORM, puis Entrée (dans TSO)	Langue
DXY n PXC1_01	–	EXPORT, puis Entrée (dans CICS)	Nom de file d'attente
DXY n PXC1_01_01	–	EXPORT QUERY, puis Entrée (dans CICS)	Nom de file d'attente
DXY n PXC1_01_02	–	EXPORT PROC, puis Entrée (dans CICS)	Nom de file d'attente
DXY n PXC1_02	–	EXPORT, puis Entrée (dans CICS)	Type de file d'attente
DXY n PXC1_02_01	–	EXPORT QUERY, puis Entrée (dans CICS)	Type de file d'attente
DXY n PXC1_02_02	–	EXPORT PROC, puis Entrée (dans CICS)	Type de file d'attente
DXY n PXC1_04	–	EXPORT, puis Entrée (dans CICS)	Attente
DXY n PXC1_04_01	–	EXPORT QUERY, puis Entrée (dans CICS)	Attente
DXY n PXC1_04_02	–	EXPORT PROC, puis Entrée (dans CICS)	Attente
DXY n PXC3_01_05	–	EXPORT REPORT, puis Entrée (dans CICS)	Nom de file d'attente
DXY n PXC3_02_05	–	EXPORT REPORT, puis Entrée (dans CICS)	Type de file d'attente
DXY n PXC3_04_05	–	EXPORT REPORT, puis Entrée (dans CICS)	Attente
DXY n PXC3_05_05	–	EXPORT REPORT, puis Entrée (dans CICS)	Form. données
DXY n PXC4_01_06	–	EXPORT DATA, puis Entrée (dans CICS)	Nom de file d'attente

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PXC4_02_06	–	EXPORT DATA, puis Entrée (dans CICS)	Type de file d'attente
DXY n PXC4_04_06	–	EXPORT DATA, puis Entrée (dans CICS)	Attente
DXY n PXC4_05_06	–	EXPORT DATA, puis Entrée (dans CICS)	Form. données
DXY n PXC4_06_06	–	EXPORT DATA, puis Entrée (dans CICS)	Mode sortie
DXY n PXC4_07_06	–	EXPORT DATA, puis Entrée (dans CICS)	En-tête
DXY n PXC5_01_03	–	EXPORT FORM, puis Entrée (dans CICS)	Nom de file d'attente
DXY n PXC5_02_03	–	EXPORT FORM, puis Entrée (dans CICS)	Type de file d'attente
DXY n PXC5_04_03	–	EXPORT FORM, puis Entrée (dans CICS)	Attente
DXY n PXC5_05_03	–	EXPORT FORM, puis Entrée (dans CICS)	Langue
DXY n PIMM_ <i>ln</i>	02 - 07	IMPORT (dans TSO)	Nom
DXY n PIMM_ <i>ln</i> _01	02 - 07	IMPORT QUERY (dans TSO)	Nom
DXY n PIMM_ <i>ln</i> _02	02 - 07	IMPORT PROC (dans TSO)	Nom
DXY n PIMM_ <i>ln</i> _03	02 - 07	IMPORT FORM (dans TSO)	Nom
DXY n PIMM_ <i>ln</i> _06	02 - 07	IMPORT DATA (dans TSO)	Nom
DXY n PIMM_ <i>ln</i> _08	02 - 07	IMPORT TABLE (dans TSO)	Nom
DXY n PIMM_ <i>ln</i>	08 - 13	IMPORT (dans TSO)	De
DXY n PIMM_ <i>ln</i> _01	08 - 13	IMPORT QUERY (dans TSO)	De
DXY n PIMM_ <i>ln</i> _02	08 - 13	IMPORT PROC (dans TSO)	De
DXY n PIMM_ <i>ln</i> _03	08 - 13	IMPORT FORM (dans TSO)	De
DXY n PIMM_ <i>ln</i> _06	08 - 13	IMPORT DATA (dans TSO)	De
DXY n PIMM_ <i>ln</i> _08	08 - 13	IMPORT TABLE (dans TSO)	De

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PIMM_14	–	IMPORT (dans TSO)	Membre
DXY n PIMM_14_01	–	IMPORT QUERY (dans TSO)	Membre
DXY n PIMM_14_02	–	IMPORT PROC (dans TSO)	Membre
DXY n PIMM_14_03	–	IMPORT FORM (dans TSO)	Membre
DXY n PIMM_14_06	–	IMPORT DATA (dans TSO)	Membre
DXY n PIMM_14_08	–	IMPORT TABLE (dans TSO)	Membre
DXY n PIQF_03	–	IMPORT, puis Entrée (dans TSO)	Commentaire
DXY n PIQF_03_01	–	IMPORT QUERY, puis Entrée (dans TSO)	Commentaire
DXY n PIQF_03_02	–	IMPORT PROC, puis Entrée (dans TSO)	Commentaire
DXY n PIQL_03_03	–	IMPORT FORM, puis Entrée (dans TSO)	Commentaire
DXY n PIQL_04_03	–	IMPORT FORM, puis Entrée (dans TSO)	Langue
DXY n PITB_01_08	–	IMPORT TABLE, puis Entrée (dans TSO)	Commentaire
DXY n PIMC_ <i>ln</i>	02 - 07	IMPORT (dans CICS)	Nom
DXY n PIMC_ <i>ln</i> _01	02 - 07	IMPORT QUERY (dans CICS)	Nom
DXY n PIMC_ <i>ln</i> _02	02 - 07	IMPORT PROC (dans CICS)	Nom
DXY n PIMC_ <i>ln</i> _03	02 - 07	IMPORT FORM (dans CICS)	Nom
DXY n PIMC_ <i>ln</i> _06	02 - 07	IMPORT DATA (dans CICS)	Nom
DXY n PIMC_ <i>ln</i> _08	02 - 07	IMPORT TABLE (dans CICS)	Nom
DXY n PIMC_08	–	IMPORT (dans CICS)	Nom de file d'attente
DXY n PIMC_08_01	–	IMPORT QUERY (dans CICS)	Nom de file d'attente

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PIMC_08_02	–	IMPORT PROC (dans CICS)	Nom de file d'attente
DXY n PIMC_08_03	–	IMPORT FORM (dans CICS)	Nom de file d'attente
DXY n PIMC_08_06	–	IMPORT DATA (dans CICS)	Nom de file d'attente
DXY n PIMC_08_08	–	IMPORT TABLE (dans CICS)	Nom de file d'attente
DXY n PIMC_09	–	IMPORT (dans CICS)	Type de file d'attente
DXY n PIMC_09_01	–	IMPORT QUERY (dans CICS)	Type de file d'attente
DXY n PIMC_09_02	–	IMPORT PROC (dans CICS)	Type de file d'attente
DXY n PIMC_09_03	–	IMPORT FORM (dans CICS)	Type de file d'attente
DXY n PIMC_09_06	–	IMPORT DATA (dans CICS)	Type de file d'attente
DXY n PIMC_09_08	–	IMPORT TABLE (dans CICS)	Type de file d'attente
DXY n PIMC_10	–	IMPORT (dans CICS)	Attente
DXY n PIMC_10_01	–	IMPORT QUERY (dans CICS)	Attente
DXY n PIMC_10_02	–	IMPORT PROC (dans CICS)	Attente
DXY n PIMC_10_03	–	IMPORT FORM (dans CICS)	Attente
DXY n PIMC_10_06	–	IMPORT DATA (dans CICS)	Attente
DXY n PIMC_10_08	–	IMPORT TABLE (dans CICS)	Attente
DXY n PLST_01	–	LIST (QUERIES, PROCS, FORMS, ANALYTICS, QMF, TABLES ou ALL)	Type
DXY n PLST_ <i>ln</i>	02 - 04	LIST (QUERIES, PROCS, FORMS, ANALYTICS, QMF, TABLES ou ALL)	Propriétaire

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PLST_ <i>ln</i>	05 - 07	LIST (QUERIES, PROCS, FORMS, ANALYTICS, QMF, TABLES ou ALL)	Nom
DXY n PLST_08	–	LIST (QUERIES, PROCS, FORMS, ANALYTICS, QMF, TABLES ou ALL)	Emplacement
DXY n PPRT_ <i>ln</i>	02 - 07	PRINT (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _01	02 - 07	PRINT QUERY (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _02	02 - 07	PRINT PROC (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _03	02 - 07	PRINT FORM (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _04	02 - 07	PRINT PROFILE (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _05	02 - 07	PRINT REPORT (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _07	02 - 07	PRINT CHART (dans TSO)	Nom
DXY n PPRT_ <i>ln</i> _08	02 - 07	PRINT TABLE (dans TSO)	Nom
DXY n PPR2_01_07	–	PRINT CHART, puis Entrée (dans TSO)	Imprimante
DXY n PPR3_01_01	–	PRINT QUERY, puis Entrée (dans TSO)	Imprimante
DXY n PPR3_01_02	–	PRINT PROC, puis Entrée (dans TSO)	Imprimante
DXY n PPR3_01_03	–	PRINT FORM, puis Entrée (dans TSO)	Imprimante
DXY n PPR3_01_04	–	PRINT PROFILE, puis Entrée (dans TSO)	Imprimante
DXY n PPR3_01_08	–	PRINT TABLE, puis Entrée (dans TSO)	Imprimante
DXY n PPR4_01_05	–	PRINT REPORT, puis Entrée (dans TSO)	Imprimante
DXY n PPR5_ <i>ln</i>	02 - 07	PRINT (dans CICS)	Nom
DXY n PPR5_ <i>ln</i> _01	02 - 07	PRINT QUERY (dans CICS)	Nom

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PPR5_02	02 - 07	PRINT PROC (dans CICS)	Nom
DXY n PPR5_03	02 - 07	PRINT FORM (dans CICS)	Nom
DXY n PPR5_04	02 - 07	PRINT PROFILE (dans CICS)	Nom
DXY n PPR5_05	02 - 07	PRINT REPORT (dans CICS)	Nom
DXY n PPR5_07	02 - 07	PRINT CHART (dans CICS)	Nom
DXY n PPR5_08	02 - 07	PRINT TABLE (dans CICS)	Nom
DXY n PPR5_08	–	PRINT (dans CICS)	Nom de file d'attente
DXY n PPR5_08_01	–	PRINT QUERY (dans CICS)	Nom de file d'attente
DXY n PPR5_08_02	–	PRINT PROC (dans CICS)	Nom de file d'attente
DXY n PPR5_08_03	–	PRINT FORM (dans CICS)	Nom de file d'attente
DXY n PPR5_08_04	–	PRINT PROFILE (dans CICS)	Nom de file d'attente
DXY n PPR5_08_05	–	PRINT REPORT (dans CICS)	Nom de file d'attente
DXY n PPR5_08_07	–	PRINT CHART (dans CICS)	Nom de file d'attente
DXY n PPR5_08_08	–	PRINT TABLE (dans CICS)	Nom de file d'attente
DXY n PPR5_09	–	PRINT (dans CICS)	Type de file d'attente
DXY n PPR5_09_01	–	PRINT QUERY (dans CICS)	Type de file d'attente
DXY n PPR5_09_02	–	PRINT PROC (dans CICS)	Type de file d'attente
DXY n PPR5_09_03	–	PRINT FORM (dans CICS)	Type de file d'attente
DXY n PPR5_09_04	–	PRINT PROFILE (dans CICS)	Type de file d'attente
DXY n PPR5_09_05	–	PRINT REPORT (dans CICS)	Type de file d'attente
DXY n PPR5_09_07	–	PRINT CHART (dans CICS)	Type de file d'attente

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PPR5_09_08	–	PRINT TABLE (dans CICS)	Type de file d'attente
DXY n PPR5_10	–	PRINT (dans CICS)	Attente
DXY n PPR5_10_01	–	PRINT QUERY (dans CICS)	Attente
DXY n PPR5_10_02	–	PRINT PROC (dans CICS)	Attente
DXY n PPR5_10_03	–	PRINT FORM (dans CICS)	Attente
DXY n PPR5_10_04	–	PRINT PROFILE (dans CICS)	Attente
DXY n PPR5_10_05	–	PRINT REPORT (dans CICS)	Attente
DXY n PPR5_10_07	–	PRINT CHART (dans CICS)	Attente
DXY n PPR5_10_08	–	PRINT TABLE (dans CICS)	Attente
DXY n PRST_01	–	RESET	Type
DXY n PRSG_01	–	RESET GLOBAL	Entrez TOUS ...
DXY n PRSG_ <i>ln</i>	02 - 11	RESET GLOBAL	Nom de la variable globale
DXY n PRUN_ <i>ln</i>	02 - 07	RUN	Nom
DXY n PRUN_ <i>ln</i> _01	02 - 07	RUN QUERY	Nom
DXY n PRUN_ <i>ln</i> _02	02 - 07	RUN PROC	Nom
DXY n PRU3_ <i>ln</i>	01 - 06	RUN QUERY, puis Entrée	Formulaire
DXY n PRU4_01	–	RUN PROC, puis Entrée	Arg
DXY n PSAV_01	–	SAVE	Type
DXY n PSA2_ <i>ln</i>	01 - 06	SAVE DATA	Nom
DXY n PSA2_08	–	SAVE DATA	Commentaire
DXY n PSA3_ <i>ln</i> _01	01 - 06	SAVE QUERY	Nom
DXY n PSA3_ <i>ln</i> _02	01 - 06	SAVE PROC	Nom
DXY n PSA3_09_01	–	SAVE QUERY	Commentaire
DXY n PSA3_09_02	–	SAVE PROC	Commentaire
DXY n PSA4_ <i>ln</i>	01 - 06	SAVE FORM	Nom
DXY n PSA4_09	–	SAVE FORM	Commentaire
DXY n PSET_01	–	SET	Type

Tableau 60. Mappage entre les variables globales DXY et les noms de zones de panneau (suite)

Nom de la variable globale (<i>n</i> représentant un identificateur de langue nationale et <i>ln</i> , un ID associé à une ligne d'une zone multiligne)	Plage de valeurs <i>ln</i>	Commande	Nom de zone
DXY n PSGL_ <i>ln</i>	01 - 19 (nombres pairs)	SET GLOBAL	Var
DXY n PSGL_ <i>ln</i>	02 - 20 (nombres pairs)	SET GLOBAL	Valeur
DXY n PSHO_01	–	SHOW	Entrez le nom ...

Remarques

Le présent document a été élaboré pour des produits et services offerts aux Etats-Unis et peut être mis à disposition par IBM dans d'autres langues. Toutefois, il peut être nécessaire de posséder une copie du produit ou de la version du produit dans cette langue pour pouvoir y accéder.

IBM se réserve le droit de ne pas proposer les produits, services ou fonctions décrits dans le présent document dans d'autres pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. Le fait qu'IBM vous fournisse ces documents ne correspond en aucune manière à une quelconque concession de licence sur ces brevets. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Pour le Canada, veuillez adresser votre courrier à :

*IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada*

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA,

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquelles ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Ces derniers sont fournis "en l'état", sans garantie d'aucune sorte. IBM ne sera en aucun cas responsable des dommages liés à l'utilisation de ces programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit :

© (nom de votre société) (année).

Des segments de code sont dérivés des Programmes exemples d'IBM Corp. © Copyright IBM Corp. _entrez l'année ou les années_.

Documentation sur l'interface de programmation

Cette publication décrit des interfaces de programmation qui permettent au Client d'écrire des programmes pouvant utiliser les services de QMF.

Marques

IBM, le logo IBM et `ibm.com` sont des marques d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible à l'adresse <http://www.ibm.com/legal/copytrade.shtml>.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Glossaire des termes et acronymes

administrateur de base de données

Personne responsable de la conception, du développement, de la sécurité, de la maintenance et de l'utilisation d'une base de données.

administrateur de QMF

Utilisateur avec les droits d'administrateur QMF.

afficher

Table logique reposant sur des données stockées dans un ensemble de tables sous-jacentes. Les données renvoyées par une vue sont déterminées par une instruction SELECT exécutée sur les tables sous-jacentes.

alias

Nom de remplacement utilisé pour identifier une table, une vue, une base de données ou un nom. Un alias peut être utilisé dans des instructions SQL pour faire référence à une table, une vue ou une base de données contenue dans le même système ou sous-système DB2 ou dans un système ou sous-système DB2 éloigné.

annulation

Restaurer les données modifiées par une instruction SQL à l'état où elles se trouvaient au dernier point de validation. En cas d'échec dans une requête contenant plusieurs instructions et aucune instruction COMMIT, toutes les instructions, sauf celles qui affectent la session QMF (comme SET), sont annulées. En cas d'échec dans une requête contenant une ou plusieurs instructions COMMIT, toutes les mises à jour ultérieures à dernière instruction COMMIT aboutie sont annulées. En tous cas, la requête s'arrête après l'échec.

APAR (rapport officiel d'analyse de programme)

Demande de correction d'un incident dans une édition prise en charge d'un programme fourni par IBM.

APF (Authorized Program Facility, fonction de programme autorisé)

Dans un environnement z/OS, fonction

permettant l'identification de programmes autorisés à utiliser des fonctions restreintes.

API (Application Program Interface, interface de programme d'application)

Interface permettant à un programme d'application écrit en langage évolué d'utiliser certaines données ou fonctions spécifiques du système d'exploitation ou d'un autre programme.

APPC (Advanced Program-to-Program Communication, communication programme à programme avancée)

Implémentation du protocole LU 6.2 de l'architecture SNA qui permet à des systèmes interconnectés de communiquer et de partager le traitement des programmes.

application

Un ou plusieurs composants logiciels ou programmes informatiques utilisant les services QMF pour fournir des fonctionnalités en support direct d'un ou plusieurs processus métier spécifique.

argument

Valeur transmise à une fonction ou une procédure, ou renvoyée par ces dernières au moment de l'exécution.

arrêt de tâche anormal (abend)

Arrêt d'une tâche, d'un travail ou d'un sous-système en raison d'une condition d'erreur qui ne peut pas être résolue pendant l'exécution par les fonctions de récupération.

Authorized Program Analysis Report (Rapport d'analyse de programme autorisé)

Voir APAR.

Authorized Program Facility (fonction de programme autorisé)

Voir APF.

base de données

Collection de données liées ou indépendantes, stockées ensemble afin de répondre à une ou plusieurs applications.

base de données relationnelle

Base de données pouvant être considérée comme un ensemble de tables et

manipulée conformément au modèle relationnel de données. Chaque base de données inclut un ensemble de tables du catalogue système décrivant la structure logique et physique des données, un fichier de configuration contenant les valeurs de paramètres allouées à la base de données et un journal de récupération consignait les transactions en cours et les transactions archivables.

base de données répartie

Base de données que les utilisateurs voient comme une entité logique, accessible localement, mais qui est composée de bases de données à différents emplacements connectées par un réseau de communications de données.

CAF (Call Attachment Facility, fonction de connexion d'appel)

Fonction de connexion DB2 for z/OS pour les programmes d'application exécutés dans TSO ou un lot z/OS. Elle peut être utilisée à la place de l'interpréteur de commandes DSN et permet de mieux contrôler l'environnement d'exécution.

Call Attachment Facility (fonction de connexion d'appel)

Voir *CAF*.

caractère de code normal

Caractère de contrôle (X' 0F') utilisé dans les systèmes EBCDIC pour indiquer que les octets suivants représentent des caractères SBCS. Voir aussi *caractère de code spécial*.

caractère de code spécial

Caractère de contrôle (X' 0E') utilisé dans les systèmes EBCDIC pour indiquer que les octets suivants représentent des caractères DBCS, jusqu'au prochain caractère de contrôle de code normal. Voir aussi *caractère de code normal*.

catalogue

Collection de tables et de vues contenant des descriptions d'objets (tables, vues et index). Voir aussi *catalogue d'objets QMF*.

catalogue d'objets QMF

Ensemble de tables de contrôle stockant les informations sur les requêtes, procédures, formats, dossiers et objets d'analyse QMF. Ces tables de contrôle

incluent Q.OBJECT_DIRECTORY, Q.OBJECT_DATA et Q.OBJECT_REMARKS.

CCSID (Coded Character Set Identifier, ID de jeu de caractères codés)

Nombre 16 bits composé de l'identificateur d'un schéma d'encodage, d'identificateurs de page de codes ainsi que d'autres informations permettant d'identifier sans ambiguïté la représentation des caractères graphiques codés. Etant donné que QMF utilise des services d'affichage fournis par GDDM, la page de code d'application GDDM doit correspondre aux CCSID utilisés dans la base de données. Voir aussi *chaîne binaire*.

chaîne binaire

Séquence d'octets qui n'est pas associée à un jeu de caractères codés et qui n'est par conséquent jamais convertie. Par exemple, les données de type BLOB sont des chaînes binaires. Voir aussi *CCSID*.

chaîne de caractères

Séquence d'octets représentant des données binaires, des caractères mono-octet, ou une combinaison de caractères mono-octet et multioctets.

chaîne graphique

Séquence de caractères provenant d'un jeu de caractères codé sur deux octets (DBCS).

CICS (Customer Information Control System)

Logiciel sous licence IBM assurant des services de traitement de transaction en ligne et la gestion des applications de gestion.

clause Dans SQL, partie distincte d'une instruction, telle qu'une clause SELECT ou WHERE.

clé Colonne ou ensemble ordonné de colonnes identifié dans la description d'une table, d'un index ou d'une contrainte référentielle. La même colonne peut faire partie de plusieurs clés.

clé externe

Dans une base de données relationnelle, clé d'une table faisant référence à la clé primaire d'une autre table.

clé primaire

Dans une base de données relationnelle,

clé qui identifie de manière unique une ligne d'une table de base de données.

CM (mode compatibilité)

Mode d'installation de QMF Version 8.1 et QMF Version 9.1 limitant les noms d'objet et propriétaire dans le catalogue d'objets QMF à 8 et 18 caractères, respectivement. Voir aussi *NFM*.

code retour SQL

SQLSTATE ou SQLCODE indiquant si l'instruction SQL précédemment exécutée a abouti, avec un ou plusieurs avertissements ou avec une erreur.

coexistence

Etat durant lequel deux éditions de QMF existent dans le même sous-système DB2. QMF Version 12.1 peut coexister avec QMF Version 9.1 nouveau mode de fonction ou QMF Version 8.1 nouveau mode de fonction uniquement.

colonne

Composant vertical d'une table de base de données. Une colonne porte un nom et un type de données particulier (un caractère décimal ou entier, par exemple).

colonne d'identité

Colonne permettant au gestionnaire de base de données DB2 de générer automatiquement une valeur numérique pour chaque ligne insérée dans une table. Les colonnes d'identité sont définies au moyen de la clause AS IDENTITY. Une table ne peut comporter qu'une seule colonne d'identité.

communication évoluée de programme à programme

Voir *APPC*.

concaténation

Jointure de deux caractères ou chaîne afin de former une seule chaîne.

condition de recherche

Critère de sélection des lignes d'une table. Une condition de recherche est constituée d'un ou plusieurs prédicats.

connexion

En matière de communication de données, association établie entre des entités pour la transmission de l'information. Voir aussi *connexion SQL*. La connectivité avec des serveurs distants n'est pas prise en charge lors de

l'exécution de QMF for TSO en procédure mémorisée DB2 for z/OS.

connexion SQL

Association entre un processus applicatif et un serveur d'applications local ou éloigné ou un serveur de bases de données. Voir aussi *unité d'oeuvre éloignée*, *unité d'oeuvre répartie*.

constante

Élément de langage indiquant une valeur constante. Les constantes sont classées en constantes de type chaîne ou numérique.

contrainte

Règle limitant les valeurs susceptibles d'être insérées, supprimées ou mises à jour dans une table.

contrainte de vérification

Contrainte définie par l'utilisateur qui spécifie les valeurs pouvant se trouver dans les colonnes spécifiques d'une table de base. Voir aussi *contrainte*.

contrainte référentielle

Règle selon laquelle les valeurs non NULL d'une clé externe spécifique sont valides uniquement si elles apparaissent également comme valeurs de la clé primaire de la table parente. La contrainte référentielle est également définie en fonction du fichier dépendant.

correctif

Voir *PTF*.

CSECT (section de contrôle)

Partie d'un programme dont le programmeur indique qu'il s'agit d'une unité pouvant être allouée à nouveau, dont tous les éléments doivent être chargés dans des emplacements du stockage principal adjacents.

curseur

Structure de contrôle nommée utilisée par un programme d'application pour indiquer et sélectionner une ligne de données dans un ensemble de lignes.

Customer Information Control System

Voir *CICS*.

DBCS (jeu de caractères codé sur deux octets)

Jeu de caractères dans lequel chaque caractère est représenté par deux octets. Ces jeux de caractères sont souvent utilisés par exemple par le japonais ou le chinois, qui comprennent des symboles ne

pouvant pas être représentés par un seul octet. Voir aussi *SBCS*.

DBMS (Database Management System)

Système logiciel servant à contrôler la création, l'organisation et la modification d'une base de données et l'accès aux données qu'elle contient.

DCT (Destination Control Table, table de contrôle de destination)

Table décrivant chacune des destinations de données transitoires utilisées dans CICS. Cette table contient une entrée pour chaque destination indirecte, extrapartition et intrapartition.

déclencheur

Objet de base de données associé à une seule table de base ou vue et qui définit une règle. La règle est constituée d'un ensemble d'instructions SQL qui s'exécute lorsqu'une opération d'insertion, de mise à jour ou de suppression se produit sur la table de base ou la vue associée.

définition des ressources en ligne

Voir *RDO*.

demandeur

Voir *demandeur d'application*.

demandeur d'application

Source d'une demande d'un SGBD relationnel activé DRDA. Seules les bases de données DB2 for z/OS peuvent fonctionner en demandeurs d'application car c'est le seul type de base de données dans lequel QMF peut démarrer.

distant

Appartenant aux bases de données, objets ou applications installé(s) ou stocké(s) ailleurs que dans le système dans lequel QMF est actuellement exécuté. Vous pouvez accéder aux objets (dont les requêtes, formats, procédures, dossiers et objets d'analyse QMF) sur un serveur distant en utilisant la commande QMF CONNECT. Vous pouvez également utiliser une commande QMF avec un nom de vue ou une table tripartite si vous souhaitez simplement accéder aux tables ou vues à un emplacement distant. L'accès distant n'est pas permis lorsque QMF for TSO est exécuté en procédure mémorisée DB2 for z/OS.

Distributed Relational Database Architecture

Voir *DRDA*.

données binaires

Données de type CHAR or VARCHAR qui ne sont pas associées à un jeu de caractères codés et qui ne sont par conséquent jamais converties.

données réparties

Données stockées sur plusieurs, disponibles aux programmes d'application et utilisateurs distants.

Double-Byte Character Set (Jeu de caractères codé sur deux octets)

Voir *DBCS*.

DRDA (Distributed Relational Database Architecture)

Architecture qui définit des formats et des protocoles permettant un accès transparent aux données éloignées. DRDA définit deux types de fonction : demandeur d'applications et serveur d'applications.

droit d'administrateur QMF

Droit permettant à un utilisateur d'insérer ou supprimer les lignes dans la table de contrôle Q.PROFILES. Les utilisateurs avec ce droit peuvent effectuer les commandes suivantes sur les requêtes, formulaires et procédures QMF appartenant aux autres utilisateurs sans forcer les propriétaires de partager ces objets avec tous les utilisateurs : SAVE, ERASE, IMPORT, EXPORT et DISPLAY. QMF vérifie les droits administrateurs de chaque ID utilisateur lors de l'initialisation ; vous pouvez désactiver cette vérification en paramétrant la variable DSQEC_DISABLEADM dans la routine d'exit DSQUOPTS ou dans un autre programme de votre choix.

éditeur de liens

Programme informatique qui permet de créer des modules de chargement à partir d'un ou de plusieurs modules objets ou modules de chargement en convertissant les références croisées entre les modules et, le cas échéant, en corrigeant les adresses.

édition de liens

Opération permettant de créer un programme à l'aide d'un éditeur de liens.

emplacement

Serveur de base de données relationnelle spécifique dans un système de base de

données relationnelle répartie. Chaque emplacement possède un nom d'emplacement unique.

emplacement en cours

Serveur d'application auquel la session QMF est actuellement connectée. Après l'établissement de la connexion, ce serveur traite toutes les instructions SQL. Lors de l'initialisation de QMF, l'emplacement en cours peut être indiqué à l'aide des paramètres de démarrage DSQSDBNM. La connectivité avec des serveurs distants n'est pas prise en charge lors de l'exécution de QMF for TSO en procédure mémorisée DB2 for z/OS.

enregistrement

Élément de stockage qui correspond à une ligne ou à d'autres données.

environnement

1. Collection nommée de ressources logiques et physiques, utilisée pour prendre en charge l'exécution d'une fonction.

espace adresse (address space)

Plage d'adresses accessible à un programme informatique ou un processus. L'espace adresse peut désigner une mémoire physique, une mémoire virtuelle ou les deux.

espace table

Unité logique de stockage dans une base de données. Dans DB2 for z/OS, l'espace table est un ensemble de pages pouvant contenir une ou plusieurs tables. Dans DB2 for Linux, UNIX and Windows, un espace de table est un ensemble de conteneurs, et les données, index, zone longue et parties LOB d'une table peuvent être stockés dans le même espace table ou dans des espaces tables distincts.

espace table segmenté

Espace table divisé en groupes de pages de taille identique, appelés segments. Les segments sont affectés aux tables afin que les lignes de tables distinctes ne soient jamais stockées dans le même segment. Voir aussi *espace table*.

extraction

Processus d'extraction des lignes d'une base de données ou d'un fichier pour créer un objet DATA QMF. QMF prend en

charge l'extraction multiligne grâce au paramètre DSQSMRFI.

file d'attente de données transitoires

Zone de stockage CICS dans laquelle les objets sont stockés pour un traitement externe ou interne. Voir aussi *file d'attente de mémoire temporaire*.

file d'attente de mémoire temporaire

Dans CICS, file d'attente d'éléments de données pouvant être lue et relue, dans n'importe quel ordre. La file d'attente est créée par une tâche et elle est conservée jusqu'à cette même tâche ou une autre tâche la supprime. Voir aussi *file d'attente de données transitoires*.

fonction d'agrégation

Groupe de fonctions récapitulant les données d'une colonne. Demandés avec ces codes usage dans les panneaux de formulaires : AVERAGE, CALC, COUNT, FIRST, LAST, MAXIMUM, MINIMUM, STDEV, SUM, CSUM, PCT, CPCT, TPCT, TCPCT.

fonction de colonne

Voir *fonction d'agrégation*.

fonction définie par l'utilisateur

Voir *UDF*.

fonction intégrée

Fonction hautes performances intégrée à la base de données DB2. Une fonction intégrée peut être référencée dans des instructions SQL, chaque fois qu'une expression est valide.

fonction scalaire

Fonction SQL qui accepte des arguments en option et ne renvoie qu'une seule valeur scalaire par appel. Une fonction scalaire peut être référencée dans une instruction SQL, chaque fois qu'une expression est valide.

fonction SQL

Fonction totalement implémentée à l'aide d'un sous-ensemble d'instructions SQL et SQL PL.

format par défaut

Format QMF créé lorsqu'aucun format sauvegardé n'est spécifié dans la commande RUN QUERY.

GDDM (Graphical Data Display Manager)

Logiciel graphique définissant et affichant

le texte et les graphiques pour la sortie sur une imprimante ou unité d'affichage.

gestionnaire d'affichage de données graphiques
Voir *GDDM*.

gestionnaire de base de données
Programme permettant de gérer les données à l'aide de divers services : contrôle centralisé, indépendance des données et structures physiques complexes. Il optimise l'accès aux données, et permet d'assurer l'intégrité et la récupération de ces dernières, le contrôle des accès concurrents, la confidentialité et la sécurité.

hôte Système de contrôle ou le plus évolué dans une configuration de communications de données.

HTML (Hypertext Markup Language)
Langage de balisage conforme à la norme SGML et principalement conçu pour prendre en charge l'affichage en ligne d'informations textuelles et graphiques, comportant des liens hypertexte.

Hypertext Markup Language (Langage de marquage hypertexte)
Voir *HTML*.

ICU (Interactive Chart Utility)
Composant géré par le menu d'un produit GDDM d'IBM qui permet aux non programmeurs d'afficher, imprimer ou tracer des graphiques et diagrammes.

ID autorisation principale
Identificateur d'autorisation permettant d'identifier un processus d'application sur DB2 pour z/OS.

ID autorisation secondaire
Dans DB2 for z/OS, identificateur d'autorisation associé à un ID autorisation principal par une routine d'exit d'autorisation. Voir aussi *ID utilisateur principal*.

ID autorisation SQL
Voir *SQLID*.

identificateur autorisation (ID autorisation)
Chaîne de caractères qui désigne un ensemble de privilèges et permettant de vérifier les droits. Un identificateur autorisation peut représenter un objet, un utilisateur individuel, un groupe organisationnel, une fonction ou un rôle de base de données. QMF authentifie l'ID

autorisation de la base de données ou, éventuellement, l'ID de connexion TSO QMF TSO, par rapport à la colonne CREATOR de la table Q.PROFILES lors de l'initialisation de QMF.

identificateur de jeu de caractères codés
Voir *CCSID*.

ID SQL (ID autorisation SQL)
Dans DB2 for z/OS, ID autorisation utilisé dans certaines conditions pour la vérification des instructions SQL dynamiques.

index Ensemble de pointeurs ordonnés logiquement en fonction de la valeur de leur clé. Les index accélèrent l'accès aux données et permettent de garantir l'unicité des lignes d'une table.

Integrated Exchange Format
Voir *IXF*.

Interactive Chart Utility
Voir *ICU*.

Interactive System Productivity Facility (ISPF)
Voir *ISPF*.

interface appelable
Interface de programmation fournissant l'accès aux services et objets QMF.

interface de commande
Interface permettant de lancer les commandes QMF. L'interface de commande vous permet d'émettre des commandes QMF à partir d'une boîte de dialogue ISPF exécutée sous QMF. Avec cette interface, QMF communique avec la boîte de dialogue via le pool de variables ISPF.

interface de procédure mémorisée
Une interface dans QMF for TSO qui vous permet de démarrer QMF en procédure mémorisée DB2 for z/OS, transmettre le nom d'une requête QMF ou procédure effectuant le travail requis et recevoir jusqu'à 21 ensembles de résultats, dont un ensemble de résultats pour la sortie de trace. QMF for TSO peut être démarré de cette manière à partir de tout produit pouvant exécuter une procédure mémorisée DB2 for z/OS.

interface de programme d'application
Voir *API*.

ISPF (Interactive System Productivity Facility)

Logiciel sous licence IBM servant d'éditeur plein écran et de gestionnaire de boîte de dialogue. Utilisé pour écrire des programmes d'application, il fournit un moyen de générer des panneaux d'affichage standard et des boîtes de dialogue interactives entre le programmeur d'application et l'utilisateur du terminal.

IVP (Installation Verification Procedure, procédure de vérification d'installation)

Procédure ou programme dont l'objectif est de vérifier qu'un produit a été correctement installé.

IXF (Integrated Exchange Format)

Protocole permettant de transférer les données tabulaires parmi les différents produits logiciels.

JCL (langage de contrôle des travaux)

Langage de contrôle qui identifie un travail à un système d'exploitation et décrit les exigences du travail.

jeu de caractères à simple octet

Voir *SBCS*.

jointure

Opération relationnelle SQL permettant l'extraction de données à partir de plusieurs tables en fonction de valeurs concordantes dans leurs colonnes.

jointure externe

Résultat d'une opération de jointure qui comporte les lignes correspondantes des deux tables jointes et conserve en l'état une partie ou l'ensemble des lignes sans correspondance des tables. Voir aussi *jointure interne*.

jointure interne

Résultat d'une opération de jointure qui comporte uniquement les lignes correspondantes des deux tables jointes. Voir aussi *jointure externe*.

journal

Collection d'enregistrements décrivant de manière séquentielle les événements se produisant sur un système.

langage de contrôle des travaux

Voir *JCL*.

langage XML

Voir *XML*.

liaison

Processus par lequel les données de sortie générées par le précompilateur du SGBD sont converties en une structure de contrôle utilisable, telle qu'un plan d'accès, un plan d'application ou un package.

ligne

Composant horizontal d'une table, composé d'une suite de valeurs (une pour chaque colonne de la table).

ligne dépendante

Ligne contenant une clé associée correspondant à la valeur d'une clé parente de la ligne parente. La valeur de la clé associée représente donc une référence de la ligne dépendante à la ligne parente.

littéral

Chaîne de caractères dont la valeur est définie par les caractères eux-mêmes. Par exemple, la constante numérique 7 possède la valeur 7 et la constante de type caractère "CHARACTERS" possède la valeur CHARACTERS.

LOB (Large Object)

Séquence d'octets avec une taille de 0 octet à 2 gigaoctets (moins 1 octet). Il existe trois types de données : Binary Large Object (BLOB), Character Large Object (CLOB, qui peut inclure des caractères mono-octets uniquement ou un mélange de caractères mono-octets et double-octets), et Double-Byte Character Large Object (DBCLOB). QMF prend en charge une taille de colonne LOB jusqu'à 32 Ko.

local

Appartenant aux bases de données, objets ou applications installé(s) ou stocké(s) dans le système dans lequel QMF est actuellement exécuté.

longueur d'enregistrement

Longueur de stockage qui correspond à une ligne ou à d'autres données.

LUW Abréviation pour Linux, UNIX et Windows.

mode compatibilité

Voir *CM*.

module

Objet de base de données de type structure de contrôle, produit lors de la préparation d'un programme, et qui peut

contenir à la fois des formes exécutables d'instructions SQL statiques, ou d'expressions Xquery, et des espaces réservés pour des formes exécutables d'instructions SQL dynamiques.

module de chargement

Programme pouvant être chargé dans la mémoire principale en vue d'être exécuté.

mot clé

Un des mots prédéfinis d'un langage de programmation ou artificiel, d'une application ou d'une commande.

National Language Feature, langue

Voir *NLF, langue*.

NFM (nouveau mode de fonction)

Mode d'installation de QMF Version 8.1 et QMF Version 9.1 qui autorisait les noms d'objet et propriétaires du catalogue d'objets QMF à avoir la longueur maximale autorisée par la base de données. QMF Version 12.1 autorise les noms d'objets et propriétaires à être aussi longs que ne l'autorise la base de données. Voir aussi *CM*.

NLF (National Language Feature, langue)

Fonction facultative disponible avec QMF. La fonction NLF de langue permet aux utilisateurs d'interagir avec QMF dans leur propre langue.

nombre en virgule flottante à simple précision

Représentation approximative d'un nombre réel sur 32 bits.

nombre en virgule flottante en double précision

Représentation approximative d'un nombre réel sur 64 bits.

nom de corrélation

Identificateur défini et utilisé dans une instruction SQL en tant que nom d'exposition pour des objets tels qu'une table, une vue, une référence à une fonction de table, une expression de table imbriquée ou une référence à une table de modification de données. Les noms de corrélations sont utiles dans une instruction SQL pour permettre deux références distinctes à la même table de base et d'utiliser un autre nom pour représenter un objet.

nom d'emplacement

Nom unique d'un serveur de base de données. Les applications utilisent le nom

d'emplacement pour accéder à un serveur de base de données DB2.

nom tripartite

Nom complet d'une table, d'une vue ou d'un alias qui est composé d'un nom d'emplacement, d'un ID autorisation et d'un nom d'objet, séparés par un point. Les commandes QMF incluant des noms tripartites peuvent être lancées uniquement à partir de bases de données DB2 for z/OS et elles peuvent être transmises à toutes les bases de données à l'exception de DB2 for VM ou VSE. Lorsque QMF for TSO a été démarré en procédure mémorisée DB2 for z/OS, les commandes QMF avec des noms tripartites ne sont pas prises en charge.

nouveau mode de fonction

Voir *NFM*.

objet Espace de stockage nommé composé d'un ensemble de caractéristiques décrivant l'espace et, dans certains cas, les données. Un objet est un élément occupant l'espace d'une mémoire, pouvant se trouver dans une bibliothèque ou un répertoire, pouvant être sécurisé et sur lequel des opérations définies peuvent être réalisées. Voir aussi *objet QMF*.

objet actuel

Objet QMF conservé dans le stockage temporaire de sorte qu'avec chaque utilisation, il peut être facilement accessible sans extraction de base de données. Il existe sept zones de stockage temporaire : QUERY, FORM, PROC, PROFILE, REPORT, DATA et CHART. Les utilisateurs peuvent accéder à toutes les zones sauf la zone DATA avec les commandes SHOW et DISPLAY. Voir aussi *stockage temporaire*.

objet LOB

Voir *LOB*.

objet QMF

Objet utilisé par les utilisateurs QMF pour interroger, formater et présenter les données ou gérer l'interaction entre QMF et la base de données. Les objets QMF comprennent les requêtes et données de résultats de requêtes, formats, procédures, rapports, graphiques et le profil QMF. Chaque objet QMF possède une zone de stockage temporaire utilisée pour afficher l'objet. Tous les objets à l'exception des

rapports et graphiques peuvent être sauvegardés dans la base de données ; les rapports et graphiques sont créés de manière dynamique en fonction de la requête utilisateur en appliquant les spécifications de formatage d'un format QMF particulier sur les données de résultats renvoyées de la base de données. Voir aussi *stockage temporaire*.

opérateur de comparaison

Dans SQL, symbole utilisé dans les expressions de comparaison pour indiquer une relation entre deux valeurs. Les opérateurs de comparaison sont les suivants = (égal à), <> (différent de), < (inférieur à), > (supérieur à), <= (inférieur ou égal à) et >= (supérieur ou égal à).

option d'exploitation en temps partagé

Voir *TSO*.

page de codes

Attribution particulière de points de code à des caractères graphiques. Dans une page de codes donnée, un point de code peut avoir une seule signification. Une page de codes identifie comment les points de code non définis sont gérés.

panneau

Affichage formaté d'informations sur un écran contenant également des zones d'entrée.

paramètre

Valeur ou référence transmise à une fonction, une commande ou un programme qui sert d'entrée ou qui contrôle des actions. La valeur est fournie par un utilisateur ou un autre programme ou processus.

paramètre à mot clé

Paramètre constitué d'un mot clé suivi d'une ou plusieurs valeurs. Voir aussi *paramètre positionnel*.

paramètre positionnel

Paramètre devant apparaître à un emplacement spécifié, relatif à d'autres paramètres. Voir aussi *paramètre mot clé*.

partition

Partie d'un ensemble de pages. Chaque partition correspond à un fichier distinct, dont la taille peut être augmentée de façon indépendante. Une partition peut atteindre une taille de 1, 2 ou 4 gigaoctets, selon le nombre de partitions contenues

dans l'ensemble de pages partitionné. Toutes les partitions d'un même ensemble de pages ont la même taille.

plan Voir *plan d'application*.

plan d'application

Structure de contrôle générée lors du processus de liaison. Le nom par défaut du plan d'application QMF Version 12.1 est QMF12.

pool de mémoire tampon

Zone de mémoire dans laquelle les pages de données sont lues, modifiées et mises en suspens lors du traitement. Voir aussi *espace adresse*.

précision

Attribut d'un nombre décrivant le nombre total de chiffres significatifs.

prédicat

Elément d'une condition de recherche qui suppose une opération de comparaison, de manière expresse ou implicite.

privilege (droit)

Dans SQL, capacité attribuée à un utilisateur via l'instruction d'octroi GRANT.

procédure

Ensemble d'instructions ou de commandes séquentiel utilisé pour effectuer une ou plusieurs tâches. Voir aussi *procédure linéaire* et *procédure avec logique*.

procédure de vérification d'installation

Voir *IVP*.

procédure linéaire

Ensemble séquentiel de commandes ou synonymes de commande QMF pouvant être utilisées pour effectuer plusieurs opérations en même temps. Voir aussi *procédure avec logique*.

procédure navigationnelle

Ensemble d'instructions effectuant une ou plusieurs tâches. Une procédure avec logique commence par un commentaire REXX et autorise une logique conditionnelle (utilisant REXX), des calculs, des chaînes de construction et des commandes CICS ou TSO. Voir aussi *procédure linéaire*.

procédure stockée

Routine pouvant être appelée à l'aide de

l'instruction SQL CALL pour effectuer des opérations pouvant inclure à la fois des instructions en langage hôte et des instructions SQL.

produit de base

Version anglaise de QMF, établie lors de l'installation de QMF. Un autre environnement linguistique est établi après l'installation en installant la fonction de langue associée à la langue.

profil Objet contenant des informations sur les caractéristiques de la session de l'utilisateur.

PTF (Program Temporary Fix)

Pour les produits System i, System p, et System z, correctif testé par IBM et rendu disponible à tous les clients.

QBE (Query-by-Example)

Composant de QMF permettant aux utilisateurs de créer des requêtes graphiques.

qualificatif

Lorsqu'on se rapporte à un objet QMF, partie du nom identifiant le propriétaire ou l'emplacement d'un objet. Si on se rapporte à un fichier TSO, partie du nom séparé du reste du nom par des points. Par exemple, 'TCK', 'XYZ' et 'QUERY' sont des qualificatifs du nom de fichier 'TCK.XYZ.QUERY'.

Query-by-Example

Voir *QBE*.

RCT (Resource Control Table, table de contrôle de ressource)

Table de contrôle DB2 définissant la relation entre les transactions CICS et les ressources DB2.

RDBMS (Relational Database Management System, système de gestion de base de données relationnelle)

Ensemble d'équipements matériels et de logiciels qui organisent et permettent l'accès à une base de données relationnelle.

RDO (Resource Definition Online)

Dans CICS, fonction permettant à l'utilisateur de définir certaines ressources CICS de manière interactive lors de l'exécution de CICS. Spécifiquement, RDO

permet à l'utilisateur de définir des terminaux, programmes et transactions de manière interactive.

réentrant

Code exécutable pouvant résider dans le stockage en copie partagée pour toutes les unités d'exécution de la base de données. Le code réentrant ne peut pas s'auto modifier ; il fournit des zones de stockage distinctes pour chaque unité d'exécution.

référence corrélée

Référence à une colonne d'une table se situant en dehors de la sous-requête.

requête

Demande d'extraction d'informations d'une base de données en fonction de conditions précises. Il peut s'agir par exemple d'une interrogation lancée sur une table pour répertorier tous les clients d'une banque dont le solde créditeur est supérieur à 1000 euros. Dans QMF, une requête se rapporte également aux instructions SQL soumises à partir du panneau de requête SQL, QBE ou création assistée de requêtes, même si ces instructions ne sont pas des requêtes pour informations (instructions SELECT).

requête assistée

Requête gérée par le menu contrôlée par les paramètres fournis par l'utilisateur.

ressource

Objet auquel s'applique un verrou ou une réclamation. Il peut s'agir d'un espace table, d'un espace index, d'une partition de données, d'une partition d'index ou d'une partition logique.

Restructured Extended Executor

Voir *REXX*.

retour à la ligne dans une colonne

Formatage de valeur dans un rapport quand les valeurs occupent plusieurs lignes d'une colonne. Le retour à la ligne dans une colonne est souvent utilisé lorsqu'une colonne contient des valeurs dont la longueur dépasse la largeur de la colonne, par exemple lors de l'affichage de données XML.

rétromigration

Processus de retour à l'édition précédente d'un programme logiciel après une migration ou une tentative de migration vers une édition en cours.

REXX (Restructured Extended Executor)

Langage de programmation haut niveau d'objectif général, particulièrement adapté aux programmes ou procédures EXEC pour les calculs personnels.

routine

Programme ou séquence d'instructions appelé(e) par un programme. La plupart du temps, une routine a un objectif général et elle est fréquemment utilisée.

routine d'exit

Programme recevant le contrôle d'un autre programme afin d'exécuter des fonctions spécifiques.

SBCS (Single-Byte Character Set)

Jeu de caractères codés dans lequel chaque caractère est représenté sur un octet. Un point de code sur un octet permet de représenter jusqu'à 256 caractères. Voir aussi *jeu de caractères à deux octets*.

section de contrôle

Voir *CSECT*.

serveur

Voir *serveur d'applications*.

serveur d'application

Cible d'une requête provenant d'un demandeur d'application. Le système de gestion de base de données sur le site du serveur d'applications traite la requête demandée. La connectivité avec des serveurs distants n'est pas prise en charge lors de l'exécution de QMF for TSO en procédure mémorisée DB2 for z/OS.

serveur de base de données

Programme logiciel qui utilise un gestionnaire de bases de données pour proposer des services de base de données à d'autres programmes logiciels ou ordinateurs.

session

Toutes les interactions entre l'utilisateur et QMF à partir du moment où l'utilisateur appelle QMF jusqu'au lancement de la commande EXIT.

sous-chaîne

Partie d'une chaîne de caractères.

sous-requête

Requête SQL complète apparaissant dans une clause WHERE ou HAVING d'une autre requête.

sous-système

Dans DB2 pour z/OS, instance distincte d'un système de gestion de base de données relationnelle (SGDB relationnel).

SQLCA (Structured Query Language Communication Area)

Ensemble de variables fournissant un programme d'application avec des informations sur l'exécution de ses requêtes ou instructions SQL du gestionnaire de la base de données. Lorsqu'une erreur est associée à un code SQL, l'aide message QMF (disponible en appuyant sur la touche Aide) affiche le contenu du SQLCA.

SQL (Structured Query Language)

Langage normalisé, conçu pour définir et manipuler des données enregistrées dans une base de données relationnelle.

stockage temporaire

Zone utilisée pour stocker temporairement un objet QMF pendant que l'utilisateur l'utilise de sorte que, avec chaque utilisation, elle puisse être accessible dans autre extraction de base de données. Il existe sept zones de stockage temporaire : QUERY, DATA, FORM, PROC, REPORT, CHART ou PROFILE. A l'exception des données de résultat de requête (objet DATA), les objets QMF de ces zones peuvent être affichés à l'aide de la commande SHOW suivis par le nom de la zone de stockage. Bien que le contenu de la zone DATA ne puisse pas être directement affiché, les utilisateurs peuvent émettre les commandes SHOW REPORT ou SHOW CHART pour voir les données de résultats de requête formatées avec les spécifications du format en cours de la zone FORM. Voir également *objet QMF, objet en cours*.

Structured Query Language Communication Area Voir *SQLCA*.**Structured Query Language (SQL)**

Voir *SQL*.

suppression en cascade

Processus permettant au gestionnaire de base de données DB2 d'appliquer des contraintes référentielles en supprimant toutes les lignes enfant d'une ligne parente supprimée.

synonyme de commande

Partie instruction ou instruction/objet d'une commande définie par un site. Après la définition et l'activation des synonymes de commande dans le profil QMF, les utilisateurs peuvent entrer les synonymes dans la ligne de commande QMF comme ils le font pour les commandes QMF standard.

syntaxe étendu

Syntaxe utilisé pour les commandes QMF SET GLOBAL et GET GLOBAL et certains appels de fonction dans une application d'interface callable. La syntaxe étendue définit les paramètres utilisés par les applications d'interface callable QMF écrites dans Assembler, C, COBOL, Fortran ou PL/I.

syntaxe linéaire

Syntaxe de commande QMF entrée dans une instruction d'un programme ou d'une procédure, ou pouvant être entrée sur la ligne de commande QMF.

système de gestion de base de données

Voir *DBMS*.

système de gestion de base de données relationnelle

Voir *RDBMS*.

table Dans une base de données relationnelle, objet de base de données composé d'un nombre spécifique de colonnes et permettant de stocker un ensemble de lignes non classées. Voir aussi *table de base*.

table auxiliaire

Table qui stocke des colonnes hors de la table qui les définit. Voir aussi *table de base*.

table de base

Table générée par l'instruction SQL CREATE TABLE et contenant des données rémanentes.

table de contrôle de ressources

Voir *RCT*.

table de contrôle des destinations

Voir *DCT*.

table dépendante

Table dépendant d'un objet. Par exemple, une table avec une clé externe dépend de la table contenant la clé primaire correspondante.

table de synonymes de commande

Table qui stocke une commande définie par le site dans chaque ligne. Vous affectez un ensemble de synonymes de commande à un utilisateur en stockant le nom de cette table dans le profil de l'utilisateur.

tables de contrôle

Ensemble de tables que QMF utilise pour stocker les informations sur les objets QMF et gérer les opérations QMF. Voir aussi *catalogue d'objets QMF*.

texte de bloc de détail

Texte du corps d'un rapport associé à une ligne particulière de données.

texte d'en-tête de détail

Texte dans l'en-tête d'un rapport.

trace Enregistrement du traitement d'un programme ou d'une transaction. Les informations recueillies à partir d'une trace peuvent servir à cerner tant les problèmes que les performances.

transaction

Unité de traitement constituée d'un ou plusieurs programmes d'application, affectant un ou plusieurs objets et lancée par une seule demande.

TSO (option de partage du temps)

Élément de base du système d'exploitation z/OS permettant aux utilisateurs de travailler de manière interactive avec le système.

type de données

Classification identifiant un des types de données. Dans SQL, le type de données est un attribut de colonnes, littéraux, variables d'hôte, registres spéciaux, paramètres et le résultat de fonctions et expressions.

type distinct

Type de données défini par l'utilisateur et qui partage une représentation commune avec des types de données intégrés.

UDF (User-Defined Function, fonction définie par l'utilisateur)

Fonction définie pour le système de base de données DB2 à l'aide de l'instruction CREATE FUNCTION et pouvant donc être référencée dans les instructions SQL. Une fonction utilisateur peut être une fonction externe ou une fonction SQL.

unicode

Codage de caractères standard prenant en charge l'échange, le traitement et l'affichage de texte rédigé dans des langages communs ainsi que certains textes graphiques et historiques. Le standard Unicode possède un jeu de caractères 16 bits défini par l'ISO 10646.

unité de récupération (UR)

Séquence d'opérations au sein d'une unité d'oeuvre entre les points de cohérence.

unité d'exécution

Structure DB2 qui décrit la connexion d'une application, analyse sa progression, exécute les fonctions liées aux ressources et délimite ses possibilités d'accès aux ressources et services DB2. La plupart des fonctions DB2 s'exécutent dans le cadre d'une unité d'exécution.

unité d'oeuvre éloignée

Forme de traitement de base de données relationnelle répartie dans laquelle un programme d'application comme QMF peut accéder aux données sur une base de données éloignée au sein d'une unité d'oeuvre. La connexion est établie par la commande QMF CONNECT. La commande CONNECT ne peut pas être utilisée lorsque QMF for TSO est exécuté en procédure mémorisée DB2 for z/OS.

unité d'oeuvre répartie

L'unité d'oeuvre éloignée est une forme de traitement de base de données relationnelle répartie dans laquelle un programme d'application peut accéder aux données sur une base de données éloignée au sein d'une unité d'oeuvre. Au sein d'une unité d'oeuvre, une application, comme QMF, exécutée sur un système peut transmettre les requêtes SQL à plusieurs systèmes de gestion de bases de données éloignées à l'aide du SQL pris en charge par ces systèmes. La requête s'effectue via une commande QMF incluant un nom de vue ou une table à trois parties. Les commandes QMF avec des noms tripartites ne peuvent pas être transmises aux bases de données DB2 for VM or VSE ou utilisées lorsque QMF for TSO a été démarré en procédure mémorisée DB2 for z/OS. Les noms tripartites des commandes QMF ne

peuvent pas se rapporter non plus à une table contenant des types de données LOB.

unité d'oeuvre (UOW)

Série d'opérations pouvant faire l'objet d'une récupération dans un processus applicatif. A un instant donné, un processus applicatif constitue une unité d'oeuvre unique, mais la durée de vie d'un processus peut impliquer la création de nombreuses unités d'oeuvre résultant d'opérations de validation ou d'annulation. Dans le cadre d'une opération de mise à jour multisite, une même unité d'oeuvre peut comporter plusieurs unités de récupération. Dans les requêtes SQL QMF comprenant plusieurs instructions et aucune instruction COMMIT, toutes les instructions comprennent une seule unité d'oeuvre, de sorte que toutes les instructions à l'exception de celles qui affectent la session (comme SET) sont annulées en cas d'échec. Dans les requêtes SQL QMF incluant plusieurs instructions et une ou plusieurs instructions COMMIT, une unité d'oeuvre est composée d'une instruction COMMIT et de toutes les instructions précédentes du début de la requête ou de la dernière instruction COMMIT. En cas d'échec, toutes les mises à jour ultérieures à dernière instruction COMMIT aboutie sont annulées.

valeur par défaut

Valeur prédéterminée, attribut ou option utilisée lorsqu'aucune valeur n'est spécifiée. Une valeur par défaut peut être définie pour les données de colonne dans les tables DB2 en spécifiant le mot clé DEFAULT dans une instruction SQL modifiant les données (comme INSERT, UPDATE et MERGE).

validation

Opération qui termine une unité d'oeuvre en libérant les verrous de sorte que les modifications apportées à la base de données par cette unité d'oeuvre puissent être perçues par d'autres processus. Les modifications ainsi validées deviennent permanentes.

validation en deux phases

Processus en deux étapes par l'intermédiaire duquel les ressources récupérables d'un sous-système externe

sont validées. Lors de la première phase, les sous-systèmes du gestionnaire de bases de données sont interrogés, ce qui permet de vérifier qu'ils sont prêts pour la validation. Si tous les sous-systèmes renvoient une réponse positive, le gestionnaire de bases de données leur demande d'exécuter la validation.

variable d'agrégation

Fonction d'agrégation placée dans un rapport avec les panneaux FORM.BREAK, FORM.CALC, FORM.DETAIL ou FORM.FINAL. Sa valeur apparaît avec le bas de page de rupture, le texte de bloc de détail ou le texte final lors de la génération du rapport.

variable de substitution

(1) Variable dans une procédure ou une requête dont la valeur est spécifiée par une variable globale ou une variable d'exécution. (2) Variable dans un format QMF dont la valeur est spécifiée par une variable globale.

variable d'exécution

Variable dans une procédure ou une requête dont la valeur est spécifiée par l'utilisateur lors de l'exécution de la procédure ou requête. La valeur d'une variable d'exécution est uniquement disponible dans la procédure ou requête en cours. Voir aussi *variable globale*.

variable globale

Entité nommée dont la valeur est conservée le temps d'une session QMF par défaut. QMF utilise des variables globales pour gérer les activités de session et de base de données. Certaines variables globales peuvent être définies avec la commande SET GLOBAL, d'autres enregistrent des informations sur l'état de la session QMF en cours et ne peuvent pas être définies.

variation de détail

Définition de formatage de données spécifiée sur un panneau FORM.DETAIL à utiliser pour formater de manière conditionnelle un rapport ou une partie de rapport.

verrou Mécanisme de sérialisation d'une séquence d'événements ou de l'accès aux données.

XML (Extensible Markup Language)

Métalangage standard pour la définition des langages de marquage SGML (Standard Generalized Markup Language).

zone de mémoire auxiliaire

Zone de travail utilisée dans le traitement conversationnel en vue de conserver des informations provenant d'un programme d'application pendant toutes les exécutions de ce dernier.

z/OS

Système d'exploitation IBM grands systèmes utilisant une mémoire réelle 64 bits.

Index

A

- accès aux données distant
 - ID utilisateur pour CONNECT 216
- ADD, commande
 - confirmation de l'éditeur de table 224
- ADDRESS, commande 14, 31
- aide en ligne
 - aide des messages QMF
 - affichage des codes SQL positifs 227
- alias
 - vue qui extrait des alias pour LIST 237
- allocation d'espace principal à l'exportation 237
- allocation d'espace secondaire à l'exportation 237
- appels de fonction
 - C 168
 - DSQCIC 168
 - DSQCICE 168
- appels de programme 17
- applications
 - bilingues 41
 - commandes 1
 - INTERACT 48
 - présentation 45
 - traitement 17
 - conditions requises pour une application ISPF 33
 - contrôle 1
 - débogage 149
 - démarrage 2
 - développement 1
 - enregistrements de données 108
 - environnement CICS 4
 - méthodes d'implémentation 3
 - procédures 7
 - procédures navigationnelles 3
 - synonyme de commande 2
 - types 1
- applications traduisibles 44
- arguments 12
- Assembler
 - appels de fonction 154
 - CICS 153
 - exemple de programme 156
 - z/OS 164
 - exemple de programme 156
 - exemples de programmes TSO 159, 164
 - High Level Assembler (HLASM) 153
 - interface en langage 153
 - macros 213
 - zone de communication 162

B

- balises CDATA dans un fichier XML exporté 98
- balises cell dans un fichier XML exporté 98
- base de données
 - codes SQL
 - Voir* codes SQL
 - ID sous-système, variable globale 216
 - lecture non validée par rapport à la lecture non reproductible 237

- BIGINT (type de données)
 - exportation 73
 - mots clés de code d'édition, format exporté 113
 - largeur de colonne lors de l'exportation 73
- BINARY, mot clé vu dans les formats exportés 113
- BINARY (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation 73
 - mots clés de code d'édition, format exporté 113
- blocs de répertoire, spécification lors de l'exportation 237

C

- CALL, instruction 14
 - spécification de l'ensemble de résultats pour le rapport 237
- caractères de contrôle chariot, suppression 237
- casse mixte, mots de passe 9
- CCSID utilisé pour les exportations XML 98
- CHANGE, commande
 - confirmation de l'éditeur de table 224
- CHAR, mot clé vu dans les formats exportés 113
- CHAR (type de données)
 - exportation
 - mots clés de code d'édition, format exporté 113
- CLEAR, commande
 - confirmation de l'éditeur de table 224
- COBOL
 - appels de fonction 178
 - CICS 183
 - conditions requises pour l'exécution 183
 - délimiteurs 183
 - DSQCOMM 181
 - exemple de programme 180
 - interface callable 177
 - ISPF 185
 - macros 213
 - TSO 184
 - zone de communication 177
- codes, SQL
 - Voir* codes SQL
- codes d'édition, mots clés vus dans les exportations 113
- codes d'édition définis par l'utilisateur
 - mot clé vu dans les formats exportés 113
- codes hexadécimaux pour les types de données exportés 113
- codes retour
 - différents de zéro 12
 - interface callable 20
 - interface de commande 27
 - message 13
- codes SQL
 - affichage à partir de la dernière commande 249
 - positifs, activation de la prise en charge des messages 227
- codes SQL positifs, prise en charge des messages 227
- colonne
 - DESCRIBE commande
 - Voir* DESCRIBE, commande
 - en-tête
 - étiquettes par rapport aux noms 227
 - enregistrements C 80
 - format de données 80

- colonne (*suite*)
 - indicateur par défaut dans l'éditeur de table 224
 - largeurs lors de l'exportation 73
 - longueur des noms à l'exportation 227
 - longueur des noms lors de l'exportation 73
 - commande, synonymes 70
 - commande CANCEL
 - confirmation de l'éditeur de table 224
 - commande CONNECT
 - DB2 pour VM ou VSE 9
 - commande DISPLAY
 - restriction de la mise à jour de la date de dernière utilisation 237
 - commande EDIT
 - éditeur par défaut 237
 - commande END
 - confirmation de l'éditeur de table 224
 - commande EXPORT
 - format, langue nationale utilisée 237
 - restriction de la mise à jour de la date de dernière utilisation 237
 - commande IMPORT
 - langue nationale utilisée, IMPORT FORM 237
 - restriction de la mise à jour de la date de dernière utilisation 237
 - tables d'accélérateur 237
 - commande LIST
 - ordre des éléments, modification 227
 - valeur par défaut du paramètre OWNER 237
 - vues sous-jacentes
 - variables globales stockant les noms de vue 237
 - commande PRINT
 - CICS
 - nom/type de la file d'attente 223
 - variables globales
 - restriction de la date de dernière utilisation 237
 - suppression des caractères de contrôle chariot 237
 - commande REFRESH
 - confirmation de l'éditeur de table 224
 - commande RUN
 - restriction de la mise à jour de la date de dernière utilisation 237
 - variables de substitution 9
 - variables de substitution imbriquées 15
 - commande SAVE
 - format, langue nationale utilisée 237
 - paramètre SHARE, variable globale définissant 237
 - restriction de la mise à jour de la date de dernière utilisation 237
 - commande SAVE DATA
 - tables d'accélérateur 237
 - commande SHOW
 - SHOW CHANGE, confirmation de l'éditeur de table 224
 - SHOW SEARCH, confirmation de l'éditeur de table 224
 - commandes
 - applications 45
 - applications bilingues 43
 - code retour 12
 - environnement 31
 - INTERACT 48
 - interface 2
 - appel à partir d'un programme 26
 - codes retour 27
 - conditions requises 4
 - description 25
 - exemple de programme 25
 - service SELECT 26
 - commandes (*suite*)
 - langue nationale, paramètre 237
 - longueur 17
 - RUN 10
 - SET GLOBAL 53
 - spécifiques au système 9
 - variable de langue 41
 - variables globales prenant en charge 216
 - variables globales stockant la sortie des messages 224
 - commentaire
 - enregistrements de données d'application 108, 122
 - formats exportés 122
 - commentaires, envoi à IBM xii
 - commentaires en retour, envoi à IBM xii
 - commutateur INTERACT (DSQAO_INTERACT) 49
 - conflits de ressources, réduction 237
 - connaissances préalables requises pour DB2 for z/OS ix
 - CONNECT, commande
 - description 45
 - exemple 45
 - mots de passe à casse mixte 9, 222
 - procédures 9
 - procédures de lancement 7
 - variable globale pour CONNECT ID 216
 - connectivité avec des serveurs distants
 - restrictions relatives à l'interface de procédure mémorisée 45
 - connexions distantes à une base de données 9
 - conventions pour la mise en évidence x
 - conversions disponibles dans QMF
 - Voir support multiculturel
 - CONVERT QUERY, commande
 - restriction de la mise à jour de la date de dernière utilisation 237
 - variables globales pour 249
 - coût de requête estimé, désactivation 227
 - curseur
 - état 216
 - stabilité, activation 237
- ## D
- D, DC, DZ, DZC, codes d'édition
 - symbole monétaire, changement 227
 - data
 - enregistrements, exportation 73
 - largeurs de type 73
 - objet
 - formats 73
 - DATA, objet
 - variables globales associées 216
 - DATE, mot clé vu dans les formats exportés 113
 - DATE (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - mots clés de code d'édition, format exporté 113
 - format ISO lors de l'exportation 73
 - date de dernière utilisation, listes d'objets 227, 237
 - date de dernière utilisation des objets
 - limitation de RUN, SAVE, IMPORT 237
 - tri des listes par 227
 - date de modification, classement des listes par 227
 - DB2 pour VM ou VSE
 - commande CONNECT 9
 - connexions distantes 9
 - débogage des applications
 - ISPF, utilisation 36

débogage des applications (*suite*)
 test de dialogue PDF 36

DECFLOAT (type de données)
 exportation
 mots clés de code d'édition, format exporté 113

DECIMAL (type de données)
 codes SQLTYPE lors de l'exportation (format QMF) 73
 exportation
 mots clés de code d'édition, format exporté 113
 largeur de colonne lors de l'exportation 73

définition de schéma dans un fichier XML exporté 98

DELETE, commande
 confirmation de l'éditeur de table 224

délimiteurs
 entre les instructions dans les requêtes SQL 237

DESCRIBE, commande
 vues prenant en charge 237

diagramme de syntaxe, mode de lecture x

documentation sur l'interface de programmation 263

données
 enregistrements, exportation 98
 enregistrements D 80
 exportation 72
 objet
 format d'exportation IXF 80

données CSV
 exportation 100

données en virgule flottante décimale de format étendu
Voir DECFLOAT (type de données)

données en virgule flottante décimale de format long
Voir type de données DECFLOAT

données LOB
 exportation 98

données numériques
 data
 symbole monétaire, changement 227

données transitoires
 par opposition à la mémoire temporaire 145
 variables globales associées à l'impression 223
 variables globales associées au fichier auxiliaire 223
 variables globales pour le traçage 223

droit d'accès administrateur, variables globales pour 237

droits d'accès administrateur, variables globales pour 216

droits d'accès administrateur QMF
Voir droits d'accès administrateur, variables globales pour

DSQABFA 159

DSQABFAC 156

DSQCIA 154

DSQCM, variables globales 224

DSQCOMM
 Assembler 162
 C 167, 172
 COBOL 177
 définition 18
 définition de zones 18
 DSQCOMM 162
 DSQCOMM 172
 texte de message 153
 traitement des erreurs 22

DSQDEBUG, journal de trace
Voir aussi fonction de trace
 consignation des codes SQL positifs 227

DSQEC_NLFCMD_LANG, variable 41, 237

DSQQC, variables globales 249

DSQQM, variables globales 249

DSQSDLG, paramètre 56

DSQUOPTS, routine d'initialisation 237

DXI, variables globales 250

E

écran d'accueil 7

écran de rupture 113

écrans
 confirmation
 écrasements dans la mémoire temporaire, éviter 237
 estimation du coût relatif, désactivation 227
 ID
 affichage 227

écrans de confirmation
 Reset Report 237

écrans de variante 113

écrasements de la mémoire temporaire, éviter 237

EDIT, commande 35

emplacement en cours 45

en-têtes, colonne
Voir en-tête

END, commande
 description 46
 interface de commande 26
 réexécution de procédures de lancement 7
 session interactive 210
 types de session 46

enregistrement d'en-tête
 exportations XML 98
 format 73
 IXF 80
 niveau d'objet 102
 objet format 113
 zones 102

enregistrement de fin d'objet (E) 108

enregistrements
 colonne (C) 80
 continuation de données (C) 112
 description de table (T) 80
 données (D) 80
 données d'application (*) 108
 en-tête 102
 format fixe 102
 format variable 104
 formats 73
 valeur de données (V) 105

enregistrements de table dans des fichiers exportés 80

ensemble de données, définition pour les exportations 237

ensemble de résultats
 enregistrements dans un fichier XML exporté 98
 spécification de l'ensemble de résultats à utiliser pour le rapport 237

ensembles de données PDS et PDSE
 définition de la mémoire d'exportation 237
 définition du type vers QMF 237

ensembles de données PS, définition pour l'exportation 237

environnement CICS
 Assembler 4
 conditions requises pour z/OS 164

CONNECT, commande 9

file d'attente de données 4
 file d'attente de données transitoires 145
 format IXF 80
 utilisation pour le transfert d'objets QMF 72

interaction avec DB2 23

interface callable 4

programmes en COBOL 183

région 23

- environnement CICS (*suite*)
 - remplacement des paramètres de démarrage du programme 23
 - variables globales associées 223
 - environnements multilingues 43
 - estimation du coût pour la requête, désactivation 227
 - état de session QMF, variables pour 216
 - étiquettes par rapport aux noms pour les en-têtes de colonne 227
 - exécution interactive de QMF, variable globale 216
 - EXIT, commande 46
 - EXPORT, commande
 - à l'aide de CICS 145
 - DATA 72
 - formats
 - mots clés de code d'édition 113
 - longueurs des noms de colonne 73, 227
 - objet données 72
 - objet table 72
 - option IXF 80
 - spécification XML utilisée 98
 - TSO
 - spécification de la mémoire 237
 - exportation
 - calculs de largeur pour les types de données 73
 - codes hexadécimaux pour les types de données 73
 - considérations liées au stockage 145
 - diagrammes 142
 - données CSV 100
 - données et tables 72
 - données LOB 98
 - formats 102
 - mots clés utilisés pour les codes d'édition 113
 - objets format 71
 - par rapport à la sauvegarde 72
 - procédures 141
 - rapports 102
 - requêtes créées de façon interactive 102
 - requêtes QBE 142
 - Requêtes SQL 141
 - taille maximale du fichier 142
 - types d'objets 71
 - extraction multiligne 56
 - incidents au niveau du nom tripartite 56
- F**
- feuille de style pour les fichiers XML exportés 98
 - fichier auxiliaire
 - utilisation de la mémoire étendue dans TSO 237
 - variables globales prenant en charge 223
 - fichier de feuille de style DSQ1STSH 98
 - fichier schéma DSQ1SCEM 98
 - fichier schéma qmf_data.xsd 98
 - file d'attente
 - variables globales d'impression vers 223
 - variables globales pour les données auxiliaires 223
 - FLOAT (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - format des données de colonne, IXF 80
 - mots clés de code d'édition, format exporté 113
 - largeur de colonne lors de l'exportation 73
 - fonction de trace
 - codes SQL positifs 227
 - niveau de trace de l'application 216
 - fonction NLF (National Language Feature)
 - conditions requises pour les écrans 39
 - environnements de session 39
 - langue 39
 - format
 - enregistrements d'en-tête lors de l'exportation 113
 - enregistrements de données lors de l'exportation 113
 - exportation 102
 - taille maximale du fichier 142
 - FORM.MAIN
 - noms de colonne par rapport aux étiquettes 227
 - FORMAT.COLONNES
 - noms de colonne par rapport aux étiquettes 227
 - importation 122
 - LAYOUT, commande 113
 - mots clés de type de données lors de l'exportation 113
 - numéros de table 113
 - numéros de zone 113
 - panneaux
 - variables globales associées 216
 - support multiculturel pour SAVE, EXPORT, IMPORT 237
 - traduction 113
 - format d'enregistrement des données
 - type de données XML 98
 - format d'exportation CSV
 - taille maximale du fichier 142
 - format d'exportation IXF
 - binaire 80
 - caractère 80
 - description 80
 - exemples 80
 - numéros de version 80
 - prise en charge de nom long 80
 - taille maximale du fichier 142
 - format HTML pour les rapports
 - exportation dans 127
 - taille maximale du fichier 142
 - format QMF pour l'exportation de données et de tables 73, 142
 - format VBS, exportations de données XML 98
 - formats 124
 - codés
 - définition 71
 - données, exportation 73
 - données de colonne 80
 - enregistrement d'en-tête 73
 - IXF 71, 80
 - objet format 113
 - objet rapport 124
 - sans informations de contrôle 129
 - objet requête créée de façon interactive 131
 - table 72
 - FORTRAN
 - appels de fonction 188
 - DSQABFF 189
 - DSQCOMM 192
 - exemple de programme 189
 - interface appellable 186
 - ISPF 194
 - macros 213
 - TSO 194
 - zone de communication 187

G

GDDM (gestionnaire d'affichage de données graphiques) 48
gestionnaire de base de données, variable globale stockant le type 216
GET GLOBAL, commande 21, 47
GRAPHIC, mot clé vu dans les formats exportés 113
GRAPHIC (type de données)
 codes SQLTYPE lors de l'exportation (format QMF) 73
 exportation
 mots clés de code d'édition, format exporté 113
 largeur de colonne lors de l'exportation 73
Graphics Data Format (GDF) 142

I

ICU (Interactive Chart Utility) 48
ID de jeu de caractères codés, exportations XML 98
ID sous-système, variable globale 216
ID utilisateur, connexions de base de données 216
IMPORT, commande
 à l'aide de CICS 145
 DATA, option 101
 définition 71
 erreurs et avertissements lors de l'exécution de 113
 spécification XML utilisée 98
importation 71, 102
 informations sur le niveau d'objet 102
 objet format 113
 objet requête créée de façon interactive 139
 tables créées en dehors de QMF 72
indicateur d'attention pour les applications 216
indicateur d'attention pour les utilisateurs 216
informations de contrôle
 suppression dans les rapports 129
informations de maintenance x
informations de support x
informations SQLSTATE, affichage 249
informations sur la migration 102
instruction ARG 12
instruction de signal d'erreur 12
instruction PARSE ARG 12
instruction PREPARE, résolution des accès simultanés 237
INTEGER (type de données)
 exportation
 format des données de colonne, IXF 80
 mots clés de code d'édition, format exporté 113
INTERACT, commande
 description 48
 format de commande 49
 session
 arrêt 46, 48
 format 48
interface callable
 appel à partir d'une procédure navigationnelle 15
 application, exécution 22
 CICS, exécution sous 23
 COBOL 177
 codes retour 20
 commandes 21
 débogage des applications 149
 description 17
 exemples de programmes 3
 Assembler 156
 C 170
 COBOL 180
 FORTRAN 186

interface callable (*suite*)
 GET GLOBAL, commande 47
 informations sur le traitement de la commande 17
 ISPF 3
 langues 4, 17, 153
 macros 213
 PL/I 196
 programme 3
 REXX
 appel avec 7
 description 206
 utilisations 3
 variables de communication 207
START, commande 3
 démarrage de QMF 21
 syntaxe 55
 zone de communication 3
 C 172
 COBOL 181
 définition 18
 définition de zones 18
 traitement des erreurs 22
interface de procédure mémorisée
 DSQDBLG, paramètre 56
 restrictions relatives à l'accès aux données éloignées 9
interface de produit, macros 213
interfaces d'accès à QMF
 interface callable
 Voir interface callable
 interface de commande
 Voir commandes, interface
 interface de procédure mémorisée
 Voir interface de procédure mémorisée
interfaces de programmation d'application
 interface callable
 Voir interface callable
 procédure mémorisée (TSO uniquement)
 Voir interface de procédure mémorisée
interfaces de programme d'application
 interface de commande
 Voir commandes, interface
ISPF (Interactive System Productivity Facility)
 interface callable 34

J

journal, trace 227

L

langage C
 appels de fonction 168
 CICS 174
 conditions requises pour l'interface 168
 exemples de programmes 170
 interface callable 167
 ISPF 175
 TSO 175
 zone de communication 167
 DSQCOMM 172
 mappage 167
langues prises en charge
 conversions
 Voir support multiculturel
 programmation de l'interface callable 4
LAYOUT, commande 113

LAYOUT, commande (*suite*)
restriction de la mise à jour de la date de dernière utilisation 237
lecture non validée, activation 237
lignes
longueur maximale 237
longueurs maximales
Requêtes SQL 237

M

macros d'interface de produit 213
mémoire
données auxiliaires
virtuelle étendue 237
étendue, données temporaires (TSO) 56
spécification lors de l'exportation 237
mémoire étendue, utilisation pour les données auxiliaires
définition de la quantité 237
paramètres de programme requis 56
mémoire étendue, utilisation pour les données temporaires
paramètres de programme requis 56
mémoire virtuelle
Voir mémoire
MESSAGE, commande
affichage de texte 50
description 50
écrans d'aide QMF 50
écrans ISPF 50
exemples 50
options 50
suppression de l'exécution d'une procédure linéaire 50
traçage 151
messages
variables globales associées
messages de la commande précédente 224
prise en charge des messages des codes SQL positifs 227
RUN QUERY, messages 249
mini-session
commandes non valides 68
commandes valides 68
rapport 68
mise en évidence, conventions x
mode compatibilité et extraction multiligne 56
mode de fonctionnement
variable globale montrant 216
mode interactif
ICU GDDM 48
procédures de lancement 7
QMF 48
mots clés 56
pour les codes d'édition, dans un fichier exporté 113
START, commande 56
mots clés SQL
requêtes à plusieurs instructions
Voir requêtes à plusieurs instructions
SELECT
options de résolution des accès simultanés 237
mots de passe à casse mixte 222
mots de passe RACF et à casse mixte 222
MR, paramètre
incidents au niveau du nom tripartite 56
multiligne, extraction 56

N

National Language Feature (NLF)
Voir support multiculturel
New Function Mode et extraction multiligne 56
NEXT, commande
confirmation de l'éditeur de table 224
niveau d'isolement des requêtes 237
NLF (National Language Feature)
Voir support multiculturel
NLF (National Language Feature, langue)
définie 39
environnements multilingues 43
ID de langue 33
nom d'emplacement
Voir aussi accès aux données distant
variable globale stockant 216
noms
classement des listes par 227
pour les colonnes, changement en étiquettes de base de données 227
vues prenant en charge la commande LIST, variables globales pour 237
noms de propriétaire
classement des listes par 227
valeur par défaut de la commande LIST 237
variables globales associées 216
noms tripartites dans les commandes QMF
échec lorsque MR=YES 56
notification des codes SQL positifs 227
NULL
valeurs
caractère par défaut, éditeur de table 224
NUMERIC, mot clé vu dans les formats exportés 113
numéro d'édition de QMF, variable globale 216
numéro de version
QMF, variable globale stockant 216
versions IXF lors de l'exportation 80

O

objet
date de dernière utilisation 227, 237
ressource partagée 237
type
classement des listes par 227
variables globales associées à l'objet actuel 216
objet DATA
incomplet, activation du panneau Reset Report 237
objet de données incomplet
activation du panneau Reset Report 237
écran d'entrée 101
objets bilingues 41
objets diagramme 142
option 64BIT, paramètre DSQSPTYP 56
option A pour le débogage 149
option FILE, paramètre DSQSPTYP 56
option L pour le débogage 149
ordre
éléments de la commande LIST, modification 227
ordre croissant des listes 227
ordre de tri pour la commande LIST 227
ordre décroissant des listes 227

P

- panneau d'estimation du coût relatif, désactivation 227
- panneau Reset Report, activation 237
- panneaux 48
 - en cours 48
 - interactifs 48
 - noms
 - variables globales associées 216
- panneaux de confirmation
 - écrasements dans la mémoire temporaire 237
- paramètre CASE de profil QMF 222
- paramètre CONFIRM de profil QMF 222
- paramètre DECIMAL de profil QMF 222
- paramètre LANGUAGE, profil QMF 222
- paramètre LENGTH de profil QMF 222
- paramètre MODEL, profil QMF 222
- paramètre RESOURCE GROUP, profil QMF 222
- paramètre SHARE de la commande SAVE 237
- paramètre SPACE, profil QMF 222
- paramètre WIDTH, profil QMF 222
- paramètres DSQ dans une commande START
 - DSQADPAN 56
 - DSQALANG 56
 - DSQSBSTG 56
 - DSQSCMD 56
 - DSQSDBCS 56
 - DSQSDBNM 56
 - DSQSDBQN 56
 - DSQSDBQT 56
 - DSQSDEBUG 56
 - DSQSIROW 56
 - DSQSMODE 56
 - DSQSMRFI 56
 - DSQSPILL 56
 - DSQSPLAN 56
 - DSQSPRID 56
 - DSQSPTYP 56
 - DSQSRSTG 56
 - DSQSRUN 56
 - DSQSSPQN 56
 - DSQSSUBS 56
- PDF 35, 36
- performances
 - options de résolution des accès simultanés 237
- PL/I
 - appels de fonction 198
 - CICS 203
 - DSQABFP 199
 - DSQCOMM 201
 - exemple de programme 199
 - interface appellable 196
 - ISPF 204
 - macros 213
 - TSO 204
 - z/OS 203
 - zone de communication 196
- plan d'application
 - ID par défaut, V12.1 56
- plateforme VM
 - restrictions de nom tripartite 56
- plateforme VSE
 - restrictions de nom tripartite 56
- PREVIOUS, commande
 - confirmation de l'éditeur de table 224
- PRINTER, paramètre
 - QMF, profil 222

- prise en charge DBCS
 - modification de l'indicateur NULL, éditeur de table 224
 - variables globales associées 216
- procédure
 - éviter l'écrasement du panneau PROC 237
 - exportation
 - taille maximale du fichier 142
 - initialisation, définition des variables pendant 216
 - invocation, nouvelle exécution 237
 - procédures mémorisées
 - Voir* procédure mémorisée
- procédure linéaire
 - STOPPROC, option 50
 - suppression 50
- procédure mémorisée
 - démarrage de QMF for TSO en tant que
 - support des variables globales 216
 - spécification de l'ensemble de résultats pour le rapport 237
- procédures de lancement
 - applications bilingues 42
 - commande CONNECT 7
 - écriture 7
 - nom, spécification 7
 - stockage 7
- profil
 - éviter l'écrasement des valeurs non enregistrées 237
 - variables globales associées 222

Q

- Q.SYSTEM_INI, procédure 216
- QBE (Query-By-Example)
 - format d'exportation 142
 - spécifications d'importation/exportation de fichier 142
 - taille maximale du fichier lors de l'exportation 142
- QMF interactif
 - variable globale pour le mode de fonctionnement 216
- QMF par lots
 - variable globale pour le mode de fonctionnement 216
- quantité de défilement, paramètre 227

R

- rapports
 - affichage de texte 50
 - données de ligne 108
 - données de table 106
 - enregistrements 102
 - enregistrements d'exportation 124
 - enregistrements de ligne (L) 109
 - exécutions des procédures mémorisées
 - Voir* procédure mémorisée
 - exemple d'en-tête 124
 - exemple de rapport 124
 - exportation 72
 - taille maximale du fichier 142
 - exportation sans informations de contrôle 129
 - exportés transversaux 130
 - format d'exportation 124
 - HTML 127
 - impression
 - caractères de contrôle chariot 237
 - longueur des lignes, paramètre 237
 - mini-session 68
 - numéros de table 124

- rapports (*suite*)
 - numéros de zone 124
 - objet 2
 - format d'exportation 124
 - numéros de table 124
 - numéros de zone 124
 - sur 124
 - panneau 2
 - suppression après l'exécution de la requête 227
 - sur 124
 - utilisations de l'exportation 124
 - recommandations
 - légales 261
 - registre spécial SQLID 216
 - requête
 - conversion
 - Voir* CONVERT QUERY, commande
 - coût estimé, désactivation 227
 - éviter l'écrasement du panneau QUERY 237
 - exécution
 - Voir* RUN QUERY, commande
 - instructions CALL
 - Voir aussi* CALL, instruction
 - spécification de l'ensemble de résultats pour le rapport 237
 - niveau d'isolement 237
 - rapport de l'exécution
 - suppression 227
 - sous-types stockés dans globales 216
 - SQL
 - Voir* requêtes SQL
 - variable globale de modèle 216
 - requête assistée
 - enregistrements de données 131
 - exportation 102, 131
 - taille maximale du fichier 142
 - format d'exportation 131
 - numéros de table 131
 - numéros de zone 131
 - spécifications d'importation/exportation de fichier 142
 - requête créée de façon interactive
 - enregistrements d'en-tête 131
 - requêtes à plusieurs instructions 237
 - requêtes SQL
 - longueurs dépassant 32 ko 237
 - Requêtes SQL 141
 - exportation
 - taille maximale du fichier 142
 - résolution des accès simultanés 237
 - RUN, commande
 - écran d'invite 10
 - variables de substitution 10
 - RUN QUERY, commande
 - requêtes à plusieurs instructions 237
 - requêtes SQL supérieures à 32 ko 237
 - tables d'accélérateur 237
 - variables globales pour les messages 249
- S**
- SAVE, option
 - EDIT TABLE, commande
 - variables globales associées 224
 - SAVE DATA, commande 72
 - SELECT, instructions
 - options de résolution des accès simultanés 237
 - service SELECT PGM, ISPF 34
 - services d'analyse et exportation XML 98
 - session, variables enregistrant l'état 216
 - session QMF de traitement par lots, variables globales
 - pour 216
 - SET GLOBAL, commande 216
 - interface callable 21, 53
 - saisie de variables 10
 - syntaxe 53
 - syntaxe étendu 53
 - SHOW, commande
 - SHOW GLOBALS 216
 - signe + dans les colonnes de l'éditeur de table,
 - modification 224
 - signe dollar dans les rapports, changement 227
 - SKIP LOCKED DATA, option des instructions SELECT 237
 - SMALLINT (type de données)
 - exportation
 - format des données de colonne, IXF 80
 - mots clés de code d'édition, format exporté 113
 - sous-routine DSQCIX 208
 - spécification pour l'exportation/importation XML 98
 - START, commande
 - démarrage de QMF 21, 55
 - erreurs de débogage 153
 - mots clés
 - DSQADPAN 56
 - DSQALANG 56
 - DSQSBSTG 56
 - DSQSCMD 56
 - DSQSDBCS 56
 - DSQSDBNM 56
 - DSQSDBQN 56
 - DSQSDBQT 56
 - DSQSDBG 56
 - DSQSIROW 56
 - DSQSMODE 56
 - DSQSMRFI 56
 - DSQSPILL 56
 - DSQSPRID 56
 - DSQSPTYP 56
 - DSQSRSTG 56
 - DSQSRUN 56
 - DSQSSPQN 56
 - DSQSSUBS 56
 - liste 56
 - syntaxe 55
 - zone de communication d'interface 18
 - stockage
 - considérations liées à l'exportation 145
 - données XML 101
 - spécification lors de l'exportation 237
 - taille maximale du fichier exporté 142
 - stockage temporaire 22
 - CICS
 - variables globales associées à l'impression 223
 - variables globales associées au fichier auxiliaire 223
 - confirmation des écrasements 237
 - file d'attente 145
 - modification 22
 - restrictions 3
 - variables globales pour le traçage 223
 - support multiculturel
 - formats (SAVE/EXPORT/IMPORT) 237
 - variables globales associées 216, 237
 - symbole monétaire, changement 227
 - symbole monétaire euro, activation 227

- synonymes de commande
 - création 70
 - définitions 222
 - exemple 2
 - table NLF 39
- syntaxe étendu
 - SET GLOBAL, commande 53
- système d'exploitation, variable globale pour 216

T

- tables
 - commande LIST
 - variables globales associées 237
 - création en dehors de QMF 80
 - enregistrements de description (T) 80, 106
 - enregistrements de ligne (R) 108
 - exportation 72
 - format, numéros 113
 - objet
 - importation 72
 - règles d'importation/exportation 101
 - spécifications d'importation/exportation de fichier 142
 - traitement 72
 - rapport, numéros 124
 - requête créée de façon interactive, numéros 131
- termes du contrat de licence VUE (variable globale) 216
- TIME, mot clé vu dans les formats exportés 113
- TIME (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - mots clés de code d'édition, format exporté 113
 - format ISO lors de l'exportation 73
 - largeur de colonne lors de l'exportation 73
- TIMEST, mot clé vu dans les formats exportés 113
- TIMESTAMP, type de données
 - exportation
 - format des données de colonne, IXF 80
- TIMESTAMP (type de données)
 - exportation
 - mots clés de code d'édition, format exporté 113
- TIMESTAMP WITH TIME ZONE (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - format des données de colonne, IXF 80
 - formats d'enregistrement de colonne 80
 - mots clés de code d'édition, format exporté 113
 - numéros de version IXF pour l'exportation 80
 - format ISO lors de l'exportation 73
 - largeur de colonne lors de l'exportation 73
- touches de fonction
 - où les définitions sont stockées 222
- traçabilité 63
- traçage
 - allocation de fichier pour 150
 - commandes ISPF 36
 - création de définitions de trace 36
 - définition 149
 - désactivation 150
 - exemple 151
 - option A 149
 - option L 149
 - paramètre de profil pour 222
 - variables globales pour 223
- TRACE, commande 63
 - exemple PL/I 64
- trace de service 63

- trace de service (*suite*)
 - exemple PL/I 64
- TSO
 - mémoire étendue pour les données temporaires 56
 - mots de passe à casse mixte sur RACF 9
 - programmes Assembler 164
 - programmes d'interface appelable en langage
 - Assembler 164
 - programmes d'interface appelable en langage C 175
 - programmes d'interface appelable en langage REXX 210
 - programmes en langage C 175
 - programmes REXX 210
- TSTMPITZ, mot clé vu dans les formats exportés 113
- type de données à virgule décimale flottante
 - Voir type de données DECFLOAT
- type de données BIGINT
 - exportation
 - format des données de colonne, IXF 80
- type de données BINARY
 - exportation
 - format des données de colonne, IXF 80
 - largeur de colonne lors de l'exportation 73
- type de données CHAR
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - format des données de colonne, IXF 80
 - largeur de colonne lors de l'exportation 73
- type de données DATE
 - exportation
 - format des données de colonne, IXF 80
 - largeur de colonne lors de l'exportation 73
- type de données DECFLOAT
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - format des données de colonne, IXF 80
 - largeur de colonne lors de l'exportation 73
- type de données DECIMAL
 - exportation
 - format des données de colonne, IXF 80
- type de données GRAPHIC
 - exportation
 - format des données de colonne, IXF 80
- type de données INTEGER
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - largeur de colonne lors de l'exportation 73
- type de données LONG VARCHAR
 - exportation
 - format des données de colonne, IXF 80
- type de données LONG VARGRAPHIC
 - exportation
 - format des données de colonne, IXF 80
- type de données SMALLINT
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - largeur de colonne lors de l'exportation 73
- type de données TIME
 - exportation
 - format des données de colonne, IXF 80
- type de données TIMESTAMP
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - format ISO lors de l'exportation 73
 - largeur de colonne lors de l'exportation 73
- type de données VARBINARY
 - largeur de colonne lors de l'exportation 73
- type de données VARCHAR
 - exportation
 - format des données de colonne, IXF 80

- type de données VARGRAPHIC
 - exportation
 - format des données de colonne, IXF 80
- type de données XML
 - exportation 98
 - considérations liées au stockage 101
 - format des enregistrements de données 98
 - taille maximale du fichier 142
- type de données XML (Extensible Markup Language)
 - Voir XML*, type de données
- types de données, considérations relatives à l'exportation 73, 113

U

- unité d'oeuvre éloignée 9
 - comportement de la commande 45
- UNKNOWN, mot clé vu dans les formats exportés 113
- USE CURRENTLY COMMITTED, option des instructions
 - SELECT 237

V

- valeurs de paramètre DATAFORMAT, commande EXPORT
 - QMF 73, 227
- valeurs monétaires, changement du symbole monétaire 227
- validation du fichier XML exporté 98
- VARBINARY (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - format des données de colonne, IXF 80
 - mots clés de code d'édition, format exporté 113
- VARCHAR (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - largeur de colonne lors de l'exportation 73
- VARGRAPHIC (type de données)
 - codes SQLTYPE lors de l'exportation (format QMF) 73
 - exportation
 - mots clés de code d'édition, format exporté 113
 - largeur de colonne lors de l'exportation 73
- variable d'indicateur d'arrêt 216
- variable globale d'environnement 216
- variable globale de contrat de licence VUE 216
- variable globale du contrat de licence pour QMF VUE 216
- variables
 - définition 9
 - globales 9, 215
 - substitution 9
 - pool 17
 - rc 12
 - règles 54
 - saisie 10
 - substitution 9
 - traitement des erreurs 22
- variables de substitution 10
 - affectation de valeurs 9
 - appels de programme REXX 15
 - syntaxe 9
 - variables globales, définition 9
- variables globales
 - affichage d'un rapport après la commande RUN
 - QUERY 227
 - caractères de contrôle chariot dans les impressions 237
 - CICS
 - données auxiliaires 223
 - impression 223

- variables globales (*suite*)
 - CICS (*suite*)
 - traçage 223
 - classes
 - DSQAO 216
 - DSQAP 222, 223
 - DSQCM 224
 - DSQCP 224
 - DSQEC 237
 - DSQQC 249
 - DSQQM 249
 - DXY 250
 - interface de procédure mémorisée 216
 - commande EXPORT, mémoire (TSO) 237
 - commande LIST
 - valeur par défaut de OWNER 237
 - vues prenant en charge 237
 - comparaison entre le fonctionnement par lots et le fonctionnement interactif 216
 - CONNECT ID 216
 - création 53
 - création de variables 53
 - date de dernière utilisation sur les objets 237
 - définition 53
 - SET GLOBAL, commande 53
 - définition à l'initialisation 216, 237
 - définition/affichage 216
 - définitions de synonyme de commande 222
 - droit d'accès administrateur 237
 - droits d'accès administrateur 216
 - écrasements dans la mémoire temporaire 237
 - ensemble de résultats pour les procédures mémorisées 237
 - état du curseur de base de données 216
 - étiquettes par rapport aux noms 227
 - gestionnaire de base de données 216
 - ID de panneau, affichage 227
 - ID sous-système 216
 - indicateur d'attention pour les utilisateurs 216
 - instructions SQL à plusieurs instructions 237
 - interface de procédure mémorisée 216
 - lignes extraites, nombre de 216
 - liste 215
 - longueur des lignes dans les rapports QMF 237
 - longueur des noms de colonne à l'exportation 227
 - longueur des noms de colonne lors de l'exportation 73
 - modèle de requête 216
 - niveau d'isolement des requêtes 237
 - niveau de trace de l'application 216
 - nom d'emplacement distant 216
 - nom de la base de données locale 216
 - nom du panneau en cours 216
 - nom du propriétaire 216
 - notification des codes SQL positifs 227
 - objet actuel 216
 - ordre de liste 227
 - panneau d'estimation du coût relatif 227
 - panneau de format en cours 216
 - panneau Reset Report, affichage 237
 - paramètre CASE de profil 222
 - paramètre CONFIRM de profil 222
 - paramètre LANGUAGE de profil 222
 - paramètre LENGTH de profil 222
 - paramètre MODEL de profil 222
 - paramètre PRINTER de profil 222
 - paramètre RESOURCE GROUP de profil 222
 - paramètre SHARE de la commande SAVE 237

- variables globales (*suite*)
 - paramètre SPACE de profil 222
 - paramètre TRACE de profil 222
 - paramètre WIDTH de profil 222
 - prise en charge DBCS 216
 - prise en charge de commande 216
 - procédure d'invocation, nouvelle exécution 237
 - QMF utilisé via RUW 222
 - quantité de défilement 227
 - règles 54
 - requêtes SQL supérieures à 32 ko 237
 - résolution des accès simultanés 237
 - RUN QUERY, messages 249
 - sortie des messages 224
 - sous-types de requête 216
 - stockage étendu pour les données auxiliaires 237
 - support multiculturel 216, 237
 - symbole monétaire 227
 - version/édition 216
- variables globales de session 250
- variables globales DSQAO 216
- variables globales DSQAP 222, 223
- variables globales DSQCP 224
- variables globales DSQEC 237
- variantes, FORM.DETAIL
 - variable globale stockant le nombre 216
- verrous sur les données
 - éviter l'escalade 237
 - options de résolution des accès simultanés 237
- vue
 - commande LIST, variables globales associées 237

W

WAIT FOR OUTCOME, option des instructions SELECT 237

Z

- zone de communication
 - COBOL 177, 181
 - définition 18
 - FORTRAN 192
 - PL/I 201
- zones de contrôle dans les objets exportés 104
 - enregistrements de fichier de format 104
 - enregistrements de fichier de rapport 104
 - Enregistrements T 106



Numéro de programme : 5615-DB2
5697-QM2

SC43-3493-00

