



**Guide d'administration système
Volume 2**

**Adaptive Server Enterprise
12.5**

Réf. du document : 38414-01-1250-01

Dernière mise à jour : Mai 2001

Cette publication concerne le logiciel de gestion de bases de données de Sybase et toutes les versions ultérieures qui ne feraient pas l'objet d'une réédition de la documentation ou de la publication de notes de mise à jour. Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis. Le logiciel décrit est fourni sous contrat de licence et il ne peut être utilisé ou copié que conformément aux termes de ce contrat.

Pour commander des ouvrages supplémentaires ou acquérir des droits de reproduction, si vous habitez aux Etats-Unis ou au Canada, appelez notre Service Clients au (001-800) 685-8225, télécopie (001-617) 229-9845.

Les clients ne résidant ni aux Etats-Unis ni au Canada et qui disposent d'un contrat de licence pour les U.S.A. peuvent joindre notre Service Clients par télécopie. Ceux qui ne bénéficient pas de cette licence doivent s'adresser à leur revendeur Sybase ou au distributeur le plus proche. Les mises à jour du logiciel ne sont fournies qu'à des dates d'édition périodiques. Tout ou partie de cette publication ne peut être reproduit, transmis ou traduit sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, manuel, optique ou autre) sans l'accord écrit préalable de Sybase, Inc.

Sybase, le logo Sybase, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, AnswerBase, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, Backup Server, ClearConnect, Client-Library, Client Services, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, E-Anywhere, E-Whatever, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, ImpactNow, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, KnowledgeBase, MainframeConnect, Maintenance Express, MAP, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, MySupport, Net-Gateway, Net-Library, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, RW-Library, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, Transact-SQL, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server et XP Server sont des marques de Sybase, Inc.

Unicode et le logo Unicode sont des marques déposées de Unicode, Inc.

Tous les autres noms de produit, société ou marque apparaissant dans ce document sont des marques ou marques déposées de leurs propriétaires respectifs.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608, Etats-Unis d'Amérique.

Table des matières

CHAPITRE 15	Présentation des sujets relatifs aux ressources disque	569
	Allocation de devices et placement d'objets.....	569
	Commandes de gestion des ressources disque	570
	Éléments à prendre en compte dans les décisions de gestion du stockage	572
	Restauration	572
	Performances.....	573
	Etat et valeurs par défaut au moment de l'installation.....	574
	Tables système gérant le stockage.....	574
	Table sysdevices	575
	Table sysusages	576
	Table syssegments	577
	Table sysindexes.....	577
CHAPITRE 16	Initialisation des devices de base de données	579
	Présentation des devices de base de données	579
	Utilisation de la commande disk init	580
	Syntaxe de disk init	581
	Exemples de commande disk init.....	581
	Spécification d'un nom de device logique à l'aide de disk init.....	581
	Spécification d'un nom de device physique à l'aide de disk init.....	581
	Choix d'un numéro de device pour disk init.....	582
	Spécification de la taille du device à l'aide de disk init	583
	Spécification du paramètre dsync à l'aide de disk init (facultatif).....	584
	Autres paramètres facultatifs pour disk init	586
	Obtention d'informations relatives aux devices	587
	Suppression de devices	589
	Désignation des devices par défaut	589
	Choix des devices par défaut et des autres devices	590

CHAPITRE 17	Mise en miroir des devices de base de données.....	593
	Présentation de la mise en miroir des disques	593
	Choix des éléments à mettre en miroir	594
	Mise en miroir avec un espace disque physique minimal	595
	Mise en miroir pour une reprise instantanée	596
	Conditions ne désactivant pas la mise en miroir	597
	Commandes de mise en miroir des disques	598
	Initialisation des miroirs	599
	Annulation de la mise en miroir d'un device	600
	Relance de la mise en miroir	602
	waitfor mirrorexit.....	602
	Mise en miroir du device master	603
	Obtention d'informations sur les devices et les miroirs	603
	Didacticiel de mise en miroir des disques	603
CHAPITRE 18	Configuration de la mémoire	607
	Détermination de la mémoire disponible pour Adaptive Server ...	607
	Allocation de la mémoire par Adaptive Server	609
	Allocation de l'espace disque	610
	Tailles de pages logiques et buffers plus importants	611
	Mémoire segmentée.....	611
	Utilisation de la mémoire par Adaptive Server	612
	Mémoire nécessitée par Adaptive Server	614
	Si vous effectuez une mise à niveau.....	615
	Paramètres de configuration concernant l'allocation de mémoire	615
	Allocation dynamique de la mémoire	617
	Démarrage impossible d'Adaptive Server	618
	Réduction dynamique des paramètres de configuration	
	de la mémoire.....	618
	Procédures système de configuration de la mémoire	622
	Utilisation de sp_configure pour définir des paramètres	
	de configuration	622
	Utilisation de sp_helpconfig pour obtenir de l'aide sur	
	les paramètres de configuration	624
	Utilisation de sp_monitorconfig pour chercher des	
	statistiques d'utilisation du cache de métadonnées.....	626
	Principales utilisations de la mémoire d'Adaptive Server.....	627
	Adaptive Server code exécutable et overhead.....	628
	Caches de données et de procédures	628
	Détermination de la taille du cache de procédures	628
	Détermination de la taille du cache de données par défaut ..	629
	Connexions utilisateur	631
	Bases de données, index et objets ouverts.....	632
	Nombre de verrous.....	633

	Devices de base de données et structures d'E/S disque	633
	Autres paramètres utilisant de la mémoire.....	634
	Traitement parallèle.....	634
	Serveurs distants.....	635
	Intégrité référentielle.....	636
	Autres paramètres ayant une incidence sur la mémoire	636
CHAPITRE 19	Configuration des caches de données	637
	Le cache de données sur Adaptive Server	638
	Commandes de configuration du cache.....	639
	Informations sur les caches de données.....	641
	Configuration des caches de données.....	643
	Configuration explicite du cache par défaut	646
	Changement de type de cache	648
	Configuration de la stratégie de remplacement d'un cache ..	649
	Division d'un cache de données en zones mémoire	650
	Taille d'E/S de journal pour le cache de journal	654
	Liaison d'objets à des caches	654
	Restrictions des liaisons de caches	656
	Obtention d'informations sur les liaisons de caches	656
	Vérification de l'overhead du cache.....	657
	Incidence de l'overhead sur l'espace total du cache	658
	Suppression de liaisons de caches.....	659
	Modification de la zone de vidage d'une zone mémoire	659
	Zone de vidage trop petite.....	661
	Zone de vidage trop grande	662
	Modification de la limite de prélecture asynchrone	
	pour une zone mémoire.....	663
	Redéfinition de la taille des caches de données nommés	664
	Augmentation de la taille d'un cache.....	664
	Diminution de la taille d'un cache	665
	Suppression de caches de données	666
	Modification de la taille des zones mémoire	667
	Déplacement d'espace de la zone mémoire	667
	Déplacement d'espace d'autres zones mémoire.....	668
	Ajout de partitions de cache	670
	Définition du nombre de partitions de cache avec	
	sp_configure	670
	Définition du nombre de partitions de cache local.....	671
	Priorité	671
	Suppression d'une zone mémoire	672
	Impossibilité de supprimer des zones lorsque	
	les pages sont en cours d'utilisation	672

	Incidence de la liaison de caches sur la mémoire et les plans de requêtes	673
	Sortie de pages du cache.....	673
	Verrouillage pour effectuer des liaisons	673
	Incidence de la liaison de caches sur les procédures et les triggers stockés.....	674
	Configuration de caches de données à l'aide du fichier de configuration	674
	Entrées sur les zones et les caches dans le fichier de configuration	674
	Conseils sur la configuration des caches	678
CHAPITRE 20	Gestion des serveurs multiprocesseur	681
	Traitement parallèle	681
	Définitions	682
	Architecture cible.....	682
	Configuration d'un environnement SMP.....	684
	Gestion des moteurs	684
	Désactivation d'un moteur à l'aide de la commande dbcc engine	686
	Gestion des connexions utilisateur.....	689
	Paramètres de configuration qui influent sur les systèmes SMP	691
CHAPITRE 21	Création et gestion des bases de données utilisateur.....	695
	Commandes de création et de gestion des bases de données utilisateur	695
	Autorisations de gestion des bases de données utilisateur	696
	Utilisation de la commande create database	697
	Syntaxe de la commande create database	697
	Fonctionnement de la commande create database	698
	Ajout d'utilisateurs aux bases de données	699
	Attribution d'espace et de devices aux bases de données	700
	Devices et taille de base de données par défaut	701
	Estimation de l'espace requis.....	702
	Placement du journal de transactions sur un device distinct	702
	Evaluation de la taille du journal de transactions	703
	Device et taille de journal par défaut.....	705
	Transfert du journal de transactions sur un autre device	705
	Utilisation de l'option for load pour la restauration des bases de données.....	706
	Utilisation de l'option with override avec create database.....	707
	Modification de la propriété d'une base de données.....	708

	Utilisation de la commande alter database	709
	Syntaxe de la commande alter database	709
	Utilisation de la commande drop database	711
	Tables système gérant l'allocation d'espace	712
	Table sysusages	712
	Obtention d'informations sur le stockage des bases de données	715
	Noms et options des devices de base de données	715
	Vérification du volume d'espace utilisé.....	716
	Interrogation d'une table système sur l'utilisation de l'espace	719
CHAPITRE 22	Définition des options de base de données	721
	Présentation des options de base de données	721
	Utilisation de la procédure sp_dboption	722
	Description des options de base de données	722
	abort tran on log full.....	723
	allow nulls by default	723
	auto identity	724
	dbo use only	724
	ddl in tran.....	724
	identity in nonunique index.....	726
	no chkpt on recovery	726
	no free space acctg	727
	read only.....	727
	select into/bulkcopy/pllsort	727
	single user	728
	trunc log on chkpt	728
	unique auto_identity index.....	729
	Modification des options de base de données	730
	Affichage des options sur une base de données	731
CHAPITRE 23	Création et utilisation de segments	733
	Présentation des segments.....	733
	Segments définis par le système	734
	Commandes et procédures nécessaires à la gestion des segments	735
	Avantages des segments.....	736
	Gestion de l'espace	736
	Amélioration des performances.....	737
	Transfert d'une table vers un autre device	739
	Création de segments	740
	Modification de la portée des segments.....	740
	Extension de la portée des segments	741
	Réduction de la portée d'un segment.....	742

Attribution d'objets de base de données aux segments.....	742
Création d'objets sur un segment.....	743
Placement des objets existants sur les segments.....	745
Stockage des pages de texte sur un device distinct	747
Création d'index clusterisés sur les segments	748
Suppression de segments.....	748
Obtention d'informations relatives aux segments.....	749
sp_helpsegment	750
sp_helpdb.....	751
sp_help et sp_helpindex.....	751
Segments et tables système	752
Didacticiel sur les segments.....	753
Segments et index clusterisés.....	757
CHAPITRE 24	Utilisation de la commande reorg
	759
Sous-commandes de reorg	759
Cas d'utilisation de la commande reorg	761
Utilisation de l'utilitaire optdiag pour évaluer le besoin d'une	
réorganisation (reorg)	761
Récupération d'espace sans la commande reorg	762
Déplacement de lignes redirigées vers leurs pages d'origine	762
Utilisation de reorg compact pour supprimer	
les redirections de lignes	763
Récupération de l'espace inutilisé provenant des	
suppressions et des mises à jour	763
Récupération de l'espace inutilisé et annulation de la redirection de	
lignes	764
Reconstruction d'une table.....	765
Conditions requises pour l'exécution de reorg rebuild.....	766
Options resume et time de réorganisation des tables	
de grande taille	767
Spécification du nb_minutes dans l'option time.....	768
Utilisation de la commande reorg rebuild sur les index.....	769
Syntaxe	769
Commentaires	769
Restrictions.....	770
Méthode de reconstruction des index par	
reorg rebuild nom_index.....	770
Espace nécessaire pour la reconstruction d'un index	771
Caractéristiques de performances	772
Messages d'état	772

CHAPITRE 25	Contrôle de la cohérence des bases de données.....	773
	Présentation de Database Consistency Checker.....	773
	Compréhension des concepts d'allocation de pages et d'objets..	774
	Fonctionnement de la table d'allocation d'objets (OAM)	777
	Fonctionnement des liens de pages.....	779
	Contrôles pouvant être effectués avec dbcc	779
	Contrôle de la cohérence des bases de données et des tables ..	780
	dbcc checkstorage	781
	dbcc checktable.....	784
	dbcc checkdb	787
	Contrôle de l'allocation de pages	788
	dbcc checkalloc	788
	dbcc indexalloc.....	790
	dbcc tablealloc.....	790
	Correction des erreurs d'allocation à l'aide des options fix nofix	791
	Génération de rapports avec dbcc tablealloc et dbcc indexalloc .	792
	Contrôle de cohérence des tables système	793
	Stratégies d'utilisation des commandes pour le contrôle	
	de cohérence.....	793
	Comparaison des performances des commandes dbcc	794
	Utilisation des E/S étendues et prélecture asynchrone.....	795
	Planification de la maintenance de la base de données	
	sur votre site	795
	Fonctionnement du résultat des commandes dbcc.....	798
	Erreurs provoquées par des problèmes de cohérence	
	dans la base de données	799
	Comparaison des erreurs légères et des erreurs graves	800
	Contrôle des erreurs à l'aide de la commande dbcc checkverify .	802
	Fonctionnement de la commande dbcc checkverify	802
	Quand utiliser dbcc checkverify.....	803
	Mode d'utilisation de dbcc checkverify	804
	Suppression d'une base de données endommagée	805
	Préparation préliminaire à l'utilisation de dbcc checkstorage.....	805
	Planification des ressources.....	807
	Configuration d'Adaptive Server pour dbcc checkstorage.....	810
	Création de la base de données dbccdb.....	815
	Mise à jour de la table dbcc_config	817
	Maintenance de dbccdb	818
	Réévaluation et mise à jour de la configuration de dbccdb...	818
	Suppression des données obsolètes dans dbccdb	819
	Suppression des espaces de travail.....	819
	Réalisation de contrôles de cohérence sur dbccdb.....	819
	Génération de rapports à partir de dbccdb	820
	Rapport de synthèse des opérations dbcc checkstorage.....	820

Rapport sur la configuration, les statistiques et les informations	821
Visualisation des informations de configuration pour une base de données cible.....	821
Comparaison des résultats des opérations dbcc checkstorage.....	822
Rapport des erreurs rencontrées dans une base de données objet	822
Rapport d'informations sur les statistiques de dbcc_counter	823

CHAPITRE 26

Elaboration d'un plan de sauvegarde et de reprise..... 825

Suivi des modifications de la base de données	826
Informations sur le journal de transactions.....	827
Synchronisation d'une base de données avec son journal :	
points de reprise	827
Définition de l'intervalle de reprise.....	827
Procédure de point de reprise automatique	828
Troncature du journal après des points de reprise automatiques	829
Points de reprise libres	830
Requête manuelle d'un point de reprise.....	830
Reprise automatique après panne ou arrêt du système	831
Choix des messages à afficher pendant une reprise	831
Ordre de reprise des bases de données défini par l'utilisateur	832
Utilisation de sp_dbrecovery_order.....	833
Modification ou suppression de la position de reprise d'une base de données	833
Affichage de l'ordre de reprise des bases de données défini par l'utilisateur	834
Isolement de panne lors de la reprise	834
Utilisation des commandes de sauvegarde et chargement...	846
Application des modifications à la base de données :	
load transaction	849
Mise à disposition des utilisateurs de la base de données :	
online database	849
Déplacement d'une base de données vers un autre Adaptive Server	849
Mise à niveau d'une base de données utilisateur.....	850
Options spéciales de dump transaction	851
Options de chargement spéciales pour identifier les fichiers sauvegardés	852
Restauration d'une base de données à partir de sauvegardes	852

Pause et reprise de la mise à jour des bases de données	855
Consignes d'utilisation de quiesce database.....	856
Rôles serveur dans une relation primaire et secondaire	858
Démarrage du serveur secondaire à l'aide de l'option -q	859
Valeur de l'enregistrement du journal de la base de données "au repos" mise à jour.....	859
Mise à jour du numéro de séquence de sauvegarde	860
Sauvegarde des devices primaires à l'aide de quiesce database	862
Marque des copies archivées pendant l'état repos	866
Désignation du responsable des sauvegardes	867
Utilisation du Backup Server pour la sauvegarde et la reprise	868
Relations entre Adaptive Server et Backup Server	869
Communication avec le Backup Server	871
Montage d'un nouveau volume	872
Démarrage et arrêt du Backup Server	873
Configuration du serveur pour l'accès à distance	873
Choix d'un support de sauvegarde.....	874
Protection des bandes de sauvegarde contre la réécriture... ..	874
Sauvegarde dans des fichiers ou sur disque	875
Création de noms de devices logiques pour les devices de sauvegarde locaux	875
Liste des noms courants de devices	876
Ajout d'un device de sauvegarde	877
Redéfinition d'un nom de device logique.....	877
Planification des sauvegardes des bases de données utilisateur	877
Planification des sauvegardes systématiques.....	878
Autres sauvegardes de bases de données à planifier	878
Planification des sauvegardes de master	880
Sauvegarde de master après chaque changement	880
Sauvegarde des scripts et des tables système	880
Troncature du journal de transactions de la base de données master.....	881
Prévention des changements de volume et des reprises.....	881
Planification des sauvegardes de la base de données model	882
Troncature du journal de transactions de la base de données model	882
Planification des sauvegardes de la base de données sybssystemprocs	882
Configuration d'Adaptive Server pour les chargements simultanés	883
Collecte des statistiques de sauvegarde.....	884

CHAPITRE 27	Sauvegarde et restauration de bases de données utilisateur . 885
	Syntaxe de la commande dump and load..... 886
	Spécification du device de sauvegarde et de base de données .. 889
	Règles de spécification des noms de bases de données 891
	Règles de spécification des devices de sauvegarde..... 891
	Détermination du device de sauvegarde (bande) par
	Backup Server 893
	Spécification de l'option de compression 894
	Fichiers de sauvegarde et sauvegardes compressées de
	Backup Server 898
	Chargement de bases de données et de journaux
	de transactions sauvegardés avec l'option compress 899
	Spécification d'un Backup Server distant 900
	Spécification de la densité de bande, de la taille de bloc et
	de la capacité 902
	Remplacement de la densité par défaut..... 903
	Remplacement de la taille de bloc par défaut 904
	Spécification de la capacité de la bande pour
	les commandes dump 905
	Fonction bande non rembobinage de Backup Server 905
	Spécification du nom de volume 907
	Chargement depuis un volume multifichier 908
	Identification d'une sauvegarde..... 908
	Amélioration des performances de sauvegarde ou
	de chargement..... 911
	Compatibilité avec les versions antérieures 912
	Etiquettes stockées au format nombre entier..... 913
	Configuration des ressources système 913
	Spécification d'autres devices de sauvegarde : clause stripe on . 917
	Sauvegarde sur plusieurs devices 918
	Chargement à partir de plusieurs devices..... 919
	Utilisation de moins de devices pour le chargement
	que pour la sauvegarde 919
	Spécification des caractéristiques de devices individuels 920
	Options d'exploitation de bande 920
	Spécification de démontage de la bande 922
	Rembobinage de la bande 922
	Protection des fichiers de sauvegarde contre l'écrasement .. 922
	Réinitialisation d'un volume avant une sauvegarde 923
	Sauvegardes multiples sur un seul volume..... 923
	Remplacement de la destination par défaut des messages 924
	Mise des bases de données en ligne avec with standby_access 927
	Dans quels cas utiliser with standby_access ? 928

Mise des bases de données en ligne avec for standby_access.....	928
Obtention d'informations sur les fichiers de sauvegarde.....	929
Requête d'informations relatives aux en-têtes de sauvegarde.....	930
Détermination de la base de données, du device, du nom de fichier et de la date	931
Copie du journal après une panne de device.....	932
Troncature d'un journal ne se trouvant pas sur un segment distinct.....	934
Troncature du journal en environnement de début de développement.....	934
Troncature d'un journal sans espace libre	935
Risques liés à l'utilisation de with truncate_only et de with no_log	936
Aménagement d'un espace suffisant pour le journal.....	936
Réponse aux requêtes de changement de volume.....	939
Syntaxe de sp_volchanged	939
Invites de changement de volume lors des sauvegardes	940
Invites de changement de volume lors des chargements	942
Restauration d'une base de données : instructions détaillées	944
Réalisation d'une sauvegarde courante du journal de transactions	945
Contrôle de l'utilisation de l'espace	945
Suppression des bases de données	947
Suppression des devices défectueux.....	947
Initialisation de nouveaux devices.....	948
Recréation des bases de données.....	948
Chargement de la base de données	949
Chargement des journaux de transactions.....	949
Activation des bases de données.....	951
Chargement des sauvegardes de bases de données effectuées avec des versions antérieures	952
Mise à niveau d'une sauvegarde vers Adaptive Server	952
Bit d'état base de données désactivée.....	954
Identificateurs de versions.....	955
Liaisons de cache et chargement de bases de données	956
Bases de données et liaisons de cache	957
Objets de base de données et liaisons de cache.....	957
Contraintes entre bases de données et chargement de bases de données	958

CHAPITRE 28	Restauration des bases de données système	961
	En quoi consiste la restauration d'une base de données système ?	961
	Symptômes indiquant que la base de données master est endommagée.....	962
	Restauration de la base de données master	962
	A propos de la restauration	963
	Résumé de la procédure de restauration.....	963
	Etape 1 : Se procurer des copies des tables système	965
	Etape 2 : Construction d'un nouveau device master.....	965
	Etape 3 : Démarrage d'Adaptive Server en mode restauration de master.....	966
	Etape 4 : Recréation des allocations de device pour master	967
	Etape 5 : Vérification des informations syssservers de Backup Server	972
	Etape 6 : Vérification de l'exécution de Backup Server.....	973
	Etape 7 : Chargement d'une sauvegarde de master.....	973
	Etape 8 : Mise à jour du paramètre de configuration number of devices	974
	Etape 9 : Redémarrage d'Adaptive Server en mode restauration de master.....	974
	Etape 10 : Comparaison des tables système pour vérifier la sauvegarde courante de master.....	975
	Etape 11 : Redémarrage d'Adaptive Server.....	975
	Etape 12 : Restauration des ID utilisateur du serveur.....	975
	Etape 13 : Restauration de la base model	976
	Etape 14 : Vérification d'Adaptive Server.....	977
	Etape 15 : Sauvegarde de master.....	977
	Restauration de la base model	977
	Restauration de la base model générique.....	978
	Restauration de model à partir d'une sauvegarde.....	978
	Restauration de model sans sauvegarde.....	978
	Restauration de la base sybssystemprocs	979
	Restauration de sybssystemprocs à l'aide d'installmaster	979
	Restauration de sybssystemprocs à l'aide de load database ..	981
	Restauration des tables système à l'aide de disk reinit et de disk refit	981
	Restauration de sysdevices à l'aide de disk reinit.....	982
	Restauration de sysusages et de sysdatabase à l'aide de disk refit	983

CHAPITRE 29	Gestion de l'espace libre avec des seuils	985
	Contrôle d'espace libre avec le seuil ultime	986
	Passage du seuil	986
	Contrôle de la fréquence d'exécution de sp_thresholdaction	987
	Enregistrement d'annulations et seuil ultime	988
	Calcul de l'espace pour les enregistrements d'annulation	989
	Détermination de l'espace utilisé par les enregistrements d'annulation	990
	Effet des enregistrements d'annulation sur le seuil ultime	990
	Seuils définis par l'utilisateur	991
	Seuil ultime et caches de journal utilisateur pour des segments de journal et de données partagés	993
	Le franchissement du seuil ultime suspend les transactions	994
	Utilisation de alter database quand la base de données master atteint le seuil ultime	996
	Annulation automatique ou suspension des processus	996
	Utilisation de abort tran on log full pour annuler les transactions.....	997
	Reprise des processus suspendus	997
	Ajout, modification et suppression de seuils	998
	Affichage d'informations sur les seuils existants	998
	Seuils et tables système.....	998
	Ajout d'un seuil d'espace libre	999
	Modification d'un seuil d'espace libre	999
	Spécification d'une nouvelle procédure de seuil ultime	1000
	Suppression d'un seuil	1001
	Création d'un seuil d'espace libre pour le segment de journal ...	1001
	Ajout d'un seuil de journalisation à 45 pour cent de la taille du journal.....	1002
	Test et réglage du nouveau seuil	1002
	Création de seuils supplémentaires sur d'autres segments.....	1005
	Détermination de la valeur du seuil	1005
	Création de procédures de seuil	1006
	Déclaration des paramètres de la procédure	1006
	Génération de messages dans le journal d'erreurs.....	1007
	Sauvegarde du journal de transactions.....	1007
	Une procédure de seuil simple.....	1008
	Une procédure plus complexe.....	1009
	Où mettre en place une procédure de seuil	1011
	Désactivation de la comptabilisation de l'espace libre pour les segments de données	1012
Index		1013

Table des matières

Présentation des sujets relatifs aux ressources disque

Ce chapitre présente certains sujets courants qui déterminent comment allouer et utiliser les ressources disque avec Adaptive Server.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Allocation de devices et placement d'objets	569
Commandes de gestion des ressources disque	570
Éléments à prendre en compte dans les décisions de gestion du stockage	572
Etat et valeurs par défaut au moment de l'installation	574
Tables système gérant le stockage	574

De nombreuses options par défaut d'Adaptive Server prennent des valeurs raisonnables du point de vue de la gestion du stockage. Par exemple, elles définissent l'emplacement des bases de données, des tables et des index ainsi que la quantité d'espace allouée à chacun de ces éléments. La responsabilité de l'allocation et de la gestion du stockage est souvent centralisée. En général, c'est l'administrateur système qui a le dernier mot en ce qui concerne l'allocation des ressources disque à Adaptive Server ainsi que l'emplacement physique des bases de données, des tables et des index sur ces ressources.

Allocation de devices et placement d'objets

Lors de la configuration d'un nouveau système, l'administrateur système doit prendre en compte plusieurs points qui ont une incidence directe sur le nombre et la taille des ressources disque. Ces sujets d'allocation de device portent sur les commandes et procédures qui ajoutent des ressources disque à Adaptive Server. Ils font l'objet d'une description dans les chapitres du tableau 15-1.

Tableau 15-1 : Sujets d'allocation de device

Tâche	Chapitre
Initialise et alloue une zone par défaut de devices de base de données.	Chapitre 16, "Initialisation des devices de base de données"
Met en miroir les devices de base de données en vue de leur restauration.	Chapitre 17, "Mise en miroir des devices de base de données"

Une fois les ressources disque initiales allouées à Adaptive Server, l'administrateur système, le propriétaire de la base de données et les propriétaires d'objets doivent s'interroger sur la façon de placer les bases de données et les objets de base de données sur des devices de base de données spécifiques. Ces questions déterminent le placement des objets de base de données sur votre système et si les objets partagent des devices. Les tâches de placement d'objets sont présentées tout au long de ce manuel, notamment dans les chapitres figurant dans le tableau 15-2.

Tableau 15-2 : Sujets de placement d'objets

Tâche	Chapitre
Place des bases de données sur des devices de base de données spécifiques.	Chapitre 21, "Création et gestion des bases de données utilisateur"
Place des tables et des index sur des devices de base de données spécifiques.	Chapitre 23, "Création et utilisation de segments"

N'allouez pas les devices indépendamment du placement d'objets. Par exemple, si vous décidez qu'une table doit résider sur une paire de devices dédiée, vous devez d'abord allouer ces devices à Adaptive Server. Les autres sections de ce chapitre donnent un aperçu de l'allocation de devices et du placement d'objets ; elles comportent également des pointeurs qui renvoient aux chapitres concernés, si nécessaire.

Commandes de gestion des ressources disque

Le tableau 15-3 recense les principales commandes utilisées par un administrateur système pour allouer des ressources disque à Adaptive Server et contient des références aux chapitres présentant ces commandes.

Tableau 15-3 : Commandes d'allocation des ressources disque

Commande	Tâche	Chapitre
<pre>disk init name = "nom_dev" physname = "nom_phys"...</pre>	Met un device physique à la disposition d'un Adaptive Server particulier. Affecte un nom de device de base de données (<i>nom_dev</i>) permettant d'identifier le device dans les autres commandes Adaptive Server.	Chapitre 16, "Initialisation des devices de base de données"
<pre>sp_deviceattr nom_logique, nom_option, valeur_option</pre>	Modifie le paramètre <i>dsync</i> d'un fichier de devices de base de données existant.	Chapitre 16, "Initialisation des devices de base de données"
<pre>sp_diskdefault "nom_dev"...</pre>	Ajoute <i>nom_dev</i> à la zone générale de l'espace de base de données par défaut.	Chapitre 16, "Initialisation des devices de base de données"
<pre>disk mirror name = "nom_dev" mirror = "nom_phys"...</pre>	Met en miroir un device de base de données sur un device physique particulier.	Chapitre 17, "Mise en miroir des devices de base de données"

Le tableau 15-4 présente les commandes utilisées dans le placement d'objets. Pour plus d'informations sur l'impact du placement d'objets sur les performances, reportez-vous au chapitre 5, "Gestion de l'emplacement physique des données", du document *Performances et optimisation*.

Tableau 15-4 : Commandes de placement d'objets sur les ressources disque

Commande	Tâche	Chapitre
<pre>create database...on nom_dev ou alter database...on nom_dev</pre>	Met les devices de base de données à la disposition d'une base de données Adaptive Server spécifique. La clause <i>log on</i> de <i>create database</i> place les journaux de base de données sur un device de base de données spécifique.	Chapitre 21, "Création et gestion des bases de données utilisateur"
<pre>create database... ou alter database...</pre>	Utilisées avec la clause <i>on nom_dev</i> , ces commandes allouent de l'espace aux devices de base de données par défaut.	Chapitre 21, "Création et gestion des bases de données utilisateur"
<pre>sp_addsegment nom_seg, nom_base, nom_dev et sp_extendsegment nom_seg, nom_base, nom_dev</pre>	Crée un segment, une collection nommée d'espace à partir des devices mis à la disposition d'une base de données particulière.	Chapitre 23, "Création et utilisation de segments"

Commande	Tâche	Chapitre
<code>create table...on nom_seg</code> ou <code>create index...on nom_seg</code>	Crée des objets de base de données et les place sur un segment particulier de l'espace disque attribué à la base de données.	Chapitre 23, "Création et utilisation de segments"
<code>create table...</code> ou <code>create index...</code>	Utilisés sans la clause <code>on nom_seg</code> , les tables et les index occupent la zone générale de l'espace alloué à la base de données (devices par défaut).	Chapitre 23, "Création et utilisation de segments"

Éléments à prendre en compte dans les décisions de gestion du stockage

L'administrateur système doit prendre des décisions concernant l'allocation physique de l'espace des bases de données d'Adaptive Server. Les principaux points à prendre en compte sont les suivants :

- **Restauration** – la mise en miroir des disques et la maintenance des journaux sur un device physique distinct fournissent deux mécanismes de restauration complète dans l'éventualité d'une défaillance des disques.
- **Performances** – pour les tables ou les bases de données dans lesquelles la vitesse de lecture et d'écriture sur disque est essentielle, le placement correct des objets de base de données sur les devices physiques est un facteur d'amélioration des performances. La mise en miroir des disques ralentit la vitesse d'écriture sur disque.

Restauration

La restauration est la principale raison justifiant l'utilisation de plusieurs devices disque. Il est possible d'effectuer une reprise instantanée en mettant en miroir les devices de base de données. Quant à la reprise complète, il suffit de stocker le journal de la base de données sur un device physique séparé.

Conservation des journaux sur un device séparé

Tant qu'un device de base de données n'est pas mis en miroir, la reprise complète nécessite le stockage du journal de transactions d'une base de données sur un device différent de celui des données réelles (index compris) d'une base de données. En cas de défaillance d'un disque dur, vous pouvez créer une base de données actualisée en chargeant une sauvegarde de la base de données et en appliquant les enregistrements de journal qui étaient stockés de façon sécurisée sur un autre device. Pour plus d'informations sur la clause `log on de create database`, reportez-vous au chapitre 21, "Création et gestion des bases de données utilisateur".

Mise en miroir

Si tous les devices Adaptive Server sont mis en miroir sur un disque physique distinct, la restauration instantanée est assurée en cas de défaillance du disque dur. Le chapitre 17, "Mise en miroir des devices de base de données", décrit le processus de mise en miroir des devices.

Performances

Vous pouvez améliorer les performances du système en plaçant les journaux et les objets de base de données sur des devices distincts :

- Le placement d'une table sur un disque dur et les index non clusterisés sur un autre se traduit par l'accélération des lectures et des écritures sur disque, puisque le travail est réparti sur deux disques.
- La répartition de tables volumineuses sur deux disques peut améliorer les performances, notamment celles des applications multi-utilisateur.
- Lorsque le journal et les données partagent les mêmes devices, la mise en buffer des enregistrements des journaux de transactions dans le cache des journaux utilisateur est désactivée.
- Le partitionnement fournit plusieurs points d'insertion pour une table sans index, ajoute un degré de parallélisme aux systèmes configurés pour effectuer le traitement parallèle des demandes et autorise la répartition des E/S d'une table sur plusieurs devices de base de données.

Pour plus d'informations sur l'impact du placement d'objets sur les performances, reportez-vous au chapitre 5, "Gestion de l'emplacement physique des données", du guide *Performances et optimisation*.

Etat et valeurs par défaut au moment de l'installation

Vous trouverez des instructions d'installation d'Adaptive Server dans le Guide d'installation pour votre plate-forme. Le programme et les scripts d'installation initialisent le device master et configurent les bases de données master, model, subsystemprocs, sybsecurity ainsi que les bases temporaires.

Lorsque vous installez Adaptive Server, les bases de données système, les segments définis par le système et les devices de base de données sont organisés de la façon suivante :

- Les bases de données master, model et tempdb sont installées sur le device master.
- La base de données subsystemprocs est installée sur un device que vous avez spécifié.
- Trois segments sont créés dans chaque base de données : system, default et logsegment.
- Le device master est le device de stockage par défaut de toutes les bases de données créées par l'utilisateur.

Remarque Après l'initialisation des nouveaux devices pour le stockage par défaut, supprimez le device master de la zone de stockage par défaut à l'aide de `sp_diskdefault`. Ne stockez pas de bases de données et des objets utilisateur sur le device master. Pour plus d'informations, reportez-vous à la section "Désignation des devices par défaut", page 589.

- Si vous installez la base de données d'audit, sybsecurity, celle-ci se trouve sur son propre device.

Tables système gérant le stockage

Deux tables système dans la base de données master et deux autres dans chaque base de données utilisateur suivent le placement des bases de données, des tables (y compris la table des journaux de transactions, syslogs) et des index. La relation entre les tables est illustrée par la figure 15-1.

Figure 15-1 : Tables système gérant le stockage

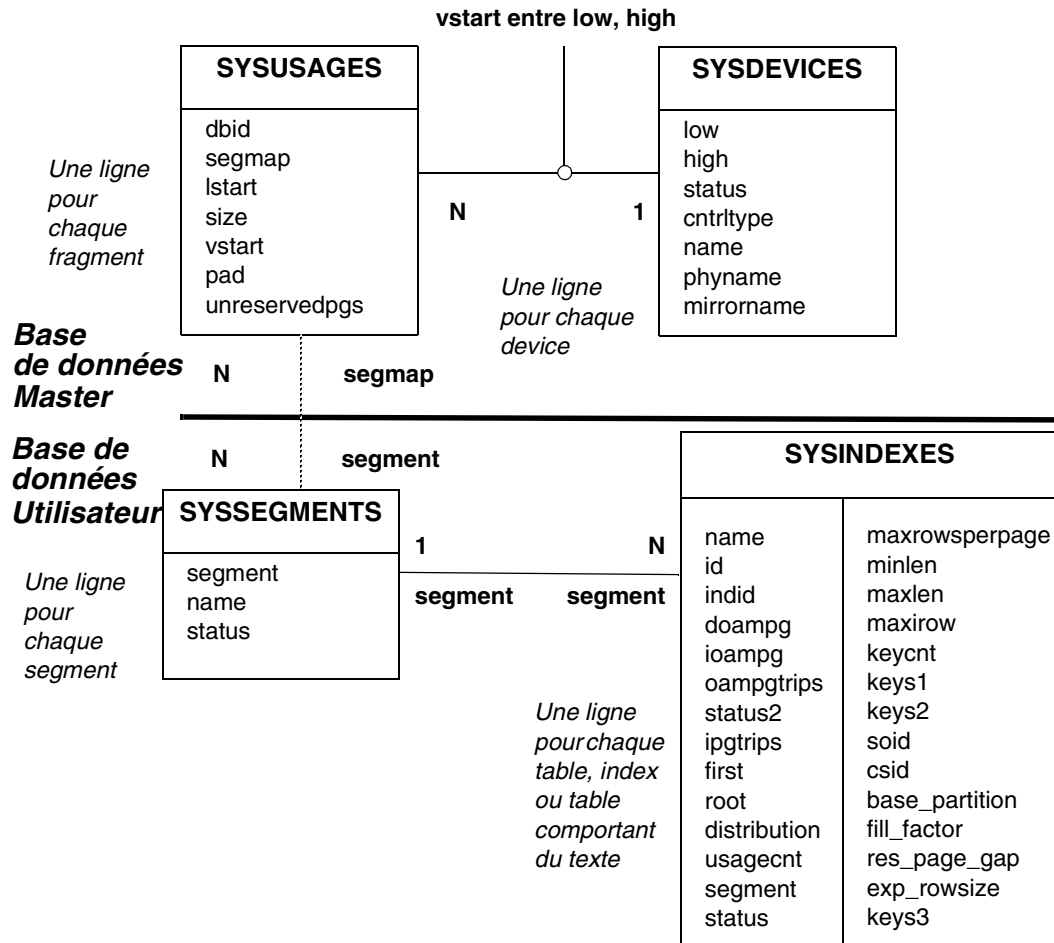


Table sysdevices

La table sysdevices de la base de données master contient une ligne pour chaque **device de base de données** et peut contenir une ligne pour chaque device de sauvegarde (fichier bande, disque ou du système d'exploitation) mis à la disposition d'Adaptive Server.

La commande `disk init` ajoute des entrées concernant les devices de base de données à la table `master.sysdevices`. Les devices de sauvegarde, ajoutés à l'ordre de la procédure système `sp_addumpdevice`, sont présentés dans le chapitre 26, "Elaboration d'un plan de sauvegarde et de reprise".

`sysdevices` stocke deux noms pour chaque device :

- Un *nom logique* ou un *nom de device*, utilisé dans toutes les commandes de gestion de stockage ultérieures, est stocké dans la colonne `name` de `sysdevices`. Il s'agit en général d'un nom convivial indiquant peut-être l'utilisation prévue du device, par exemple "logdev" ou "userdbdev".
- Le *nom physique* est celui défini pour le device dans le système d'exploitation. Il n'est utilisé que dans la commande `disk init` ; ensuite, toutes les commandes de stockage des données d'Adaptive Server utilisent le nom logique.

Vous placez une base de données ou un journal de transactions sur un ou plusieurs devices en spécifiant le nom logique du device dans l'instruction `create database` ou `alter database`. La clause `log on` de `create database` place un journal de transactions de base de données sur un device différent pour assurer la restauration complète. Le device de journal doit également avoir une entrée dans `sysdevices` avant que vous ne puissiez utiliser `log on`.

Une base de données peut résider sur un ou plusieurs devices et un device peut stocker une ou plusieurs bases de données. Pour plus d'informations sur la création de bases de données sur des devices de base de données, reportez-vous au chapitre 21, "Création et gestion des bases de données utilisateur".

Table `sysusages`

La table `sysusages` de la base de données `master` garde la trace de tout l'espace que vous attribuez à l'ensemble des bases de données Adaptive Server.

`create database` et `alter database` allouent de l'espace à la base de données en ajoutant une ligne à `sysusages` pour chaque device de base de données ou fragment de device. Lorsque vous n'allouez qu'une partie de l'espace sur un device à l'aide de `create` ou de `alter database`, cette partie s'appelle un *fragment*.

Les procédures système `sp_addsegment`, `sp_dropsegment` et `sp_extendsegment` modifient la colonne `segmap` dans `sysusages` pour le device qui est mappé ou non mappé à un segment. Le chapitre 23, "Création et utilisation de segments", présente ces procédures en détail.

Table *syssegments*

La table `syssegments`, une dans chaque base de données, répertorie les segments dans une base de données. Un **segment** est une collection de devices de base de données et/ou de fragments mis à la disposition d'une base de données spécifique. Les tables et les index peuvent être attribués à un segment particulier et donc à un device physique particulier ou être répartis sur un ensemble de devices physiques.

`create database` crée des entrées par défaut dans `syssegments`. Les procédures système `sp_addsegment` et `sp_dropsegment` ajoutent et suppriment des lignes dans `syssegments`.

Table *sysindexes*

La table `sysindexes` répertorie chaque table et chaque index ainsi que le segment dans lequel chaque table, index clusterisé, index non clusterisé et chaîne de pages de texte est stocké(e). Elle contient également d'autres informations comme le paramètre `max_rows_per_page` de la table ou de l'index.

Les commandes `create table`, `create index` et `alter table` créent de nouvelles lignes dans `sysindexes`. Le partitionnement d'une table modifie la fonction des entrées de `sysindexes` pour la table, comme le décrit le chapitre 5, "Gestion de l'emplacement physique des données", du guide *Performances et optimisation*.

Initialisation des devices de base de données

Ce chapitre explique comment initialiser des devices de base de données et attribuer des devices à la zone de devices par défaut.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Présentation des devices de base de données	579
Utilisation de la commande disk init	580
Syntaxe de disk init	581
Obtention d'informations relatives aux devices	587
Suppression de devices	589
Désignation des devices par défaut	589

Présentation des devices de base de données

Un device de base de données stocke les objets qui constituent les bases de données. Le terme *device* ne se rapporte pas nécessairement à un device physique spécifique : il peut faire référence à une partie de disque (telle qu'une partition) ou à un fichier du système de fichiers utilisé pour le stockage des bases de données et de leurs objets.

Chaque device ou fichier de base de données doit être préparé et connu d'Adaptive Server avant de pouvoir être utilisé pour le stockage des bases de données. Ce processus est appelé **initialisation**.

Une fois initialisé, un device de base de données peut être :

- alloué à la zone de devices par défaut pour les commandes `create` et `alter database` ;
- attribué à la zone d'espace disponible d'une base de données utilisateur ;

- attribué à une base de données utilisateur et utilisé pour le stockage d'un ou plusieurs objets de base de données ;
- attribué au stockage des journaux de transactions d'une base de données.

Utilisation de la commande *disk init*

Un administrateur système initialise de nouveaux devices de base de données à l'aide de la commande *disk init* qui effectue les tâches suivantes :

- mappage du device de disque physique ou du fichier du système d'exploitation avec un nom de *device de base de données* ;
- insertion du nom du nouveau device dans *master..sysdevices* ;
- préparation du device pour le stockage de la base de données.

Remarque Avant d'exécuter la commande *disk init*, reportez-vous au Guide d'installation pour votre plate-forme pour choisir un device de base de données et le préparer pour l'utiliser avec Adaptive Server. Vous pouvez repartitionner les disques de votre ordinateur afin d'obtenir des performances optimales avec les bases de données Sybase.

disk init divise les devices de base de données en **unités** d'allocation. La taille de l'unité d'allocation dépend de la taille de la page logique configurée pour votre serveur (2, 4, 8 ou 16 ko). Dans chaque unité d'allocation, la commande *disk init* initialise la première page comme une page d'allocation qui comprendra les informations sur la base de données résidant (le cas échéant) sur l'unité d'allocation.

Avertissement ! Après l'exécution de la commande *disk init*, sauvegardez la base de données master. Si la base master est endommagée, sa restauration sera ainsi plus simple et plus sûre. Reportez-vous au chapitre 28, "Restauration des bases de données système".

Syntaxe de *disk init*

La syntaxe de *disk init* est la suivante :

```
disk init
  name = "nom_device",
  physname = "nom_physique",
  [vdevno = numéro_device_virtuel, ]
  size = taille_du_device
  [, vstart = adresse_virtuelle,
   cntrltype = numéro_contrôleur]
  [, dsync = {true | false}]
```

Exemples de commande *disk init*

Sous UNIX :

```
disk init
  name = "user_disk",
  physname = "/dev/rxy1a",
  size = "10M"
```

Sous Windows NT :

```
disk init
  name = "user_disk",
  physname = "d:\devices\userdisk.dat",
  size = "10M"
```

Spécification d'un nom de device logique à l'aide de *disk init*

Le *nom_device* doit être un identificateur correct. Ce nom est utilisé dans les commandes *create database* et *alter database*, ainsi que dans les procédures système qui gèrent les segments. Le nom du device logique n'est connu que d'Adaptive Server ; il ne l'est pas du système d'exploitation sur lequel s'exécute le serveur.

Spécification d'un nom de device physique à l'aide de *disk init*

Le *nom_physique* du device de base de données est le nom d'une partition de disque (UNIX), d'un device externe ou d'un fichier du système d'exploitation. Sur les plates-formes PC, vous ne pouvez utiliser que des noms de fichier du système d'exploitation pour *nom_physique*.

Choix d'un numéro de device pour *disk init*

Adaptive Server spécifie automatiquement le texte disponible identifiant le numéro du device de base de données. Il s'agit du numéro de device virtuel (vdevno). Il n'est pas nécessaire de spécifier ce numéro lors du lancement de la commande `disk init`.

Si vous choisissez de sélectionner manuellement la commande `vdevno`, celle-ci doit être unique parmi les devices utilisés par Adaptive Server. Le numéro de device 0 représente le device master. Le numéro le plus élevé doit être inférieur d'un chiffre au nombre de devices de base de données pour lequel votre système est configuré. Par exemple, pour un système ayant une configuration par défaut de 10 devices, les numéros de device admis sont compris entre 1 et 9. Pour connaître la valeur de la configuration de votre système, exécutez la commande `sp_configure "nombre devices"` et vérifiez la valeur d'exécution :

```
sp_configure "nombre devices"
Parameter name  Default  Memory Used  Config Value  Run Value
-----
number of devices  10      0            10            10
```

Pour connaître les numéros déjà utilisés pour `vdevno`, consultez la colonne `device_number` du rapport provenant de `sp_helpdevice` ou utilisez la requête ci-après pour afficher la liste de tous les numéros de device déjà utilisés :

```
select distinct low/16777216
       from sysdevices
       order by low
```

Adaptive Server est configuré par défaut pour 10 devices. Les contraintes du système d'exploitation peuvent vous obliger à définir un nombre moins élevé de devices. Reportez-vous à la présentation de la commande `sp_configure`, servant à modifier les paramètres de configuration, au chapitre 5, "Définition des paramètres de configuration".

Spécification de la taille du device à l'aide de *disk init*

Vous pouvez utiliser les spécificateurs d'unité suivants pour indiquer la taille du device : 'k' ou 'K' indique les kilo-octets, 'm' ou 'M' les méga-octets et 'g' ou 'G' les giga-octets. Bien que ce soit facultatif, Sybase recommande d'insérer systématiquement le spécificateur d'unité dans les commandes *disk init* et *create database* afin d'éviter toute confusion dans le nombre de pages réelles allouées. Ce spécificateur doit figurer entre apostrophes ou guillemets.

Les points suivants s'appliquent à la syntaxe de *disk init* :

- Vous pouvez indiquer la taille comme valeur flottante ou l'arrondir à la valeur entière la plus proche. Par exemple, si vous indiquez une taille de 3,75 Go, celle-ci est arrondie à 3 Go.
- Si vous ne spécifiez pas de taille :
 - Les commandes *disk init* et *disk reinit* utilisent la taille de base de la page disque de 2 ko.
 - L'argument de taille des commandes *create database* et *alter database* est exprimé en méga-octets de partie de disque. Cette valeur est convertie en nombre correspondant à la taille de la page logique avec lequel le device master a été construit.
 - Taille minimale d'une base de données. Vous ne pouvez pas modifier la taille d'un device de base de données après avoir exécuté la commande *disk init*.
 - Si vous envisagez d'utiliser le nouveau device pour créer une base de données, la valeur minimale de *size* dépend de la taille de page logique utilisée par le serveur, décrite dans tableau 16-1 :

Tableau 16-1 : Taille minimale de la base de données

Taille de la page logique	Taille minimale de la base de données
2 ko	2 méga-octets
4 ko	4 méga-octets
8 ko	8 méga-octets
16 ko	16 méga-octets

Si vous initialisez un device de base de données pour un journal de transactions ou afin de stocker des tables ou des index de petite taille sur un segment, la valeur de *size* peut n'être que de 512 blocs (1 Mo).

Si vous initialisez un device non formaté, déterminez la taille du device à partir de votre système d'exploitation, en suivant les instructions de la Guide d'installation pour votre plate-forme. Utilisez la taille totale disponible, jusqu'à une valeur maximale pour votre plate-forme. Lorsque le disque est initialisé à l'usage d'Adaptive Server, vous ne pouvez pas utiliser cet espace disque à d'autres fins.

disk init utilise *size* pour calculer la valeur du numéro de page virtuel supérieur dans *sysdevices.high*.

Avertissement ! Si le device physique ne contient pas le nombre de blocs spécifié par le paramètre *size*, la commande *disk init* échoue. Si vous utilisez le paramètre facultatif *vstart*, le device physique doit contenir la somme des blocs spécifiée par les paramètres *vstart* et *size* ; sinon, la commande échoue.

Spécification du paramètre *dsync* à l'aide de *disk init* (facultatif)

Pour les devices initialisés dans des fichiers du système d'exploitation UNIX, le paramètre *dsync* contrôle si les écritures d'accès à ces fichiers sont ou non stockées dans le buffer. Lorsque le paramètre *dsync* est activé, Adaptive Server ouvre un fichier de devices de base de données à l'aide du drapeau *dsync* UNIX. Le drapeau *dsync* garantit que les écritures d'accès au fichier de devices se produisent directement sur le support de stockage physique et Adaptive Server peut restaurer les données sur le device en cas de panne système.

Lorsque *dsync* est désactivé, les écritures d'accès au fichier de devices peuvent être stockées dans le buffer par le système de fichiers UNIX et la restauration des données sur le device ne peut pas être assurée. Le paramètre *dsync* ne doit être désactivé que dans le cas où l'intégrité des données n'est pas requise ou lorsque l'administrateur système exige des performances et un fonctionnement similaires aux versions antérieures d'Adaptive Server.

Remarque Le paramètre *dsync* est ignoré dans le cas de devices initialisés sur des partitions de disque et dans des fichiers Windows NT. Dans les deux cas, les écritures d'accès au device de la base de données prennent place directement sur le support physique.

Conséquences des performances de *dsync*

L'utilisation de *dsync* avec les fichiers de devices de base de données implique les concessions de performances suivantes :

- HP-UX et Digital UNIX ne supportent pas les E/S asynchrones dans les fichiers du système d'exploitation. Si les fichiers de devices de base de données sur ces plates-formes utilisent l'option *dsync*, le moteur Adaptive Server qui émet les écritures d'accès au fichier de devices se bloque jusqu'à ce que l'opération d'écriture soit arrivée à son terme. Cela peut entraîner un ralentissement des performances au cours de l'opération de mise à jour.
- Lorsque *dsync* est activé, il peut y avoir ralentissement des opérations d'écriture d'accès aux fichiers de devices de base de données, par rapport aux versions antérieures d'Adaptive Server (là où *dsync* n'est pas supporté). En effet, Adaptive Server doit écrire les données sur le disque au lieu de se contenter de copier les données du cache dans le buffer du système de fichiers UNIX.

Dans le cas où des performances d'écriture plus élevées sont requises (mais où l'intégrité des données après une panne système n'est pas requise), la désactivation de *dsync* entraîne des performances au niveau du fichier de devices identiques à celles des précédentes versions d'Adaptive Server. Par exemple, vous pouvez envisager de stocker *tempdb* dans un fichier de devices dédié avec l'option *dsync* désactivée, si les performances ne sont pas acceptables lors de l'utilisation de *dsync*.

- Le temps de réponse pour les opérations de lecture est généralement meilleur pour les devices stockés dans les fichiers du système d'exploitation UNIX, par rapport aux devices stockés sur les partitions de disque. Les données issues des fichiers de devices peuvent profiter du cache du système de fichiers UNIX ainsi que du cache d'Adaptive Server et d'autres lectures sont alors possibles sans qu'un accès au disque physique ne soit requis.
- La commande *disk init* requiert plus de temps pour achever l'opération avec les versions précédentes d'Adaptive Server car l'espace disque requis est alloué au cours de la phase d'initialisation du device.

Limites et restrictions de *dsync*

Les limites et restrictions suivantes s'appliquent lors de l'utilisation du paramètre *dsync* :

- La valeur définie pour *dsync* est toujours "vrai" pour le fichier de devices master. Vous ne pouvez pas modifier le paramètre *dsync* pour le device master. Si vous tentez de désactiver *dsync* pour le device master, Adaptive Server affiche un message d'avertissement.
- Si vous modifiez un paramètre *dsync* de fichier de devices à l'aide de la procédure *sp_deviceattr*, vous devez réinitialiser Adaptive Server pour que les modifications prennent effet.
- Lors de la mise à niveau d'une version précédente d'Adaptive Server à partir de la version 12.x, la valeur définie pour *dsync* est "vrai" pour le seul fichier de devices master. Vous devez utiliser la procédure *sp_deviceattr* pour modifier le paramètre *dsync* pour tout autre fichier de devices.
- Adaptive Server ignore le paramètre *dsync* pour les devices de base de données stockés sur des partitions de disque. Les écritures d'accès aux devices stockés sur des partitions de disque se font toujours directement sur le support physique.
- Adaptive Server ignore également le paramètre *dsync* pour les devices de base de données stockés dans les fichiers du système d'exploitation Windows NT. Adaptive Server sous Windows NT utilise automatiquement une fonctionnalité équivalente à *dsync* pour tous les fichiers de devices de base de données.

Autres paramètres facultatifs pour *disk init*

vstart constitue l'adresse virtuelle de départ ou l'offset permettant à Adaptive Server de commencer à utiliser le device de base de données. *vstart* accepte les spécificateurs d'unité facultatifs suivants : 'k' ou 'K' (kilo-octets), 'm' ou 'M' (méga-octets) et 'g' ou 'G' (giga-octets). La taille de l'offset dépend de la façon dont vous avez tapé la valeur de *vstart*.

- Si vous ne spécifiez pas de taille d'unité, *vstart* utilise des pages de 2 ko pour son adresse de départ. Par exemple, si vous spécifiez *vstart* = 13, Adaptive Server utilise des pages de 13 * 2 ko comme valeur d'offset pour l'adresse de départ.

- Si vous spécifiez une taille d'unité, vstart s'en sert comme adresse de départ. Par exemple, si vous spécifiez vstart = "13M", Adaptive Server définit l'offset de l'adresse de départ à 13 méga-octets.

La valeur par défaut (qui est également la valeur conseillée) de vstart est 0. Si les devices spécifiés ne contiennent pas la somme des blocs vstart + size disponible, la commande disk init échoue.

Le mot-clé cntrltype facultatif spécifie le contrôleur de disques. La valeur par défaut est 0. Réinitialisez-le uniquement si vous y êtes invité.

Remarque Pour effectuer l'initialisation du disque, l'utilisateur ayant démarré Adaptive Server doit disposer des autorisations appropriées du système d'exploitation sur le device en cours d'initialisation.

Obtention d'informations relatives aux devices

La procédure système sp_helpdevice fournit des informations sur les devices dans la table sysdevices.

Lorsque sp_helpdevice est utilisée sans nom de device, elle répertorie tous les devices disponibles sur Adaptive Server. Utilisée avec le nom d'un device, elle donne des informations sur ce device. Ici, sp_helpdevice donne des informations sur le device master :

```

                                sp_helpdevice master
device_name  physical_name  description
-----
master      d_master      special, default disk, physical disk, 20 MB

status      cntrltype    device_number    low    high
-----
3           0             0                0      9999
    
```

Chaque ligne de master.sysdevices décrit :

- un device de sauvegarde (bande, disque ou fichier) pour la sauvegarde des bases de données ou,
- un device de base de données pour le stockage des bases de données.

Le contenu initial de sysdevices est déterminé par le système d'exploitation. Les entrées de sysdevices sont généralement les suivantes :

- une entrée pour le device master ;
- une entrée pour la base de données sybssystemprocs que vous pouvez utiliser pour stocker des bases de données supplémentaires, comme pubs2 et sybsyntax, ou pour les bases de données et les journaux utilisateur ;
- deux entrées pour les devices de sauvegarde de type bande.

Si vous avez installé le système d'audit, vous trouverez également un device distinct pour sybsecurity.

Les champs low et high représentent les numéros de page attribués au device. Pour les devices de sauvegarde, ils correspondent à la capacité du device.

Le champ status de sysdevices est un bitmap qui indique le type de device, si un device de type disque est utilisé comme device de stockage par défaut lorsque les utilisateurs exécutent une commande create ou alter database sans spécifier de device de base de données, d'informations sur la mise en miroir de disques et de paramètres dsync. Les bits d'état et leur signification sont répertoriés dans le tableau 16-2 :

Tableau 16-2 : Bits d'état dans sysdevices

Bit	Signification
1	Disque par défaut (peut être utilisé par toute commande create ou alter database ne spécifiant pas d'emplacement)
2	Disque physique
4	Disque logique (non utilisé)
8	En-tête à ignorer (utilisé avec les devices de sauvegarde de type bande)
16	Device de sauvegarde
32	Écritures en série
64	Device mis en miroir
128	Lectures mises en miroir
256	Miroir secondaire uniquement
512	Miroir activé
2048	Usage interne ; défini après disk unmirror, side = retain
4096	Le miroir du device principal doit être désactivé (utilisé en interne)
8192	Le miroir du device secondaire doit être désactivé (utilisé en interne)
16384	Le device de fichier UNIX utilise le paramètre dsync (l'écriture se passe directement sur le support physique)

Pour plus d'informations sur les devices de sauvegarde et `sp_addumpdevice`, reportez-vous au chapitre 26, "Elaboration d'un plan de sauvegarde et de reprise".

Suppression de devices

Pour supprimer des devices de base de données et de sauvegarde, utilisez la commande `sp_dropdevice`. La syntaxe est la suivante :

```
sp_dropdevice nom_logique
```

Vous ne pouvez pas supprimer un device utilisé par une base de données. Vous devez d'abord supprimer cette dernière.

La commande `sp_dropdevice` supprime le nom du device de `sysdevices`. Elle ne supprime pas le fichier du système d'exploitation : elle ne fait que le rendre inaccessible à Adaptive Server. Vous devez utiliser les commandes du système d'exploitation pour supprimer ce fichier après avoir utilisé `sp_dropdevice`.

Désignation des devices par défaut

Afin de créer une zone de devices de base de données par défaut qui sera utilisée par tous les utilisateurs d'Adaptive Server pour la création de bases de données, utilisez `sp_diskdefault` après l'initialisation des devices. La commande `sp_diskdefault` marque ces devices comme devices par défaut dans `sysdevices`. Lorsqu'un utilisateur crée une base de données (ou en modifie une) sans spécifier de device de base de données, une nouvelle quantité d'espace disque est allouée dans la zone de l'espace disque par défaut.

La syntaxe de la commande `sp_diskdefault` est la suivante :

```
sp_diskdefault nom_logique, {defaulton | defaultoff}
```

Vous utiliserez certainement l'option `defaultoff` pour supprimer le device master de la zone d'espace par défaut :

```
sp_diskdefault master, defaultoff
```

La commande ci-ci-dessous fait de sprocdev (le device contenant la base de données sybsystemprocs) un device par défaut :

```
sp_diskdefault sprocdev, defaulton
```

Adaptive Server peut comporter plusieurs devices par défaut. Ils sont utilisés dans l'ordre dans lequel ils apparaissent dans la table sysdevices (dans l'ordre alphabétique). Lorsque le premier device par défaut est rempli, le second est sélectionné, et ainsi de suite.

Remarque Après avoir initialisé un ensemble de devices de base de données, vous pouvez décider de les affecter à des bases de données ou à des objets de base de données particuliers plutôt que de les ajouter à la zone de devices par défaut. Par exemple, vous pouvez vouloir vous assurer qu'une table ne dépasse jamais la taille d'un device particulier.

Choix des devices par défaut et des autres devices

sp_diskdefault permet de prévoir avec précision l'utilisation de l'espace afin d'optimiser les performances et la reprise, tout en autorisant les utilisateurs à créer ou à modifier les bases de données.

Assurez-vous que les devices ci-dessous *ne sont pas* des devices par défaut :

- le device master (utilisez sp_diskdefault pour définir defaultoff après avoir ajouté des devices utilisateur) ;
- le device pour sybsecurity ;
- tout device dédié aux journaux ;
- les devices sur lesquels résident les bases de données à hautes performances, utilisant éventuellement des segments.

Vous pouvez utiliser le device contenant sybssystemprocs pour d'autres bases de données utilisateur.

Remarque Si vous utilisez la mise en miroir de disques ou de segments, choisissez avec précaution les devices que vous ajoutez à la liste par défaut à l'aide de `sp_diskdefault`. Dans la plupart des cas, les devices devant être mis en miroir ou les bases de données qui contiendront des objets placés sur des segments doivent allouer des devices de manière explicite, plutôt que d'être inclus dans le stockage par défaut.

Mise en miroir des devices de base de données

Ce chapitre décrit la création et la gestion des disques miroirs.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Présentation de la mise en miroir des disques	593
Choix des éléments à mettre en miroir	594
Commandes de mise en miroir des disques	598
Didacticiel de mise en miroir des disques	603

Présentation de la mise en miroir des disques

La **mise en miroir des disques** permet une reprise instantanée en cas de panne de support. La commande `disk mirror` entraîne la duplication d'un device de base de données d'Adaptive Server, c'est-à-dire que toutes les écritures du device sont dupliquées sur un device physique séparé. Même si un device échoue, l'autre possède une copie à jour de toutes les transactions.

En cas d'échec en lecture ou en écriture sur un device mis en miroir, Adaptive Server annule la mise en miroir de ce dernier et génère des messages d'erreur.

Choix des éléments à mettre en miroir

Lorsque vous mettez un device en miroir, vous devez prendre en compte les facteurs comme les coûts de temps d'arrêt système, l'impact sur les performances et le coût des supports de stockage. L'examen de ces problèmes permettra de décider si vous mettez en miroir les journaux de transactions, tous les devices d'un serveur ou seulement quelques-uns.

Remarque Vous ne pouvez pas mettre en miroir un device de sauvegarde.

Vous devez mettre en miroir tous les devices de base de données par défaut afin de protéger ces derniers de l'incidence d'une commande `create` ou `alter database` sur l'un des devices de base de données figurant dans la liste par défaut.

Vous devez non seulement mettre en miroir les devices de base de données, mais également placer les journaux de transactions qui leur sont associés sur un device de base de données distinct. Pour bénéficier d'une protection accrue, vous pouvez en outre mettre en miroir le device de base de données utilisé pour le stockage des journaux de transactions.

Pour stocker le journal de transactions d'une base de données (c'est-à-dire, la table système `syslogs`) sur un autre device que le reste de la base, indiquez le nom du device de base de données et du device de journal lors de la création de la base de données. Vous pouvez également modifier la base de données en lui ajoutant un second device, puis exécuter la procédure système `sp_logdevice`.

Les trois exemples suivants impliquent des coûts et des compromis différents en matière de performances :

- *Rapidité de la reprise* – vous pouvez effectuer une reprise instantanée lorsque la base master et les bases de données utilisateur (avec les journaux) sont mises en miroir. La restauration des journaux de transactions est alors inutile.
- *Espace de stockage* – une reprise immédiate nécessite une redondance totale (toutes les bases de données et tous les journaux sont mis en miroir), ce qui occupe beaucoup d'espace disque.
- *Impacts sur les performances* – la mise en miroir des bases de données utilisateur (figure 17-2 et figure 17-3) augmente le temps nécessaire pour l'écriture des transactions sur les deux disques.

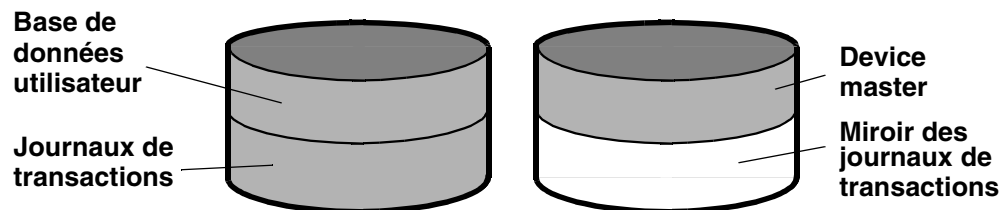
Mise en miroir avec un espace disque physique minimal

La figure 17-1 représente la "configuration minimale garantie" pour une restauration de base de données en cas de panne matérielle. Le device master et le miroir du journal de transactions de la base de données utilisateur sont stockés sur des partitions séparées sur un disque physique. L'autre disque stocke la base de données utilisateur et son journal de transactions sur deux partitions de disque séparées.

Si le disque comportant la base de données utilisateur tombe en panne, vous pouvez restaurer cette base sur un autre disque en utilisant vos sauvegardes et le journal de transactions mis en miroir.

Si le disque comportant le device master tombe en panne, vous pouvez restaurer le device master à partir d'une sauvegarde de la base de données master et remettre en miroir le journal de transactions de la base de données utilisateur.

Figure 17-1 : Mise en miroir avec un espace disque physique minimal



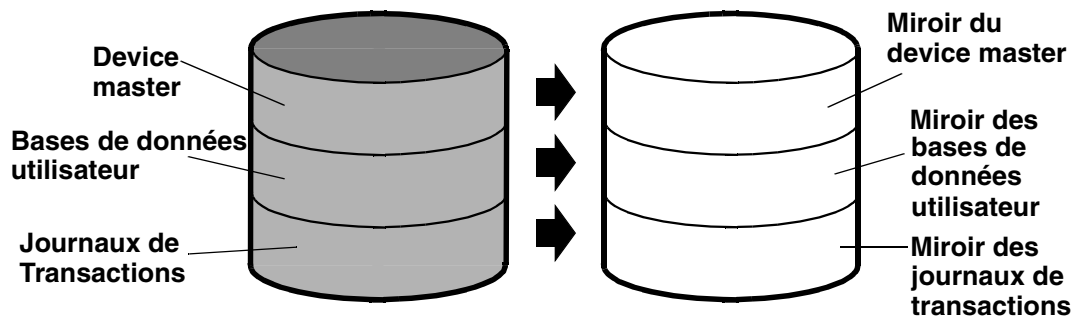
Cette configuration réduit au minimum l'espace de stockage disque nécessaire. Elle permet une reprise totale, même si le disque stockant la base de données utilisateur et le journal de transactions est endommagé, car le miroir de ce dernier le permet. Cependant la reprise instantanée n'est pas possible car la base master et les bases de données utilisateur ne sont pas mises en miroir et doivent être restaurées à partir des sauvegardes.

Mise en miroir pour une reprise instantanée

La figure 17-2 représente une autre configuration de mise en miroir. Dans ce cas, le device master, les bases de données utilisateur et le journal de transactions sont tous stockés sur des partitions différentes appartenant au même device physique et tous mis en miroir sur un deuxième device physique.

La configuration représentée sur la figure 17-2 permet une reprise instantanée en cas de panne matérielle. Les copies de travail de la base master, du journal et des bases de données utilisateur présentes sur le disque primaire sont toutes mises en miroir et la panne d'un des disques n'interrompt pas le travail des utilisateurs Adaptive Server.

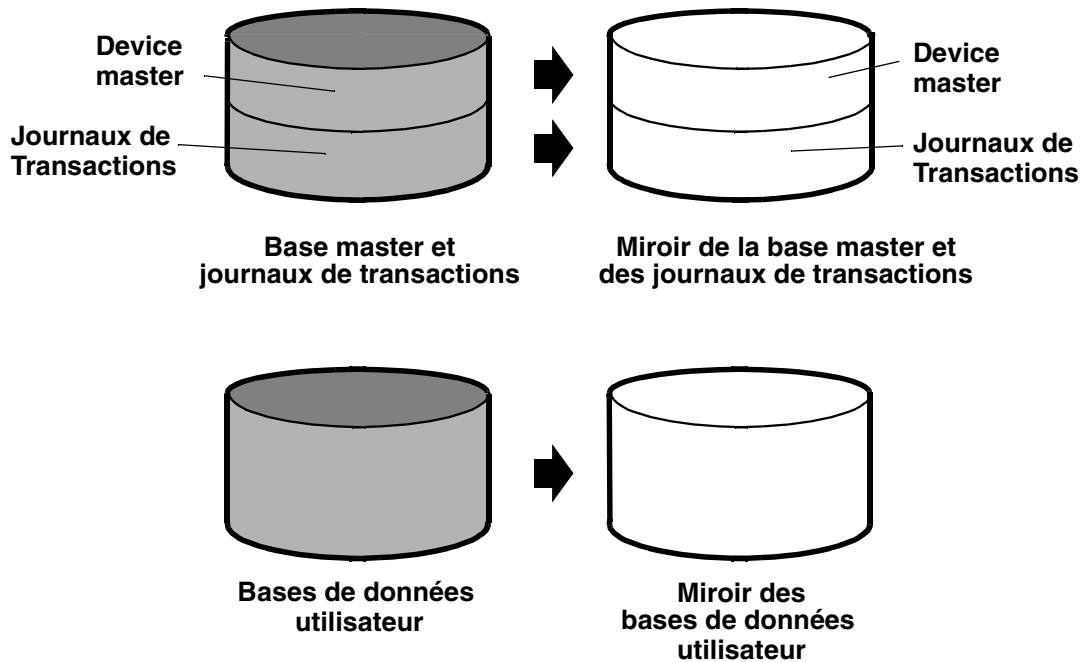
Figure 17-2 : Mise en miroir du disque pour une reprise rapide



Avec cette configuration, toutes les données sont écrites en double, une fois sur le disque primaire et une fois sur le miroir. Les applications nécessitant de nombreuses écritures peuvent devenir plus lentes à cause de la mise en miroir.

La figure 17-3 représente une configuration avec un haut niveau de redondance. Dans ce cas, les trois devices de base de données sont mis en miroir, mais la configuration nécessite quatre disques au lieu de deux. Cette configuration permet d'améliorer les performances pendant les transactions d'écriture car le journal de transactions de la base de données est stocké sur un device différent de celui des bases de données utilisateur et le système peut accéder aux deux avec un temps de positionnement des têtes de disque réduit.

Figure 17-3 : Mise en miroir des disques : journaux de transactions sur un disque séparé



Conditions ne désactivant pas la mise en miroir

Adaptive Server ne désactive un miroir que s'il trouve une erreur d'E/S sur un device mis en miroir. Par exemple, si Adaptive Server essaye d'écrire sur un bloc incorrect sur le disque, une erreur se produit qui désactive la mise en miroir pour le device. Cependant, le traitement n'est pas interrompu sur le device miroir.

Les conditions suivantes *ne* désactivent *pas* un miroir :

- Un bloc inutilisé sur un device est incorrect. Adaptive Server ne détecte pas d'erreur d'E/S et ne désactive pas la mise en miroir tant qu'il n'a pas accédé au bloc incorrect.

- Les données sur un device sont écrasées. Cela peut se produire si un device mis en miroir est monté comme un système de fichiers UNIX et qu'UNIX écrase les données Adaptive Server. Cette situation altère la base de données mais ne désactive pas la mise en miroir, car Adaptive Server ne trouve pas d'erreur d'E/S.
- Des données incorrectes sont écrites sur les devices primaire et secondaire.
- Les droits d'accès aux fichiers sont modifiés sur un device actif. Certains administrateurs système (SA) essayent de tester la mise en miroir des disques en modifiant les droits d'accès sur un device, espérant déclencher des erreurs d'E/S et désactiver la mise en miroir de l'autre device. Cependant, le système d'exploitation UNIX ne vérifie pas les droits d'accès sur un device après son ouverture ; il ne se produit donc pas d'erreur d'E/S tant que le device n'est pas de nouveau ouvert.

La mise en miroir des disques ne détecte pas et n'empêche pas l'altération de la base de données. Certains scénarios décrits peuvent altérer la base de données ; vous devez donc effectuer régulièrement des contrôles de cohérence sur toutes les bases de données, avec `dbcc checkalloc` et `dbcc checkdb`. Pour plus d'informations sur ces commandes, reportez-vous au chapitre 25, "Contrôle de la cohérence des bases de données".

Commandes de mise en miroir des disques

Les commandes `disk mirror`, `disk unmirror` et `disk remirror` gèrent la mise en miroir des disques. Toutes ces commandes peuvent être exécutées pendant que les devices sont actifs ; vous pouvez donc lancer ou arrêter une mise en miroir de device de base de données lors de l'utilisation des bases de données.

Remarque Les commandes `disk mirror`, `disk unmirror` et `disk remirror` modifient la table `sysdevices` de la base de données `master`. Après avoir exécuté une de ces commandes, sauvegardez la base de données `master` pour pouvoir la restaurer ultérieurement si elle est altérée.

Initialisation des miroirs

disk mirror lance la mise en miroir des disques. N'initialisez *pas* le device miroir avec disk init. Un device de base de données et son miroir constituent un device logique. La commande disk mirror permet d'ajouter le nom du miroir à la colonne mirrorname dans la table sysdevices.

Remarque Afin de pouvoir continuer à utiliser les E/S asynchrones, mettez toujours en miroir entre eux les devices pouvant effectuer des E/S asynchrones. Dans la plupart des cas, cela suppose une mise en miroir des partitions de disque entre elles et des systèmes d'exploitation entre eux. Si le système d'exploitation ne peut pas effectuer d'E/S asynchrones sur des fichiers, la mise en miroir d'une partition de disque vers un fichier normal génère un message d'erreur. Cette mise en miroir fonctionne mais n'utilise pas d'E/S asynchrones.

La syntaxe de disk mirror est la suivante :

```
disk mirror
  name = "nom_device" ,
  mirror = "nom_physique"
  [ , writes = { serial | noserial }]
```

nom_device est le nom du device que vous mettez en miroir, tel qu'il figure dans sysdevices.name (par disk init). Utilisez la clause mirror "*nom_physique*" pour spécifier le chemin d'accès du device mis en miroir, entre guillemets ou apostrophes. Si le device miroir est un fichier, "*nom_physique*" doit identifier sans ambiguïté le chemin d'accès sur lequel le fichier sera créé ; il ne peut pas s'agir d'un fichier existant du système d'exploitation.

Sur les systèmes supportant les E/S asynchrones, l'option writes permet de spécifier si les écritures sur le premier device doivent se terminer avant le début de celles sur le second device (serial) ou si les deux requêtes d'E/S doivent s'exécuter en parallèle, une sur chaque côté du miroir (noserial). Dans les deux cas, si une écriture n'est pas terminée, l'erreur d'E/S peut entraîner l'arrêt de la mise en miroir du device.

Les écritures serial (en série) sont la valeur par défaut. Les écritures sur les devices se font de manière consécutive, c'est-à-dire que les premières se terminent avant que les secondes ne débutent. Ces écritures constituent une garantie en cas de panne de courant. En effet, l'une des écritures peut être détériorée, mais rarement les deux. Généralement, les écritures serial sont plus lentes que les écritures noserial.

Dans l'exemple suivant, tranlog est le nom du device logique pour une partition de disque. Le device tranlog a été initialisé avec disk init et est utilisé comme un device de journal de transactions (comme dans create database...log on tranlog). La commande suivante met en miroir le device du journal de transactions :

```
disk mirror
  name = "tranlog",
  mirror = "/dev/rxyle"
```

Annulation de la mise en miroir d'un device

La mise en miroir est automatiquement désactivée lorsque l'un des deux devices physiques tombe en panne. En cas d'échec en lecture ou en écriture sur un device mis en miroir, Adaptive Server génère des messages d'erreur. Adaptive Server continue de fonctionner, sans mise en miroir. Pour relancer la mise en miroir, vous devez exécuter la commande disk remirror.

Utilisez la commande disk unmirror pour arrêter la mise en miroir pendant une maintenance sur le matériel.

```
disk unmirror
  name = "Nom_périphérique"
  [, side = { "primary" | secondary }]
  [, mode = { retain | remove }]
```

L'option side de la commande disk unmirror permet de spécifier le côté du miroir que vous voulez désactiver. primary (entre guillemets) est le device présent dans la colonne name de la table sysdevices ; secondary (sans guillemets) est le device présent dans la colonne mirrorname de la table sysdevices. secondary est la valeur par défaut.

L'option mode indique si l'annulation de la mise en miroir doit être temporaire (retain) ou permanente (remove). retain est la valeur par défaut.

Désactivation temporaire d'un device

Par défaut (mode=retain), Adaptive Server désactive temporairement le device spécifié pour vous permettre de le réactiver ultérieurement. Cette opération est similaire à ce qui se produit lorsqu'un device tombe en panne et qu'Adaptive Server active son miroir :

- Les E/S sont dirigées uniquement vers le device restant de la paire mise en miroir.

- La colonne status de sysdevices est modifiée pour indiquer que la fonction de mise en miroir a été désactivée.
- Les entrées correspondant aux disques primaires (phynome) et secondaire (mirrorname) demeurent inchangées.

Désactivation permanente d'un miroir

Utilisez mode=remove pour désactiver la mise en miroir. Cette option élimine des tables système toutes les références au device mis en miroir, mais *ne* supprime *pas* le fichier du système d'exploitation ayant été utilisé comme miroir.

Si vous spécifiez mode=remove :

- La colonne status est modifiée pour indiquer que la fonction de mise en miroir doit être ignorée.
- La colonne phynome est remplacée par le nom du device secondaire dans la colonne mirrorname si le device primaire est celui qui est désactivé.
- La colonne mirrorname est définie comme NULL.

Impact sur les tables système

L'option mode modifie la colonne status dans la table sysdevices, pour indiquer que la mise en miroir est désactivée (reportez-vous au tableau 16-2, page 588). Son influence sur les colonnes phynome et mirrorname de la table sysdevices dépend également de l'argument side, comme indiqué dans le tableau 17-1

Tableau 17-1 : Impact des options mode et side sur la commande disk mirror

		side	
		primary	secondary
mode	remove	Le nom dans mirrorname passe dans phynome et mirrorname est défini comme NULL ; status est modifié.	Le nom dans mirrorname est supprimé ; status est modifié.
	retain	Les noms restent inchangés ; status est modifié pour indiquer quel device est désactivé.	

L'exemple suivant interrompt le fonctionnement du device primaire :

```
disk unmirror
  name = "tranlog",
  side = "primary"
```

Relance de la mise en miroir

Utilisez la commande `disk remirror` pour relancer un processus de mise en miroir interrompu, à cause d'une panne de device ou par l'utilisation de `disk unmirror`. La syntaxe est la suivante :

```
disk remirror
  name = "nom_device"
```

Cette commande copie le device de base de données sur son miroir.

waitfor mirrorexit

Une panne de disque pouvant remettre en cause la sécurité du système, la commande `waitfor mirrorexit` peut être incluse dans une application pour effectuer des tâches spécifiques lorsque le disque n'est plus en miroir :

```
begin
  waitfor mirrorexit
  commandes à exécuter
end
```

Les commandes dépendent de vos applications. Vous pouvez ajouter des avertissements dans des applications effectuant des mises à jour ou utiliser `sp_dboption` pour rendre certaines bases de données accessibles en lecture seule uniquement si le disque ne fonctionne plus en miroir.

Remarque Adaptive Server n'est averti qu'un device n'est plus en miroir que lorsqu'il tente d'effectuer des E/S sur le device miroir. Pour les bases de données mises en miroir, cela se produit à un point de reprise ou lorsqu'un buffer Adaptive Server doit être écrit sur disque. Pour les journaux mis en miroir, les E/S apparaissent lorsqu'un processus écrit des opérations dans le journal, ceci incluant toute transaction validée effectuant des modifications de données, un point de reprise ou une sauvegarde de base de données.

waitfor mirroredit (et les messages d'erreur affichés sur la console et le journal d'erreurs en cas d'échec du miroir) ne sont activés que par ces événements.

Mise en miroir du device master

Si vous mettez en miroir le device contenant la base de données master dans un environnement UNIX, vous devez modifier le fichier runserver de votre Adaptive Server, pour que le device mis en miroir démarre lors de l'initialisation du serveur.

Sous UNIX, ajoutez l'argument -r et le nom du device mis en miroir :

```
dataserver -d /dev/rsdlf -r /dev/rs0e -e/sybase/install/errorlog
```

Pour obtenir des informations sur la mise en miroir du device master sous Windows NT, reportez-vous au *Guide Utilitaires*.

Obtention d'informations sur les devices et les miroirs

Pour obtenir des informations sur l'ensemble des devices Adaptive Server de votre système (les bases de données utilisateur et leurs miroirs, ainsi que les devices de sauvegarde), exécutez sp_helpdevice.

Didacticiel de mise en miroir des disques

Les étapes suivantes présentent l'utilisation des commandes de mise en miroir des disques et leurs conséquences sur certaines colonnes de master..sysdevices. Le numéro du status et son caractère hexadécimal équivalent pour chaque entrée de sysdevices figurent entre parenthèses :

Etape 1

Initialisez un nouveau device test à l'aide de :

```
disk init name = "test",  
physname = "/usr/sybase/test.dat",  
size=5120
```

Cette commande insère les valeurs suivantes dans les colonnes de master..sysdevices :

name	phyname	mirrorname	status
test	/usr/sybase/test.dat	NULL	16386

Status 16386 indique que le device est physique (2, 0x00000002) et que toutes les écritures se font vers un fichier UNIX (16384, 0x00004000). Puisque la colonne mirrorname est NULL, la mise en miroir n'est pas activée sur ce device.

Etape 2

Mettez en miroir le device test à l'aide de :

```
disk mirror name = "test",  
mirror = "/usr/sybase/test.mir"
```

Cette commande modifie les colonnes de master..sysdevices :

name	phyname	mirrorname	status
test	/usr/sybase/test.dat	/usr/sybase/test.mir	17122

Status 17122 indique que la mise en miroir est activée (512, 0x00000200) sur ce device. Les lectures sont mises en miroir (128, 0x00000080) ainsi que les écritures dans un device de fichier UNIX (16384, 0x00004000), le device est mis en miroir (64, 0x00000040) ainsi que les écritures serial (32, 0x00000020). Le device est un disque physique (2, 0x00000002).

Etape 3

Désactivez le device mis en miroir (côté secondaire), mais conservez ce miroir :

```
disk unmirror name = "test",  
side = secondary, mode = retain
```

name	phyname	mirrorname	status
test	/usr/sybase/test.dat	/usr/sybase/test.mir	18658

Status 18658 indique que le device est mis en miroir (64, 0x00000040) et que le device miroir est conservé (2048, 0x00000800), mais sa mise en miroir est désactivée (bit 512 désactivé) et seul le device primaire est utilisé (bit 256 désactivé). Les lectures sont mises en miroir (128, 0x00000080), les écritures sont mises en miroir dans un fichier UNIX (16384, 0x00004000) et serial (32, 0x00000020). Le device est un disque physique (2, 0x00000002).

Etape 4

Mettez à nouveau le device test en miroir :

```
disk remirror name = "test"
```

Cette commande modifie les colonnes de master..sysdevices :

name	phyname	mirrorname	status
test	/usr/sybase/test.dat	/usr/sybase/test.mir	17122

Status 17122 indique que la mise en miroir est activée (512, 0x00000200) sur ce device. Les lectures sont mises en miroir (128, 0x00000080) ainsi que les écritures dans un device de fichier UNIX (16384, 0x00004000), le device est mis en miroir (64, 0x00000040) ainsi que les écritures serial (32, 0x00000020). Le device est un disque physique (2, 0x00000002).

Etape 5

Désactivez le device test (côté primaire), mais conservez le miroir :

```
disk unmirror name = "test",
side = "primary", mode = retain
```

Cette commande modifie les colonnes de master.sysdevices :

name	phyname	mirrorname	status
test	/usr/sybase/test.dat	/usr/sybase/test.mir	16866

Status 16866 indique que le device est mis en miroir (64, 0x00000040) mais sa mise en miroir est désactivée (bit 512 désactivé) et seul le device secondaire est utilisé (256, 0x00000100). Les lectures sont mises en miroir (128, 0x00000080), les écritures sont mises en miroir dans un fichier UNIX (16384, 0x00004000) et sont serial (32, 0x00000020). Le device est un disque physique (2, 0x00000002).

Etape 6

Mettez à nouveau le device test en miroir :

```
disk remirror name = "test"
```

Cette commande modifie les colonnes de master.sysdevices :

name	phyname	mirrorname	status
test	/usr/sybase/test.dat	/usr/sybase/test.mir	17122

Status 17122 indique que la mise en miroir est activée (512, 0x00000200) sur ce device. Les lectures sont mises en miroir (128, 0x00000080) ainsi que les écritures dans un device de fichier UNIX (16384, 0x00004000), le device est mis en miroir (64, 0x00000040) ainsi que les écritures serial (32, 0x00000020). Le device est un disque physique (2, 0x00000002).

Etape 7

Désactivez le device test (côté primaire) et supprimez ce miroir :

```
disk unmirror name = "test", side = "primary",
mode = remove
```

Cette commande modifie les colonnes de master.sysdevices :

name	phyname	mirrorname	status
test	/usr/sybase/test.mir	NULL	16386

L'état 16386 indique que le device est physique (2, 0x00000002) et que toutes les écritures se font dans un fichier UNIX (16384, 0x00004000). Puisque la colonne mirrorname est NULL, la mise en miroir n'est pas activée sur ce device.

Étape 8

Supprimez ce device test pour mettre fin à ce didacticiel :

```
sp_dropdevice test
```

Cette commande supprime toutes les entrées du device test dans master..sysdevices.

Configuration de la mémoire

Ce chapitre explique comment Adaptive Server utilise la mémoire et comment maximiser la mémoire mise à la disposition d'Adaptive Server sur votre système.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Détermination de la mémoire disponible pour Adaptive Server	607
Utilisation de la mémoire par Adaptive Server	612
Procédures système de configuration de la mémoire	622
Principales utilisations de la mémoire d'Adaptive Server	627
Autres paramètres utilisant de la mémoire	634

Détermination de la mémoire disponible pour Adaptive Server

Plus la mémoire disponible est importante, plus Adaptive Server dispose de ressources pour les caches et buffers internes. Lorsque la mémoire disponible pour les caches est suffisante, Adaptive Server lit moins souvent les données du disque pour des informations ou des plans de procédures.

Lorsque vous configurez Adaptive Server pour qu'il utilise le maximum de mémoire mise à disposition sur votre ordinateur, les performances n'en sont pas diminuées. Toutefois, assurez-vous d'évaluer correctement les autres spécifications mémoire de votre système, puis configurez Adaptive Server pour qu'il n'utilise que la mémoire disponible restante. Si Adaptive Server ne peut pas obtenir la quantité de mémoire pour lequel il est configuré, il peut être dans l'impossibilité de démarrer.

Pour déterminer la quantité maximale de mémoire disponible pour Adaptive Server sur votre système, procédez comme suit :

- 1 Déterminez la quantité totale de mémoire physique sur votre système.
- 2 A ce total, soustrayez la mémoire requise pour le système d'exploitation.
- 3 Soustrayez la mémoire requise pour Backup Server, Monitor Server ou les autres logiciels liés à Adaptive Server qui doivent fonctionner sur la même machine.
- 4 Si la machine n'est pas dédiée à Adaptive Server, soustrayez également les spécifications mémoire des autres composants du système.

Par exemple, soustrayez la quantité de mémoire qui sera utilisée par les applications clientes s'exécutant sur la machine Adaptive Server. Les systèmes à multifenêtrage, tels que X Windows, nécessitent une quantité importante de mémoire et peuvent interférer avec les performances d'Adaptive Server lorsqu'ils sont utilisés sur la même machine qu'Adaptive Server.

La mémoire restante après ces soustractions relatives aux spécifications du système d'exploitation et aux autres applications correspond à la quantité totale de mémoire disponible pour Adaptive Server. La valeur du paramètre de configuration max memory indique la quantité maximale de mémoire sur laquelle Adaptive Server peut être configuré. Pour plus d'informations sur la configuration d'Adaptive Server pour utiliser cette mémoire, reportez-vous à la section "Paramètres de configuration concernant l'allocation de mémoire", page 615.

Allocation de la mémoire par Adaptive Server

Toutes les pages d'objets base de données sont dimensionnées selon la **taille de la page logique** indiquée lorsque vous créez un device master. toutes les bases de données (et tous les objets des différentes bases de données) utilisent la même taille de page logique. la taille des pages logiques d'Adaptive Server (2, 4, 8 ou 16 ko) détermine l'allocation de l'espace au sein du serveur. chaque page d'allocation, page de table d'allocation d'objets (oam), page de date, page d'index, page de texte et ainsi de suite, est construite sur une page logique. par exemple, si la taille de la page logique d'Adaptive Server est 8 ko, chaque type de page a une taille de 8 ko. toutes ces pages consomment la totalité de l'espace spécifié par la taille de la page logique. des pages logiques plus grandes vous permettent de créer des lignes plus grandes, ce qui augmente vos performances car Adaptive Server peut accéder à davantage de données chaque fois qu'il lit une page. par exemple, une page de 16 ko peut contenir huit fois plus de données qu'une page de 2 ko, une page de 8 ko, quatre fois plus qu'une page de 2 ko et ainsi de suite, pour toutes les tailles de pages logiques.

La taille des pages logiques est un paramètre global du serveur ; il est impossible d'avoir des pages logiques de tailles différentes au sein du même serveur. Toutes les tables ont des tailles appropriées de façon que la taille de la ligne ne soit pas plus grande que la taille de la page actuelle du serveur. Ceci signifie que les lignes ne peuvent pas s'étaler sur plusieurs pages.

Quelle que soit la taille des pages logiques pour laquelle il est configuré, Adaptive Server alloue de l'espace pour les objets (tables, indices, chaînes de page de texte) par extents, chacun étant composé de huit pages logiques. En d'autres termes, si un serveur est configuré pour des pages logiques de 2 ko, il alloue un extent, 16 ko, à chacun de ces objets. S'il est configuré pour des pages logiques de 16 ko, il alloue un extent, 128 ko, à chacun de ces objets.

Cela est vrai également pour les tables système. Si votre serveur contient un grand nombre de petites tables, la consommation d'espace peut être relativement importante si le serveur utilise des pages logiques plus grandes. Par exemple, pour un serveur configuré pour des pages logiques de 2 ko, systypes (avec environ 31 lignes courtes, un index clusterisé et un index non clusterisé) réserve trois extents ou 48 ko de mémoire. Si vous faites migrer le serveur pour qu'il utilise des pages de 8 ko, l'espace réservé à systypes correspond toujours à trois extents, soit 192 ko de mémoire. Pour un serveur configuré pour des pages logiques de 16 ko, systypes nécessite un espace disque de 384 ko. Pour les petites tables, l'espace inutilisé dans le dernier extent peut devenir significatif sur les serveurs utilisant des tailles de pages logiques plus importantes.

Les bases de données sont, elles aussi, concernées par les tailles de pages logiques. Chaque base de données comprend des catalogues système et leurs index. Si vous procédez à une migration d'une taille de page logique plus petite vers une taille plus importante, vous devez prendre en compte le volume de l'espace disque nécessaire à chaque base de données. Le tableau 18-1 répertorie la taille minimale pour une base de données avec les différentes tailles de pages logiques.

Tableau 18-1 : Tailles minimales des bases de données

Taille de page logique	Taille minimale de la base de données
2 ko	2 Mo
4 ko	4 Mo
8 ko	8 Mo
16 ko	16 Mo

Notez que la taille des pages logiques ne correspond pas à la taille des pages d'allocation de mémoire. La taille des pages d'allocation de mémoire prend toujours la valeur 2 ko, quelle que soit la taille des pages logiques qui peut elle être de 2 ko, 4 ko, 8 ko ou 16 ko. Le plus souvent, les paramètres de configuration associés à la mémoire utilisent des unités de 2 ko comme taille des pages de mémoire. Ces paramètres de configuration sont les suivants :

- max memory
- total logical memory
- total physical memory
- procedure cache size
- size of process object heap
- size of shared class heap
- size of global fixed heap

Allocation de l'espace disque

Notez que la taille des pages logiques ne correspond pas à la taille des pages d'allocation de mémoire. C'est l'unité dans laquelle l'espace disque est alloué et Adaptive Server a alloué des pages de 2 ko à cet espace. Certains paramètres de configuration utilisent cette taille de page de 2 ko comme unités d'allocation.

Tailles de pages logiques et buffers plus importants

Adaptive Server alloue des zones de buffer en unités de pages logiques. Par exemple, sur un serveur qui utilise des pages logiques de 2 ko, 8 Mo sont alloués au cache de données par défaut, ce qui constitue environ 2 048 buffers. Si vous avez alloué les mêmes 8 Mo au cache de données par défaut sur un serveur utilisant une taille de pages logiques de 16 ko, le cache de données par défaut représente approximativement 256 buffers. Sur un système connaissant un trafic important, la conséquence de ce petit nombre de buffers peut être que le buffer se trouve toujours dans la zone de vidage, ce qui entraîne un ralentissement pour les tâches exigeant des buffers propres. En général, pour obtenir les mêmes caractéristiques de gestion de buffer sur des tailles de pages supérieures à 2 ko, vous devez mettre à l'échelle la taille des caches afin d'atteindre une taille de page plus importante. Ainsi, si vous multipliez la taille des pages logiques par quatre, les tailles du cache et des zones doivent être quatre fois plus importantes.

Adaptive Server alloue généralement la mémoire de manière dynamique et pour le traitement des lignes lorsque cela est nécessaire, et attribue la taille maximale à ces buffers même si des buffers importants ne sont pas nécessaires. Ces demandes de gestion de mémoire peuvent entraîner des performances moindres au niveau d'Adaptive Server lors de la gestion de données de caractères étendus.

Mémoire segmentée

Une zone de mémoire segmentée est une zone de mémoire interne créée au démarrage et utilisée par des tâches pour allouer la mémoire de manière dynamique en fonction des besoins. Cette zone de mémoire est utilisée par des tâches qui nécessitent une quantité importante de mémoire dans la pile, comme les tâches qui utilisent des colonnes larges. Par exemple, si vous apportez une modification à une colonne ou à une ligne large, le buffer temporaire de cette tâche peut être supérieur à 16 ko, ce qui correspond à une quantité trop importante à allouer dans la pile. Adaptive Server alloue la mémoire de manière dynamique et la libère pendant l'exécution de la tâche. La zone de mémoire segmentée réduit considérablement la taille de la pile prédéclarée pour chaque tâche tout en améliorant l'efficacité de l'utilisation de la mémoire sur le serveur. La mémoire segmentée utilisée par la tâche est renvoyée à la zone de mémoire segmentée lorsque la tâche est terminée.

Définissez la mémoire segmentée avec le paramètre de configuration heap memory per user.

La taille de la zone de mémoire dépend du nombre de connexions utilisateur. Sybase vous recommande de définir la mémoire segmentée par utilisateur de sorte qu'elle corresponde à trois fois la taille de la page logique.

Utilisation de la mémoire par Adaptive Server

Dans Adaptive Server, la mémoire est exprimée sous forme de mémoire logique totale et de mémoire physique totale :

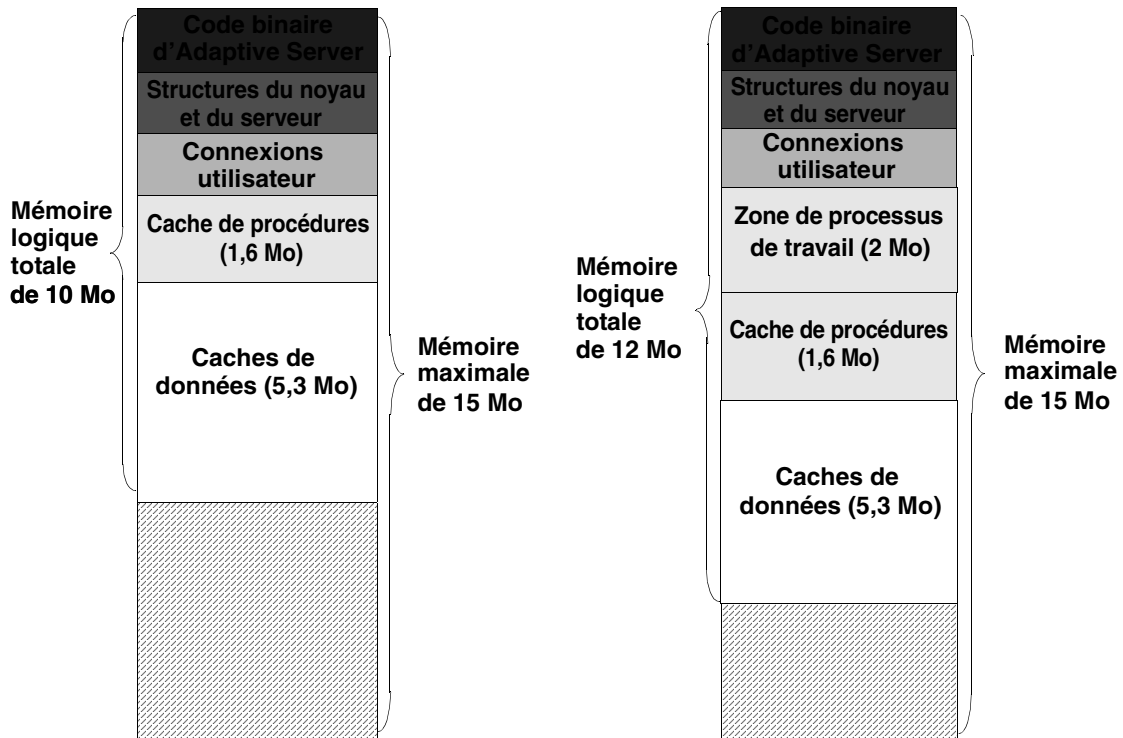
- La mémoire logique totale correspond au total de la mémoire nécessaire à tous les paramètres sp_configure. La mémoire logique totale doit être disponible, mais elle peut être ou ne pas être utilisée à un moment donné. La valeur de la mémoire logique totale peut varier en raison des modifications apportées aux valeurs des paramètres de configuration.
- La mémoire physique totale correspond au total de tous les segments de mémoire partagée dans Adaptive Server. En d'autres termes, la mémoire physique totale correspond à la quantité de mémoire utilisée par Adaptive Server à un moment donné. Vous pouvez vérifier cette valeur avec le paramètre de configuration en lecture seule total physical memory. La valeur de total physical memory ne peut qu'augmenter car Adaptive Server ne réduit pas les zones de mémoire une fois qu'elles sont allouées. Vous pouvez réduire la quantité de mémoire physique totale en modifiant les paramètres de configuration et en relançant Adaptive Server.

Au démarrage, Adaptive Server alloue de la mémoire pour les composants suivants :

- Mémoire utilisée par Adaptive Server pour les structures de données non configurables.
- Mémoire pour tous les paramètres configurables par l'utilisateur, y compris le cache de données, le cache de procédures et le cache de données par défaut.

La figure 18-1 illustre la manière dont Adaptive Server alloue la mémoire lorsque vous modifiez des paramètres de configuration de mémoire :

Figure 18-1 : Gestion des modifications de la configuration de mémoire par Adaptive Server



En cas d'ajout d'une zone de processus de travail de 2 Mo à la configuration de la mémoire d'Adaptive Server, les caches de procédures et de données conservent leurs tailles configurées initiales, respectivement de 1,6 Mo et 5,3 Mo. Dans la mesure où la taille de max memory est supérieure de 5 Mo à la taille de total logical memory, elle absorbe facilement la zone de mémoire ajoutée. Si la nouvelle zone de processus de travail porte la taille du serveur au-delà de la limite de max memory, toute commande transmise en vue d'augmenter la zone du processus de travail échoue. Si c'est le cas, la mémoire logique totale nécessaire à la nouvelle configuration est indiquée dans le message d'échec de `sp_configure`. Définissez la valeur de max memory sur une valeur supérieure à celle du paramètre total logical memory requis par la nouvelle configuration. Réessayez ensuite votre demande `sp_configure`.

La taille des caches de données et de procédures par défaut influent de manière significative sur les performances générales. Pour obtenir des recommandations sur l'optimisation de la taille du cache de procédures, reportez-vous au chapitre 14, "Utilisation et performances de la mémoire", du document *Performances et optimisation*.

Mémoire nécessaire par Adaptive Server

La mémoire totale nécessaire au lancement d'Adaptive Server correspond au *total de tous les paramètres de configuration de mémoire* auquel s'ajoutent la *taille du cache de procédures* et la *taille du cache de buffers*, exprimées en nombres arrondis plutôt qu'en pourcentages. La taille du cache de procédures et la taille du cache de buffers *ne* dépendent *pas* de la mémoire totale configurée. Vous pouvez configurer indépendamment la taille du cache de procédures et la taille du cache de buffers. Pour obtenir des informations comme la taille totale de chaque cache, le nombre de zones de chaque cache, etc., utilisez `sp_cacheconfig`.

Pour déterminer la quantité de mémoire utilisée par Adaptive Server à un moment donné, utilisez `sp_configure`. Exemple :

```
1> sp_configure "total logical memory"
Parameter Name      Default      Memory Used Config Value Run Value
Unit                Type
-----
total logical memory      33792      127550      63775      63775
memory pages (2k)      read-only
```

La valeur de la colonne Memory Used (mémoire utilisée) est exprimée en kilo-octets, tandis que la valeur de la colonne Config Value (valeur configurée) est exprimée en pages de 2 ko.

La valeur config indique la mémoire logique totale utilisée par Adaptive Server à l'exécution. La colonne de la valeur run indique la mémoire logique totale utilisée par la configuration actuelle d'Adaptive Server. Si vous exécutez cet exemple, vous observerez un autre résultat car il n'existe pas deux versions d'Adaptive Server configurées exactement de la même manière.

Pour plus d'informations sur `sp_configure`, reportez-vous au *Manuel de Référence d'Adaptive Server Enterprise - Volume 3: Procédures*.

Si vous effectuez une mise à niveau

Si vous effectuez une mise à niveau vers la version 12.5 ou supérieure d'Adaptive Server, les valeurs de configuration de total logical memory, procedure cache percent, et min online engines des versions d'Adaptive Server inférieures à 12.5 servent à calculer les nouvelles valeurs de procedure cache size et de number of engines at startup. Adaptive Server calcule la taille du cache de données par défaut lors de la mise à niveau et écrit cette valeur dans le fichier de configuration. Si la taille calculée pour le cache de données ou pour le cache de procédures est inférieure à la taille par défaut, sa valeur par défaut est restaurée. Lors de la mise à niveau, max memory prend la valeur de total logical memory spécifiée dans le fichier de configuration.

Restaurez la valeur de max memory afin de vous conformer aux besoins en termes de ressources.

Vous pouvez utiliser l'option verify de sp_configure pour vérifier les modifications apportées au fichier de configuration sans avoir à relancer Adaptive Server. Respectez la syntaxe suivante :

```
sp_configure "configuration file", 0, "verify", "full_path_to_file"
```

Paramètres de configuration concernant l'allocation de mémoire

Lorsque vous définissez la configuration de la mémoire d'Adaptive Server, indiquez les différentes spécifications mémoire par une valeur absolue, à l'aide de sp_configure. Vous pouvez également spécifier la taille des caches de données et de procédures sous forme de valeur absolue.

Trois paramètres de configuration affectent la manière dont la mémoire est allouée. Ces paramètres sont les suivants :

max memory

Le paramètre de configuration max memory permet d'établir un réglage maximal pour la quantité de mémoire qui peut être allouée à Adaptive Server. Si vous définissez max memory sur une valeur légèrement plus élevée que la valeur nécessaire actuellement, vous mettez à la disposition d'Adaptive Server de la mémoire supplémentaire en cas d'augmentation de ses besoins.

allocate max shared memory

La manière dont Adaptive Server alloue la mémoire spécifiée par max memory dépend de la configuration de allocate max shared memory et de dynamic allocation on demand.

Le paramètre allocate max shared memory permet soit d'allouer toute la mémoire spécifiée par max memory au démarrage, soit d'allouer uniquement la mémoire nécessaire à la spécification de la mémoire logique totale lors du démarrage.

Sur certaines plates-formes, si le nombre de segments de mémoire partagée alloués à une application est supérieur à un nombre optimal spécifique à la plate-forme, il peut se produire une dégradation des performances. Si c'est le cas, définissez la mémoire maximale sur la quantité maximale disponible pour ASE. Définissez la mémoire partagée maximale sur un et redémarrez le serveur. Ainsi, toute la mémoire de la mémoire maximale est allouée par ASE au démarrage avec le nombre de segments le moins élevé.

Par exemple, si vous définissez allocate max shared memory sur 0 (valeur par défaut) et max memory sur 500 Mo, et que la configuration du serveur ne nécessite que 100 Mo de mémoire au démarrage, Adaptive Server alloue les 400 Mo restants en cas de besoin accru de mémoire. En revanche, si vous définissez allocate max shared memory sur 1, Adaptive Server alloue l'intégralité des 500 Mo au démarrage.

L'avantage d'allouer toute la mémoire au démarrage, en définissant allocate max shared memory sur 1, réside dans le fait qu'il n'y a pas de dégradation des performances lors du réajustement du serveur en fonction de la mémoire supplémentaire. Cependant, si vous ne prévoyez pas correctement l'augmentation de la mémoire et que max memory prend une valeur élevée, vous pouvez perdre de la mémoire physique. Dans la mesure où vous ne pouvez pas réduire les paramètres de configuration de mémoire de manière dynamique, il est important de prendre en compte les autres spécifications mémoire.

dynamic allocation on demand

Le paramètre dynamic allocation on demand permet de déterminer si les ressources mémoire sont allouées à la demande ou en cas de besoin. Si vous définissez dynamic allocation on demand sur 1, les variations de mémoire sont allouées en fonction des besoins. Si vous définissez ce paramètre sur 0, la mémoire configurée demandée dans la variation de configuration de mémoire est allouée lors de la reconfiguration de la mémoire.

Par exemple, si vous définissez la valeur de `dynamic allocation on demand` sur 1 et que vous donnez à `number of user connections` la valeur 1024, la mémoire logique totale correspond à 1024 multiplié par la quantité de mémoire par utilisateur. Si la quantité de mémoire par utilisateur est de 112 ko, la mémoire des connexions utilisateur est de 112 Mo (1024 x 112).

C'est la quantité maximale de mémoire que le paramètre de configuration `number of user connections` peut utiliser. En revanche, si seulement 500 utilisateurs sont connectés au serveur, la quantité de mémoire physique totale utilisée par le paramètre `number of user connections` prend la valeur 56 Mo (500 x 112).

A présent, considérez que la valeur de `dynamic allocation on demand` soit 0, lorsque vous donnez à `number of user connections` la valeur 1024, toutes les ressources des connexions utilisateur sont immédiatement configurées.

Le mieux serait d'organiser la mémoire d'Adaptive Server de sorte que la quantité de mémoire physique totale soit inférieure à la quantité de mémoire logique totale, soit moins que `max memory`. Cela est possible, en partie, en donnant à `dynamic allocation on demand` la valeur 1 et à `allocate max shared memory` la valeur 0.

Allocation dynamique de la mémoire

Adaptive Server alloue la mémoire physique de manière dynamique. Cela permet aux utilisateurs de modifier la configuration de la mémoire d'Adaptive Server sans avoir à redémarrer le serveur.

Remarque Adaptive Server alloue la mémoire de manière dynamique, mais ne la réduit pas de la même manière. Il est important d'évaluer avec exactitude les spécifications de votre système car il est possible que vous ayez à redémarrer votre serveur si vous réduisez les paramètres de configuration de la mémoire et que vous souhaitez libérer la mémoire physique utilisée auparavant. Pour plus d'informations, reportez-vous à la section "Réduction dynamique des paramètres de configuration de la mémoire", page 618.

La valeur du paramètre de configuration `max_memory` doit être modifiée dans les cas suivants :

- lorsque vous modifiez la quantité de mémoire vive (RAM) sur votre machine ;
- lorsque le mode d'utilisation de votre machine change ;
- lorsque la configuration échoue en raison de la valeur insuffisante de `max_memory`.

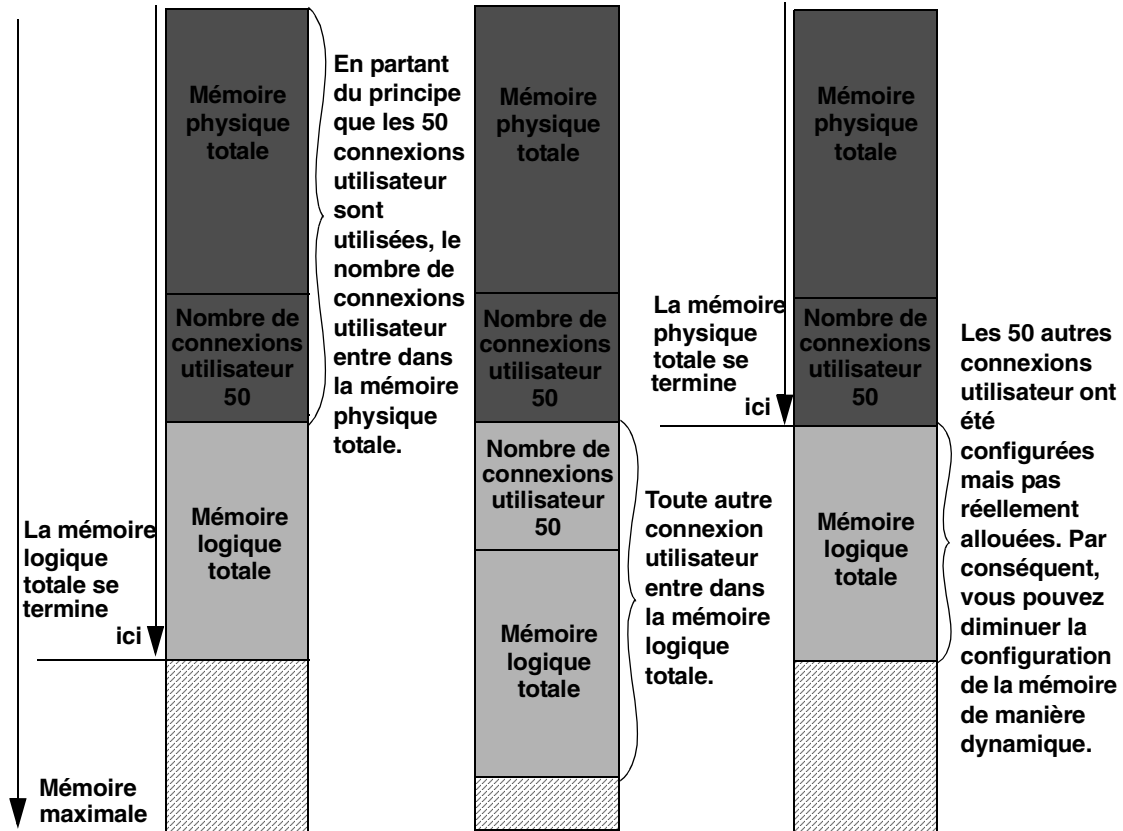
Démarrage impossible d'Adaptive Server

Lorsqu'Adaptive Server démarre, il doit occuper la quantité de mémoire spécifiée par le paramètre `total logical memory` dans le système d'exploitation. Si Adaptive Server ne peut pas démarrer car il ne parvient pas à occuper suffisamment de mémoire, diminuez les spécifications mémoire en réduisant les valeurs des paramètres de configuration qui utilisent de la mémoire. Vous pouvez également avoir besoin de réduire les valeurs des paramètres de configuration qui nécessitent des quantités importantes de mémoire. Ensuite, relancez Adaptive Server pour utiliser les nouvelles valeurs. Pour plus d'informations sur l'utilisation des fichiers de configuration, reportez-vous au chapitre 5, "Définition des paramètres de configuration",.

Réduction dynamique des paramètres de configuration de la mémoire

Si vous redéfinissez les paramètres de configuration sur une valeur inférieure, toute la mémoire utilisée n'est pas libérée de manière dynamique. Pour constater la réduction des modifications de configuration de la mémoire, reportez-vous aux figure 18-2 et figure 18-3.

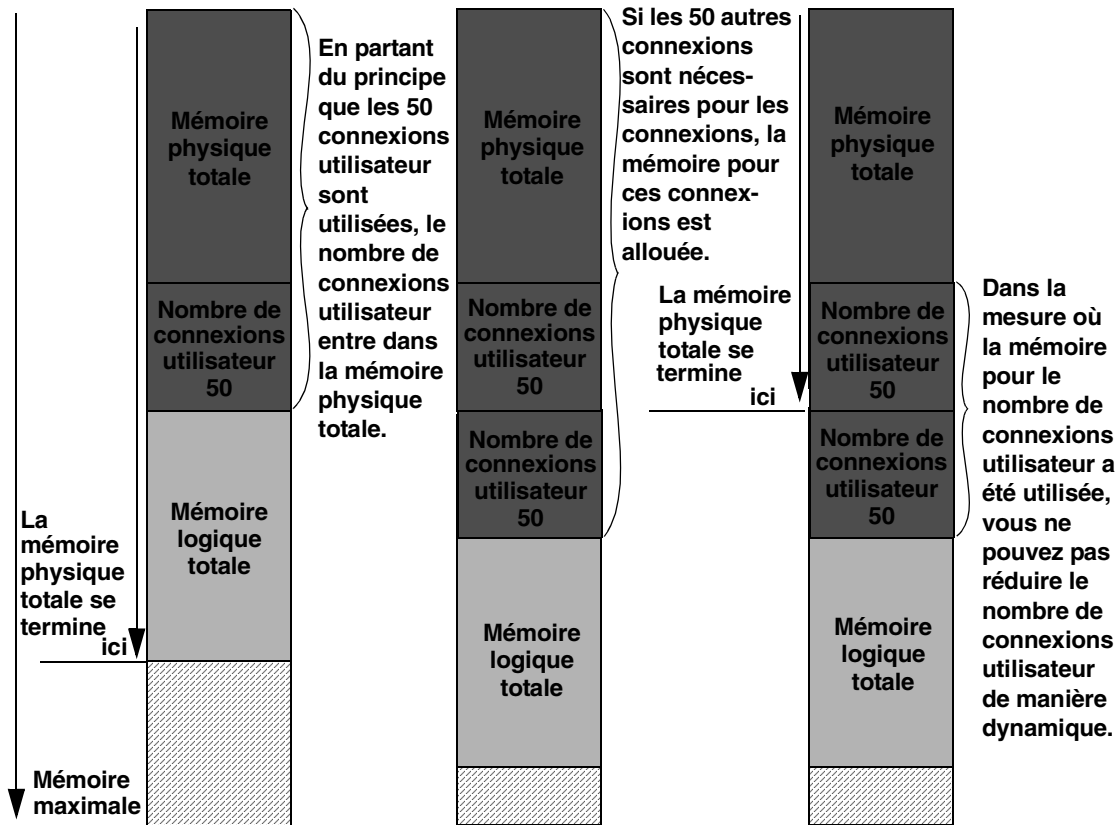
Figure 18-2 : Allocation dynamique à la demande définie sur 1 sans nouvelle connexion utilisateur



Sur la figure 18-2, comme le paramètre `dynamic allocation on demand` prend la valeur 1, la mémoire est dorénavant utilisée uniquement lorsqu'un événement déclenche un besoin d'utiliser de la mémoire supplémentaire. Dans cet exemple, cet événement peut prendre la forme, par exemple, d'une demande d'autres connexions utilisateur lorsqu'un client tente de se connecter à Adaptive Server.

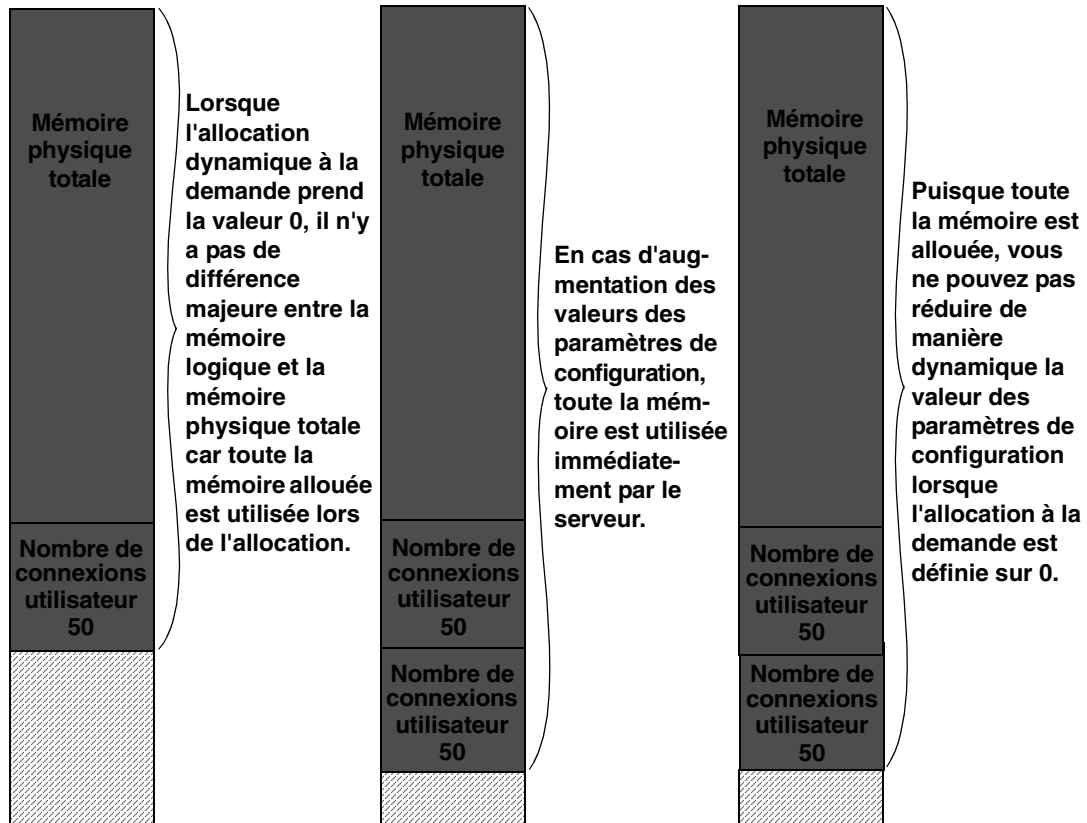
Vous pouvez réduire le paramètre `number of user connections` à un nombre supérieur ou égal au nombre de connexions utilisateur actuellement allouées car le paramètre `dynamic allocation on demand` prenant la valeur 1 et sans augmentation réelle de la demande de connexions utilisateur, le serveur ne demande pas de mémoire supplémentaire.

Figure 18-3 : Allocation dynamique à la demande définie sur 1 avec de nouvelles connexions utilisateur



La figure 18-3 part du principe que chacune des 50 autres connexions utilisateur est réellement utilisée. Vous ne pouvez pas réduire la valeur du paramètre number of user connections car la mémoire est utilisée. Vous pouvez utiliser sp_configure pour spécifier un changement des paramètres de configuration de la mémoire mais ce changement ne prendra effet qu'au redémarrage du serveur.

Figure 18-4 : Allocation dynamique à la demande définie sur 0



Remarque Théoriquement, lorsque le paramètre `dynamic allocation on demand` prend la valeur 0, il ne devrait pas y avoir de différence entre la mémoire logique totale et la mémoire physique totale. Cependant, il y a des écarts entre la manière dont Adaptive Server estime les spécifications mémoire et la manière dont la mémoire est effectivement requise pour l'utilisation. Pour cette raison, vous pouvez constater une différence entre les deux à l'exécution.

Lorsque le paramètre `dynamic allocation on demand` prend la valeur 0, toutes les spécifications mémoire configurées sont allouées immédiatement. Vous ne pouvez pas réduire la configuration de la mémoire de manière dynamique.

Sur les figure 18-3 et figure 18-4, les utilisateurs peuvent modifier la valeur des paramètres de configuration de la mémoire afin de leur donner une valeur inférieure et correcte. Même si cette modification ne prend pas effet de manière dynamique, elle interdit d'utiliser la nouvelle mémoire. Par exemple, si vous avez configuré le paramètre `number of user connections` pour permettre 100 connexions utilisateur et que vous portez cette valeur à 50 connexions utilisateur, dans les situations représentées par les figure 18-3 et figure 18-4, vous pouvez réduire la valeur du paramètre `number of user connections` à 50. Cette modification n'affecte pas la mémoire utilisée par Adaptive Server tant que le serveur n'a pas été redémarré mais elle empêche la connexion au serveur de tout nouvel utilisateur.

Procédures système de configuration de la mémoire

Les trois procédures système nécessaires pour configurer la mémoire d'Adaptive Server sont les suivantes :

- `sp_configure`
- `sp_helpconfig`
- `sp_monitorconfig`

Utilisation de `sp_configure` pour définir des paramètres de configuration

Pour plus d'informations sur la syntaxe complète et l'utilisation de la procédure `sp_configure`, ainsi que sur chaque paramètre de configuration, reportez-vous au chapitre 5, "Définition des paramètres de configuration". La suite de ce chapitre fournit des instructions pertinentes sur la configuration des paramètres qui utilisent la mémoire d'Adaptive Server.

Exécutez la procédure `sp_configure` en spécifiant le groupe "Memory Use" afin de visualiser ces paramètres sur votre serveur.

```
sp_configure "Memory Use"
```

La présence du signe "#" dans la colonne "Memory Used" indique que ce paramètre est un composant d'un autre paramètre et que la mémoire qu'il utilise est incluse dans celle de cet autre composant. Par exemple, comme la mémoire utilisée par stack size et stack guard size fait partie intégrante des spécifications mémoire de chaque connexion utilisateur et de chaque processus de production, la valeur correspondante est incluse dans la quantité de mémoire requise par number of user connections et par number of worker processes, si elle dépasse 200.

Certaines des valeurs de cette liste sont des valeurs calculées. Elles ne peuvent pas être définies directement à l'aide de la procédure sp_configure, mais elles sont mentionnées afin d'indiquer où est allouée la mémoire. Par exemple, total data cache size est une valeur calculée.

Mémoire disponible à augmenter de manière dynamique

L'exécution de la procédure sp_configure memory affiche tous les paramètres de mémoire et détermine la différence entre la mémoire maximale et la mémoire logique totale qui correspond à la quantité de mémoire disponible pour une augmentation dynamique. Exemple :

```

1> sp_configure memory
Msg 17411, Level 16, State 1:
Procédure 'sp_configure', Line 187:
Configuration option is not unique.
Parameter Name          Default      Memory Used Config Value Run Value
Unit                    Type
-----
additional network memory      0           0           0           0
bytes                      dynamic
allocate max shared memory    0           0           0           0
switch                      dynamic
heap memory per user          4096        0           4096        4096
bytes                      dynamic
lock shared memory            0           0           0           0
switch                      static
max memory                   33792       300000      150000      150000
memory pages (2k)            dynamic
memory alignment boundary    16384        0           16384       16384
bytes                      static
memory per worker process     1024         4           1024        1024
bytes                      dynamic
shared memory starting address 0            0           0           0
not applicable                static

```

Procédures système de configuration de la mémoire

total logical memory		33792	110994	55497	55497
memory pages(2k)	read-only				
total physical memory		0	97656	0	48828
memory pages(2k)	read-only				

189 006 ko de mémoire sont disponibles pour la reconfiguration. C'est la différence entre le paramètre 'max memory' et le paramètre 'total logical memory'.

Utilisation de *sp_helpconfig* pour obtenir de l'aide sur les paramètres de configuration

sp_helpconfig évalue la quantité de mémoire requise pour un paramètre de configuration et une valeur donnés. Elle fournit en outre une brève description du paramètre, des informations concernant les valeurs minimales, maximale et par défaut, la valeur d'exécution, ainsi que la quantité de mémoire utilisée à la valeur d'exécution courante. La procédure *sp_helpconfig* s'avère particulièrement utile lorsque vous planifiez de modifier considérablement un serveur, par exemple en chargeant des bases de données volumineuses à partir d'autres serveurs et que vous voulez évaluer la quantité de mémoire nécessaire.

Pour connaître la quantité de mémoire nécessaire à la configuration d'un paramètre, tapez tout ou partie du nom de ce paramètre pour l'identifier sans ambiguïté, suivi de la valeur que vous souhaitez configurer :

```
sp_helpconfig "worker processes", "50"
```

```
number of worker processes is the maximum number of worker processes that can be in use Server-wide at any one time.
```

Minimum Value	Maximum Value	Default Value	Current Value	Memory Used
Unit	Type			
0	2147483647	0	0	0
number	dynamic			

```
Configuration parameter, 'number of worker processes', will consume 7091K of memory if configured at 50.
```

```
Changing the value of 'number of worker processes' to '50' increases the amount of memory ASE uses by 7178 K.
```


Vous pouvez également utiliser la procédure `sp_helpconfig` pour déterminer la valeur à utiliser pour `sp_configure`, si vous connaissez la quantité de mémoire que vous souhaitez allouer à une ressource spécifique :

```
sp_helpconfig "user connections", "5M"
number of user connections sets the maximum number of user connections that can
be connected to SQL Server at one time.
Minimum Value Maximum Value Default Value Current Value Memory Used
Unit Type
-----
5 2147483647 25 25 3773
number dynamic
Configuration parameter, 'number of user connections', can be configured to 33
to fit in 5M of memory.
```

La principale différence entre la syntaxe de ces deux instructions réside dans l'utilisation, dans le seconde exemple, d'une unité de mesure indiquant à la procédure que la valeur est une taille et non une valeur de configuration. Les unités de mesure admises sont les suivantes :

- P : pages (pages Adaptive Server de 2 ko),
- K : kilo-octets,
- M : méga-octets,
- G : giga-octets.

Il arrive que la syntaxe ne soit pas pertinente pour le type de paramètre ou qu'Adaptive Server ne parvienne pas à calculer la quantité de mémoire utilisée. Dans ce genre de cas, la procédure `sp_helpconfig` génère un message d'erreur. Par exemple, si vous tentez de spécifier une taille pour un paramètre alternatif, comme `allow resource limits`, la procédure `sp_helpconfig` imprime le message qui décrit la fonction du paramètre pour tous les paramètres de configuration qui n'utilisent pas la mémoire.

Utilisation de *sp_monitorconfig* pour chercher des statistiques d'utilisation du cache de métadonnées

La procédure *sp_monitorconfig* affiche des statistiques concernant l'utilisation du cache de métadonnées sur certaines ressources serveur partagées, notamment :

- le nombre de bases de données, d'objets et d'index pouvant être ouverts simultanément ;
- le nombre de descripteurs de balayage auxiliaires utilisés par des requêtes d'intégrité référentielle ;
- le nombre de descripteurs disponibles et actifs ;
- le pourcentage de descripteurs actifs ;
- le nombre maximal de descripteurs utilisés depuis le démarrage du serveur ;
- la taille actuelle du cache de procédures et la quantité réellement utilisée.

Supposons, par exemple, que vous ayez affecté la valeur 500 au paramètre de configuration *number of open indexes*. En période de pointe, vous pouvez exécuter la procédure *sp_monitorconfig* de la manière suivante pour obtenir des statistiques précises sur l'utilisation réelle du cache de métadonnées pour les descripteurs d'index. Exemple :

```
sp_monitorconfig "number of open indexes"
Usage information at date and time: Aug 14 1997 8:54AM.
Name           # Free  # Active  % Active  # Max Ever Used  Re-used
-----
number of open 217    283     56.60    300             No
objects
```

Dans ce rapport, le nombre maximal d'index ouverts qui ont été utilisés depuis le dernier démarrage du serveur est de 300, bien qu'Adaptive Server soit configuré pour 500 index. Vous pouvez donc remplacer la valeur courante du paramètre de configuration *number of open indexes* par 330, afin de prendre en charge le maximum de 300 descripteurs d'index utilisés, plus 10 % d'espace supplémentaire.

Vous pouvez également déterminer la taille actuelle du cache de procédures avec le paramètre de taille du cache de procédures `sp_monitorconfig`. Ce paramètre décrit la quantité d'espace du cache de procédures actuellement configurée et l'espace maximal jamais utilisé. Par exemple, le cache de procédures sur le serveur suivant est configuré pour 20 000 pages :

```
1> sp_monitorconfig "procedure cache size"
option_name          config_value run_value
-----
procedure cache size          3271          3271
```

Cependant, lorsque vous exécutez la procédure `sp_monitorconfig "procedure cache size"`, vous découvrez que le cache de procédures n'a jamais utilisé plus de 14 241 pages. Par conséquent, vous pouvez réduire la valeur d'exécution du cache de procédures, ce qui permet d'économiser de la mémoire :

```
1> sp_monitorconfig "procedure cache size"
Usage information at date and time: May  1 2001 10:30AM.
Name                # Free      # Active    % Active    # Max Ever
Used  Re-used
-----
procedure cache size          5878          14122          70.61          14241
No
```

Principales utilisations de la mémoire d'Adaptive Server

Cette section décrit les paramètres de configuration qui utilisent une grande partie de la mémoire d'Adaptive Server, ainsi que ceux qui sont fréquemment modifiés sur de nombreuses installations Adaptive Server. Ces paramètres doivent être contrôlés par les administrateurs système qui configurent un Adaptive Server pour la première fois. De même, ils doivent être révisés chaque fois que la configuration du système change, qu'Adaptive Server est mis à niveau ou que les autres variables de configuration qui font appel à la mémoire sont modifiées.

Pour plus d'informations sur les paramètres de configuration qui utilisent le moins de mémoire ou qui sont le moins utilisés, reportez-vous à la section "Autres paramètres utilisant de la mémoire", page 634.

Adaptive Server code exécutable et overhead

La taille du code exécutable entre dans la valeur du paramètre total logical memory. Cette taille varie en fonction de la plate-forme et de la version, mais elle est généralement comprise entre 6 et 8 Mo. Afin de connaître la taille de l'exécutable et de l'overhead d'Adaptive Server pour votre plate-forme, utilisez la procédure `sp_configure` pour afficher la valeur du paramètre de configuration `executable codesize + overhead`. Pour plus d'informations, reportez-vous à la section "executable codesize + overhead", page 155.

Lorsque vous activez Component Integration Services avec le paramètre de configuration `enable cis` et que vous redémarrez ensuite Adaptive Server, la taille du code exécutable et celle de l'overhead augmentent. Les autres paramètres de configuration de Component Integration Services utilisent l'espace de la zone de mémoire générale.

Caches de données et de procédures

Comme il a été expliqué dans la section "Utilisation de la mémoire par Adaptive Server", page 612, vous spécifiez la taille des caches de données et de procédures. Le fait de réserver un espace mémoire suffisant aux caches de données et de procédures est l'un des principaux facteurs de performances. Cette section explique la différence entre les deux caches, ainsi que la manière de contrôler la taille des caches.

Détermination de la taille du cache de procédures

Le paramètre `procedure cache size` spécifie la taille du cache de procédures, exprimée en pages de 2 ko, quelle que soit la taille des pages logiques sur le serveur. Exemple :

```
sp_configure "procedure cache size"
1> sp_configure "procedure cache size"
Parameter Name          Default      Memory Used Config Value
Run Value   Unit          Type
-----
procedure cache size    3271        6914        3271
                        3271 memory pages(2k) dynamic
```

La quantité de mémoire utilisée par le cache de procédures est de 8,248 Mo. Pour affecter une autre taille au cache de procédures, exécutez la commande suivante :

```
sp_configure "procedure cache size", new_size
```

Cet exemple restaure la taille du cache de procédures à 10 000 pages de 2 ko (20 Mo) :

```
sp_configure "procedure cache size", 10000
```

Détermination de la taille du cache de données par défaut

Les procédures `sp_cacheconfig` et `sp_helpcache` affichent toutes deux le cache de données par défaut actuel, exprimé en méga-octets. Par exemple, le code ci-dessous présente une version d'Adaptive Server configurée avec un cache de données par défaut de 19,86 Mo :

```
sp_cacheconfig
```

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	0.00 Mb	19.86Mb
Total			0.00Mb	19.86 Mb

```

=====
Cache: default data cache, Status: Active, Type: Default
      Config Size: 0.00 Mb, Run Size: 19.86 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition: 1, Run Partition: 1
IO Size   Wash Size   Config Size   Run Size     APF Percent
-----
2 Kb      4066 Kb      0.00 Mb      19.86 Mb     10
    
```

Pour modifier le cache de données par défaut, exécutez la procédure `sp_cacheconfig` et spécifiez "default data cache". Par exemple, pour faire passer le cache de données par défaut à 25 Mo, tapez :

```
sp_cacheconfig "default data cache", "25M"
```

Pour que cette modification prenne effet, vous devez relancer Adaptive Server.

La taille du cache de données par défaut est une valeur absolue et la taille minimale est de 256 fois la taille des pages logiques. Pour 2 ko, la valeur minimale est de 512 ko et pour 16 ko, la valeur minimale est de 4 Mo. La valeur par défaut est de 8 Mo. Lors du processus de mise à niveau, Adaptive Server définit la taille du cache de données par défaut sur la valeur du cache de données par défaut contenue dans le fichier de configuration.

Contrôle de l'espace du cache

Vous pouvez contrôler l'espace disponible dans les caches de données et de procédures à l'aide de la procédure `sp_configure` :

```
sp_configure "total data cache size"
```

Contrôle de la taille des caches à l'aide du journal d'erreurs

Un autre moyen de déterminer la manière dont Adaptive Server utilise la mémoire consiste à consulter les messages relatifs à la mémoire dans le journal d'erreurs d'Adaptive Server, au démarrage de ce dernier. Ces messages indiquent précisément la quantité de mémoire allouée respectivement aux caches de données et de procédures, le nombre d'objets compilés pouvant résider simultanément dans le cache, ainsi que la taille de la zone de buffer.

Ces messages fournissent des informations très précises concernant l'allocation des caches sur Adaptive Server. Comme il a été expliqué précédemment, la quantité de mémoire allouée aux caches de procédures dépend de la valeur d'exécution du paramètre de configuration `procedure cache size`.

Chacun de ces messages est décrit ci-dessous.

Messages relatifs au cache de procédures

Deux messages fournissent des informations sur le cache de procédures dans le journal d'erreurs.

```
server: Number of proc buffers allocated: 556
```

Ce message indique le nombre total de buffers de procédures (proc buffers) alloués dans le cache de procédures.

```
server: Number of blocks left for proc headers: 629
```

Ce message indique le nombre total d'en-têtes de procédures (proc headers) disponibles dans le cache de procédures.

proc buffer

Un *buffer de procédures* est une structure de données utilisée pour gérer les objets compilés dans le cache de procédures. Un buffer de procédures est utilisé pour chaque exemplaire d'un objet compilé stocké dans le cache de procédures. Lorsqu'Adaptive Server démarre, il détermine le nombre de buffers de procédures requis et multiplie cette valeur par la taille d'un seul buffer (76 octets) pour obtenir la quantité totale de mémoire requise.

proc header

Un *en-tête de procédure* correspond au lieu de stockage d'un objet compilé dans le cache de procédures. En fonction de la taille de l'objet à stocker, un ou plusieurs en-têtes de procédures peuvent être requis. Le nombre total d'objets compilés pouvant être stockés dans le cache de procédures est limité par la valeur la plus faible entre le nombre d'en-têtes ou de buffers de procédures disponibles.

La taille totale du cache de procédures correspond à la somme de la mémoire totale allouée aux buffers de procédures (arrondie à la limite de page supérieure la plus proche) et de la mémoire allouée aux en-têtes de procédures.

Messages relatifs au cache de données

Au démarrage, Adaptive Server enregistre dans le journal d'erreurs la taille totale de chaque cache et celle de chacune des zones du cache. L'exemple suivant illustre le cache de données par défaut et un cache défini par l'utilisateur, comportant chacun deux zones de mémoire :

```
Memory allocated for the default data cache cache: 8030 Kb
Size of the 2K memory pool: 7006 Kb
Size of the 16K memory pool: 1024 Kb
Memory allocated for the tuncache cache: 1024 Kb
Size of the 2K memory pool: 512 Kb
Size of the 16K memory pool: 512 Kb
```

Connexions utilisateur

La quantité de mémoire requise par connexion utilisateur dépend de la plate-forme utilisée et elle change lorsque vous modifiez les autres variables de configuration, c'est-à-dire :

- default network packet size
- stack size et stack guard size
- user log cache size

La modification de ces paramètres entraîne la modification de la quantité d'espace utilisé par chaque connexion utilisateur : vous devez multiplier la différence de taille par le nombre de connexions utilisateur. Par exemple, en présence de 300 connexions utilisateur et si vous envisagez d'augmenter la valeur du paramètre stack size de 34 ko à 40 ko, la nouvelle valeur nécessite 1800 ko de mémoire en plus.

Bases de données, index et objets ouverts

Les trois paramètres de configuration qui contrôlent le nombre total de bases de données, d'index et d'objets pouvant être ouverts simultanément sont gérés par des caches spéciaux, appelés **caches de métadonnées**. Les caches de métadonnées résident dans la zone de mémoire d'Adaptive Server réservée aux structures du noyau et aux structures serveur. Pour configurer l'espace alloué à chacun de ces caches, utilisez les paramètres suivants :

- number of open databases
- number of open indexes
- number of open objects

Lorsqu'il ouvre une base de données ou qu'il accède à un index ou à un objet, Adaptive Server doit lire les informations qui s'y rapportent dans les tables système correspondantes : sysdatabases, sysindexes et sysobjects. Les caches de métadonnées associés aux bases de données, aux index et aux objets permettent à Adaptive Server d'accéder directement aux informations correspondantes sur la ligne de sysdatabases, sysindexes ou sysobjects de sa structure de mémoire interne. Ainsi, les performances sont améliorées car Adaptive Server évite des appels coûteux qui requièrent des accès disque. La synchronisation et le conflit de verrous sont réduits lorsqu'Adaptive Server doit extraire des informations de base de données, d'index ou d'objet lors de l'exécution.

La gestion de caches de métadonnées individuels pour les bases de données, les index ou les objets est profitable dans le cas d'une base de données contenant un grand nombre d'index et d'objets et faisant l'objet d'une forte concurrence d'accès entre les utilisateurs. Pour plus d'informations sur la configuration du nombre de caches de métadonnées, reportez-vous aux sections "number of open databases", page 155, "number of open indexes", page 158 et "number of open objects", page 159.

Nombre de verrous

Tous les processus Adaptive Server partagent une zone de structures de verrouillage. Lors d'une première estimation relative à la configuration du nombre de verrous, multipliez par 20 le nombre d'utilisateurs simultanés attendus *plus* la valeur du paramètre number of worker processes. Le nombre de verrous requis par les requêtes peut varier considérablement. Pour plus d'informations, reportez-vous à la section "number of locks", page 147. Pour plus d'informations sur l'utilisation de la mémoire par les processus de production, reportez-vous à la section "Processus de production", page 634.

Devices de base de données et structures d'E/S disque

Le paramètre de configuration number of devices contrôle le nombre de devices de base de données pouvant être utilisés par Adaptive Server pour le stockage des données. Pour plus d'informations, reportez-vous à la section "number of devices", page 122.

Lorsqu'un processus utilisateur a besoin d'exécuter une E/S physique, il la place en file d'attente dans une structure d'E/S disque. Pour plus d'informations, reportez-vous à la section "disk i/o structures", page 121.

Autres paramètres utilisant de la mémoire

Cette section décrit les paramètres de configuration qui utilisent une quantité de mémoire raisonnable ou qui sont rarement utilisés.

Traitement parallèle

Le traitement parallèle requiert davantage de mémoire que le traitement en série. Les paramètres de configuration qui influent sur le traitement parallèle sont les suivants :

- number of worker processes
- memory per worker process
- partition groups
- number of mailboxes et number of messages

Processus de production

Le paramètre de configuration number of worker processes définit le nombre total de processus de production disponibles simultanément sur Adaptive Server. Chaque processus de production requiert approximativement la même quantité de mémoire qu'une connexion utilisateur.

Le fait de changer l'un des paramètres suivants modifie la quantité de mémoire requise par chaque processus de production :

- default network packet size
- stack size et stack guard size
- user log cache size
- memory per worker process

Le paramètre de configuration memory per worker process contrôle la quantité de mémoire supplémentaire qui est placée dans une zone pour l'ensemble des processus de production. Cette mémoire supplémentaire stocke l'overhead des structures de données diverses et les buffers de communication interprocessus. Pour plus d'informations sur la définition du paramètre memory per worker process, reportez-vous au manuel *Performances et optimisation*.

Requêtes parallèles et cache de procédures

Chaque processus de production crée son propre exemplaire du plan d'exécution de requête dans l'espace emprunté au cache de procédures. De plus, le processus de coordination conserve en mémoire deux exemplaires du plan d'exécution de requête.

Groupes de partitions

Vous ne devez reconfigurer la valeur que si vous utilisez un nombre très important de partitions dans les tables de votre serveur. Pour plus d'informations, reportez-vous à la section "partition groups", page 222.

Serveurs distants

Certains des paramètres de configuration qui permettent à Adaptive Server de communiquer avec d'autres serveurs Sybase, tels que Backup Server, Component Integration Services ou Serveur XP font appel à la mémoire.

Les paramètres de configuration qui concernent les serveurs distants et qui font appel à la mémoire sont les suivants :

- number of remote sites
- number of remote connections
- number of remote logins
- remote server pre-read packets

Nombre de sites distants

Attribuez au paramètre de configuration number of remote sites une valeur correspondant au nombre de sites simultanés avec lesquels vous devez communiquer depuis votre serveur. Si vous utilisez uniquement Backup Server et aucun autre serveur distant, vous pouvez augmenter l'espace disponible dans vos caches de données et de procédures en affectant la valeur 1 à ce paramètre.

La connexion entre Adaptive Server et XP Server utilise un site distant unique.

Autres paramètres de configuration pour les appels de procédure à distance

Les paramètres suivants de configuration des communications distantes n'utilisent qu'une petite partie de la mémoire pour chaque connexion :

- number of remote connections
- number of remote logins

Chacune des connexions simultanées entre Adaptive Server et XP Server pour l'exécution des ESP utilise une connexion distante et un login distant.

Comme le paramètre remote server pre-read packets accroît l'espace requis par chacune des connexions spécifiées par le paramètre number of remote connections, l'augmentation du nombre de paquets prélués peut avoir un impact significatif sur l'utilisation de la mémoire.

Intégrité référentielle

Si les tables de votre base de données utilisent un grand nombre de contraintes référentielles et que le nombre de connexions utilisateur dépasse la valeur par défaut, vous pouvez être amené à ajuster le paramètre number of aux scan descriptors. Toutefois, la valeur par défaut est suffisante dans la plupart des cas. Si une connexion utilisateur dépasse la valeur courante du paramètre, Adaptive Server génère un message d'erreur qui vous conseille d'augmenter la valeur du paramètre number of aux scan descriptors.

Autres paramètres ayant une incidence sur la mémoire

Les autres paramètres de configuration ayant une incidence sur la mémoire sont répertoriés ci-dessous. Lorsque vous modifiez ces paramètres, vérifiez la quantité de mémoire qu'ils utilisent, ainsi que l'impact de cette modification sur vos caches de procédures et de données.

- | | |
|--------------------------------|-------------------------------|
| • additional network memory | • max SQL text monitored |
| • allow resource limits | • number of alarms |
| • audit queue size | • number of large i/o buffers |
| • event buffers per engine | • permission cache entries |
| • max number network listeners | |
| • max online engines | |

Configuration des caches de données

Ce chapitre décrit la procédure de création et d'administration de caches nommés sur Adaptive Server.

L'administration des caches de données vise principalement à les reconfigurer en vue d'améliorer les performances. Le présent chapitre s'intéresse surtout aux mécanismes de travail avec les caches de données. Le chapitre 14, "Utilisation et performances de la mémoire" du document *Performances et optimisation* aborde les concepts de performances des caches de données.

Sujet	Page
Le cache de données sur Adaptive Server	638
Commandes de configuration du cache	639
Informations sur les caches de données	641
Configuration des caches de données	643
Division d'un cache de données en zones mémoire	650
Liaison d'objets à des caches	654
Obtention d'informations sur les liaisons de caches	656
Suppression de liaisons de caches	659
Modification de la zone de vidage d'une zone mémoire	659
Modification de la limite de prélecture asynchrone pour une zone mémoire	663
Redéfinition de la taille des caches de données nommés	664
Suppression de caches de données	666
Ajout de partitions de cache	670
Suppression d'une zone mémoire	672
Incidence de la liaison de caches sur la mémoire et les plans de requêtes	673
Configuration de caches de données à l'aide du fichier de configuration	674
Conseils sur la configuration des caches	678
Erreurs de fichier de configuration	680

Le cache de données sur Adaptive Server

Le cache de données contient les pages de journal, d'index et de données actuellement utilisées par Adaptive Server, ainsi que les pages qu'il a récemment utilisées. Lors de sa première installation, Adaptive Server possède un cache de données par défaut, utilisé pour l'ensemble des activités portant sur les journaux, les index et les données. Sa taille par défaut est de 8 ko. La création d'autres caches ne réduit pas la taille du cache de données par défaut. Vous pouvez également créer des zones au sein des caches nommés et du cache par défaut en vue d'effectuer des E/S étendues. Vous pouvez ensuite lier une base de données, une table (y compris la table syslogs), un index ou une chaîne de pages de type texte ou image à un cache de données nommé.

Avec des tailles d'E/S étendues, Adaptive Server peut effectuer une prélecture des données lorsque l'optimiseur de requêtes détermine que les performances en seraient améliorées. Par exemple, une taille d'E/S de 128 ko sur un serveur configuré avec des pages logiques de 16 ko signifie qu'Adaptive Server peut lire un extent entier –8 pages–simultanément, au lieu d'effectuer 8 E/S distinctes. Pour plus d'informations sur l'optimiseur, reportez-vous au chapitre 19, "Outils d'optimisation de la requête" du document *Performances et optimisation*.

Les tris peuvent également tirer parti des zones de buffer configurées pour les tailles d'E/S étendues.

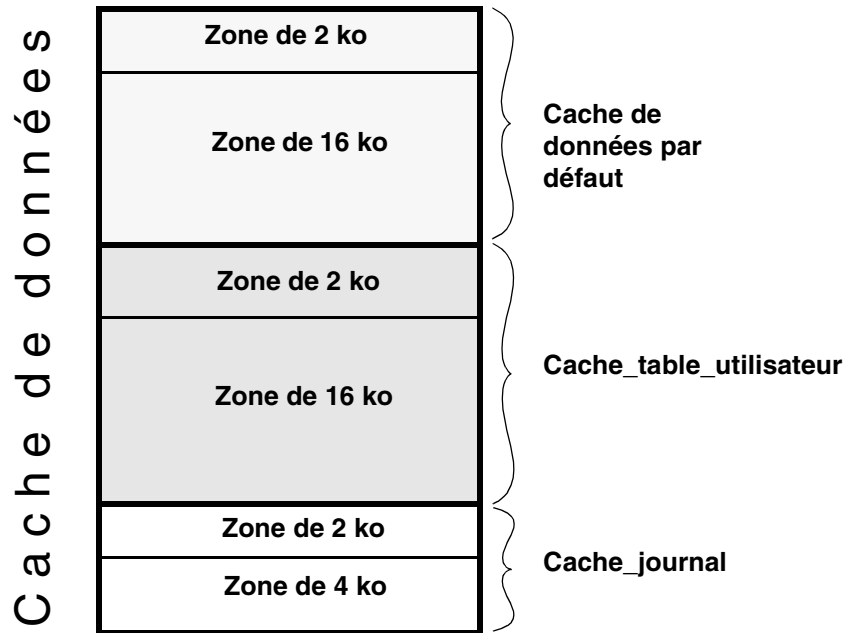
La configuration des caches de données nommés ne divise pas le cache par défaut en structures distinctes. Les caches de données nommés que vous créez ne peuvent être utilisés que par les bases de données ou les objets de base qui leur sont explicitement liés. Tous les objets qui ne sont pas liés de manière explicite aux caches de données nommés utilisent le cache de données par défaut.

Adaptive Server fournit des caches de données configurables par l'utilisateur afin d'améliorer les performances, notamment dans le cas de serveurs multiprocesseur. Reportez-vous à la section "Cache de données", page 325 du document *Performances et optimisation*.

La figure 19-1 présente un cache de données comportant le cache par défaut et deux caches de données nommés. Ce serveur utilise des pages logiques de 2 ko.

Le cache de données par défaut contient une zone de 2 ko et une zone de 16 ko. Le cache_table_utilisateur possède également une zone de 2 ko et une de 16 ko. Le cache_journal a une zone de 2 ko et une de 4 ko.

Figure 19-1 : Cache de données avec le cache par défaut et deux caches nommés



Commandes de configuration du cache

La table suivante répertorie les commandes de configuration des caches de données nommés, permettant de lier des objets à des caches et d'annuler ces liaisons, et de fournir des informations sur les liaisons de caches. Elle indique également les procédures permettant de contrôler la taille des objets de votre base de données et les commandes contrôlant l'utilisation du cache au niveau des sessions, des commandes ou des objets.

Commandes de configuration du cache

Commande	Fonction
sp_cacheconfig	Crée ou supprime des caches nommés et modifie la taille, le type, la police ou le nombre de partitions de cache.
sp_poolconfig	Crée et supprime des zones d'E/S et modifie leur taille, leur taille de vidage, ainsi que la limite de prélecture asynchrone en pourcentage.
sp_bindcache	Lie les bases de données ou les objets de bases de données à un cache.
sp_unbindcache	Annule la liaison entre des objets ou des bases de données et un cache.
sp_unbindcache_all	Annule la liaison entre tous les objets et un cache spécifié.
sp_helpcache	Fournit des informations sur les caches de données et répertorie les bases de données et les objets de base de données liés aux caches.
sp_cachestrategy	Indique les stratégies de cache définies pour une table ou un index et active ou désactive la prélecture ou la stratégie MRU.
sp_logiosize	Change la taille des E/S par défaut pour le journal.
sp_spaceused	Fournit des informations sur la taille des tables et des index ou sur l'espace utilisé dans une base de données.
sp_estspace	Évalue la taille des tables et des index, en fonction du nombre de lignes que la table contiendra.
sp_help	Indique le cache auquel une table est liée.
sp_helpindex	Indique le cache auquel un index est lié.
sp_helpdb	Indique le cache auquel une base de données est liée.
set showplan on	Fournit la taille des E/S et les stratégies d'utilisation des caches pour une requête.
set statistics io on	Indique le nombre de lectures effectuées pour une requête.
set prefetch [on off]	Active ou désactive la prélecture pour la durée d'une session.
select... (prefetch...lru mru)	Oblige le serveur à employer la taille d'E/S ou la stratégie de remplacement MRU spécifiée.

Non seulement on peut utiliser les commandes pour configurer les caches de données nommés de manière interactive, mais il est également possible de modifier le fichier de configuration situé dans le répertoire `$SYBASE`. Pour plus d'informations, reportez-vous à la section "Configuration de caches de données à l'aide du fichier de configuration", page 674.

Informations sur les caches de données

sp_cacheconfig permet de créer et de configurer des caches de données nommés. Lors de sa première installation, Adaptive Server possède un cache de données unique, nommé default data cache. Pour voir les informations relatives aux caches, tapez :

sp_cacheconfig

Les résultats de sp_cacheconfig ressemblent à ce qui suit :

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	0.00 Mb	59.44 Mb
Total			0.00 Mb	59.44 Mb

```

=====
Cache: default data cache,   Status: Active,   Type: Default
      Config Size: 0.00 Mb,   Run Size: 59.44 Mb
      Config Replacement: strict LRU,   Run Replacement: strict LRU
      Config Partition:      1,   Run Partition:      1
    
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	12174 Kb	0.00 Mb	59.44 Mb	10

Les informations récapitulatives pour chaque cache apparaissent dans un bloc en haut du rapport et s'achèvent avec la taille totale de tous les caches configurés. Pour chaque cache, un bloc d'informations indique la configuration des zones mémoire dans le cache.

Les colonnes sont les suivantes :

- "Cache Name" indique le nom du cache.
- "Status" indique si le cache est actif ou non. Les valeurs possibles sont :
 - "Pend/Act" : le cache vient d'être créé et sera actif après un redémarrage.
 - "Active" : le cache est actif.
 - "Pend/Del" : le cache est actif, mais il sera supprimé au prochain redémarrage du serveur. La taille du cache a été réinitialisée à 0 de manière interactive. Pour plus d'informations, reportez-vous à la section "Suppression de caches de données", page 666.

- "Type" indique si le cache peut stocker des pages de journal et de données ("Mixed") ou uniquement des pages de journal ("Log Only"). Seul le cache par défaut possède le type "Default". Vous ne pouvez pas modifier le type de default data cache, ni changer en "Default" le type d'un autre cache.
- "Config Value" affiche la taille du cache au prochain redémarrage d'Adaptive Server. Dans le résultat de l'exemple précédent, la taille de default data cache est de 0 car il n'a pas encore été configuré de manière explicite.
- "Run Value" affiche la taille utilisée actuellement par Adaptive Server. Dans le cas du default data cache, cette taille correspond toujours à l'intégralité de l'espace des caches de données qui n'a pas été explicitement configuré pour un autre cache.

Le deuxième bloc des résultats commence par trois lignes d'informations décrivant le cache. Les deux premières lignes reprennent les données du premier bloc. Sur la troisième ligne, "Config Replacement" et "Run Replacement" indiquent la stratégie du cache, qui est soit "strict LRU", soit "relaxed LRU". Le paramètre d'exécution est le paramètre actif ; si la règle a été modifiée depuis le dernier démarrage du serveur, le paramètre de configuration sera différent du paramètre d'exécution.

sp_cacheconfig fournit ensuite une ligne d'informations sur chaque zone du cache :

- "IO Size" indique la taille des buffers dans la zone. La taille par défaut de la zone est celle de la page logique du serveur. Lorsque vous configurez un cache pour la première fois, l'intégralité de l'espace est attribuée à la zone. Les tailles admises sont 2 ko, 4 ko, 8 ko et 16 ko.
- "Wash Size" indique la taille de la zone de vidage. Pour plus d'informations, reportez-vous à la section "Modification de la zone de vidage d'une zone mémoire", page 659.
- "Config Size" et "Run Size" indiquent respectivement la taille configurée et la taille actuellement employée. Ces valeurs sont différentes pour la zone car vous ne pouvez pas configurer sa taille de manière explicite. Elles peuvent également varier pour d'autres zones si vous avez tenté de transférer de l'espace entre elles et si une partie de l'espace n'a pas pu être libérée.

- "Config Partition" et "Run Partition" affichent le nombre configuré de partitions de cache et le nombre de partitions en cours d'utilisation. Ces nombres peuvent varier si vous avez modifié le nombre de partitions depuis la dernière réinitialisation.
- "APF Percent" affiche le pourcentage de la zone pouvant contenir les buffers inutilisés qui ont été récupérés lors d'une prélecture asynchrone.

Une ligne récapitulative indique la taille totale du ou des caches affichés.

Configuration des caches de données

Le default data cache et le cache de procédure d'Adaptive Server sont spécifiés à l'aide d'une valeur absolue. Lorsque vous planifiez et mettez en oeuvre la configuration des caches, la première étape consiste à définir le paramètre de configuration max memory. Après avoir défini max memory, déterminez la quantité d'espace que vous souhaitez allouer aux caches de données sur votre serveur. Bien que la taille d'un cache de données ne soit limitée que par l'accès à la mémoire sur le système, max memory doit être plus grand que total logical memory. Vous devez fournir une valeur absolue pour la taille de default data cache et tous les autres caches définis par l'utilisateur. Pour une présentation de l'utilisation de la mémoire d'Adaptive Server, reportez-vous au chapitre 18, "Configuration de la mémoire".

Vous pouvez configurer les caches de données de deux façons :

- de manière interactive avec sp_cacheconfig et sp_poolconfig,
- en modifiant votre fichier de configuration.

Les sections suivantes décrivent la procédure d'utilisation de sp_cacheconfig et de sp_poolconfig. Pour plus d'informations sur l'utilisation du fichier de configuration, reportez-vous à la section "Configuration de caches de données à l'aide du fichier de configuration", page 674.

Chaque fois que vous exécutez `sp_cacheconfig` ou `sp_poolconfig`, Adaptive Server inscrit les informations relatives au nouveau cache ou à la nouvelle zone dans le fichier de configuration, et copie l'ancienne version de ce fichier dans un fichier de sauvegarde. Un message indiquant le nom de ce fichier de sauvegarde est envoyé dans le journal d'erreurs.

La syntaxe de création d'un cache est la suivante :

```
sp_cacheconfig cache_name, "size [P|K|M|G] "
```

Vous pouvez spécifier les unités de taille suivantes :

- P – pages (taille de page logique d'Adaptive Server)
- K – kilooctets (par défaut)
- M – mégaoctets
- G – gigaoctets

La taille maximale d'un cache de données n'est limitée que par le volume de mémoire disponible sur votre système.

La commande suivante configure un cache de 10 Mo nommé `pubs_cache` :

```
sp_cacheconfig pubs_cache, "10M"
```

Cette commande apporte les modifications dans les tables système et écrit les nouvelles valeurs dans le fichier de configuration, mais elle n'active pas le cache. Pour que ces modifications soient prises en compte, vous devez redémarrer Adaptive Server.

Si vous exécutez `sp_cacheconfig` pour connaître la configuration avant de redémarrer le serveur, vous obtiendrez des valeurs différentes pour "Config" et "Run" :

```
sp_cacheconfig pubs_cache
```

Cache Name	Status	Type	Config Value	Run Value
pubs_cache	Pend/Act	Mixed	10.00 Mb	0.00 Mb
		Total	10.00 Mb	0.00 Mb

La mention "Pend/Act" dans la colonne Status indique que le cache `pubs_cache` sera configuré après un redémarrage. "Config Value" est à 10 Mo et "Run Value" à 0. Les valeurs de configuration et d'exécution sont également différentes lorsque vous supprimez des caches ou que vous modifiez leur taille.

La partie des résultats fournissant des détails sur les zones n'apparaît pas lorsque les caches ne sont pas actifs.

Après avoir redémarré Adaptive Server, sp_cacheconfig affiche les valeurs suivantes :

sp_cacheconfig

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	0.00 Mb	49.37 Mb
pubs_cache	Active	Mixed	10.00 Mb	10.00 Mb
Total			10.00 Mb	59.37 Mb

```

=====
Cache: default data cache, Status: Active, Type: Default
      Config Size: 0.00 Mb, Run Size: 49.37 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition: 1, Run Partition: 1
    
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	10110 Kb	0.00 Mb	49.37 Mb	10

```

=====
Cache: pubs_cache, Status: Active, Type: Mixed
      Config Size: 10.00 Mb, Run Size: 10.00 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition: 1, Run Partition: 1
    
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	2048 Kb	0.00 Mb	10.00 Mb	10

Le cache pubs_cache est désormais actif et l'espace est alloué à la zone. La taille du cache par défaut a été réduite de 10 Mo. Le reste de la différence entre la taille du cache par défaut et le volume total de cache disponible provient de la modification des valeurs d'overhead.

Voici le même exemple pour un serveur à pages logiques de 16 ko :

1> sp_cacheconfig pubs_cache

Cache Name	Status	Type	Config Value	Run Value
pubs_cache	Active	Mixed	10.00 Mb	10.00 Mb
Total			10.00 Mb	10.00 Mb

Configuration des caches de données

```
Cache: pubs_cache, Status: Active, Type: Mixed
      Config Size: 10.00 Mb, Run Size: 10.00 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition: 1, Run Partition: 1
```

```
IO Size  Wash Size  Config Size  Run Size      APF Percent
-----  -
16 Kb    2048 Kb        0.00 Mb      10.00 Mb      10
```

Pour obtenir des exemples, reportez-vous à la section "Incidence de l'overhead sur l'espace total du cache", page 658.

Vous pouvez créer autant de caches que vous le voulez avant de redémarrer Adaptive Server. Cependant, vous devez redémarrer Adaptive Server avant de configurer des zones ou lier des objets aux caches nouvellement créés.

Configuration explicite du cache par défaut

Vous devez configurer explicitement la taille du default data cache parce qu'il nécessite une valeur absolue. Exécutez `sp_helpcache` pour voir la quantité de mémoire que le cache peut encore utiliser. Exemple :

```
sp_helpcache
Cache Name          Config Size      Run Size        Overhead
-----
default data cache  25.00 Mb        25.00 Mb        0.22 Mb
pubs_cache          10.00 Mb        10.00 Mb        0.11 Mb
pubs_log            31.25 Mb        31.25 Mb        0.27 Mb

Memory Available For  Memory Configured
Named Caches         To Named Caches
-----
66.44 Mb             66.25 Mb

----- Cache Binding Information: -----

Cache Name  Entity Name Type  Index Name          Status
-----
```

Pour spécifier la taille absolue du default data cache, exécutez `sp_cacheconfig` avec `default data cache` et une valeur de taille. Cette commande fixe la taille du default data cache à 25 Mo :

```
sp_cacheconfig "default data cache", "25M"
```

Après le redémarrage du serveur, Config Value présente cette valeur.

sp_cacheconfig

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	25.00 Mb	49.37 Mb
pubs_cache	Active	Mixed	10.00 Mb	10.00 Mb
Total			10.00 Mb	59.37 Mb

```

=====
Cache: default data cache, Status: Active, Type: Default
      Config Size: 25.00 Mb, Run Size: 49.37 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition:      1, Run Partition:      1
  
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	10110 Kb	00.00 Mb	49.37 Mb	10

```

=====
Cache: pubs_cache, Status: Active, Type: Mixed
      Config Size: 10.00 Mb, Run Size: 10.00 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition:      1, Run Partition:      1
  
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	2048 Kb	0.00 Mb	10.00 Mb	10

Vous pouvez modifier la taille du default data cache avec sp_cacheconfig. Les changements apportés à la taille du default data cache sont sans incidence sur la taille d'un autre cache. De la même manière, une fois la taille du default data cache spécifiée, la configuration d'autres caches définis par l'utilisateur ne la modifie pas. Lors de la création d'un cache défini par l'utilisateur, la mémoire est prise dans max memory, sans modification de la taille du default data cache.

Remarque Si, après la configuration du default data cache, vous réduisez max memory à un niveau définissant total logical memory à une valeur supérieure à celle de max memory, Adaptive Server ne démarre pas. Modifiez votre fichier de configuration pour augmenter la taille d'autres caches et augmentez la valeur des paramètres de configuration nécessitant de la mémoire afin de créer un environnement dans lequel total logical memory est supérieur à max memory. Pour plus d'informations, reportez-vous au chapitre 18, "Configuration de la mémoire".

Le default data cache et tous les caches définis par l'utilisateur sont configurés explicitement avec une valeur absolue. En outre, de nombreux paramètres de configuration utilisent de la mémoire. Pour maximiser les performances et éviter les erreurs, fixez la valeur de max memory à un niveau suffisamment élevé pour convenir à tous les caches et à tous les paramètres de configuration utilisant de la mémoire.

Adaptive Server envoie un message d'avertissement si la valeur de max memory est inférieure à celle de total logical memory.

Changement de type de cache

Pour réserver un cache pour le journal de transactions, vous devez changer son type en "logonly". Dans l'exemple suivant, le pubs_log est créé avec le type "logonly" :

```
sp_cacheconfig pubs_log, "7M", "logonly"
```

Avant redémarrage, l'état du cache est alors le suivant :

Cache Name	Status	Type	Config Value	Run Value
pubs_log	Pend/Act	Log Only	7.00 Mb	0.00 Mb
Total			7.00 Mb	0.00 Mb

Vous pouvez modifier le type d'un cache mixte ("mixed") existant, à condition qu'aucun objet autre que de journal ne lui soit lié :

```
sp_cacheconfig pubtune_cache, logonly
```

En général, dans les environnements où s'effectuent de très nombreuses transactions, Adaptive Server fournit de meilleures performances si vous allouez deux pages logiques aux journaux de transactions. Lorsque les pages sont de grande taille (4, 8 et 16 ko), exécutez sp_sysmon pour trouver la configuration la mieux adaptée à votre site. Pour plus d'informations sur la configuration des caches en vue d'améliorer les performances du journal, reportez-vous à la section "Taille d'E/S de journal pour le cache de journal", page 654.

Configuration de la stratégie de remplacement d'un cache

Si un cache est dédié à une table ou à un index et qu'il comporte peu ou pas de remplacements de buffer lorsque le système atteint un état stable, vous pouvez définir une stratégie de remplacement LRU souple. Cette stratégie améliore les performances des caches dans lesquels les remplacements de buffer sont très réduits ou inexistant, ainsi que celles de la plupart des caches de journal. Reportez-vous au chapitre 14, "Utilisation et performances de la mémoire" du document *Performances et optimisation* pour obtenir des informations complémentaires. Pour définir une stratégie de remplacement LRU souple, entrez la commande suivante :

```
sp_cacheconfig pubs_log, relaxed
```

La valeur par défaut est "strict".

Vous pouvez créer un cache et spécifier le type et la stratégie de remplacement qui lui sont associés, à l'aide d'une seule commande. Les exemples suivants créent deux caches, pubs_log et pubs_cache :

```
sp_cacheconfig pubs_log, "3M", logonly, relaxed
sp_cacheconfig pubs_cache, "10M", mixed, strict
```

Vous devez redémarrer Adaptive Server pour que la stratégie de remplacement du cache soit prise en compte.

Après redémarrage, le résultat est le suivant :

```
sp_cacheconfig
Cache Name          Status   Type      Config Value Run Value
-----
default data cache  Active  Default   25.00 Mb    42.29 Mb
pubs_cache          Active  Mixed     10.00 Mb    10.00 Mb
pubs_log            Active  Log Only   7.00 Mb     7.00 Mb
-----
Total               42.00 Mb   59.29 Mb
=====
Cache: default data cache,  Status: Active,  Type: Default
      Config Size: 25.00 Mb,  Run Size: 42.29 Mb
      Config Replacement: strict LRU,  Run Replacement: strict LRU
      Config Partition:      1,  Run Partition:      1

IO Size  Wash Size  Config Size  Run Size      APF Percent
-----
  2 Kb   8662 Kb    0.00 Mb     42.29 Mb     10
=====
```

Division d'un cache de données en zones mémoire

```
Cache: pubs_cache, Status: Active, Type: Mixed
Config Size: 10.00 Mb, Run Size: 10.00 Mb
Config Replacement: strict LRU, Run Replacement: strict LRU
Config Partition: 1, Run Partition: 1
```

```
IO Size Wash Size Config Size Run Size APF Percent
-----
2 Kb 2048 Kb 0.00 Mb 10.00 Mb 10
```

```
=====  
Cache: pubs_log, Status: Active, Type: Log Only  
Config Size: 7.00 Mb, Run Size: 7.00 Mb  
Config Replacement: relaxed LRU, Run Replacement: relaxed LRU  
Config Partition: 1, Run Partition: 1
```

```
IO Size Wash Size Config Size Run Size APF Percent
-----
2 Kb 1432 Kb 0.00 Mb 7.00 Mb 10
```

Division d'un cache de données en zones mémoire

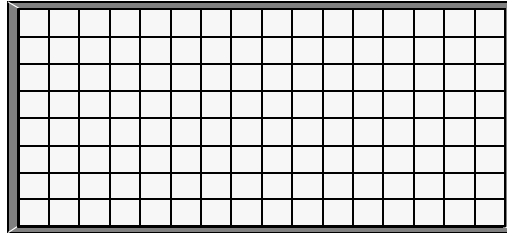
Après avoir créé un cache de données, vous pouvez le diviser en zones mémoire, chacune d'elles ayant une taille d'E/S différente. Un cache ne peut comporter qu'une zone par taille d'E/S. La taille minimum d'une zone mémoire est celle de la page logique du serveur. Les zones mémoire de plus grande taille doivent être à la puissance deux et leur taille maximum peut atteindre la taille d'un extent.

Lorsqu'Adaptive Server effectue des E/S étendues, plusieurs pages à la fois sont lues dans le cache. Ces pages sont toujours traitées ensemble ; elles vieillissent dans le cache et sont écrites sur disque comme s'il s'agissait d'une seule.

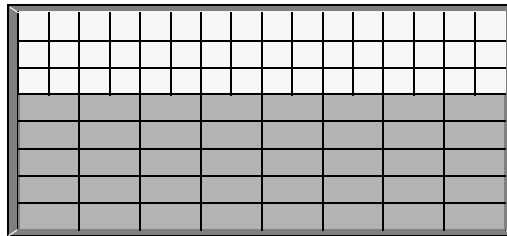
Par défaut, lorsque vous créez un cache de données nommé, la totalité de son espace est allouée à la zone mémoire par défaut. Lorsque vous créez d'autres zones, une partie de cet espace leur est allouée, ce qui réduit la taille de la zone mémoire par défaut. Par exemple, si vous créez un cache de données avec 50 Mo d'espace, l'intégralité de cet espace est attribuée à la zone de 2 ko. Si vous configurez ensuite une zone de 4 ko avec 30 Mo d'espace dans ce cache, la zone de 2 ko passe à 20 Mo.

Figure 19-2 : Configuration d'un cache et d'une zone mémoire de 4 ko

Création d'un cache de 50 Mo



Création d'une zone de 4 ko avec transfert de 30 Mo depuis la zone de 2 ko :

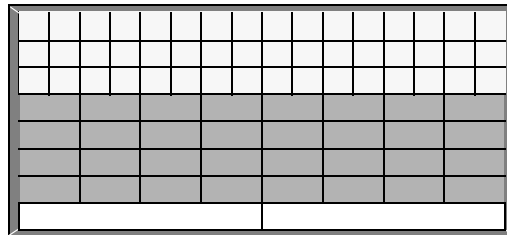


- Zone de 2 ko
- Zone de 4 ko

Une fois que vous avez créé les zones, vous pouvez transférer de l'espace entre elles. Par exemple, dans un cache comportant une zone de 2 ko de 20 Mo et une zone de 4 ko de 30 Mo, vous pouvez configurer une zone de 16 ko en prenant 10 Mo d'espace sur la zone de 4 ko.

Figure 19-3 : Transfert d'espace entre une zone existante et une nouvelle zone

Création d'une zone de 16 ko avec transfert de 10 Mo depuis la zone de 4 ko :



- Zone de 2 ko
- Zone de 4 ko
- Zone de 16 ko

Lorsque vous exécutez des commandes pour transférer de l'espace entre les zones d'un cache, vous n'avez pas besoin de redémarrer Adaptive Server ; vous pouvez donc reconfigurer des zones en fonction de l'évolution des charges des applications, sans que l'activité du serveur en soit considérablement affectée.

Outre la création de zones dans les caches que vous configurez, vous pouvez ajouter au default data cache des zones mémoire pour les E/S d'une taille maximale de 16 ko.

La syntaxe de configuration des zones mémoire est la suivante :

```
sp_poolconfig cache_name, "memsize [P|K|M|G] ",
"config_poolK" [, "affected_poolK"]
```

config_pool est configuré à la taille indiquée dans la commande. Cette dernière implique une deuxième zone (la affected_pool en transférant de l'espace depuis ou vers cette zone. Si vous ne spécifiez pas de affected_pool, l'espace est pris ou alloué sur la zone de 2 ko. La taille minimale d'une zone est de 512 ko.

Dans l'exemple suivant, une zone de 7 Mo de pages de 16 ko est créée dans le cache de données pubs_cache :

```
sp_poolconfig pubs_cache, "7M", "16K"
```

Cette commande diminue la taille de la zone mémoire. Pour obtenir la configuration courante, exécutez sp_cacheconfig en indiquant uniquement le nom du cache :

```
sp_cacheconfig pubs_cache
```

Cache Name	Status	Type	Config Value	Run Value
pubs_cache	Active	Mixed	10.00 Mb	10.00 Mb
Total			10.00 Mb	10.00 Mb

```

=====
Cache: pubs_cache, Status: Active, Type: Mixed
Config Size: 10.00 Mb, Run Size: 10.00 Mb
Config Replacement: strict LRU, Run Replacement: strict LRU
Config Partition: 1, Run Partition: 1

```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	2048 Kb	0.00 Mb	3.00 Mb	10
16 Kb	1424 Kb	7.00 Mb	7.00 Mb	10

Vous pouvez également créer des zones mémoire dans default data cache.

Dans l'exemple ci-après, vous commencez avec la configuration de cache suivante :

```
Cache Name          Status    Type      Config Value Run Value
-----
default data cache  Active   Default   25.00 Mb    42.29 Mb
-----
                                Total      25.00 Mb    42.29 Mb
=====
Cache: default data cache, Status: Active, Type: Default
      Config Size: 25.00 Mb, Run Size: 42.29 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition:      1, Run Partition:      1
```

```
IO Size  Wash Size Config Size Run Size    APF Percent
-----
  2 Kb   8662 Kb    0.00 Mb    42.29 Mb    10
```

Cette commande crée une zone de 16 ko dans le default data cache qui est de 8 Mo :

```
sp_poolconfig "default data cache", "8M", "16K"
```

La configuration devient la suivante, avec diminution de la valeur d'exécution ("Run Size") de la zone de 2 ko :

```
Cache Name          Status    Type      Config Value Run Value
-----
default data cache  Active   Default   25.00 Mb    42.29 Mb
-----
                                Total      25.00 Mb    42.29 Mb
=====
Cache: default data cache, Status: Active, Type: Default
      Config Size: 25.00 Mb, Run Size: 42.29 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition:      1, Run Partition:      1
```

```
IO Size  Wash Size Config Size Run Size    APF Percent
-----
  2 Kb   8662 Kb    0.00 Mb    34.29 Mb    10
 16 Kb  1632 Kb    8.00 Mb     8.00 Mb     10
```

Vous n'avez pas besoin de configurer la taille de la zone mémoire de 2 ko dans les caches que vous créez. Sa valeur d'exécution ("Run Size") représente la totalité de la mémoire qui n'a pas été configurée explicitement pour d'autres zones du cache.

Taille d'E/S de journal pour le cache de journal

Si vous créez un cache pour le journal de transactions d'une base de données, vous devez configurer l'espace de ce cache de sorte qu'il corresponde à la taille des E/S du journal. La valeur par défaut est deux fois la taille de la page logique du serveur (4 ko pour une page logique de 2 ko, 8 ko pour une page logique de 4 ko, etc.). En l'absence d'une zone de 4 ko, Adaptive Server utilise une E/S de 2 ko pour le journal. Vous pouvez modifier la taille des E/S du journal à l'aide de `sp_logiosize`. La taille des E/S du journal de chaque base de données est indiquée dans le journal d'erreurs au démarrage d'Adaptive Server, mais vous pouvez aussi contrôler cette taille pour une base de données en vous plaçant dans cette base et en exécutant `sp_logiosize` sans paramètres.

L'exemple suivant crée une zone de 4 ko dans le cache `pubs_log` :

```
sp_poolconfig pubs_log, "3M", "4K"
```

Vous pouvez également créer une zone mémoire de 4 ko dans le `default data cache`, pour qu'elle serve aux journaux de transactions de toute base n'étant pas liée à un autre cache :

```
sp_poolconfig "default data cache", "2.5M", "4K"
```

Reportez-vous au document *Performances et optimisation* pour obtenir des informations complémentaires sur le réglage de la taille des E/S de journal.

Liaison d'objets à des caches

`sp_bindcache` permet d'attribuer à un cache une base de données, une table, un index ou un objet de type texte ou image. Pour qu'une entité puisse être liée à un cache, les conditions suivantes doivent être remplies :

- Le cache nommé doit exister et son état doit être "Active".
- La base de données ou l'objet de base de données doit exister.
- Pour lier des tables, index ou objets, vous devez utiliser la base de données dans laquelle ces éléments sont stockés.
- Pour la liaison de tables système, y compris la table du journal de transactions `syslogs`, la base de données doit être en mode mono-utilisateur.

- Pour lier une base de données, vous devez vous trouver dans la base master et la base à lier doit être en mode mono-utilisateur.
- Pour lier une base de données, une table utilisateur, un index ou un objet de type texte ou image à un cache, celui-ci doit posséder le type "Mixed" (mixte). Seule la table syslogs peut être liée à un cache de type "Log Only".
- Vous devez être propriétaire de l'objet ou de la base de données ou être administrateur système.

Pour lier des objets à des caches, vous devez redémarrer Adaptive Server après la création des caches. En revanche, les liaisons sont prises en compte immédiatement et un redémarrage n'est pas nécessaire.

La syntaxe de liaison d'objets à des caches est la suivante :

```
sp_bindcache cache_name, dbname [, [owner.]tablename  
[, indexname | "text only" ] ]
```

Le nom du propriétaire est facultatif si la table appartient au "dbo".

La commande suivante lie la table titles au cache nommé pubs_cache :

```
sp_bindcache pubs_cache, pubs2, titles
```

Pour lier un index sur titles, ajoutez le nom de cet index en troisième paramètre :

```
sp_bindcache pubs_cache, pubs2, titles, titleind
```

Dans les exemples précédents, il n'est pas nécessaire d'indiquer le nom du propriétaire car les objets de la base de données pubs2 appartiennent au "dbo". Cependant, si vous voulez spécifier une table appartenant à tout autre utilisateur, vous devez indiquer le nom de son propriétaire. Vous devez placer ce paramètre entre guillemets, car le point qu'il comporte constitue un caractère spécial :

```
sp_bindcache pubs_cache, pubs2, "fred.sales_east"
```

La commande suivante lie le journal de transactions syslogs au cache pubs_log :

```
sp_bindcache pubs_log, pubs2, syslogs
```

La base de données doit être en mode mono-utilisateur pour que vous puissiez lier des tables système, y compris le journal de transactions syslogs, à un cache. Exécutez sp_dboption depuis la base master, puis la commande use database et enfin checkpoint :

```
sp_dboption pubs2, single, true  
use pubs2  
checkpoint
```

Les colonnes text et image d'une table sont stockées dans une structure de données distincte dans la base de données. Pour lier ce type d'objet à un cache, ajoutez le paramètre "text_only" :

```
sp_bindcache pubs_cache, pubs2, au_pix, "text only"
```

La commande suivante, exécutée depuis la base master, lie la base de données tempdb à un cache :

```
sp_bindcache tempdb_cache, tempdb
```

Vous pouvez recréer des liaisons d'objets sans avoir besoin de supprimer les liaisons existantes.

Restrictions des liaisons de caches

Vous ne pouvez pas lier un objet de base de données ou annuler sa liaison lorsque :

- des lectures de données modifiées (dirty read) sont actives sur cet objet,
- un curseur est ouvert sur l'objet.

De plus, Adaptive Server doit verrouiller l'objet pendant la liaison ou l'annulation de la liaison ; la commande peut donc avoir un temps de réponse important car elle attend que les verrous soient libérés. Pour plus d'informations, reportez-vous à la section "Verrouillage pour effectuer des liaisons", page 673.

Obtention d'informations sur les liaisons de caches

sp_helpcache, exécutée avec un nom de cache, fournit des informations sur ce cache et les entités qui lui sont liées :

```
sp_helpcache pubs_cache
```

Cache Name	Config Size	Run Size	Overhead
pubs_cache	10.50 Mb	10.50 Mb	0.56 Mb

----- Cache Binding Information: -----

Cache Name	Entity Name	Type	Index Name	Status
------------	-------------	------	------------	--------

-----	-----	----	-----	-----
pubs_cache	pubs2.dbo.titles	index	titleind	V
pubs_cache	pubs2.dbo.au_pix	index	tau_pix	V
pubs_cache	pubs2.dbo.titles	table		V
pubs_cache	pubs2.fred.sales_east	table		V

Si vous exécutez `sp_helpcache` sans fournir de nom de cache, cette procédure affiche des informations sur tous les caches configurés sur Adaptive Server et sur tous les objets qui leur sont liés.

`sp_helpcache` peut effectuer une recherche de chaîne correspondant au nom du cache, à l'aide de `%cachename%`. Ainsi, la chaîne "pubs" correspond aux caches "pubs_cache" et "pubs_log".

La colonne "Status" indique si la liaison d'un cache est correcte ("V") ou incorrecte ("I"). Si une base de données ou un objet est lié(e) à un cache et que ce cache est supprimé, les informations sur cette liaison restent dans les tables système, mais la liaison est marquée comme étant incorrecte. Tous les objets dont la liaison est incorrecte utilisent default data cache. Si vous créez ensuite un autre cache du même nom, la liaison devient correcte lorsque ce cache est activé par un redémarrage d'Adaptive Server.

Vérification de l'overhead du cache

`sp_helpcache` peut indiquer le volume d'overhead requis pour gérer un cache de données nommé d'une certaine taille. Lorsque vous créez un cache de données nommé, la totalité de l'espace que vous demandez à l'aide de `sp_cacheconfig` est mise à la disposition du cache. La mémoire nécessaire à la gestion du cache est prise sur default data cache.

Pour obtenir l'overhead requis pour gérer un cache, indiquez la taille proposée. Vous pouvez utiliser les unités de taille P pour pages, K pour kilooctets, M pour mégaoctets ou G pour gigaoctets. La commande suivante contrôle l'overhead pour 20 000 pages :

```
sp_helpcache "20000P"

2.08Mb of overhead memory will be needed to manage a
cache of size 20000P
```

Notez que vous ne perdez pas d'espace de cache lorsque vous configurez des caches utilisateur. Environ 5 % de mémoire sont requis pour les structures qui stockent et maintiennent les pages en mémoire, que vous utilisiez un cache de données de grande taille ou plusieurs caches plus petits.

Incidence de l'overhead sur l'espace total du cache

L'exemple décrit dans la section "Informations sur les caches de données", page 641 concerne un default data cache disposant de 59,44 Mo d'espace avant création de caches définis par l'utilisateur. Le serveur dans cet exemple utilise une page logique de 2 ko. Après la création du cache pubs_cache de 10 Mo et le redémarrage d'Adaptive Server, la procédure sp_cacheconfig indique une taille totale de cache de 59,44 Mo.

La configuration d'un cache de données peut sembler augmenter ou diminuer le cache total disponible. Cela tient au volume d'overhead requis pour gérer un cache d'une taille donnée et au fait que l'overhead n'est pas inclus dans les valeurs affichées par sp_cacheconfig.

L'exécution de sp_helpcache pour contrôler l'overhead du cache par défaut de 59,44 Mo et le nouveau cache de 10 Mo vous permet de constater que la modification de l'espace provient du changement de la taille d'overhead. La commande suivante affiche l'overhead du default data cache avant toute modification :

```
sp_helpcache "59.44M"

3.04Mb of overhead memory will be needed to manage a
cache of size 59.44M
```

La commande suivante affiche l'overhead pubs_cache :

```
sp_helpcache "10M"

0.53Mb of overhead memory will be needed to manage a
cache of size 10M
```

Les calculs suivants ajoutent l'overhead requis pour gérer l'espace de cache d'origine, puis soustraient de l'espace l'overhead de pubs_cache :

Taille totale de cache d'origine (overhead non inclus)	59,44
Overhead pour le cache par défaut de 59,44 Mo	+3,04
Espace de cache total, overhead inclus	62,48
pubs_cache de 10 Mo et overhead de 0,53 Mo	-10,53
Espace restant	51,95
Overhead pour le cache de 51,95 Mo	-2,69
Taille utilisable pour le cache par défaut	49,26

Les tailles de cache sont arrondies à deux décimales lorsqu'elles sont affichées par sp_cacheconfig et l'overhead, à deux décimales par sp_helpcache, donc les résultats comportent très peu d'erreurs d'arrondi.

Suppression de liaisons de caches

Deux commandes permettent d'annuler des liaisons de caches :

- `sp_unbindcache` annule la liaison entre une entité et un cache ;
- `sp_unbindcache_all` annule la liaison entre tous les objets et un cache.

La syntaxe de `sp_unbindcache` est la suivante :

```
sp_unbindcache dbname [, [owner.]tablename  
[, indexname | "text only" ] ]
```

La commande ci-après annule la liaison de l'index `titleidind` sur la table `titles` de la base de données `pubs2` :

```
sp_unbindcache pubs2, titles, titleidind
```

Pour annuler les liaisons entre tous les objets et un cache, utilisez `sp_unbindcache_all` en indiquant le nom du cache :

```
sp_unbindcache_all pubs_cache
```

Remarque Vous ne pouvez pas utiliser `sp_unbindcache_all` si plus de huit bases de données et/ou des objets de huit bases de données sont liés au cache. Vous devez utiliser `sp_unbindcache` sur des bases de données ou des objets individuels afin de réduire à huit ou moins le nombre de bases de données impliquées.

Lorsque vous supprimez la liaison d'un cache à un objet, toutes les pages actuellement en mémoire sont supprimées du cache.

Modification de la zone de vidage d'une zone mémoire

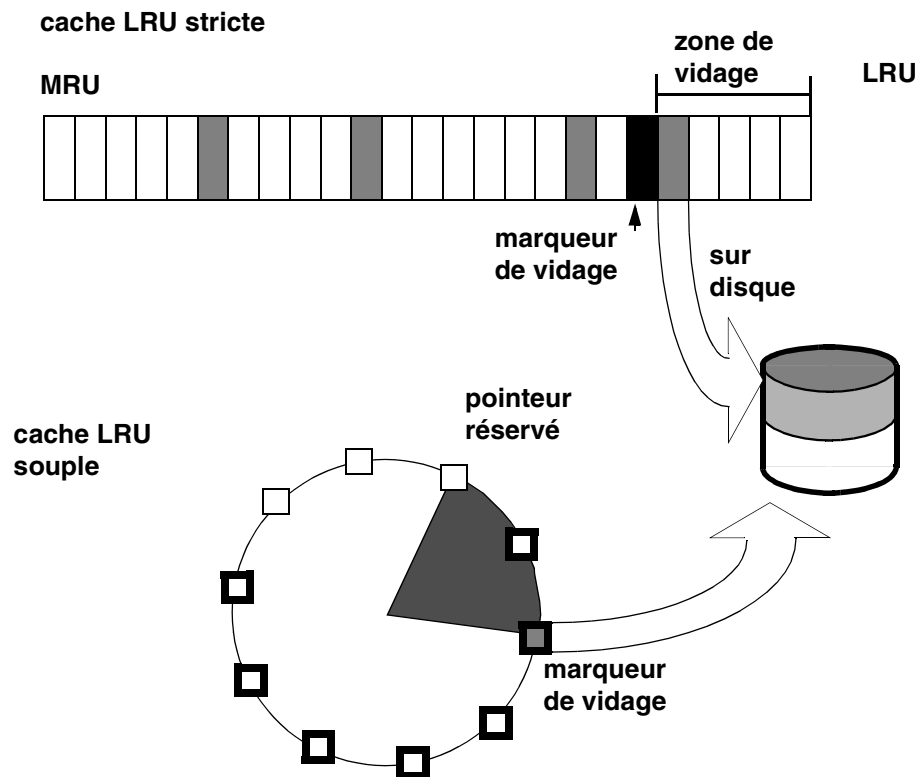
Lorsqu'il a besoin de lire un buffer dans le cache, Adaptive Server le place :

- à l'extrémité LRU (utilisée le moins fréquemment) de chaque zone mémoire dans un cache disposant d'une stratégie de remplacement LRU stricte,
- au niveau du pointeur réservé, dans un cache disposant d'une stratégie de remplacement LRU souple. Si le bit récemment utilisé du buffer situé au niveau du pointeur réservé est défini, ce dernier est transféré au buffer suivant dans la zone.

Une partie de chaque zone est configurée comme **zone de vidage**. Lorsque les pages modifiées (dans le cache) dépassent le marqueur de vidage et entrent dans la zone de vidage, Adaptive Server lance une E/S asynchrone sur ces pages. A l'issue de ces écritures, les pages sont marquées comme étant vides et restent disponibles dans le cache jusqu'à ce qu'elles atteignent l'extrémité LRU.

L'espace alloué à la zone de vidage doit être suffisant pour que les E/S du buffer prennent fin avant que la page ait besoin d'être remplacée. La figure 19-4 illustre la manière dont la zone de vidage d'une zone de buffer fonctionne respectivement avec un cache LRU stricte et souple.

Figure 19-4 : Zone de vidage d'une zone de buffer



Par défaut, la taille de la zone de vidage d'une zone de mémoire est configurée de la manière suivante :

- Si la taille de la zone est inférieure à 300 Mo, la taille de vidage par défaut correspond à 20 % des buffers de la zone.

- Si la taille de la zone est supérieure à 300 Mo, la taille de vidage par défaut correspond à 20 % du nombre de buffers dans ces 300 Mo.

La taille minimale de la zone de vidage correspond à 10 buffers. La taille maximale de la zone de vidage correspond à 80 % de la taille de la zone mémoire.

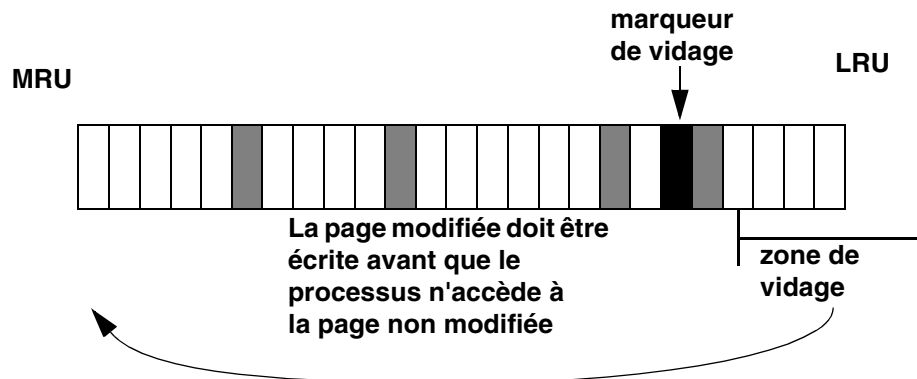
Un buffer correspond à un bloc de pages égal à la taille d'E/S de la zone. Chaque buffer est traité comme une unité : toutes les pages du buffer sont lues dans le cache, écrites sur le disque et évoluent dans le cache en une unité. Pour obtenir la taille du bloc, multipliez le nombre de buffers par la taille de la zone—pour une zone de 2 ko, 256 buffers = 512 ko, pour une zone de 16 ko, 256 buffers = 4 096 ko.

Par exemple, si vous configurez une zone de 16 ko avec 1 Mo d'espace, la zone comporte 64 buffers ; 20 % de 64 donnent 12,8. Cette valeur est arrondie à l'unité inférieure, ce qui signifie que 12 buffers, soit 192 ko, sont alloués à la zone de vidage.

Zone de vidage trop petite

Si la zone de vidage est trop petite pour pouvoir être utilisée dans une zone de buffer, les opérations qui requièrent un buffer non modifié risquent de devoir attendre la fin des E/S sur le buffer modifié situé à l'extrémité LRU de la zone ou au niveau du pointeur réservé. Ce problème, qui s'appelle une **monopolisation de buffer**, peut altérer considérablement les performances. La figure 19-5 représente une monopolisation de buffer sur un cache comportant une stratégie de remplacement stricte.

Figure 19-5 : Monopolisation de buffer due à une zone de vidage trop petite



Vous pouvez utiliser `sp_sysmon` pour déterminer si des problèmes de monopolisation de buffer se produisent dans vos zones mémoire. Exécutez la procédure `sp_sysmon` lorsque le cache enregistre un trafic important en matière d'E/S et un grand nombre d'opérations de mise à jour car la monopolisation de buffer résulte généralement de l'association de plusieurs pages modifiées et d'un taux élevé de remplacement au sein du cache.

Si le résultat "Buffers Grabbed Dirty" de la section récapitulative concernant le cache affiche une valeur différente de 0 dans la colonne "Count", contrôlez la ligne "Grabbed Dirty" pour chaque zone afin de déterminer l'origine du problème. Augmentez la taille de la zone de vidage pour la zone mémoire concernée. La commande suivante alloue 8 ko à la zone de vidage de la zone mémoire de 720 ko :

```
sp_poolconfig pubs_cache, "8K", "wash=720K"
```

Si la zone mémoire est très petite, vous pouvez être amené à augmenter sa taille, en particulier si le résultat de la procédure `sp_sysmon` indique que cette zone connaît des taux de rotation importants.

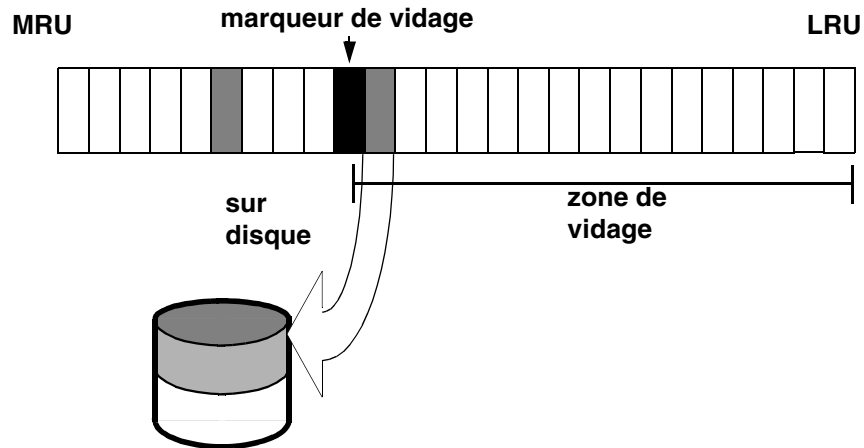
Pour plus d'informations sur cette procédure, reportez-vous au document *Performances et optimisation*.

Zone de vidage trop grande

Si la zone de vidage est trop grande dans une zone mémoire, les buffers circulent trop rapidement dans le cache après le "marqueur de vidage" et une opération d'écriture asynchrone est lancée sur tous les buffers modifiés, comme l'indique la figure 19-6. Le buffer est marqué comme "non modifié" et il reste dans la zone de vidage de la chaîne MRU/LRU jusqu'à ce qu'il ait atteint l'extrémité LRU. Si une autre requête modifie une page du buffer, Adaptive Server doit exécuter des E/S supplémentaires pour écrire à nouveau cette page sur le disque.

Si la procédure `sp_sysmon` révèle un pourcentage important de buffers "Found in Wash" pour un cache avec une stratégie de remplacement stricte alors qu'il n'y a pas de monopolisation de buffer, vous pouvez tenter de réduire la taille de la zone de vidage. Pour plus d'informations sur cette procédure, reportez-vous au document *Performances et optimisation*.

Figure 19-6 : Conséquences d'une zone de vidage trop grande



Modification de la limite de prélecture asynchrone pour une zone mémoire

La limite de prélecture asynchrone indique le pourcentage de la zone mémoire pouvant être alloué au stockage des pages qui ont été placées dans le cache par une prélecture asynchrone, mais qui n'ont encore été utilisées par aucune requête. La valeur par défaut pour le serveur est définie à l'aide du paramètre de global `async prefetch limit`. Les limites de zone mémoire, définies à l'aide de la procédure `sp_poolconfig`, remplacent la limite par défaut d'une zone individuelle.

La commande suivante définit sur 20 le pourcentage de la zone de 2 ko dans `pubs_cache` :

```
sp_poolconfig pubs_cache, "2K", "local async prefetch limit=20"
```

Les modifications apportées à la limite de prélecture sont immédiatement prises en compte et ne nécessitent pas le redémarrage d'Adaptive Server. Les valeurs admises sont comprises entre 0 et 100. Si vous définissez cette limite sur 0 pour une zone mémoire, la prélecture asynchrone est désactivée dans une zone. Pour obtenir des informations sur l'incidence d'une prélecture asynchrone sur les performances, reportez-vous au chapitre 25, "Optimisation de la prélecture asynchrone" dans le document *Performances et optimisation*.

Redéfinition de la taille des caches de données nommés

Pour modifier la taille d'un cache existant, exécutez `sp_cacheconfig` en indiquant une nouvelle taille totale pour le cache. Lorsque vous augmentez la taille d'un cache, tout l'espace supplémentaire est ajouté à la zone. Lorsque vous diminuez la taille d'un cache, l'espace est pris dans la zone par défaut. Vous ne pouvez pas diminuer la taille de la zone de 2 ko en dessous de 512 ko.

Augmentation de la taille d'un cache

`sp_cacheconfig` indique que `pubs_cache` est actuellement configuré avec 10 Mo d'espace :

```
sp_cacheconfig pubs_cache
```

Cache Name	Status	Type	Config Value	Run Value
pubs_cache	Active	Mixed	10.00 Mb	10.00 Mb
Total			10.00 Mb	10.00 Mb

```
=====  
Cache: pubs_cache, Status: Active, Type: Mixed  
Config Size: 10.00 Mb, Run Size: 10.00 Mb  
Config Replacement: strict LRU, Run Replacement: strict LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	720 Kb	0.00 Mb	3.00 Mb	20
16 Kb	1424 Kb	7.00 Mb	7.00 Mb	10

Pour augmenter la taille de ce cache et de sa zone de 2 ko, indiquez la nouvelle taille totale du cache :

```
sp_cacheconfig pubs_cache, "20M"
```

Le résultat ci-après fournit la configuration de `pubs_cache` avant un redémarrage :

Cache Name	Status	Type	Config Value	Run Value
pubs_cache	Active	Mixed	20.00 Mb	10.00 Mb
Total			20.00 Mb	10.00 Mb


```
=====
Cache: pubs_cache, Status: Active, Type: Mixed
      Config Size: 10.00 Mb, Run Size: 10.00 Mb
      Config Replacement: strict LRU, Run Replacement: strict LRU
      Config Partition: 1, Run Partition: 1

IO Size  Wash Size  Config Size  Run Size      APF Percent
-----  -
      2 Kb   720 Kb      0.00 Mb      3.00 Mb       20
      16 Kb  1424 Kb     7.00 Mb      7.00 Mb       10
```

Les 10 Mo supplémentaires ont été configurés et seront disponibles dans la zone de 2 ko après le prochain redémarrage.

Diminution de la taille d'un cache

Vous pouvez également diminuer la taille d'un cache. Par exemple, la commande suivante fournit des informations sur le cache pubs_log :

```
sp_cacheconfig pubs_log

Cache Name      Status      Type      Config Value  Run Value
-----
pubs_log        Active      Log Only      7.00 Mb      7.00 Mb
-----
                        Total      7.00 Mb      7.00 Mb
=====
Cache: pubs_log, Status: Active, Type: Log Only
      Config Size: 7.00 Mb, Run Size: 7.00 Mb
      Config Replacement: relaxed LRU, Run Replacement: relaxed LRU
      Config Partition: 1, Run Partition: 1

IO Size  Wash Size  Config Size  Run Size      APF Percent
-----  -
      2 Kb   920 Kb      0.00 Mb      4.50 Mb       10
      4 Kb   512 Kb      2.50 Mb      2.50 Mb       10
```

La commande suivante diminue la taille du cache pubs_log, réduisant la taille de la zone par défaut :

```
sp_cacheconfig pubs_log, "6M"
```

Suppression de caches de données

Après redémarrage d'Adaptive Server, `sp_cacheconfig` fournit les valeurs suivantes :

Cache Name	Status	Type	Config Value	Run Value
pubs_log	Active	Log Only	6.00 Mb	6.00 Mb
Total			6.00 Mb	6.00 Mb

=====
Cache: pubs_log, Status: Active, Type: Log Only
Config Size: 6.00 Mb, Run Size: 6.00 Mb
Config Replacement: relaxed LRU, Run Replacement: relaxed LRU
Config Partition: 1, Run Partition: 1

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	716 Kb	0.00 Mb	3.50 Mb	10
4 Kb	512 Kb	2.50 Mb	2.50 Mb	10

Lorsque vous diminuez la taille d'un cache de données, la totalité de l'espace à supprimer doit être disponible dans la zone de 2 ko. Vous pouvez avoir besoin de transférer de l'espace vers la zone par défaut à partir d'autres zones avant de réduire la taille du cache de données. Ainsi, dans le dernier exemple, si vous vouliez réduire la taille du cache à 3 Mo, vous devriez utiliser `sp_poolconfig` pour transférer de la mémoire depuis la zone de 4 ko vers celle de 2 ko. La mémoire est déplacée vers "memory available for named caches" (mémoire disponible pour les caches nommés). Pour plus d'informations, reportez-vous à la section "Modification de la taille des zones mémoire", page 667.

Suppression de caches de données

Pour supprimer intégralement un cache de données, réinitialisez sa taille à 0 :

```
sp_cacheconfig pubs_log, "0"
```

L'état (status) de ce cache prend alors la valeur "Pend/Del". Pour que cette modification prenne effet, vous devez redémarrer Adaptive Server. Tant que vous ne le faites pas, le cache reste actif et tous les objets qui lui sont liés continuent de l'utiliser pour les E/S.

Si vous supprimez un cache de données et que des objets lui sont liés, les liaisons de ce cache sont notées comme étant incorrectes (invalid) lors du prochain redémarrage d'Adaptive Server. Tous les objets dont la liaison est incorrecte utilisent le default data cache. Des messages d'avertissement sont inscrits dans le journal d'erreurs lorsque les liaisons sont marquées comme étant incorrectes. Par exemple, si la table titles de la base de données pubs2 est liée à un cache et que ce dernier est supprimé, le message inscrit dans le journal est le suivant :

```
Cache binding for database '5', object '208003772',
index '0' is being marked invalid in Sysattributes.
```

Si vous recréez le cache et que vous redémarrez Adaptive Server, les liaisons redeviennent correctes (valid).

Vous ne pouvez pas supprimer default data cache.

Modification de la taille des zones mémoire

Pour modifier la taille d'une zone mémoire, exécutez `sp_poolconfig` en indiquant le nom du cache, la nouvelle taille de la zone, la taille des E/S de la zone à modifier et la taille des E/S de la zone à partir de laquelle les buffers doivent être extraits. Si vous n'indiquez pas ce dernier paramètre, l'intégralité de l'espace serait extrait de ou alloué à la zone.

Déplacement d'espace de la zone mémoire

La commande suivante vérifie la configuration existante du cache `pubs_log` (le résultat de cet exemple s'appuie sur les exemples des sections précédentes) :

```
sp_cacheconfig pubs_log
```

Cache Name	Status	Type	Config Value	Run Value
pubs_log	Active	Log Only	6.00 Mb	6.00 Mb
Total			6.00 Mb	6.00 Mb

```
=====  
Cache: pubs_log, Status: Active, Type: Log Only  
Config Size: 6.00 Mb, Run Size: 6.00 Mb  
Config Replacement: relaxed LRU, Run Replacement: relaxed LRU
```

Modification de la taille des zones mémoire

Config Partition: 1, Run Partition: 1

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	716 Kb	0.00 Mb	3.50 Mb	10
4 Kb	512 Kb	2.50 Mb	2.50 Mb	10

La commande suivante augmente à 5 Mo la taille de la zone de 4 ko, en prenant l'espace requis depuis la zone de 2 ko :

```
sp_poolconfig pubs_log, "5M", "4K"
```

```
sp_cacheconfig pubs_log
```

Cache Name	Status	Type	Config Value	Run Value
pubs_log	Active	Log Only	6.00 Mb	6.00 Mb
Total			6.00 Mb	6.00 Mb

```
=====  
Cache: pubs_log, Status: Active, Type: Log Only  
Config Size: 6.00 Mb, Run Size: 6.00 Mb  
Config Replacement: relaxed LRU, Run Replacement: relaxed LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	716 Kb	0.00 Mb	1.00 Mb	10
4 Kb	1024 Kb	5.00 Mb	5.00 Mb	10

Déplacement d'espace d'autres zones mémoire

Pour transférer de l'espace depuis une zone autre que la zone de 2 ko, vous devez indiquer le nom du cache, ainsi qu'une taille d'E/S cible et une taille d'E/S d'origine. La configuration courante de default data cache est la suivante :

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	25.00 Mb	29.28 Mb
Total			25.00 Mb	29.28 Mb

```
=====  
Cache: default data cache, Status: Active, Type: Default  
Config Size: 25.00 Mb, Run Size: 29.28 Mb  
Config Replacement: strict LRU, Run Replacement: strict LRU
```

Config Partition: 1, Run Partition: 1

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	3844 Kb	0.00 Mb	18.78 Mb	10
4 Kb	512 Kb	2.50 Mb	2.50 Mb	10
16 Kb	1632 Kb	8.00 Mb	8.00 Mb	10

La commande suivante augmente la taille de la zone de 4 ko de 2,5 Mo à 4 Mo, en prenant l'espace depuis la zone de 16 ko :

```
sp_poolconfig "default data cache", "4M", "4K", "16K"
```

La configuration est alors la suivante :

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	25.00 Mb	29.28 Mb
Total			25.00 Mb	29.28 Mb

```
=====  
Cache: default data cache, Status: Active, Type: Default  
Config Size: 25.00 Mb, Run Size: 29.28 Mb  
Config Replacement: strict LRU, Run Replacement: strict LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	3844 Kb	0.00 Mb	18.78 Mb	10
4 Kb	512 Kb	4.00 Mb	4.00 Mb	10
16 Kb	1632 Kb	6.50 Mb	6.50 Mb	10

Lorsque vous exécutez une commande pour transférer des buffers entre des zones d'un cache, Adaptive Server ne peut transférer que les buffers disponibles. Il ne peut pas transférer des buffers en cours d'utilisation ni des buffers comportant des modifications qui n'ont pas encore été écrites sur disque.

Lorsqu'Adaptive Server ne peut pas transférer le nombre de buffers requis, il affiche un message d'information indiquant la taille demandée et la taille obtenue de la zone mémoire.

Ajout de partitions de cache

Sur les serveurs à plusieurs moteurs, plusieurs tâches peuvent tenter d'accéder simultanément au cache. Par défaut, chaque cache possède un seul verrou d'attente. Ainsi, une seule tâche à la fois peut modifier le cache ou y accéder. Si le conflit de verrous d'attente du cache dépasse 10 %, l'augmentation du nombre de partitions de cache pour un cache peut réduire le conflit de verrous d'attente et améliorer les performances.

Vous pouvez configurer le nombre de partitions de cache :

- pour l'ensemble des caches de données, à l'aide du paramètre de configuration global `cache partition number`,
- pour un seul cache, avec `sp_cacheconfig`.

Le nombre de partitions contenues dans un cache est toujours une puissance de 2 comprise entre 1 et 64. Aucune zone de n'importe quelle partition de cache ne peut avoir une taille inférieure à 512 ko. Dans la plupart des cas, étant donné que la taille des caches est fonction des besoins en stockage des objets individuels, vous devez utiliser le paramètre local pour le cache particulier dont le conflit de verrous d'attente pose problème.

Reportez-vous au chapitre 14, à la section "Réduction des conflits de verrous avec les partitions de caches", page 339 page 14-18 du document *Performances et optimisation* pour obtenir des informations sur le choix du nombre de partitions pour un cache.

Définition du nombre de partitions de cache avec `sp_configure`

Pour définir le nombre de partitions de cache pour l'ensemble des caches d'un serveur, exécutez `sp_configure`. Par exemple, pour définir le nombre de partitions de cache sur 2, tapez :

```
sp_configure "global cache partition number",2
```

Vous devez redémarrer le serveur pour que la modification prenne effet.

Définition du nombre de partitions de cache local

Exécutez `sp_cacheconfig` ou le fichier de configuration pour définir le nombre de partitions de cache local. Cette commande permet de définir la valeur 4 pour le nombre de partitions de cache de default data cache :

```
sp_cacheconfig "default data cache", "cache_partition=4"
```

Vous devez réinitialiser le serveur pour que la modification prenne effet.

Priorité

Le paramétrage de la partition de cache local a toujours la priorité sur la valeur globale de partition de cache.

Les commandes ci-après définissent la valeur 4 pour la partition à l'échelle du serveur et la valeur 2 pour le nombre de partitions `pubs_cache` :

```
sp_configure "global cache partition number", 4
sp_cacheconfig "pubs_cache", "cache_partition=2"
```

Le nombre de partitions de cache local a la priorité sur le nombre global de partitions de cache. Ainsi, `pubs_cache` utilise 2 partitions. Tous les autres caches configurés possèdent 4 partitions.

Pour supprimer le paramètre local pour `pubs_cache` et utiliser à la place la valeur globale, utilisez la commande :

```
sp_cacheconfig "pubs_cache", "cache_partition=default"
```

Pour réinitialiser le nombre global de partitions de cache et lui attribuer la valeur par défaut, utilisez :

```
sp_configure "global cache partition number", 0, "default"
```

Suppression d'une zone mémoire

Pour supprimer intégralement une zone mémoire, réinitialisez sa taille à 0. Cette commande supprime la zone de 16 ko, en ajoutant l'intégralité de l'espace à la zone de 2 ko :

```
sp_poolconfig "default data cache", "0", "16K"  
sp_cacheconfig "default data cache"
```

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	25.00 Mb	29.28 Mb
Total			25.00 Mb	29.28 Mb

=====
Cache: default data cache, Status: Active, Type: Default
Config Size: 25,00 Mb, Run Size: 29,28 Mb
Config Replacement: strict LRU, Run Replacement: strict LRU
Config Partition: 1, Run Partition: 1

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	3844 Kb	6.50 Mb	25.28 Mb	10
4 Kb	512 Kb	4.00 Mb	4.00 Mb	10

Si vous n'indiquez pas de zone affectée (16 ko dans l'exemple précédent), l'intégralité de l'espace est placée dans la zone par défaut. Vous ne pouvez pas supprimer la zone par défaut d'un cache.

Impossibilité de supprimer des zones lorsque les pages sont en cours d'utilisation

Si la zone mémoire que vous tentez de supprimer contient des pages en cours d'utilisation ou des pages qui ont été modifiées mais pas écrites sur le disque, Adaptive Server transfère autant de pages que possible vers la zone spécifiée et affiche un message d'information qui indique la taille de la zone restante. Si celle-ci est inférieure à la taille minimale autorisée de la zone, vous recevez également un message d'avertissement indiquant que la zone a été marquée comme indisponible. Si vous exécutez la procédure `sp_cacheconfig` après avoir reçu un message d'avertissement de ce type, la section "pool detail" relative à la zone concernée contient une colonne "Status" supplémentaire qui porte la mention "Unavailable/too small" (indisponible/trop petite) ou "Unavailable/deleted" (indisponible/supprimée).

Vous pouvez réexécuter la commande ultérieurement pour finir de supprimer la zone. Les zones qui portent la mention "Unavailable/too small" ou "Unavailable/deleted" sont également supprimées lorsque vous redémarrez Adaptive Server.

Incidence de la liaison de caches sur la mémoire et les plans de requêtes

Les liaisons et annulations de liaisons d'objets peuvent modifier les performances. Lorsque vous liez une table ou un index, ou que vous annulez cette liaison :

- les pages de l'objet sont rejetées du cache ;
- l'objet doit être verrouillé pour que la liaison soit effectuée ;
- tous les plans d'exécution de requête des procédures et des triggers doivent être recompilés.

Sortie de pages du cache

Lorsque vous liez un objet ou une base de données à un cache, toutes les pages de l'objet déjà stockées en mémoire sont supprimées du cache source. La prochaine fois qu'une requête a besoin de ces pages, elles sont lues dans le nouveau cache. De même, lorsque vous annulez des liaisons d'objets, les pages du cache sont supprimées du cache configuré par l'utilisateur et lues dans le cache par défaut la prochaine fois qu'une requête en a besoin.

Verrouillage pour effectuer des liaisons

Pour lier ou annuler des liaisons de tables utilisateur, d'index ou d'objets de type texte ou image, les commandes de liaison de cache doivent poser un verrou de table exclusif sur l'objet. Si un utilisateur a posé un verrou sur une table et que vous exécutez la procédure `sp_bindcache`, `sp_unbindcache` ou `sp_unbindcache_all` sur l'objet, la procédure système est mise en veille jusqu'à ce qu'elle puisse à son tour poser son verrou.

Dans le cas de bases de données, tables système et index sur des tables système, la base de données doit être en mode mono-utilisateur, afin qu'aucun autre utilisateur ne puisse poser un verrou sur l'objet.

Incidence de la liaison de caches sur les procédures et les triggers stockés

Les liaisons de caches et les tailles d'E/S font partie du plan d'exécution de requête pour les procédures stockées et les triggers. Lorsque vous modifiez la liaison d'un objet à un cache, toutes les procédures stockées faisant référence à cet objet seront recompilées lors de leur prochaine exécution. Lorsque vous modifiez la liaison d'une base de données à un cache, toutes les procédures stockées faisant référence à des objets de cette base qui ne sont pas explicitement liés à un cache seront recompilées lors de leur prochaine exécution.

Configuration de caches de données à l'aide du fichier de configuration

Vous pouvez ajouter ou supprimer des caches de données nommés et reconfigurer des caches existants, ainsi que leurs zones mémoire, en modifiant le fichier de configuration utilisé lorsque vous démarrez Adaptive Server.

Remarque Vous ne pouvez pas reconfigurer de caches et de zones sur un serveur tant qu'il est en fonctionnement. Si vous tentez de lire un fichier de configuration contenant des informations sur les caches et les zones différentes de celles qui sont configurées sur le serveur, la lecture échoue.

Entrées sur les zones et les caches dans le fichier de configuration

A chaque cache de données configuré sur le serveur correspond un bloc d'informations dans le fichier de configuration :

```
[Named Cache:cache_name]
  cache size = {size | DEFAULT}
  cache status = {mixed cache | log only | default data cache}
  cache replacement policy = {DEFAULT |
    relaxed LRU replacement | strict LRU replacement }
```

Vous pouvez spécifier les unités de taille suivantes :

- P – pages (pages Adaptive Server)
- K – kilo-octets (par défaut)
- M – méga-octets
- G – giga-octets

Par exemple, l'entrée du fichier de configuration pour le default data cache est la suivante :

```
[Named Cache:default data cache]
cache size = DEFAULT
cache status = default data cache
cache replacement policy = strict LRU replacement
```

Seule l'entrée default data cache est impérativement requise pour qu'Adaptive Server démarre. Elle doit comporter la taille et l'état du cache, qui doit être "default data cache".

Si ce cache comporte d'autres zones configurées en plus de la zone, le bloc de l'exemple précédent est suivi d'un bloc d'informations pour chaque zone :

```
[16K I/O Buffer Pool]
pool size = size
wash size = size
local async prefetch limit = DEFAULT
```

Remarque Il arrive que l'entrée du fichier de configuration pour la zone mémoire d'un cache soit inexistante. Si vous modifiez le pourcentage de prélecture asynchrone à l'aide de la procédure `sp_poolconfig`, la modification est effectuée uniquement dans les tables système, et non dans le fichier de configuration.

La procédure `sp_cacheconfig` fournit les résultats ci-dessous, suivis des entrées du fichier de configuration correspondant à cette configuration de zones et de caches :

Cache Name	Status	Type	Config Value	Run Value
default data cache	Active	Default	25.00 Mb	29.28 Mb
pubs_cache	Active	Mixed	20.00 Mb	20.00 Mb
pubs_log	Active	Log Only	6.00 Mb	6.00 Mb
tempdb_cache	Active	Mixed	4.00 Mb	4.00 Mb
		Total	55.00 Mb	59.28 Mb

Configuration de caches de données à l'aide du fichier de configuration

```
=====  
Cache: default data cache, Status: Active, Type: Default  
Config Size: 25.00 Mb, Run Size: 29.28 Mb  
Config Replacement: strict LRU, Run Replacement: strict LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	3844 Kb	6.50 Mb	25.28 Mb	10
4 Kb	512 Kb	4.00 Mb	4.00 Mb	10

```
=====  
Cache: pubs_cache, Status: Active, Type: Mixed  
Config Size: 20.00 Mb, Run Size: 20.00 Mb  
Config Replacement: strict LRU, Run Replacement: strict LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	2662 Kb	0.00 Mb	13.00 Mb	10
16 Kb	1424 Kb	7.00 Mb	7.00 Mb	10

```
=====  
Cache: pubs_log, Status: Active, Type: Log Only  
Config Size: 6.00 Mb, Run Size: 6.00 Mb  
Config Replacement: relaxed LRU, Run Replacement: relaxed LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	716 Kb	0.00 Mb	1.00 Mb	10
4 Kb	1024 Kb	5.00 Mb	5.00 Mb	10

```
=====  
Cache: tempdb_cache, Status: Active, Type: Mixed  
Config Size: 4.00 Mb, Run Size: 4.00 Mb  
Config Replacement: strict LRU, Run Replacement: strict LRU  
Config Partition: 1, Run Partition: 1
```

IO Size	Wash Size	Config Size	Run Size	APF Percent
2 Kb	818 Kb	0.00 Mb	4.00 Mb	10

Les informations correspondantes dans le fichier de configuration sont les suivantes :

```
[Named Cache:default data cache]  
cache size = 25M  
cache status = default data cache  
cache replacement policy = DEFAULT
```

```
local cache partition number = DEFAULT

[2K I/O Buffer Pool]
pool size = 6656.0000k
wash size = 3844 K
local async prefetch limit = DEFAULT

[4K I/O Buffer Pool]
pool size = 4.0000M
wash size = DEFAULT
local async prefetch limit = DEFAULT

[Named Cache:pubs_cache]
cache size = 20M
cache status = mixed cache
cache replacement policy = strict LRU
replacement
local cache partition number = DEFAULT

[16K I/O Buffer Pool]
pool size = 7.0000M
wash size = DEFAULT
local async prefetch limit = DEFAULT

[Named Cache:pubs_log]
cache size = 6M
cache status = log only
cache replacement policy = relaxed LRU
replacement
local cache partition number = DEFAULT

[4K I/O Buffer Pool]
pool size = 5.0000M
wash size = DEFAULT
local async prefetch limit = DEFAULT

[Named Cache:tempdb_cache]
cache size = 4M
cache status = mixed cache
cache replacement policy = DEFAULT
local cache partition number = DEFAULT
```

Pour obtenir des informations complémentaires sur le fichier de configuration, reportez-vous au chapitre 5, "Définition des paramètres de configuration".

Avertissement ! Vérifiez la valeur du paramètre de configuration max memory et assurez-vous qu'il reste suffisamment de mémoire pour satisfaire les autres besoins d'Adaptive Server. Si vous attribuez trop de mémoire aux caches de données dans votre fichier de configuration, Adaptive Server ne démarrera pas. Si cela se produit, modifiez le fichier de configuration de manière à diminuer le volume d'espace dans les caches de données ou à augmenter la max memory allouée à Adaptive Server. Pour plus d'informations sur le contrôle de la taille des caches, reportez-vous au chapitre 18, "Configuration de la mémoire",.

Conseils sur la configuration des caches

Les caches définis par l'utilisateur permettent d'améliorer les performances d'Adaptive Server. Ce chapitre traite uniquement de la configuration des caches et des zones mémoire, et de la liaison d'objets aux caches. Les informations relatives aux performances et les conseils en matière de stratégies de test de l'utilisation des caches sont fournis au Chapitre 14, "Utilisation et performances de la mémoire" du document *Performances et optimisation*.

Voici quelques consignes d'ordre général :

- La taille du default data cache ne diminue pas lorsque vous configurez d'autres caches.
- Assurez-vous que la taille de votre default data cache est suffisante pour l'ensemble des activités de cache sur les tables et index non liés. En effet, tous les objets qui ne sont pas liés de manière explicite à un cache utilisent le cache par défaut. Il s'agit des tables système des bases de données utilisateur, des tables système de la base master et de tout autre objet.
- Au cours de la reprise, seule la zone mémoire de 2 ko du cache par défaut est active. Les journaux de transactions sont lus dans la zone de 2 ko du cache par défaut. Toutes les transactions devant être annulées doivent lire les pages de données dans le default data cache. Si la taille de default data cache est trop petite, la reprise peut prendre plus longtemps.

- Veillez à ce que la taille de la zone de 2 ko d'un cache soit suffisante. Dans de nombreux types d'accès aux données, des E/S étendues ne sont pas nécessaires. Par exemple, une requête utilisant un index pour renvoyer une seule ligne à l'utilisateur peut utiliser quatre ou cinq E/S de 2 ko et ne tirer aucun avantage d'une E/S de 16 ko.
- Certaines commandes n'effectuent que des E/S de 2 ko : disk init, certaines commandes dbcc et drop table. dbcc checktable peut effectuer des E/S de grande taille et dbcc checkdb effectue des E/S de grande taille sur les tables et des E/S de 2 ko sur les index.
- Dans le cas de caches utilisés par les journaux de transactions, configurez une zone d'E/S correspondant à la zone d'E/S du journal par défaut. Cette taille est définie pour une base de données à l'aide de sp_logiosize. La valeur par défaut est de 4 ko.
- Si vous tentez de gérer chaque index et objet séparément, ainsi que leur mise en cache, vous perdrez de l'espace de cache. Si vous avez créé des caches ou des zones qui ne sont pas utilisés au mieux par les tables ou les index qui leur sont liés, de l'espace est perdu et des E/S supplémentaires sont créées dans d'autres caches.
- Si la base tempdb est très utilisée par votre application, liez-la à son propre cache. Notez que vous ne pouvez lier que l'intégralité de la base de données tempdb ; vous ne pouvez pas lier des objets individuels de tempdb.
- Dans le cas de caches comportant des taux importants de mise à jour et de remplacement, veillez à ce que la taille de vidage soit suffisamment élevée.
- Sur des systèmes multi-CPU, répartissez les tables et leurs index les plus utilisés sur plusieurs caches pour éviter les conflits de verrous d'attente.
- Envisagez de reconfigurer les caches ou les zones mémoire des caches en fonction de l'évolution des charges de travail. La reconfiguration des caches nécessite le redémarrage du serveur, ce qui n'est pas le cas de la reconfiguration des zones mémoire.

Par exemple, si votre système effectue principalement, pendant le mois, des traitements de transactions en ligne (OLTP) et que l'activité DSS (système décisionnel) est élevée pendant quelques jours, envisagez de transférer de l'espace de la zone de 2 ko vers la zone de 16 ko pour le DSS et de redéfinir la taille des zones pour l'OLTP lorsque la charge du DSS diminue.

Erreurs de fichier de configuration

Si vous modifiez manuellement votre fichier de configuration, contrôlez les tailles de vidage, de zone et de cache. Certaines erreurs dans le fichier de configuration peuvent empêcher le démarrage :

- La taille totale des caches ne peut pas être supérieure au volume de max memory, moins les autres besoins en mémoire d'Adaptive Server.
- La taille totale des zones d'un cache ne peut pas être supérieure à la taille du cache.
- La taille de vidage ne peut pas être trop petite (moins de 20 % de la taille de la zone avec 10 buffers minimum) et ne peut pas dépasser 80 % des buffers de la zone.
- L'état du default data cache doit être "default data cache" et sa taille doit être indiquée, soit par une valeur numérique, soit avec "DEFAULT".
- L'état et la taille de tout cache doivent être spécifiés.
- La taille d'une zone et la taille de vidage des zones supérieures à 2 ko doivent être fournies.
- Les caches définis par l'utilisateur doivent avoir l'état "mixed cache" ou "log only".
- La stratégie de remplacement de cache et le pourcentage de prélecture asynchrone sont facultatifs mais, s'ils sont indiqués, ils doivent porter des valeurs correctes ou "DEFAULT".

Dans la plupart des cas, lorsqu'il manque des données, un message signale une erreur de type "unknown format" sur les lignes suivant l'entrée dont la taille, l'état ou toute autre valeur a été omise. Lorsqu'il s'agit d'une autre erreur, le nom du cache impliqué est indiqué, ainsi que le type de l'erreur. Par exemple, si la taille de vidage d'une zone n'est pas indiquée correctement, le message suivant apparaît :

```
The wash size for the 4k buffer pool in cache
pubs_cache has been incorrectly configured. It must
be a minimum of 10 buffers and a maximum of 80
percent of the number of buffers in the pool.
```


Gestion des serveurs multiprocesseur

Ce chapitre fournit des instructions pour l'administration d'Adaptive Server en environnement multiprocesseur.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Traitement parallèle	681
Définitions	682
Architecture cible	682
Configuration d'un environnement SMP	684

Traitement parallèle

Adaptive Server implémente Sybase Virtual Server Architecture™, qui permet de bénéficier de la fonction de traitement parallèle des systèmes multiprocesseur symétriques (SMP). Vous pouvez exécuter Adaptive Server comme processus unique ou comme processus coopératifs multiples, en fonction du nombre de CPU disponibles et des demandes placées sur la machine du serveur. Ce chapitre aborde les sujets suivants :

- Architecture de la machine cible pour l'environnement SMP d'Adaptive Server
- Architecture Adaptive Server pour les environnements SMP
- Gestion des tâches Adaptive Server en environnement SMP
- Gestion de plusieurs moteurs

Pour plus d'informations sur la conception d'applications pour les systèmes SMP, reportez-vous au chapitre 3, "Utilisation des moteurs et des CPU", du manuel *Performances et optimisation*.

Définitions

Les termes suivants sont employés dans ce chapitre :

- **Processus** : environnement d'exécution programmé sur des CPU physiques par le système d'exploitation.
- **Moteur** : processus exécutant un Adaptive Server qui communique avec les autres processus Adaptive Server par l'intermédiaire de la mémoire partagée. Un moteur peut être considéré comme la puissance de traitement d'une CPU. Il *ne* représente *pas* une CPU particulière. Également appelé **moteur serveur**.
- **Tâche** : environnement d'exécution dans Adaptive Server, programmé sur des moteurs par Adaptive Server.
- **Spécialisation** : décrit un processus au cours duquel une tâche Adaptive Server s'exécute uniquement sur un certain moteur (*spécialisation de tâche*), un moteur traite les E/S réseau pour une certaine tâche (*spécialisation d'E/S réseau*) ou un moteur s'exécute uniquement sur une certaine CPU (*spécialisation de moteur*).
- **Transfert de gestion des E/S réseau** : traitement de transfert de la gestion des E/S réseau d'un moteur à un autre. Les systèmes SMP supportant ce transfert permettent à Adaptive Server de répartir la charge des E/S réseau sur tous ses moteurs.

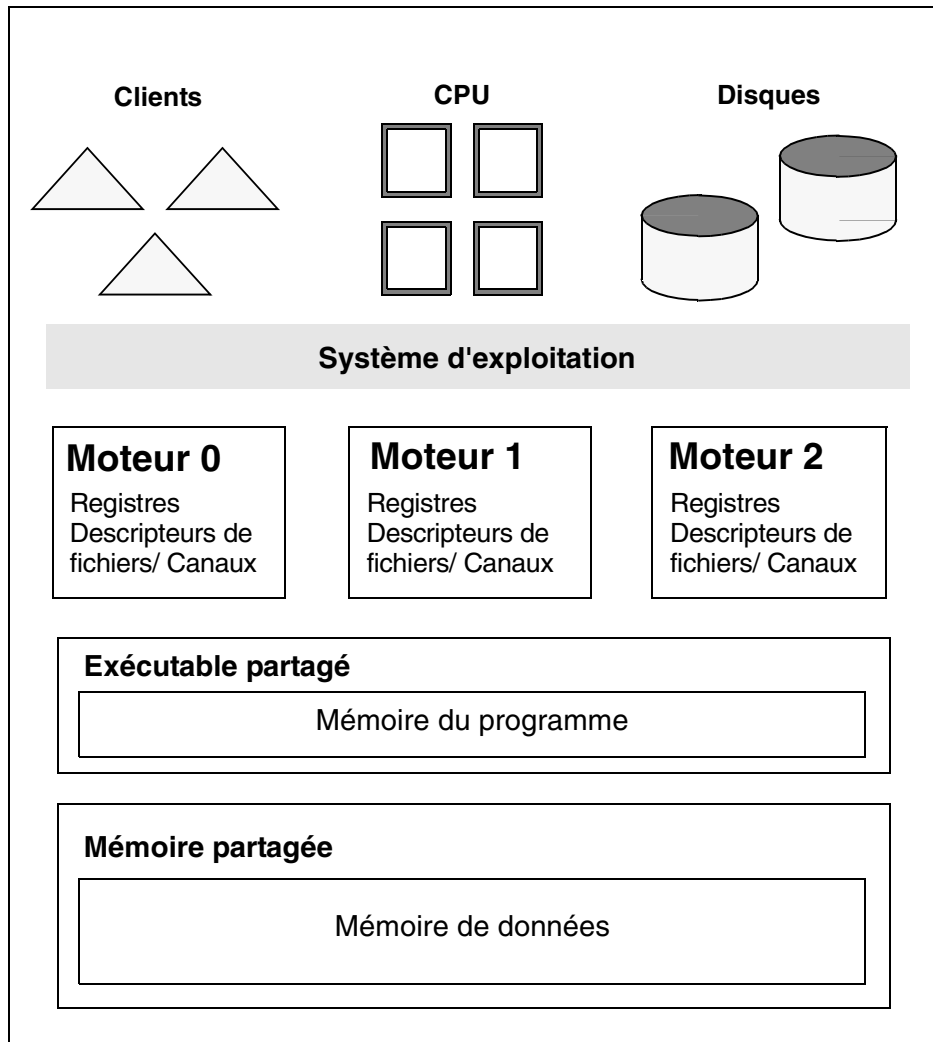
Architecture cible

L'environnement SMP est destiné aux machines présentant les caractéristiques suivantes :

- un système d'exploitation avec multiprocesseurs symétriques,
- une mémoire partagée sur un bus commun,
- 1 à 128 processeurs,
- aucun processeur maître,
- un débit très élevé.

Adaptive Server se compose d'un ou plusieurs processus coordonnés (appelés moteurs) exécutant tous, en parallèle, le programme du serveur. Pour plus d'informations, reportez-vous à la figure 20-1.

Figure 20-1 : Architecture d'un environnement SMP



En cas de connexion de clients à Adaptive Server, les connexions clientes sont transférées vers chacun des moteurs à tour de rôle. Ainsi, tous les moteurs partagent-ils la tâche de gestion des E/S réseau pour les clients. Tous les moteurs sont équivalents et communiquent par le biais de la mémoire partagée.

Les moteurs effectuent l'ensemble des fonctions de base de données, notamment la mise à jour et la journalisation. Adaptive Server, et non le système d'exploitation, programme dynamiquement les tâches clientes sur les moteurs disponibles.

Le système d'exploitation programme les processus des moteurs sur des processeurs physiques. Toute CPU disponible est utilisée pour un moteur quelconque ; il n'y a pas de *spécialisation de moteur*. Le traitement est dit *symétrique* lorsque l'absence de spécialisation entre processus et CPU crée une charge symétrique équilibrée.

Configuration d'un environnement SMP

La configuration d'un environnement SMP est sensiblement la même que celle d'un environnement monoprocesseur, bien que les machines SMP soient plus puissantes et gèrent davantage d'utilisateurs. En outre, l'environnement SMP offre la possibilité de contrôler le nombre de moteurs.

Gestion des moteurs

Pour obtenir les performances optimales d'un système SMP, vous devez conserver le nombre adéquat de moteurs.

Un moteur représente une certaine quantité de puissance CPU. Il s'agit d'une ressource configurable comme la mémoire.

Remarque Si vos connexions au serveur utilisent Component Integration Services, elles sont spécialisées sur un seul moteur et ne seront pas en mesure de migrer d'un moteur vers un autre. Adaptive Server utilise un algorithme de répartition de la charge pour répartir de manière égale la charge entre les moteurs.

Redéfinition du nombre de moteurs

Lors de la première installation d'un Adaptive Server, le système est configuré pour un seul moteur. Pour exploiter plusieurs moteurs, vous devez redéfinir le nombre de moteurs lorsque vous relancez le serveur pour la première fois (ou ultérieurement). Vous pouvez également redéfinir le nombre de moteurs à d'autres moments.

Par exemple :

- Vous pouvez *accroître* le nombre de moteurs si les performances actuelles ne sont pas adaptées à une application *et* que la machine dispose de suffisamment de CPU.
- Vous pouvez *diminuer* le nombre de moteurs si une panne matérielle désactive les CPU de la machine.

Vous devez donc redémarrer le serveur pour redéfinir le nombre de moteurs.

Le paramètre de configuration `max online engines` définit le nombre de moteurs utilisés par Adaptive Server. Redéfinissez ce paramètre à l'aide de `sp_configure`. Par exemple, pour définir 3 moteurs, procédez comme suit :

1 Lancez la commande suivante :

```
sp_configure "max online engines", 3
```

2 Arrêtez et redémarrez le serveur.

Répétez ces étapes chaque fois que vous devez modifier le nombre de moteurs. Les moteurs autres que le moteur 0 sont mis en ligne une fois la reprise achevée.

Choix du nombre approprié de moteurs

Il est important de définir le nombre adéquat de moteurs pour Adaptive Server. Voici quelques conseils :

- Le nombre de moteurs *ne doit jamais* dépasser celui des CPU. Cela pourrait ralentir les performances. Si une CPU est désactivée, réduisez d'une unité le nombre de moteurs (paramètre `max online engines`) à l'aide de `sp_configure` et redémarrez Adaptive Server.

- Le nombre de moteurs doit être égal à celui des CPU *utilisables*. Si le volume de traitements effectués par le client ou d'autres processus non-Adaptive Server est important, un moteur par CPU peut s'avérer excessif. Souvenez-vous également que le système d'exploitation peut occuper une partie d'une CPU.
- Prévoyez un nombre *suffisant* de moteurs. Il est préférable de commencer avec peu de moteurs, puis d'en ajouter lorsque les CPU existantes sont presque entièrement utilisées. S'il y a trop peu de moteurs, la capacité des moteurs existants sera dépassée, ce qui risque d'entraîner des goulets d'étranglement.

Désactivation d'un moteur à l'aide de la commande *dbcc engine*

Vous pouvez modifier dynamiquement le nombre de moteurs utilisés par Adaptive Server à l'aide de la commande *dbcc engine* pour activer ou désactiver un moteur. Cela permet à l'administrateur système de reconfigurer les ressources CPU compte tenu du fait que les besoins en traitement fluctuent avec le temps.

Deux paramètres de configuration limitent le nombre de moteurs disponibles sur le serveur :

- *max online engines* – lors de l'initialisation du serveur, le nombre de moteurs spécifié par *max online engines* est démarré. Le nombre de moteurs ne peut jamais excéder *max online engines*.
- *min online engines* – définit le nombre minimal de moteurs. Lorsque vous désactivez un moteur à l'aide de la commande *dbcc engine*, vous ne pouvez pas réduire le nombre de moteurs en dessous de la valeur définie par *min online engines*.

Etant donné les limites du système d'exploitation quant au nombre de descripteurs de fichier par processus, la réduction du nombre de moteurs entraîne la réduction du nombre de connexions réseau que le serveur peut supporter.

Il n'existe aucun moyen de transférer une connexion réseau créée pour des appels de procédure distante de serveur à serveur, comme des connexions à Replication Server et à XP Server. Vous ne pouvez donc pas désactiver un moteur qui gère l'une de ces connexions.

Syntaxe et utilisation de *dbcc engine*

La syntaxe de *dbcc engine* est la suivante :

```
dbcc engine(offline , [enginenum] )  
dbcc engine("online")
```

Si *enginenum* n'est pas spécifié, le moteur qui a le numéro le plus élevé est mis hors ligne.

Selon votre système d'exploitation et la charge présente sur Adaptive Server, la désactivation d'un moteur est une opération qui peut prendre plusieurs minutes. Pour mener à terme l'opération de désactivation d'un moteur, vous devez procéder comme suit :

- Toutes les E/S en attente pour le moteur doivent être terminées.
- Toutes les tâches associées au moteur doivent être transférées vers d'autres moteurs.
- Le système d'exploitation et l'élimination des données superflues doivent libérer toutes les structures.

Si le transfert des tâches ne peut pas s'opérer en 5 minutes environ, les tâches sont supprimées.

Avertissement ! Si vous utilisez *dbcc engine(offline)* en cas d'utilisation importante de la CPU sur le serveur, il se peut qu'Adaptive Server ne soit pas en mesure de transférer toutes les tâches avant l'expiration du délai imparti. Les tâches qui ne peuvent pas être transférées dans ce délai sont supprimées.

Etats et messages pendant *dbcc engine(offline)*

Lorsqu'un administrateur système émet une commande *dbcc engine(offline)*, des messages sont transmis au journal d'erreurs. Voici par exemple les messages sur Sun Solaris :

```
00:00000:00000:1999/04/08 15:09:01.13 kernel engine  
5, os pid 19441 offline
```

renvoyés immédiatement par *dbcc engine(offline)* ; vous devez surveiller le journal d'erreurs ou vérifier l'état du moteur dans *sysengines* pour savoir si la tâche du moteur désactivé est terminée.

Un moteur avec des connexions réseau ouvertes et utilisant Client Library ne peut pas être désactivé. Toute tentative de désactivation du moteur est suivie du message suivant contenu dans le journal d'erreurs :

```
00:00000:00000:1999/04/08 15:30:42.47 kernel
ueoffline: engine 3 has outstanding ct-lib
connections and cannot be offlined.
```

S'il existe des tâches ne pouvant pas être transférées vers un autre moteur en l'espace de quelques minutes, ces tâches sont supprimées et un message semblable au message suivant est alors transmis au journal d'erreurs :

```
00:00000:00000:1999/04/08 15:20:31.26 kernel
Process 57 is killed due to engine offline.
```

Surveillance de l'état du moteur

Les valeurs contenues dans la colonne status de sysengines assurent le suivi de l'exécution des commandes dbcc engine :

- online – indique que le moteur est activé.
- in offline – indique que dbcc engine(offline) a été exécutée. Le moteur est toujours alloué au serveur, mais ses tâches sont en cours de transfert sur d'autres moteurs.
- in destroy – indique que le transfert de toutes les tâches depuis le moteur a réussi et que le serveur attend sur la tâche au niveau du système d'exploitation pour libérer le moteur.
- in create – indique qu'un moteur est en voie d'activation.

La commande suivante montre le numéro du moteur, son état, le nombre de tâches spécialisées, ainsi que le temps d'activation :

```
select engine, status, affinitied, starttime
from sysengines
```

engine	status	affinitied	starttime
0	online	12	Mar 5 1999 9:40PM
1	online	9	Mar 5 1999 9:41PM
2	online	12	Mar 5 1999 9:41PM
3	online	14	Mar 5 1999 9:51PM
4	online	8	Mar 5 1999 9:51PM
5	in offline	10	Mar 5 1999 9:51PM



Gestion de processus logique et *dbcc engine(offline)*

Si vous utilisez la gestion de processus logique pour relier des connexions ou des applications particulières à des groupes de moteurs, utilisez *dbcc engine(offline)* avec précaution. Si vous désactivez tous les moteurs d'un groupe de moteurs :

- la connexion ou l'application peut s'exécuter sur n'importe quel moteur,
- un message d'information est transmis à la connexion en cours sur le serveur.

Etant donné que l'attribution de la spécialisation de moteur se fait au moment de la connexion d'un client, les utilisateurs déjà connectés ne sont pas transférés si les moteurs du groupe de moteurs sont de nouveau activés à l'aide de *dbcc engine("online")*.

Contrôle de l'occupation des CPU

Pour conserver le nombre approprié de moteurs, gérez l'utilisation de la CPU à l'aide d'un utilitaire du système d'exploitation. Pour connaître l'utilitaire approprié pour votre système d'exploitation, reportez-vous au Manuel de configuration pour votre plate-forme.

Gestion des connexions utilisateur

Si le système SMP supporte le transfert de gestion des E/S réseau, chaque moteur gère les E/S réseau correspondant à ses connexions. Lors de la connexion, Adaptive Server transfère la tâche de connexion cliente du moteur 0 vers le moteur traitant le moins de connexions. La tâche du client gère les E/S réseau sur ce moteur (*spécialisation de réseau*) jusqu'à ce que la connexion soit terminée. Pour déterminer si votre système SMP supporte cette migration, reportez-vous au Manuel de configuration pour votre plate-forme.

En répartissant les E/S réseau sur ses moteurs, Adaptive Server peut traiter davantage de connexions utilisateur. En effet, le nombre de connexions n'est ainsi plus limité par le nombre maximal de descripteurs de fichiers ouverts par processus. L'ajout de moteurs entraîne une augmentation linéaire du nombre maximal de descripteurs de fichiers, stocké dans la variable globale `@@max_connections`.

Lorsque vous augmentez le nombre de moteurs, Adaptive Server affiche la nouvelle valeur de `@@max_connections` à l'écran (console) et dans le journal d'erreurs après redémarrage du serveur. Vous pouvez obtenir cette valeur à l'aide de la requête suivante :

```
select @@max_connections
```

Ce nombre correspond au nombre maximal de descripteurs de fichiers autorisé par le système d'exploitation pour votre processus, moins les descripteurs de fichiers suivants, qui sont utilisés par Adaptive Server :

- un descripteur pour chaque récepteur réseau master sur le moteur 0 (un par ligne "master" de l'entrée du fichier d'interface correspondant à l'Adaptive Server concerné),
- un descripteur par moteur pour la sortie standard,
- un descripteur par moteur pour le journal d'erreurs,
- deux descripteurs par moteur pour la voie de migration de spécialisation réseau,
- un descripteur par moteur pour la configuration,
- un descripteur par moteur pour le fichier d'interface.

Par exemple, si Adaptive Server est configuré pour un moteur et que la valeur de `@@max_connections` est égale à 1019, l'ajout d'un second moteur fait passer la valeur de `@@max_connections` à 2039 (s'il n'y a qu'un récepteur réseau master).

Pour tirer profit de l'élévation de la limite de `@@max_connections`, vous pouvez configurer le paramètre `number of user connections`. Toutefois, chaque fois que vous diminuez le nombre de moteurs à l'aide de `max online engines`, vous devez également modifier la valeur de `number of user connections`. La reconfiguration de `max online engines` ou de `number of user connections` n'est pas dynamique ; vous devez donc redémarrer le serveur pour modifier ces valeurs de configuration. Pour plus d'informations sur la configuration de `number of user connections`, reportez-vous au chapitre 5, "Définition des paramètres de configuration".

Paramètres de configuration qui influent sur les systèmes SMP

Le chapitre 5, "Définition des paramètres de configuration", répertorie les paramètres de configuration pour Adaptive Server. Certains de ces paramètres, comme les taux de verrous d'attente, ne peuvent être appliqués qu'aux systèmes SMP.

Configuration des paramètres de taux de verrous d'attente

Les paramètres du taux de verrous d'attente indiquent le nombre de ressources système internes, comme les lignes dans une table ou un cache interne, qui sont protégées par un **spinlock**. Un verrou d'attente est un mécanisme simple de verrouillage qui empêche un processus d'accéder aux ressources système utilisées par un autre processus. Tous les processus qui essayent d'accéder à ces ressources doivent attendre que le verrou soit libéré.

Les paramètres de configuration du taux de verrous d'attente ne sont utilisés que dans les systèmes multiprocesseur. Un Adaptive Server configuré avec un moteur unique ne comprend qu'un verrou d'attente, quelle que soit la valeur spécifiée pour le paramètre de configuration du taux de verrous d'attente.

Le tableau 20-1 répertorie les ressources système protégées par des verrous d'attente et les paramètres de configuration que vous pouvez utiliser pour modifier le taux de verrous d'attente par défaut.

Tableau 20-1 : Paramètres de configuration du taux de verrous d'attente

Paramètre de configuration	Ressource système protégée
lock spinlock ratio	Nombre de compartiments de hachage de verrouillage
open index hash spinlock ratio	Tables de hachage des descripteurs d'index de métadonnées
open index spinlock ratio	Descripteurs d'index de métadonnées
open object spinlock ratio	Descripteurs de métadonnées d'objets
partition spinlock ratio	Lignes dans les caches de partition internes
user log cache spinlock ratio	Caches de journaux utilisateur

La valeur spécifiée pour un paramètre de taux de verrous d'attente définit la quantité d'une ressource particulière pour un verrou, et non le nombre de verrous. Par exemple, si le taux de verrous d'attente a la valeur 100, Adaptive Server alloue un verrou d'attente toutes les 100 ressources. Le nombre de verrous d'attente alloués par Adaptive Server dépend du nombre total des ressources, ainsi que du taux spécifié. Plus le taux de verrous d'attente spécifié est bas, plus le nombre de verrous sera grand.

Vous pouvez attribuer des verrous d'attente à des ressources système de deux manières différentes :

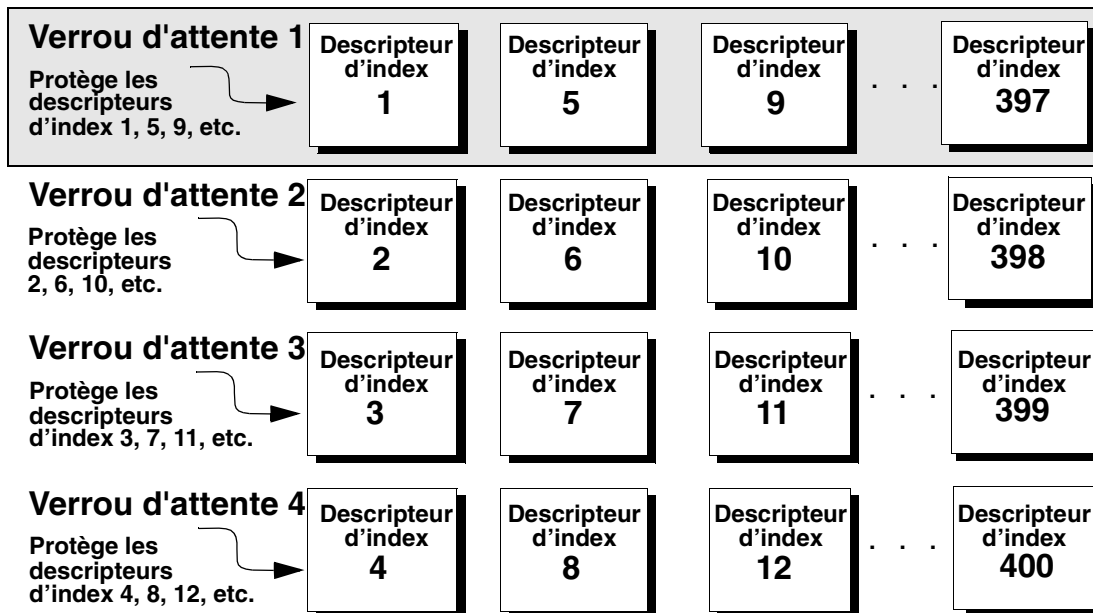
- par une attribution circulaire,
- par une attribution séquentielle,

Attribution circulaire

Les verrous de caches de métadonnées (configurés par les paramètres open index hash spinlock ratio, open index spinlock ratio et open object spinlock ratio) utilisent la méthode d'attribution circulaire.

La figure 20-2 illustre la méthode d'attribution circulaire et montre la relation entre les index et les descripteurs d'index de métadonnées.

Figure 20-2 : Relations entre les verrous d'attente et les descripteurs d'index



Supposons qu'il existe 400 descripteurs d'index de métadonnées, soit 400 lignes dans la table interne des descripteurs d'index. Le taux a la valeur 100. Cela signifie qu'il y aura 4 verrous d'attente en tout : le verrou 1 protège la ligne 1, le verrou 2 protège la ligne 2, le verrou 3 protège la ligne 3 et le verrou 4 protège la ligne 4. Ensuite, le verrou 1 protège le descripteur d'index disponible suivant, le descripteur d'index 5, jusqu'à ce que chaque descripteur d'index soit protégé par un verrou d'attente. Cette méthode d'attribution de descripteur circulaire réduit les risques de conflits entre verrous d'attente.

Attribution séquentielle

Les verrous d'attente des verrous de table, configurés par le paramètre `table lock spinlock ratio`, utilisent la méthode d'attribution séquentielle. La configuration par défaut affecte au taux de verrous d'attente de verrous de table la valeur 20, ce qui correspond à 20 lignes d'une table de hachage interne pour chaque verrou d'attente. Les lignes sont divisées de manière séquentielle : le premier verrou d'attente protège les 20 premières lignes, le second protège les 20 lignes suivantes, etc.

En théorie, l'allocation d'un verrou d'attente à une seule ressource permet de réduire les risques de conflits et d'améliorer la concurrence d'accès. Dans la plupart des cas, la valeur par défaut de ces taux de verrous d'attente est probablement la meilleure pour votre système. Ne modifiez ce taux qu'en cas de conflits entre les verrous d'attente.

Utilisez `sp_sysmon` pour obtenir un rapport sur les conflits entre verrous d'attente. Pour plus d'informations sur les conflits entre les verrous d'attente, reportez-vous au document *Performances et optimisation*.

Création et gestion des bases de données utilisateur

Ce chapitre explique comment créer et gérer des bases de données utilisateur.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Commandes de création et de gestion des bases de données utilisateur	695
Autorisations de gestion des bases de données utilisateur	696
Utilisation de la commande create database	697
Attribution d'espace et de devices aux bases de données	700
Placement du journal de transactions sur un device distinct	702
Utilisation de l'option for load pour la restauration des bases de données	706
Utilisation de l'option with override avec create database	707
Modification de la propriété d'une base de données	708
Utilisation de la commande alter database	709
Utilisation de la commande drop database	711
Tables système gérant l'allocation d'espace	712
Obtention d'informations sur le stockage des bases de données	715

Commandes de création et de gestion des bases de données utilisateur

Le tableau 21-1 répertorie les commandes servant à créer, modifier et supprimer des bases de données utilisateur, ainsi que leurs journaux de transactions.

Tableau 21-1 : Commandes de gestion des bases de données utilisateur

Commande	Tâche
create database...on <i>nom_dev</i> ou alter database...on <i>nom_dev</i>	Met des devices de base de données à la disposition d'une base de données Adaptive Server spécifique. Utilisées sans la clause sur <i>nom_dev</i> , ces commandes allouent de l'espace depuis le groupe par défaut des devices de base de données.
dbcc checktable(syslogs)	Indique la taille du journal.
sp_logdevice	Spécifie le device qui doit stocker le journal lorsque le device de journal courant est saturé.
sp_helpdb	Fournit des informations sur les devices et la taille d'une base de données.
sp_spaceused	Renvoie un résumé sur le volume d'espace de stockage utilisé par une base de données.

Autorisations de gestion des bases de données utilisateur

Par défaut, seul l'administrateur système (SA) dispose de l'autorisation d'exécuter `create database`. Il peut toutefois octroyer à d'autres utilisateurs l'autorisation d'utiliser cette commande. Cependant, dans de nombreux cas, l'administrateur système conserve le monopole de l'utilisation de la commande `create database` afin de contrôler le placement des bases de données et l'allocation des devices de base de données. L'administrateur système crée alors des bases de données pour le compte d'autres utilisateurs et en transfère ensuite le propriétaire aux utilisateurs appropriés.

Pour créer une base de données et transférer le propriétaire à un autre utilisateur, l'administrateur système procède de la façon suivante :

- 1 Il exécute la commande `create database`.
- 2 Il bascule dans cette nouvelle base à l'aide de la commande `use basededonnées`.
- 3 Il exécute la commande `sp_changedbowner`, telle que décrite dans la section "Modification de la propriété d'une base de données", page 708.

Lorsqu'un administrateur système octroie l'autorisation de créer des bases de données, l'utilisateur qui se voit octroyer l'autorisation doit être un utilisateur reconnu de la base de données master, car toutes les bases de données sont créées à partir de la base master.

Bien que l'administrateur système semble opérer en dehors du système de protection, il s'agit en fait d'une précaution quant à la sécurité. Par exemple, si un propriétaire de base de données (DBO) oublie son mot de passe ou supprime par mégarde toutes les entrées de sysusers, l'administrateur système peut rétablir la situation en utilisant les sauvegardes effectuées régulièrement.

L'autorisation d'exécuter alter database ou drop database est octroyée par défaut au propriétaire de la base de données et est automatiquement transférée avec le propriétaire de la base. Les commandes grant et revoke ne permettent pas de modifier l'autorisation d'exécuter alter database ou drop database.

Utilisation de la commande *create database*

Use create database permet de créer des bases de données utilisateur. Pour utiliser cette commande, vous devez disposer de l'autorisation de l'exécuter et vous devez être un utilisateur reconnu de la base master. Avant de créer une nouvelle base de données, entrez toujours use master.

Remarque Chaque fois que vous exécutez la commande create database, sauvegardez la base de données master. En effet, si la base master est endommagée, sa restauration sera ainsi plus simple et plus sûre. Pour plus d'informations, reportez-vous à la section chapitre 28, "Restauration des bases de données système".

Syntaxe de la commande *create database*

La syntaxe de create database est la suivante :

```
create database nom_basededonnées
  [on {default | device_basededonnées} [= taille]
  [, device_basededonnées [= taille]...]
  [log on device_basededonnées [= taille]
  [, device_basededonnées [= taille]...]
  [with {override | default_location = "chemin d'accès"}]
  [for {load | proxy_update}]
```

Le nom d'une base de données doit respecter les règles relatives aux identificateurs. Vous ne pouvez créer qu'une base de données à la fois.

Dans sa plus simple forme, la commande *create database* crée une base de données sur le device de base de données par défaut indiqué dans *master..sysdevices* :

```
create database newpubs
```

Vous pouvez contrôler diverses caractéristiques d'une nouvelle base de données en utilisant les clauses suivantes de *create database* :

- La clause *on* spécifie le nom d'un ou plusieurs devices de base de données, ainsi que l'allocation d'espace, en méga-octets, pour chacun de ces devices. Pour plus d'informations, reportez-vous à la section "Attribution d'espace et de devices aux bases de données", page 700.
- La clause *log* place le **journal de transactions** (c'est-à-dire la table *syslogs*) sur un device de base de données distinct, en lui attribuant la taille par défaut ou la taille spécifiée. Pour plus d'informations, reportez-vous à la section "Placement du journal de transactions sur un device distinct", page 702.
- *for load* indique à Adaptive Server d'omettre l'étape d'initialisation des pages lors de la création de la base de données. Utilisez cette clause lorsque vous avez l'intention de restaurer une sauvegarde dans cette nouvelle base à l'issue de sa création. Pour plus d'informations, reportez-vous à la section "Utilisation de l'option *for load* pour la restauration des bases de données", page 706.
- *with override* permet à Adaptive Server, sur des machines dont l'espace est limité, de conserver les journaux sur des fragments de device autres que ceux des données. Utilisez cette option *uniquement* lorsque vous placez journaux et données sur le même device logique. Pour plus d'informations, reportez-vous à la section "Utilisation de l'option *with override* avec *create database*", page 707.
- *taille* est dans les spécificateurs d'unité suivants : 'k' ou 'K' (kilo-octets), 'm' ou 'M' (méga-octets) et 'g' ou 'G' (giga-octets).

Fonctionnement de la commande *create database*

Lorsqu'un utilisateur disposant des autorisations requises exécute *create database*, Adaptive Server :

- vérifie que le nom de la base de données indiqué est unique ;
- contrôle que les noms de devices de base de données spécifiés sont disponibles ;

- recherche un numéro d'identification disponible pour la nouvelle base de données ;
- alloue de l'espace à la base de données sur les devices spécifiés et met à jour la table master..sysusages en fonction de ces allocations ;
- insère une ligne dans sysdatabases ;
- effectue une copie de la base de données model dans l'espace de la nouvelle base, créant ainsi les tables système de cette base ;
- initialise toutes les pages restantes sur le device de base de données. Si vous créez une base de données pour y charger une sauvegarde, for load omet l'initialisation des pages effectuée à l'issue du chargement.

La nouvelle base de données contient un ensemble de tables système dont les entrées décrivent les tables elles-mêmes. Cette base comporte toutes les modifications que vous avez apportées à la base de données model. Il peut s'agir de :

- l'ajout de noms d'utilisateurs ;
- l'ajout d'objets.
- la définition d'options de base de données. A l'origine, les options dans la base model sont désactivées (off). Si vous voulez que toutes vos bases de données héritent d'options particulières, modifiez ces options dans la base model à l'aide de sp_dboption. Pour plus d'informations sur la base model, reportez-vous au chapitre 2, "Bases de données système et facultatives",. Pour plus d'informations sur la modification des options de base de données, reportez-vous au chapitre 22, "Définition des options de base de données".

Ajout d'utilisateurs aux bases de données

Après avoir créé une base de données, l'administrateur système ou le propriétaire de base de données peut ajouter manuellement des utilisateurs à cette base, à l'aide de sp_adduser. Le responsable de la sécurité du système (SSO) peut apporter son aide si de nouveaux logins Adaptive Server sont requis. Pour plus d'informations sur la gestion des logins et des utilisateurs de base de données Adaptive Server, reportez-vous au chapitre 9, "Administration de la sécurité",.

Attribution d'espace et de devices aux bases de données

Adaptive Server alloue de l'espace de stockage aux bases de données lorsque les utilisateurs entrent la commande `create database` ou `alter database`. `create database` permet d'indiquer un ou plusieurs devices de base de données, ainsi que le volume d'espace sur chacun d'eux à allouer à la nouvelle base.

Remarque Vous pouvez aussi utiliser la clause `log on` pour placer sur un device distinct le journal de transactions d'une base de données de production. Pour plus d'informations, reportez-vous à la section "Placement du journal de transactions sur un device distinct", page 702.

Si vous utilisez le mot-clé `default` ou si vous omettez la clause `on`, Adaptive Server place la base de données sur un ou plusieurs devices de base de données par défaut, spécifiés dans `master..sysdevices`. Pour plus d'informations sur le groupe de devices par défaut, reportez-vous à la section "Désignation des devices par défaut", page 589.

Pour affecter une taille (4 Mo dans l'exemple ci-dessous) à une base de données devant être stockée dans l'emplacement par défaut, utilisez `on default = size` de la façon suivante :

```
create database newpubs
on default = "4M"
```

Pour placer la base de données sur des devices de base de données particuliers, indiquez le nom de ces devices. Vous pouvez demander qu'une base de données soit stockée sur plusieurs devices de base de données, en occupant un volume d'espace différent sur chacun d'eux. Tous les devices de base de données mentionnés dans la commande `create database` doivent figurer dans la table `sysdevices`. Cela signifie que vous devez déjà les avoir initialisés à l'aide de `disk init`. Pour des instructions sur l'utilisation de `disk init`, reportez-vous au chapitre 16, "Initialisation des devices de base de données".

L'instruction suivante crée la base de données `newdb` et lui alloue 3 Mo sur `mydata` et 2 Mo sur `newdata`. La base de données et son journal de transactions résident sur le même device :

```
create database newdb
on mydata = "3M, newdata = "2M"
```

Avertissement ! Placez toujours le journal sur un device de base de données distinct, à moins que vous ne créiez une base de données de petite taille ou peu consultée. Pour créer des bases de données de production, reportez-vous à la section "Placement du journal de transactions sur un device distinct", page 702.

Si le volume d'espace demandé sur un device de base de données n'est pas disponible, Adaptive Server crée la base en lui allouant autant d'espace que possible sur chaque device et affiche ensuite un message indiquant la quantité d'espace allouée sur chacun des devices de base de données. Il ne s'agit pas d'une erreur. Si l'espace disponible sur le device de base de données spécifié est inférieur au minimum requis pour une base de données, `create database` échoue.

Lorsque vous créez (ou modifiez) une base de données sur un device UNIX qui n'utilise pas le paramètre `dsync`, Adaptive Server affiche un message d'erreur dans le journal des erreurs. Par exemple, si vous créez le device "mydata" dans l'exemple donné précédemment qui n'utilise pas `dsync`, un message semblable à celui-ci s'affichera :

Attention : la base de données 'newdb' utilise un device virtuel non sécurisé : 'mydata'. La restauration de cette base ne peut pas être garantie.

Devices et taille de base de données par défaut

Si vous omettez le paramètre `taille` dans la clause `(on)`, Adaptive Server crée la base de données avec la quantité d'espace par défaut. Il s'agit alors de la valeur la plus élevée entre la taille spécifiée par le paramètre de configuration `default database size` et la base de données `model`.

La taille de la base `model` et le paramètre `default database size` ont à l'origine la valeur de la page logique du serveur. Vous pouvez augmenter la taille de `model` en lui allouant plus d'espace à l'aide de la commande `alter database`. Pour modifier la valeur du paramètre de configuration `default database size`, utilisez `sp_configure`. Vous pouvez alors attribuer aux nouvelles bases de données toute taille par défaut comprise entre la taille de la page logique du serveur et 10 000 Mo. Pour des instructions complètes, reportez-vous à la section "default database size", page 199.

Si vous omettez intégralement la clause `on`, la taille de la base de données est celle par défaut, comme décrit ci-avant. L'espace est alloué depuis les devices de base de données par défaut indiqués dans `master..sysdevices` par ordre alphabétique des noms de devices.

Pour obtenir les noms logiques des devices de base de données par défaut, entrez :

```
select name
       where status & 1 = 1
       order by name
```

`sp_helpdevice` affiche également le disque par défaut (`default disk`) dans la description des devices de base de données.

Estimation de l'espace requis

Les choix que vous effectuez en matière d'allocation sont très importants car il est difficile de récupérer de l'espace de stockage une fois que celui-ci a été alloué. Vous avez toujours la possibilité d'ajouter de l'espace ; toutefois, vous ne pouvez libérer l'espace qui a été alloué à une base de données, à moins que vous ne supprimiez cette base au préalable.

Vous pouvez évaluer la taille des tables et index de votre base de données à l'aide de `sp_estspace` ou en la calculant vous-même. Reportez-vous au chapitre 15, "Détermination de la taille des tables et des index", du *Performances et optimisation* pour obtenir des instructions.

Placement du journal de transactions sur un device distinct

La clause `log on` de la commande `create database` place le journal de transactions (c'est-à-dire la table `syslogs`) sur un device de base de données distinct. Sauf dans le cas où vous créez des bases de données très petites qui ne présentent pas d'importance vitale, placez toujours le fichier journal sur un autre device de base de données. En plaçant les journaux sur un device distinct, vous bénéficiez des avantages suivants :

- Vous pouvez utiliser `dump transaction` (sauvegarde des transactions), plutôt que `dump database` (sauvegarde de l'intégralité de la base de données) et ainsi économiser du temps et des bandes.

- Vous pouvez définir une taille fixe pour le journal afin de l'empêcher d'utiliser l'espace requis par d'autres activités de base de données.
- Cette opération crée une gestion par défaut des seuils d'espace disponible sur le segment de journal et permet de créer une autre gestion de l'espace libre sur le journal et les données de la base. Pour plus d'informations, reportez-vous au chapitre 29, "Gestion de l'espace libre avec des seuils".
- Les performances sont améliorées.
- La reprise totale est garantie en cas de panne de disque dur. Un argument spécial de dump transaction permet de sauvegarder le journal de transactions, même si le device de données réside sur un disque endommagé.

Pour spécifier la taille et le device du journal de transactions, utilisez la clause `log on device = taille` de `create database`. La taille est exprimée en 'k' ou 'K' (kilo-octets), 'm' ou 'M' (méga-octets) et 'g' ou 'G' (giga-octets). Par exemple, l'instruction suivante crée la base de données `newdb`, lui alloue 8 Mo sur `mydata` et 4 Mo sur `newdata`, et place un journal de transactions de 3 Mo sur un troisième device de base de données nommé `tranlog` :

```
create database newdb
on mydata = "8M", newdata = "4M"
log on tranlog = "3M"
```

Evaluation de la taille du journal de transactions

La taille du journal de transactions est déterminée par :

- le volume des activités de mise à jour dans la base de données,
- la fréquence des sauvegardes du journal de transactions.

Ceci est vrai, que vous sauvegardiez manuellement le journal de transactions ou que vous utilisiez des procédures de seuil pour automatiser cette tâche. D'une manière générale, allouez au journal entre 10 et 25 pour cent de l'espace total alloué à la base de données.

Les insertions, suppressions et mises à jour augmentent la taille du journal. La commande `dump transaction` la diminue en écrivant les transactions validées sur le disque et en les supprimant du journal. Comme les instructions `update` nécessitent la journalisation des images "avant" et "après" d'une ligne, vous devez envisager, dans le cas d'applications mettant à jour de nombreuses lignes simultanément, d'attribuer au journal de transactions une taille au moins égale au double du nombre de lignes modifiées simultanément, ou deux fois supérieure à la taille de votre plus grande table. Vous pouvez également lancer les mises à jour en **batch** plus petits, en sauvegardant les transactions entre chaque batch.

Dans des bases de données dont le volume d'insertion et de mise à jour est important, les journaux peuvent être rapidement saturés. Contrôlez régulièrement la croissance du journal pour déterminer la taille requise. Vous pourrez ainsi sélectionner des seuils plus appropriés pour le journal et programmer la fréquence des sauvegardes du journal de transactions. Pour contrôler l'espace occupé par le journal de transactions d'une base de données, utilisez d'abord la base de données. Entrez ensuite :

```
dbcc checktable(syslogs)
```

`dbcc` renvoie le nombre de pages de données utilisées par le journal. Si votre journal réside sur un device distinct, `dbcc checktable` indique également le volume d'espace utilisé et le volume d'espace disponible. Par exemple, pour un journal de 2 Mo, le résultat est le suivant :

Vérification de `syslogs`

Le nombre total de pages de données dans cette table est 199.

*** AVERTISSEMENT : L'espace utilisé sur le segment de journal est 0,39 Mo, 19,43 %.

*** AVERTISSEMENT : L'espace libre sur le segment de journal est 1,61 Mo, 80,57 %.

La table contient 1661 lignes de données.

Vous pouvez également utiliser l'instruction `Transact-SQL` suivante pour contrôler la croissance du journal :

```
select count(*) from syslogs
```

Exécutez ces commandes régulièrement pour suivre l'évolution de la taille du journal.

Device et taille de journal par défaut

Si vous omettez le paramètre *taille* dans la clause `log on`, Adaptive Server alloue une page logique de stockage sur le device de journal spécifié. Si vous omettez intégralement la clause `log on`, Adaptive Server place le journal de transactions sur le même device de base de données que les tables de données.

Transfert du journal de transactions sur un autre device

Si vous n'avez pas utilisé la clause `log on` de `create database`, suivez les instructions de cette section pour transférer votre journal de transactions sur un autre device de base de données.

`sp_logdevice` permet de transférer sur un device distinct le journal de transactions d'une base de données dont le journal et les données résident sur le même device. Toutefois, le journal de transactions reste sur le device d'origine jusqu'à ce que la page allouée soit remplie et que le journal ait été sauvegardé.

Remarque Si le journal et la base de données partagent le même device, les exécutions suivantes de `sp_logdevice` n'ont d'incidence que sur les écritures ultérieures dans le journal ; les quelques premières pages écrites dans le journal au moment de la création de la base de données ne sont pas déplacées immédiatement. Des problèmes peuvent alors surgir dans certains cas de reprise et cela n'est pas recommandé.

La syntaxe de `sp_logdevice` est la suivante :

```
sp_logdevice nom_basededonnées, nom_device
```

Le device de base de données que vous spécifiez doit avoir été initialisé à l'aide de `disk init` et alloué à la base de données à l'aide de `create` ou `alter database`.

Pour transférer l'intégralité du journal de transactions sur un autre device :

- 1 Exécutez `sp_logdevice` en indiquant le nom du nouveau device de base de données.

- 2 Exécutez des transactions de manière à remplir la page actuellement utilisée. La quantité d'espace à mettre à jour dépend de la taille de vos pages logiques. Vous pouvez exécuter l'instruction `dbcc checktable(syslogs)` avant et après avoir commencé la mise à jour, afin de savoir quand une nouvelle page est utilisée.
- 3 Attendez que toutes les transactions actives soient achevées. Vous pouvez définir la base de données en mode mono-utilisateur à l'aide de `sp_dboption`.
- 4 Exécutez la commande `dump transaction`, qui permet de supprimer toutes les pages du journal écrites sur le disque. Comme il ne reste plus de transaction active sur la partie journal de l'ancien device, toutes les pages seront supprimées. Pour plus d'informations, reportez-vous au chapitre 26, "Elaboration d'un plan de sauvegarde et de reprise".
- 5 Exécutez `sp_helplog` pour vérifier que l'intégralité du journal réside désormais sur le nouveau device de journal.

Remarque Lorsque vous déplacez un journal de transactions, l'espace qu'il n'utilise plus devient disponible pour les données. Toutefois, le transfert d'un journal de transactions ne vous permet pas de réduire le volume d'espace alloué à un device.

Le chapitre 26, "Elaboration d'un plan de sauvegarde et de reprise", présente en détail les journaux de transactions.

Utilisation de l'option *for load* pour la restauration des bases de données

Lors de la création d'une base de données, Adaptive Server initialise généralement toutes les pages non utilisées sur le device de base de données. Cette opération peut durer de quelques secondes à quelques minutes, selon la taille de la base de données et la vitesse de votre système.

Utilisez l'option `for load` si vous envisagez d'utiliser la base de données pour effectuer une restauration depuis une sauvegarde, que ce soit dans le cas d'une restauration après une panne de disque ou d'un transfert d'une base de données entre deux machines. Lorsque l'option `for load` est utilisée avec `create database`, l'initialisation des pages est omise. Une base de données cible est alors créée et ne peut être utilisée *que* pour le chargement d'une sauvegarde.

Si vous créez une base de données en utilisant `for load`, vous pouvez uniquement exécuter les commandes suivantes dans cette nouvelle base avant d'y charger une sauvegarde :

- `alter database...for load`
- `drop database`
- `load database`

Lorsque vous chargez une sauvegarde de base de données, les allocations du nouveau device de cette base doivent correspondre aux allocations utilisées dans la base sauvegardée. Pour plus d'informations sur la duplication des allocations d'espace, reportez-vous au chapitre 27, "Sauvegarde et restauration de bases de données utilisateur".

Après avoir chargé la sauvegarde dans la nouvelle base de données, vous pouvez utiliser toutes les commandes de votre choix.

Utilisation de l'option *with override avec create database*

Sur des machines dont l'espace est limité, cette option permet de stocker les journaux sur des fragments de device distincts de ceux des données. Utilisez cette option *uniquement* lorsque vous placez journaux et données sur le même device logique. Cette opération n'est pas recommandée dans la pratique, mais c'est souvent la seule option disponible sur des machines dont l'espace de stockage est limité, notamment lorsque vous devez remettre les bases de données en ligne après une panne de disque dur.

Vous aurez toujours la possibilité de sauvegarder votre journal de transactions, mais en cas de panne de disque, vous ne pourrez pas accéder au journal courant du fait qu'il réside sur le même device que les données. Vous ne pourrez effectuer la restauration que jusqu'à la dernière sauvegarde du journal de transactions ; toutes les transactions exécutées entre cette sauvegarde et la panne seront perdues.

Dans l'exemple suivant, le journal et les données résident sur des fragments distincts du même device logique :

```
create database littledb
  on diskdev1 = "4M"
  log on diskdev1 = "1M"
  with override
```

La taille minimale de la base de données que vous pouvez créer est la taille de model.

Modification de la propriété d'une base de données

Un administrateur système peut créer une base de données utilisateur et en attribuer le propriétaire à un autre utilisateur, après avoir effectué certaines tâches. `sp_changedbowner` permet de modifier le propriétaire d'une base de données. Cette procédure ne peut être exécutée que par l'administrateur système, qui doit se trouver dans la base dont le propriétaire doit être modifiée. La syntaxe est la suivante :

```
sp_changedbowner nom_login [, true ]
```

Dans l'exemple suivant, l'utilisateur "albert" devient propriétaire de la base de données courante et les alias des utilisateurs pouvant agir en tant qu'ancien "dbo" sont supprimés.

```
sp_changedbowner albert
```

Le nouveau propriétaire doit déjà avoir un nom de login Adaptive Server, mais ne peut pas être utilisateur de la base de données ni avoir un alias dans cette base. Aussi, avant de modifier le propriétaire d'une base de données, vous pouvez être amené à exécuter `sp_dropuser` ou `sp_dropalias`. Pour plus d'informations sur la modification du propriétaire, reportez-vous au chapitre 9, "Administration de la sécurité",.

Pour transférer les alias et autorisations au nouveau propriétaire de la base de données, indiquez le deuxième paramètre, true.

Remarque Vous ne pouvez pas modifier le propriétaire de la base de données master. Elle appartient toujours au login "sa".

Utilisation de la commande *alter database*

Lorsque votre base de données ou son journal de transactions croît jusqu'à remplir l'espace alloué à l'aide de *create database*, vous pouvez utiliser *alter database* afin d'ajouter de l'espace de stockage. Vous pouvez alors ajouter de l'espace pour les objets de la base, pour le journal de transactions ou pour les deux. Vous pouvez également utiliser *alter database* pour préparer la restauration d'une base de données à partir d'une sauvegarde.

L'autorisation d'exécuter *alter database* est octroyée par défaut au propriétaire de la base de données et est automatiquement transférée à chaque changement de propriétaire. Pour plus d'informations, reportez-vous à la section "Modification de la propriété d'une base de données", page 708. Les commandes *grant* et *revoke* ne permettent pas de modifier l'autorisation d'exécuter *alter database*.

Syntaxe de la commande *alter database*

Pour augmenter la taille d'une base de données et spécifier l'emplacement où doit être ajouté l'espace de stockage, utilisez la syntaxe complète de la commande *alter database* :

```
alter database nom_basededonnées
  [on {default | device_basededonnées} [= taille]
  [, device_basededonnées [= taille]]...]
  [log on {default | device_basededonnées} [= taille]
  [, device_basededonnées [= taille]]...]
  [with override]
  [for load]
  [for proxy_update]
```

Dans sa forme la plus simple, `alter database` ajoute une page logique à partir des devices de base de données par défaut. Si le journal et les données de votre base sont stockés séparément, l'espace ajouté n'est destiné qu'aux données. Pour connaître les noms des devices de base de données par défaut, utilisez la procédure `sp_helpdevice`.

Par exemple, pour ajouter une page logique à la base de données `newpubs` à partir d'un device de base de données par défaut, entrez la commande suivante :

```
alter database newpubs
```

Les clauses `on` et `log on` fonctionnent de la même façon que dans `create database`. Vous pouvez spécifier de l'espace sur un device de base de données par défaut ou sur tout autre device, et vous pouvez indiquer le nom de plusieurs devices de base de données. Si vous utilisez `alter database` pour augmenter la taille de la base de données `master`, vous ne pouvez le faire que sur le device `master`. La valeur minimale pour une augmentation de l'espace est de 1 Mo ou une unité d'allocation.

Pour ajouter 3 Mo à l'espace alloué à la base `newpubs` sur le device de base de données `pubsdata1`, entrez la commande suivante :

```
alter database newpubs  
on pubsdata1 = "3M"
```

Si Adaptive Server ne peut pas allouer la taille requise, il alloue autant d'espace que possible sur chaque device de base de données, avec une allocation minimale de 0,5 Mo (soit 256 pages de 2 ko) par device. Lorsque la commande `alter database` s'achève, un message apparaît indiquant le volume d'espace alloué ; par exemple :

```
Extension de la base de données de 1536 pages sur  
le disque pubsdata1
```

Consultez tous les messages pour vérifier que le volume d'espace requis a été ajouté.

La commande suivante ajoute 2 Mo à l'espace alloué à la base `newpubs` sur `pubsdata1`, 3 Mo sur un nouveau device nommé `pubsdata2` et 1 Mo pour le journal sur `tranlog` :

```
alter database newpubs  
on pubsdata1 = "2M", pubsdata2 = " 3M"  
log on tranlog
```

Remarque Sauvegardez la base de données master chaque fois que vous exécutez la commande `alter database`.

Utilisez `with override` pour créer un fragment de device, comportant l'espace de journalisation, sur un device contenant déjà des données, ou pour créer un fragment de données sur un device déjà utilisé pour le journal. Utilisez cette option uniquement lorsque vous n'avez pas d'autres possibilités de stockage et que la restauration des dernières transactions importe peu.

Utilisez `for load` uniquement après `create database for load` pour recréer l'allocation d'espace de la base de données chargée, depuis une sauvegarde, dans la nouvelle base. Pour plus d'informations sur la duplication des allocations d'espace lors du chargement d'une sauvegarde dans une nouvelle base, reportez-vous au chapitre 27, "Sauvegarde et restauration de bases de données utilisateur".

Utilisation de la commande *drop database*

Utilisez `drop database` pour supprimer une base de données d'Adaptive Server, ainsi que tous ses objets. Cette commande :

- libère l'espace de stockage alloué à la base de données,
- supprime les références à cette base dans les tables système de la base de données master.

Une base de données ne peut être supprimée que par son propriétaire (DBO), et ce, depuis la base master. En outre, vous ne pouvez pas supprimer une base de données qu'un utilisateur a ouverte en lecture ou écriture.

La syntaxe est la suivante :

```
drop database nom_basededonnées [, nom_basededonnées]...
```

Vous pouvez supprimer plusieurs bases de données en une seule instruction, par exemple :

```
drop database newpubs, newdb
```

Avant de supprimer un device de base de données, vous devez supprimer toutes les bases de données qu'il contient. Pour supprimer un device, utilisez la procédure `sp_dropdevice`.

Après avoir supprimé une base de données, sauvegardez la base master afin de garantir sa restauration au cas où elle serait endommagée.

Tables système gérant l'allocation d'espace

Pour créer une base de données sur un device de base de données et lui allouer une certaine quantité de mémoire, Adaptive Server crée d'abord une entrée pour la nouvelle base de données dans `sysdatabases`. Ensuite, il vérifie `master..sysdevices` pour s'assurer que les noms de device spécifiés dans `create database` existent vraiment et sont des devices de base de données. Si vous n'avez indiqué aucun device de base de données ou si vous avez utilisé l'option `default`, Adaptive Server recherche dans `master..sysdevices` et `master..sysusages` l'espace disponible sur l'ensemble des devices pouvant être utilisés pour le stockage par défaut. Cette recherche s'effectue par ordre alphabétique des noms de devices.

L'espace de stockage à partir duquel Adaptive Server recueille le volume de stockage spécifié ne doit pas nécessairement être contigu. L'espace de stockage d'une base de données peut même être extrait de plusieurs devices de base de données. Une base de données est considérée comme un ensemble logique, même si elle est stockée sur plusieurs devices.

Chaque partie de stockage d'une base de données doit représenter au moins une unité d'allocation. La première page de chaque unité d'allocation correspond à la page d'allocation. Contrairement aux autres pages, elle ne contient pas de lignes de la base de données, mais un tableau représentant l'utilisation du reste des pages.

Table *sysusages*

La table `master..sysusages` contient des informations sur le stockage des bases de données. Chaque ligne de cette table représente une allocation d'espace pour une base de données. Ainsi, toute base de données comporte une ligne dans `sysusages` chaque fois que la commande `create database` ou `alter database` lui attribue un fragment d'espace disque.

Lorsque vous installez Adaptive Server, sysusages contient des lignes pour les ID de bases de données (dbids) suivants :

- 1, base de données master,
- 2, base de données temporaire tempdb,
- 3, base de données model,
- 4, base de données sybssystemprocs.

Si vous avez installé le système d'audit, la base sybsecurity aura le dbid 5.

A mesure que des bases de données sont créées ou que la taille des bases de données existantes est augmentée, de nouvelles lignes sont ajoutées à sysusages afin de représenter ces nouvelles allocations.

Sur un Adaptive Server comportant cinq bases de données système et deux bases de données utilisateur (dbid 6 et 7), la table sysusages se présenterait de la façon suivante. Les bases utilisateur ont été créées avec l'option log on. De l'espace de stockage supplémentaire a été attribué à la base de données de dbid 7, à l'aide de deux commandes alter database :

```
select dbid, segmap, lstart, size, vstart
from sysusages
```

dbid	segmap	lstart	size	vstart
1	7	0	1536	4
2	7	0	1024	2564
3	7	0	1024	1540
4	7	0	5120	16777216
5	7	0	10240	33554432
6	3	0	512	1777216
6	4	512	512	3554432
7	3	0	2048	67108864
7	4	2048	1024	50331648
7	3	3072	512	67110912
7	3	3584	1024	67111424

(10 rows affected)

Colonne *segmap*

La colonne *segmap* constitue un masque binaire lié à la colonne *segment* de la table *syssegments* de la base de données utilisateur. Comme le *segment* du journal (*logsegment*) dans toute base utilisateur est le *segment* 2 et que les journaux de ces bases résident sur des devices distincts, la colonne *segmap* contient la valeur 4 (2^2) pour les devices nommés dans l'instruction *log on* et la valeur 3 pour le *segment* de données contenant le *segment* système ($2^0 = 1$) + le *segment* par défaut ($2^1 = 2$).

Les valeurs possibles pour des *segments* contenant des données ou des journaux sont les suivantes :

Valeur	Segment
3	Données uniquement (<i>segments system</i> et <i>default</i>)
4	Journal uniquement
7	Données et journal

Les valeurs supérieures à 7 correspondent à des *segments* définis par l'utilisateur. La colonne *segmap* est expliquée plus en détail à la section du didacticiel sur les *segments* dans le chapitre 23, "Création et utilisation de *segments*".

Colonnes *lstart*, *size* et *vstart*

- Colonne *lstart* – le premier numéro de page dans la base de données de cette unité d'allocation. Toute base de données commence à l'adresse logique 0. La colonne *lstart* prend en compte toute allocation supplémentaire effectuée pour une base, comme dans le cas du *dbid* 7.
- Colonne *size* – le nombre de pages contiguës qui ont été allouées à la même base de données. La dernière adresse logique de cette partie de la base peut être déterminée en ajoutant les valeurs de *lstart* et de *size*.
- Colonne *vstart* – l'adresse à laquelle commence la partie allouée à la base de données. Les 4 bits supérieurs stockent le numéro du device virtuel (*vdevno*) et les 4 bits inférieurs, le numéro du bloc virtuel. Pour obtenir le numéro du device virtuel, divisez la valeur de *sysusages.vstart* ou de *sysdevices.low* par 16 777 216, soit 2^{24} . La valeur de *vstart* identifie le device de base de données contenant ce numéro de page, car elle est comprise entre les valeurs des colonnes *low* et *high* de *sysdevices* pour le device concerné.

Obtention d'informations sur le stockage des bases de données

Cette section explique comment connaître le nom des devices de base de données alloués aux bases et comment déterminer le volume d'espace utilisé par chaque base de données.

Noms et options des devices de base de données

Pour connaître le nom des devices de base de données sur lesquels une base réside, exécutez `sp_helpdb` en indiquant le nom d'une base :

```

                                sp_helpdb pubs2
name      db_size  owner      dbid created      status
-----
pubs2     2.0 MB    sa          5 Aug 25, 1997  no options set

device_fragments  size      usage      free kbytes
-----
pubdev           2.0 MB    data and log      288

device      segment
-----
pubdev      default
pubdev      logsegment
pubdev      system
    
```

`sp_helpdb` indique la taille et l'emploi des devices par la base de données indiquée. La colonne "status" répertorie les options de base de données. Ces options sont décrites au chapitre 22, "Définition des options de base de données".

Si vous utilisez la base de données indiquée, `sp_helpdb` fournit également des informations sur les segments de la base et les devices nommés par les segments. Pour plus d'informations, reportez-vous au chapitre 23, "Création et utilisation de segments".

Obtention d'informations sur le stockage des bases de données

Utilisée sans argument, `sp_helpdb` renvoie des informations sur toutes les bases de données d'Adaptive Server :

```
          sp_helpdb
name      db_size  owner dbid created      status
-----
master    3.0 MB sa      1 Jan 01, 1900 no options set
model     2.0 MB sa      3 Jan 01, 1900 no options set
mydata    4.0 MB sa      7 Aug 25, 1997 no options set
pubs2     2.0 MB sa      6 Aug 23, 1997 no options set
sybsecurity 20.0 MB sa      5 Aug 18, 1997 no options set
sybssystemprocs 10.0 MB sa      4 Aug 18, 1997 trunc log on chkpt
tempdb    2.0 MB sa      2 Aug 18, 1997 select into/
                                     bulkcopy/pllsort
```

Vérification du volume d'espace utilisé

`sp_spaceused` génère :

- un rapport sur l'espace utilisé dans la base de données,
- un rapport sur l'espace utilisé par une table et ses index, ainsi que par le stockage des données de type text/image,
- un rapport sur l'espace utilisé par une table et des informations séparées sur ses index et le stockage des données de type text/image.

Vérification de l'espace utilisé dans une base de données

Pour obtenir un résumé du volume d'espace de stockage utilisé par une base de données, exécutez `sp_spaceused` dans cette base :

```
sp_spaceused
database_name      database_size
-----
pubs2              2.0 MB

reserved    data      index_size    unused
-----
-
1720 KB     536 KB     344 KB       840 KB
```

Le tableau 21-2 décrit les colonnes de ce rapport.

Tableau 21-2 : Colonnes du résultat de `sp_spaceused`

Colonne	Description
database_name	Nom de la base de données étudiée.
database_size	Volume total de l'espace alloué à la base de données à l'aide de <code>create database</code> ou <code>alter database</code> .
réservé	Volume d'espace alloué aux tables et index créés dans la base de données. (L'espace est alloué aux objets d'une base de données par incrément d'1 extent, soit 8 pages, à la fois).
data, index_size	Volume d'espace utilisé par les données et les index.
unused	Volume d'espace réservé aux tables et index existants et non encore utilisé.

La somme de valeurs des colonnes `unused`, `index_size` et `data` doit correspondre à la valeur de la colonne `reserved`. Pour obtenir le volume d'espace non réservé, soustrayez la valeur de `reserved` à `database_size`. Cet espace est disponible pour de nouveaux objets ou pour les objets existants dont la taille devient supérieure à l'espace qui leur est réservé.

En exécutant régulièrement `sp_spaceused`, vous pouvez contrôler le volume d'espace disponible de la base de données. Par exemple, si la valeur de la colonne `reserved` est proche de celle de `database_size`, cela signifie que vous manquerez d'espace pour de nouveaux objets. Si la valeur de `unused` est également peu élevée, vous manquerez d'espace pour ajouter des données.

Vérification des informations d'une table

Vous pouvez également utiliser `sp_spaceused` avec un nom de table comme paramètre :

```
sp_spaceused titles
name  rowtotal reserved  data    index_size unused
-----
titles 18      48 KB    6 KB    4 KB    38 KB
```

La valeur de la colonne rowtotal peut être différente de la valeur obtenue par l'exécution de `select count(*)` sur la table. En effet, `sp_spaceused` calcule cette valeur à l'aide de la fonction intégrée `rowcnt`. Cette fonction utilise les valeurs stockées dans les pages d'allocation. Or, ces valeurs ne sont pas mises à jour régulièrement ; dans le cas de tables sur lesquelles l'activité est très élevée, ces valeurs peuvent donc être très différentes de la réalité. `update statistics`, `dbcc checktable` et `dbcc checkdb` mettent à jour le nombre de lignes par page, la valeur de la colonne rowtotal sera plus précise si vous exécutez d'abord l'une de ces commandes.

Il est judicieux d'exécuter régulièrement `sp_spaceused` sur `syslogs`, car la taille du journal de transactions peut augmenter rapidement si la base de données est souvent modifiée. En effet, si le journal de transactions ne réside pas sur un device distinct, il peut occuper de l'espace destiné à la base de données.

Vérification des informations sur une table et ses index

Pour afficher les informations sur l'espace utilisé par des index individuels, entrez :

```
sp_spaceused titles, 1
index_name      size      reserved  unused
-----
titleidind      2 KB      32 KB     24 KB
titleind        2 KB      16 KB     14 KB

name      rowtotal  reserved  data  index_size unused
-----
titles           18      46 KB    6 KB    4 KB     36 KB
```

L'espace occupé par le stockage des pages de type `text/image` est indiqué à part de l'espace utilisé par la table. Pour le stockage de type `text/image`, le nom de l'objet commence toujours par "t", suivi du nom de la table :

```
sp_spaceused blurbs,1
index_name      size      reserved  unused
-----
blurbs          0 KB      14 KB     12 KB
tblurbs         14 KB     16 KB      2 KB

name      rowtotal  reserved  data  index_size unused
-----
blurbs           6      30 KB    2 KB    14 KB     14 KB
```

Interrogation d'une table système sur l'utilisation de l'espace

Vous pouvez créer vos propres requêtes pour obtenir des informations complémentaires sur le stockage physique. Par exemple, pour déterminer le nombre total de blocs de 2 ko d'espace de stockage sur Adaptive Server, vous pouvez interroger sysdevices :

```
select sum(high - low)
from sysdevices
where status in (2, 3)
-----
              7168
```

Dans la colonne `status`, 2 représente un device physique et 3 représente un device physique qui est aussi un device par défaut.

Définition des options de base de données

Ce chapitre décrit l'utilisation des options de base de données.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Présentation des options de base de données	721
Utilisation de la procédure sp_dboption	722
Description des options de base de données	722
Modification des options de base de données	730
Affichage des options sur une base de données	731

Présentation des options de base de données

Les options de base de données gèrent différents aspects des bases de données, tels que :

- les transactions,
- les valeurs par défaut des colonnes de tables,
- les restrictions concernant l'accès des utilisateurs,
- les performances des opérations de reprise et de bcp,
- les journaux.

L'administrateur système (SA) et le propriétaire d'une base de données (DBO) peuvent utiliser les options de base de données pour configurer les paramètres d'une base de données. Ces options diffèrent des paramètres sp_configure qui conditionnent le serveur dans son intégralité et des options set qui ne concernent que la session ou la procédure stockée courante.

Utilisation de la procédure *sp_dboption*

Utilisez la procédure système *sp_dboption* pour modifier les paramètres d'une base de données dans son intégralité. Les options restent actives jusqu'à ce qu'elles soient modifiées. La procédure *sp_dboption* permet :

- d'afficher une liste exhaustive des options de base de données, lorsqu'elle est utilisée sans paramètre ;
- de modifier une option de base de données, lorsqu'elle est utilisée avec des paramètres.

Vous pouvez modifier les options des bases de données utilisateur uniquement. Vous ne pouvez pas modifier les options de la base de données master. Pour modifier une option de base de données utilisateur (ou pour afficher une liste des options de base de données), exécutez *sp_dboption* tout en utilisant la base de données master.

La syntaxe est la suivante :

```
sp_dboption [nom_base, nom_option, {true | false}]
```

Pour qu'une ou plusieurs options soient appliquées à chaque nouvelle base de données, modifiez les options dans la base de données model.

Description des options de base de données

Pour afficher une liste des options de base de données, tous les utilisateurs ayant accès à la base de données master peuvent exécuter la procédure *sp_dboption* sans paramètres. Le résultat de *sp_dboption* se présente sous la forme suivante :

```
sp_dboption
Options de base de données définissables.
-----
abort tran on log full
allow nulls by default
auto identity
dbo use only
ddl in tran
identity in nonunique index
no chkpt on recovery
no free space acctg
read only
```

```
select into/bulkcopy/pllsort
single user
trunc log on chkpt
trunc. log on chkpt.
unique auto_identity index
```

Pour que le rapport indique les options définies dans une base de données particulière, exécutez la procédure système `sp_helpdb` dans cette base de données.

Les sections suivantes décrivent de manière plus détaillée chaque option de base de données.

abort tran on log full

L'option `abort tran on log full` détermine le traitement d'une transaction lorsque le seuil ultime est dépassé. La valeur par défaut, `false`, signifie que la transaction est interrompue et n'est reprise que lorsque l'espace a été libéré. Si vous attribuez la valeur `true` à cette option, toutes les requêtes utilisateur devant écrire dans le journal de transactions sont annulées tant que vous n'avez pas libéré de l'espace.

allow nulls by default

Lorsque vous attribuez la valeur `true` à l'option `allow nulls by default`, vous activez l'admission par défaut des valeurs null dans les colonnes, conformément aux normes SQL. La valeur par défaut Transact-SQL d'une colonne est `not null`, ce qui signifie que les valeurs NULL ne sont pas admises, à moins de spécifier `null` dans la définition de la colonne pour les commandes `create table` ou `alter table`.

auto identity

Lorsque vous attribuez la valeur true à l'option auto identity, une colonne IDENTITY de 10 chiffres est définie dans chaque table créée sans indication de clé primaire (primary), de contrainte d'unicité (unique) ou de colonne IDENTITY. La colonne IDENTITY est créée uniquement lorsque vous exécutez une commande create table, non lorsque vous exécutez une commande select into. Cette colonne n'est pas visible lorsque vous sélectionnez toutes les colonnes avec l'instruction select *. Pour l'extraire, vous devez spécifier explicitement son nom, SYB_IDENTITY_COL, dans la liste de sélection.

Pour définir la précision de cette colonne IDENTITY, utilisez le paramètre de configuration size of auto identity.

Vous pouvez activer (true) l'option auto identity dans la base tempdb ; toutefois, cette valeur n'est ni reconnue ni utilisée, et les tables temporaires créées ne comportent pas automatiquement la colonne IDENTITY.

dbo use only

Lorsque vous attribuez la valeur true (activée) à l'option dbo use only, la base de données ne peut être utilisée que par son propriétaire.

ddl in tran

Lorsque vous attribuez la valeur true à l'option ddl in tran, les commandes suivantes peuvent être utilisées dans une transaction définie par l'utilisateur :

- alter table (d'autres clauses que partition et unpartition sont autorisées)
- create default
- create index
- create procedure
- create rule
- create schema
- create table
- create trigger

- create view
- drop default
- drop index
- drop procedure
- drop rule
- drop table
- drop trigger
- drop view
- grant
- revoke

Les instructions de définition des données verrouillent les tables système pour la durée d'une transaction, ce qui peut entraîner des problèmes de performances. Utilisez-les uniquement dans des transactions courtes.

Les commandes suivantes ne peuvent en aucun cas être utilisées dans une transaction définie par l'utilisateur :

- alter database
- alter table...partition
- alter table...unpartition
- create database
- disk init
- dump database
- dump transaction
- drop database
- load transaction
- load database
- select into
- truncate table
- update statistics

identity in nonunique index

L'option `identity in nonunique index` inclut automatiquement une colonne `IDENTITY` dans les clés d'index des tables, de sorte que tous les index créés sur ces tables soient uniques. Cette option rend uniques au niveau interne les index qui ne le sont pas de façon logique et leur permet d'être utilisés pour traiter les curseurs modifiables et les lectures de niveau d'isolement 0.

La table doit déjà disposer d'une colonne `IDENTITY` pour que l'option `identity in nonunique index` fonctionne. Pour cela, utilisez une instruction `create table` ou attribuez à l'option de base de données `auto_identity` la valeur `true` avant de créer cette table.

L'option `identity in nonunique index` permet d'utiliser les curseurs et d'effectuer des lectures de niveau d'isolement 0 sur des tables ayant des index non uniques. Un index unique garantit que le curseur est positionné sur la ligne correcte lors de la prochaine lecture (`fetch`) exécutée sur ce curseur.

Ne confondez pas l'option `identity in nonunique index` avec `unique auto_identity index` qui permet d'ajouter une colonne `IDENTITY` à une nouvelle table possédant un index unique non clusterisé.

no chkpt on recovery

L'option `no chkpt on recovery` a la valeur `true` (activée) pour qu'une copie actualisée d'une base de données soit conservée. Dans ce cas, il existe une base de données "primaire" et une base de données "secondaire". A l'origine, la base de données primaire est sauvegardée puis chargée dans la base secondaire. Ensuite, le journal de transactions de la base de données primaire est régulièrement sauvegardé et chargé dans la base secondaire.

Lorsque la valeur `false` (désactivée, il s'agit du paramètre par défaut) est attribuée à cette option, un enregistrement de point de reprise est ajouté à la base de données après sa restauration au redémarrage d'Adaptive Server. Ce point de reprise, qui contrôle que le processus de restauration n'est pas réexécuté inutilement, modifie le numéro de séquence de la base de données. Si ce numéro a été changé sur la base de données secondaire, il est impossible de charger dans cette base une sauvegarde du journal de transactions de la base de données primaire.

L'activation de cette option pour la base de données secondaire empêche cette dernière d'obtenir, lors de la reprise, un point de reprise permettant que toute autre sauvegarde du journal de transactions de la base de données primaire puisse y être chargée.

no free space acctg

L'option *no free space acctg* annule la comptabilisation de l'espace disponible et l'exécution des actions relatives aux seuils pour les segments non utilisés par le journal. Le temps de restauration est ainsi réduit, car l'espace disponible de ces segments n'est pas recalculé. La valeur indiquant les lignes par page stockée pour chaque table est désactivée, ce qui peut entraîner des valeurs incorrectes dans les résultats fournis par les procédures système qui calculent l'utilisation de l'espace.

read only

L'option *read only* signifie que les utilisateurs peuvent extraire des données de la base, mais ne peuvent pas les modifier.

select into/bulkcopy/pllsort

L'option *select into/bulkcopy/pllsort* doit être activée pour que vous puissiez effectuer des opérations qui ne conservent pas un enregistrement complet des transactions dans le journal. Ces actions sont les suivantes :

- lancer l'utilitaire *writetext* ;
- exécuter *select into* dans une table permanente ;
- effectuer un **bulkcopy** "rapide" à l'aide de *bcp*. *bcp* en mode rapide est utilisé par défaut sur les tables sans index ;
- effectuer un tri parallèle.

Adaptive Server n'effectue qu'une journalisation minimale de ces commandes, n'enregistrant que les allocations et les annulations d'allocation de pages, mais pas les modifications réalisées sur les pages de données.

Il n'est pas nécessaire d'activer (on) l'option `select into/bulkcopy/plsort` pour exécuter `select into` sur une table temporaire, puisque `tempdb` n'est jamais restaurée. De même, cette option ne nécessite pas d'être définie pour que vous puissiez exécuter `bcp` sur une table possédant des index, car les insertions sont journalisées.

Après avoir exécuté la commande `select into` ou effectué un `bulkcopy` sur une base de données, vous ne pouvez pas effectuer de sauvegarde normale du journal de transactions. Après avoir apporté à votre base de données des modifications non journalisées, vous devez exécuter `dump database`, puisqu'il est impossible de restaurer ces dernières à partir des journaux de transactions.

La définition de l'option `select into/bulkcopy/plsort` n'empêche pas la sauvegarde du journal, mais les modifications de données non journalisées empêchent l'utilisation normale de `dump transaction`. Cependant, vous pouvez toujours utiliser `dump transaction...with no_log` et `dump transaction...with truncate_only`.

Par défaut, l'option `select into/bulkcopy/plsort` est désactivée pour les nouvelles bases de données. Pour modifier ce paramètre par défaut, activez cette option dans la base de données `model`.

single user

Lorsque la valeur `true` (activée) est attribuée à l'option `single user`, un seul utilisateur à la fois peut accéder à la base de données. Vous ne pouvez pas activer l'option `single user` dans `tempdb`.

trunc log on chkpt

Lorsque la valeur `true` (activée) est attribuée à l'option `trunc log on chkpt`, le journal de transactions est tronqué (les transactions validées sont supprimées) chaque fois que le processus de vérification checkpoint se produit (généralement plusieurs fois par minute), si au moins 50 lignes ont été consignées dans le journal. Le journal n'est *pas* tronqué s'il contient moins de 50 lignes ou si le propriétaire de la base de données exécute manuellement la commande `checkpoint`.

Il peut s'avérer utile d'activer cette option lors des tâches de développement durant lesquelles il n'est pas nécessaire de sauvegarder le journal de transactions. Si cette option est désactivée (valeur par défaut) et que le journal de transactions n'est jamais sauvegardé, ce dernier se remplit de plus en plus et vous risquez de ne plus avoir d'espace dans votre base de données.

Lorsque l'option `trunc log on chkpt` est activée, vous ne pouvez pas sauvegarder le journal de transactions car il est impossible de restaurer les modifications apportées à vos données à partir des sauvegardes du journal de transactions. Utilisez plutôt la commande `dump database`.

Par défaut, l'option `trunc log on chkpt` est désactivée dans les nouvelles bases de données. Pour modifier ce paramètre par défaut, activez cette option dans la base de données `model`.

Avertissement ! Si vous activez l'option `trunc log on chkpt` dans la base de données `model` et que vous souhaitez charger un ensemble de bases de données et de journaux de transactions dans une nouvelle base de données, assurez-vous que vous avez désactivé cette option dans la nouvelle base.

unique auto_identity index

Lorsque vous attribuez la valeur `true` à l'option `unique auto_identity index`, une colonne `IDENTITY` avec un index unique non clusterisé est ajoutée aux nouvelles tables. Par défaut, la colonne `IDENTITY` est de type numérique à 10 chiffres, mais vous pouvez modifier cette valeur à l'aide du paramètre de configuration `size of auto identity column`.

Vous pouvez activer (`true`) l'option `unique auto_identity index` dans la base `tempdb` ; toutefois, cette valeur n'est ni reconnue ni utilisée, et les tables temporaires créées ne comportent pas automatiquement la colonne `IDENTITY`.

L'option `unique auto_identity index` permet de créer des tables possédant automatiquement une colonne `IDENTITY` avec un index unique qui peut être utilisé avec les curseurs modifiables. Un index unique garantit que le curseur sera positionné sur la ligne correcte lors de la prochaine lecture (`fetch`) exécutée sur ce curseur. Utilisez l'option `identity in nonunique index` lorsque vous effectuez des lectures de niveau d'isolement 0 et que vous devez rendre uniques au niveau interne les index qui ne le sont pas de façon logique, afin de permettre l'utilisation de curseurs modifiables.

Dans certains cas, l'option unique `auto_identity index` peut éviter le problème Halloween. Les raisons possibles sont les suivantes :

- Les utilisateurs ne peuvent pas mettre à jour la colonne `IDENTITY` ; elle ne peut donc pas être utilisée pour la modification du curseur.
- La colonne `IDENTITY` est automatiquement créée avec un index unique non clusterisé de sorte qu'elle peut être utilisée pour le balayage du curseur modifiable.

Pour plus d'informations sur le problème Halloween, les colonnes `IDENTITY` et les curseurs, consultez le *Guide de l'utilisateur Transact-SQL*.

Ne confondez pas l'option unique `auto_identity index` avec l'option `identity in nonunique index` qui est utilisée pour rendre uniques tous les index dans une table en ajoutant une colonne `IDENTITY` dans les clés d'index de cette table.

Modification des options de base de données

Seul un administrateur système (SA) ou le propriétaire de la base de données (DBO) peut modifier les options d'une base utilisateur en exécutant `sp_dboption`. Les utilisateurs affectés à un alias du propriétaire de la base de données ne peuvent pas modifier les options d'une base à l'aide de `sp_dboption`.

Pour pouvoir exécuter `sp_dboption`, vous devez vous trouver dans la base de données master. Pour que les modifications soient prises en compte, vous devez ensuite exécuter la commande `checkpoint` sur la base de données pour laquelle l'option a été modifiée.

Souvenez-vous qu'aucune option de la base de données master ne peut être modifiée.

Pour que la base de données `pubs2` ne soit accessible qu'en lecture seule :

```
use master
sp_dboption pubs2, "read only", true
```

Exécutez ensuite la commande `checkpoint` dans la base de données modifiée :

```
use pubs2
checkpoint
```

Pour le paramètre *nom_option* de `sp_dboption`, Adaptive Server comprend toute chaîne unique qui appartient au nom de l'option. Ainsi, pour activer l'option `trunc log on chkpt`, vous pouvez exécuter cette commande :

```
use master
sp_dboption pubs2, trunc, true
```

Si vous entrez une valeur ambiguë pour le paramètre *nom_option*, un message d'erreur s'affiche. Par exemple, il existe deux options de base de données `dbo use only` et `read only`. Si vous utilisez "only" pour le paramètre *nom_option*, un message s'affiche car "only" peut s'appliquer aux deux options. Les noms complets correspondant à la chaîne fournie apparaissent pour que vous spécifiez plus précisément le paramètre *nom_option*.

Plusieurs options de base de données peuvent être activées en même temps. Vous ne pouvez pas modifier les options d'une base de données dans une transaction définie par l'utilisateur.

Affichage des options sur une base de données

Utilisez la procédure `sp_helpdb` pour déterminer les options définies pour une base de données spécifique. `sp_helpdb` répertorie, dans la colonne `status` du résultat, toutes les options de base de données actives.

L'exemple suivant montre que l'option `read only` est activée dans `mydb` :

```

                                sp_helpdb mydb
name db_size owner dbid created      status
-----
mydb  2.0 MB  sa    5    Mar 05, 1995  read only
device_fragments      size      usage      free kbytes
-----
master                2.0 MB  data and log      576
device                segment
-----
master                default
master                logsegment
master                system
name  attribute_class attribute  int_value char_value
      comments
-----
pubs2  buffer manager  cache name      NULL cache for database mydb
      NULL

```

Affichage des options sur une base de données

Pour afficher un résumé des options de toutes les bases de données, utilisez `sp_helpdb` sans spécifier de base de données :

sp_helpdb			
name	db_size	owner	dbid
created	status		

mydb	2.0 MB	sa	5
May 10, 1997	read only		
master	3.0 MB	sa	1
Jan 01, 1997	no options set		
model	2.0 MB	sa	3
Jan 01, 1997	no options set		
sybssystemprocs	2.0 MB	sa	4
Mar 31, 1995	trunc log on chkpt		
tempdb	2.0 MB	sa	2
May 04, 1997	select into/bulkcopy/pllsort		

Création et utilisation de segments

Ce chapitre décrit les procédures système et les commandes nécessaires à l'utilisation de segments (ensembles nommés de devices) dans les bases de données.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Présentation des segments	733
Commandes et procédures nécessaires à la gestion des segments	735
Avantages des segments	736
Création de segments	740
Modification de la portée des segments	740
Attribution d'objets de base de données aux segments	742
Suppression de segments	748
Obtention d'informations relatives aux segments	749
Segments et tables système	752
Didacticiel sur les segments	753

Pour plus d'informations sur l'amélioration des performances du système à l'aide de segments, reportez-vous au chapitre 5, "Gestion de l'emplacement physique des données", dans le manuel *Performances et optimisation*.

Présentation des segments

Un segment est une étiquette pointant sur un ou plusieurs devices de base de données. Les noms de segments sont utilisés avec les commandes `create table` et `create index` pour placer des tables ou des index sur des devices de base de données spécifiques. L'utilisation de segments permet d'améliorer les performances d'Adaptive Server et facilite la gestion par l'administrateur système (SA) ou le propriétaire de base de données (DBO), de l'emplacement, la taille et l'utilisation de l'espace des objets de base de données.

Vous créez des segments dans une base de données pour décrire les devices de base de données alloués à cette base. Chaque base de données Adaptive Server peut contenir, au maximum, 32 segments, y compris les segments définis par le système (reportez-vous à la section "Segments définis par le système", page 734). Avant d'attribuer les noms des segments, vous devez initialiser les devices de base de données à l'aide de disk init, puis les attribuer à la base de données avec create database ou alter database.

Segments définis par le système

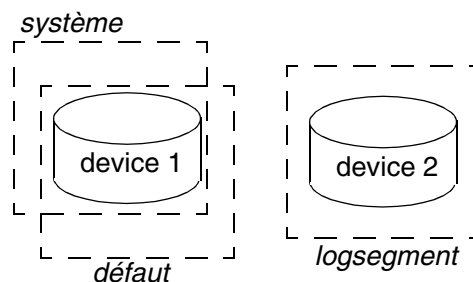
Lorsque vous créez pour la première fois une base de données, Adaptive Server crée automatiquement dans celle-ci trois segments, décrits dans le tableau 23-1.

Tableau 23-1 : Segments définis par le système

Segment	Fonction
système	Stocke les tables système de la base de données.
logsegment	Stocke le journal de transactions de la base de données.
défaut	Stocke tous les autres objets de base de données, à moins que vous n'ayez créé d'autres segments afin d'y stocker les tables ou les index à l'aide de create table...on <i>nom_segment</i> ou de create index...on <i>nom_segment</i> .

Si vous créez une base de données sur un device de base de données unique, les segments system, default et logsegment désignent le même device. Si vous utilisez la clause log on pour placer le journal de transactions sur un device distinct, les segments sont semblables à ceux de la figure 23-1.

Figure 23-1 : Segments définis par le système



Vous pouvez ajouter ou supprimer des segments définis par l'utilisateur, mais vous ne pouvez pas supprimer les segments système, par défaut et de journal d'une base de données. Une base de données doit contenir au moins un segment de chaque type.

Commandes et procédures nécessaires à la gestion des segments

Le tableau 23-2 décrit les commandes et les procédures système permettant de gérer les segments.

Tableau 23-2 : Commandes et procédures nécessaires à la gestion des segments

Commande ou procédure	Fonction
sp_addsegment	Définir un segment dans une base de données.
create table et create index	Créer un objet de base de données sur un segment.
sp_dropsegment	Supprimer un segment d'une base de données ou un device de la portée d'un segment.
sp_extendsegment	Ajouter des devices à un segment existant.
sp_placeobject	Allouer de l'espace sur un segment particulier pour une table ou un index.
sp_helpsegment	Afficher l'allocation du segment pour une base de données ou des données sur un segment particulier.
sp_helpdb	Afficher les segments sur chaque device de base de données. Pour obtenir des exemples, reportez-vous au chapitre 21, "Création et gestion des bases de données utilisateur",.
sp_help	Afficher des informations concernant une table, y compris le segment sur lequel la table réside.
sp_helpindex	Afficher des informations concernant les index d'une table, y compris les segments sur lesquels ils résident.

Avantages des segments

Lorsque vous ajoutez un nouveau device à une base de données, Adaptive Server place ce device dans une zone d'espace par défaut (les segments default et system de la base de données). L'espace disponible de la base de données se trouve ainsi augmenté, mais les objets devant occuper ce nouvel espace ne sont pas définis. Une table ou un index peut s'étendre au point de saturer cette zone d'espace, ne laissant aucune place pour les tables indispensables. Plusieurs tables ou index très sollicités peuvent, en outre, être placés sur un device physique unique dans cette zone par défaut, entraînant une baisse des performances des E/S.

Lorsque vous créez un objet sur un segment, il peut utiliser tous les devices de base de données disponibles sur le segment, mais aucun autre. Les segments permettent de gérer l'espace disponible pour chaque objet.

Les sections suivantes décrivent comment utiliser les segments pour gérer l'utilisation de l'espace disque et améliorer les performances. La section "Transfert d'une table vers un autre device", page 739 explique comment déplacer une table d'un device sur un autre en utilisant les segments et les index clusterisés.

Gestion de l'espace

Si vous attribuez des objets non indispensables à un segment, ces objets ne peuvent s'étendre au-delà de l'espace disponible sur les devices du segment. Inversement, si vous attribuez une table indispensable à un segment et que les devices du segment ne sont pas disponibles pour d'autres segments, aucun autre objet ne peut se voir allouer cet espace.

Lorsque les devices d'un segment sont saturés, vous pouvez agrandir le segment pour y inclure des devices supplémentaires ou des fragments de device, si nécessaire. Vous pouvez également utiliser des seuils, pour être averti lorsque l'espace devient trop restreint sur un segment particulier de base de données.

Si vous créez des segments supplémentaires pour les données, vous pouvez créer des procédures associées au seuil pour chacun d'eux. Pour plus d'informations sur les seuils, reportez-vous au chapitre 29, "Gestion de l'espace libre avec des seuils".

Amélioration des performances

Dans un environnement Adaptive Server comportant plusieurs bases de données et/ou disques, une allocation correcte de l'espace aux bases de données et un emplacement judicieux des objets de base de données sur les devices physiques contribuent à l'amélioration des performances. Normalement, chaque base de données possède l'exclusivité sur les devices de base de données qu'elle utilise, c'est-à-dire qu'elle ne partage pas un disque physique avec une autre base. Dans la plupart des cas, vous pouvez améliorer les performances en plaçant les objets de base de données très utilisés sur des disques physiques dédiés ou en "fractionnant" des tables volumineuses sur plusieurs disques physiques.

Les sections suivantes décrivent ces différents moyens d'améliorer les performances. Pour plus d'informations sur l'amélioration des performances à l'aide des segments, reportez-vous au manuel *Performances et optimisation*.

Séparation des tables, des index et des journaux

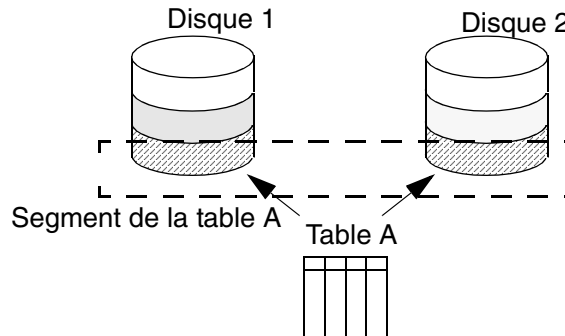
Généralement, le placement d'une table sur un device physique, ses index non clusterisés sur un deuxième device physique et le journal de transactions sur un troisième permet d'améliorer les performances. L'utilisation de devices physiques distincts (contrôleurs de disques) réduit le temps nécessaire aux lectures et écritures sur le disque. Si vous ne pouvez pas organiser tous les devices comme cela, rassemblez au moins tous les index non clusterisés sur un device physique dédié.

L'option `log on` de la commande `create database` (ou `sp_logdevice`) place le journal de transactions sur un disque physique séparé. Utilisez des segments pour placer les tables et les index sur des devices physiques spécifiques. Pour plus d'informations sur l'emplacement des tables et des index sur des segments, reportez-vous à la section "Attribution d'objets de base de données aux segments", page 742.

Fractionnement de tables

La répartition d'une table, volumineuse et très utilisée, entre plusieurs devices situés sur des contrôleurs de disque séparés peut améliorer les performances de lecture d'une table. Lorsqu'une table volumineuse est répartie sur plusieurs devices, il est probable que les lectures brèves simultanées s'effectueront sur des disques différents. La figure 23-2 montre une table partagée entre deux devices sur son segment.

Figure 23-2 : Partitionnement d'une table sur plusieurs devices physiques



Pour fractionner une table sur plusieurs devices, vous pouvez utiliser l'une des trois méthodes suivantes, toutes basées sur l'utilisation de segments :

- utilisez le partitionnement de table ;
- si la table ne possède pas d'index clusterisé, utilisez le chargement partiel ;
- si la table contient des types de données text ou image, séparez la chaîne de texte des autres données.

Partitionnement de tables

Le partitionnement d'une table crée plusieurs chaînes de pages dans la table et entraîne la répartition de ces chaînes de pages sur tous les devices du segment de la table (voir la figure 23-2). Le partitionnement d'une table améliore les performances d'insertions et de lecture, car plusieurs chaînes de pages sont disponibles pour les insertions.

Avant de partitionner une table, vous devez la créer sur un segment contenant le nombre souhaité de devices. La suite de ce chapitre décrit comment créer et modifier des segments. Pour plus d'informations sur le partitionnement des tables à l'aide de la commande `alter table`, reportez-vous au chapitre 5, "Gestion de l'emplacement physique des données", dans le manuel *Performances et optimisation*.

Chargement partiel

Pour fractionner une table ayant un index clusterisé, vous devez utiliser `sp_placeobject` avec plusieurs commandes `load` pour charger les différentes parties de la table sur plusieurs segments. Cette méthode n'est pas facile à utiliser et à gérer, mais elle fournit un moyen de fractionner les tables et leurs index clusterisés sur des devices physiques. Pour plus d'informations, notamment sur la syntaxe à utiliser, reportez-vous à la section "Placement des objets existants sur les segments", page 745.

Séparation des colonnes de type text et image

Adaptive Server stocke les données des colonnes de type text et image sur une chaîne de pages de données distincte. Par défaut, la chaîne de texte est placée sur le même segment que les autres données de la table. Etant donné que la lecture d'une colonne de texte nécessite une opération de lecture effectuée par le pointeur de texte dans la table sous-jacente ainsi qu'une opération de lecture supplémentaire sur la page de texte dans la chaîne de texte distincte, le placement de la chaîne de texte et des données de la table sous-jacente sur des devices physiques distincts permet d'améliorer les performances. Pour plus d'informations, notamment sur la syntaxe à utiliser, reportez-vous à la section "Stockage des pages de texte sur un device distinct", page 747.

Transfert d'une table vers un autre device

Vous pouvez également utiliser des segments pour déplacer une table d'un device sur un autre à l'aide de la commande `create clustered index`. Les index clusterisés, dans lesquels les données réelles se trouvent au niveau inférieur, ou *niveau feuille*, de l'index, résident sur le même segment que la table. Vous pouvez donc déplacer une table entière en supprimant son index clusterisé (le cas échéant), puis créer ou recréer un index clusterisé sur le segment souhaité. Pour plus d'informations, notamment sur la syntaxe à utiliser, reportez-vous à la section "Création d'index clusterisés sur les segments", page 748.

Création de segments

Pour créer un segment dans une base de données vous devez :

- initialiser le device physique à l'aide de disk init ;
- rendre le device de base de données accessible à la base de données en spécifiant la clause on avec la commande create database ou alter database. Ainsi, le nouveau device est automatiquement ajouté aux segments default et system de la base de données.

Une fois que le device de base de données est créé et qu'il est disponible pour la base de données, définissez le segment dans la base à l'aide de la procédure stockée sp_addsegment. La syntaxe est la suivante :

```
sp_addsegment nom_segment, nom_base, nom_device
```

où :

- nom_segment représente un identificateur correct. Attribuez aux segments des noms décrivant leur fonction et utilisez des extensions telles que "_seg".
- nom_base représente le nom de la base de données où le segment sera créé.
- nom_device représente le nom du device de base de données (nom utilisé avec les commandes disk init, create et alter database).

L'instruction suivante crée le segment seg_mydisk1 sur le device de base de données mydisk1 :

```
sp_addsegment seg_mydisk1, mydata, mydisk1
```

Modification de la portée des segments

Lors de l'utilisation de segments, vous devez également gérer leur portée, c'est-à-dire le nombre de devices de base de données sur lesquels chaque segment pointe. Vous pouvez :

- étendre la portée d'un segment, en le faisant pointer sur un ou plusieurs devices supplémentaires, ou
- réduire la portée d'un segment, en le faisant pointer sur un nombre inférieur de devices.

Extension de la portée des segments

Vous pouvez étendre la portée d'un segment si l'objet ou les objets de base de données attribués au segment n'ont plus assez d'espace. La procédure système `sp_extendsegment` permet d'agrandir la taille d'un segment, en ajoutant des devices de base de données à un segment existant. La syntaxe est la suivante :

```
sp_extendsegment nom_segment, nom_base, nom_device
```

Avant d'étendre un segment vous devez :

- spécifier le device de base de données dans `sysdevices` ;
- rendre disponible le device de base de données dans la base de données souhaitée ;
- inclure le nom du segment dans la base de données courante.

L'exemple suivant ajoute le device de base de données `pubs_dev2` à un segment existant appelé `bigseg` :

```
sp_extendsegment bigseg, pubs2, pubs_dev2
```

Pour agrandir le segment `default` dans votre base de données, le mot "default" doit apparaître entre guillemets dans la commande :

```
sp_extendsegment "default", mydata, newdevice
```

Extension automatique de la portée d'un segment

Si vous utilisez `alter database` pour ajouter de l'espace sur un device de base de données nouveau pour cette base, les segments `system` et `default` sont étendus pour inclure ce nouvel espace. Ainsi, la portée des segments `system` et `default` est étendue chaque fois que vous ajoutez un nouveau device à la base de données.

Si vous utilisez `alter database` pour ajouter de l'espace sur un device de base de données existant, tous les segments représentés sur ce device sont étendus pour inclure le nouveau fragment de device. Par exemple, si vous avez initialisé un device de 4 Mo appelé `newdev`, que vous avez alloué 2 Mo de ce device à `mydata` et que vous avez attribué 2 Mo au segment `testseg`, alors :

```
alter database mydata on newdev = "2M"  
sp_addsegment testseg, mydata, newdev
```

Si vous modifiez mydata ultérieurement pour utiliser l'espace restant sur newdev, le fragment d'espace restant est automatiquement représenté sur le segment testseg :

```
alter database mydata on newdev = "2M"
```

Pour d'autres exemples sur l'attribution par Adaptive Server de nouveaux fragments de device à des segments, reportez-vous à la section "Didacticiel sur les segments", page 753.

Réduction de la portée d'un segment

Vous pouvez réduire la portée d'un segment, s'il inclut des devices de base de données que vous souhaitez réserver à d'autres segments. Par exemple, si vous ajoutez un nouveau device de base de données qui doit être exclusivement utilisé par une table, vous devez réduire la portée des segments default et system de sorte que ceux-ci ne pointent plus sur le nouveau device.

Utilisez `sp_dropsegment` pour supprimer uniquement un device de base de données d'un segment et, par conséquent, réduire la portée du segment :

```
sp_dropsegment nom_segment, nom_base, device
```

Avec trois arguments, `sp_dropsegment` supprime uniquement de la portée du segment le device spécifié. Vous pouvez également utiliser `sp_dropsegment` pour supprimer un segment entier de la base de données. Pour ce faire, reportez-vous à la section "Suppression de segments", page 748.

L'exemple suivant supprime le device de base de données `pubs_dev2` de la portée de `bigseg` :

```
sp_dropsegment bigseg, pubs2, pubs_dev2
```

Attribution d'objets de base de données aux segments

Cette section explique comment attribuer des objets de base de données nouveaux ou existants à des segments définis par l'utilisateur pour :

- rassembler les nouveaux objets sur un ou plusieurs devices de base de données ;

- placer une table et son index sur des devices distincts, afin d'améliorer les performances ;
- fractionner un objet existant sur plusieurs devices de base de données.

Création d'objets sur un segment

Pour placer un nouvel objet sur un segment, vous devez d'abord créer ce segment. Vous pouvez également modifier la portée de ce segment (ou d'autres segments), afin qu'il ne pointe que sur les devices de base de données souhaités. Tout nouveau device de base de données ajouté à une base de données est automatiquement ajouté à la portée des segments default et system.

Après avoir défini le segment dans la base de données courante, utilisez `create table` ou `create index` avec la clause facultative `on nom_segment` pour créer l'objet sur le segment. La syntaxe est la suivante :

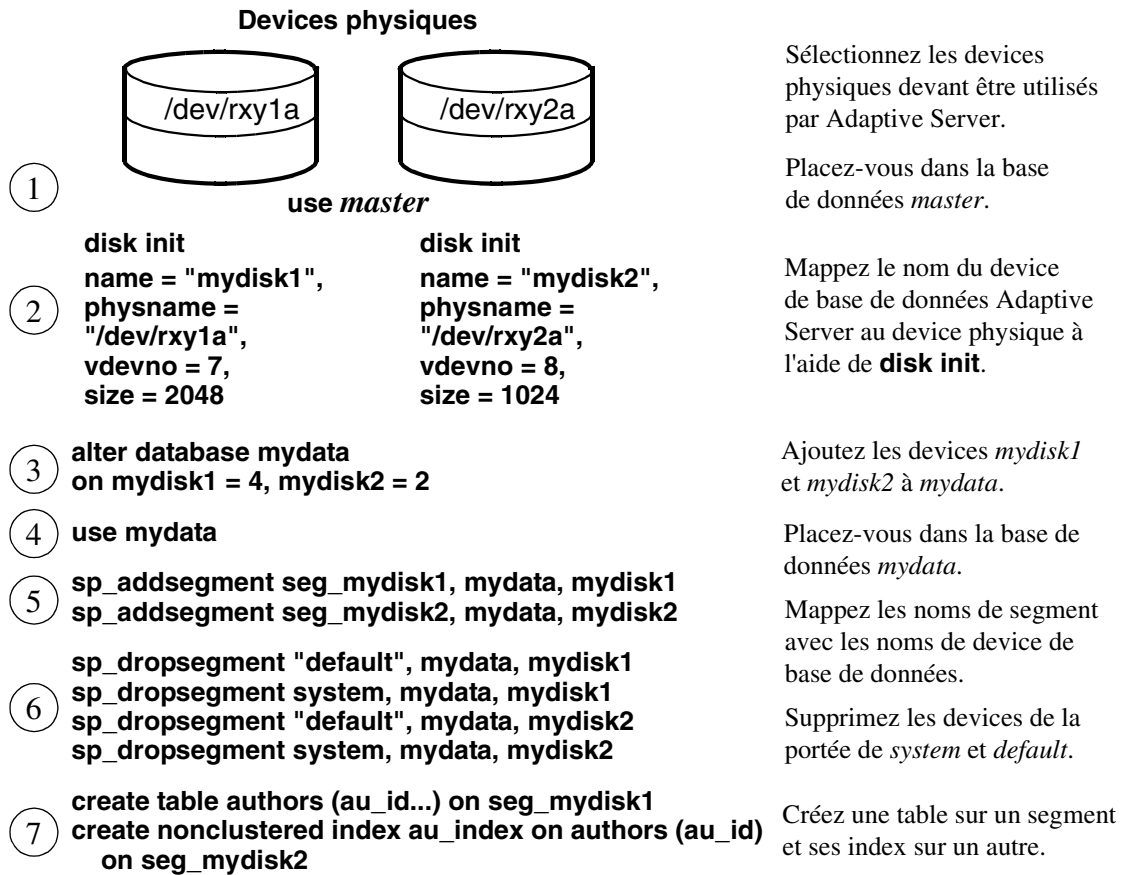
```
create table nom_table (nom_colonne type_données ... )  
  [on nom_segment]  
  
create [ clustered | nonclustered ] index nom_index  
  on nom_table(nom_colonne)  
  [on nom_segment]
```

Remarque Les index clusterisés, dans lesquels les données réelles se trouvent au niveau inférieur, ou niveau feuille, de l'index, résident par définition sur le même segment que la table. Reportez-vous à la section "Création d'index clusterisés sur les segments", page 748.

Exemple : création d'une table et d'un index sur des segments séparés

La figure 23-3 résume la séquence de commandes utilisées pour créer des tables et des index sur des disques physiques spécifiques d'un serveur utilisant une taille de page logique de 2 ko.

Figure 23-3 : Création d'objets sur des devices spécifiques à l'aide des segments



- 1 Commencez par vous placer dans la base de données *master*.
- 2 Initialisez les disques physiques.
- 3 Allouez les nouveaux devices de base de données à une base de données.
- 4 Placez-vous dans la base de données *mydata* par la commande *use basededonnées*.
- 5 Créez deux segments, chacun pointant vers un des nouveaux devices.
- 6 Réduisez la portée des segments *default* et *system* pour qu'ils ne pointent pas vers les nouveaux devices.
- 7 Créez les objets, en donnant des noms aux nouveaux segments.

Placement des objets existants sur les segments

`sp_placeobject` ne supprime pas un objet du segment qui lui est alloué. Cependant, toutes les allocations de disque s'appliquant à cet objet se feront sur le nouveau segment spécifié. La syntaxe est la suivante :

```
sp_placeobject nom_segment, nom_objet
```

Avec la commande suivante, toutes les allocations de disque s'appliquant par la suite à la table `mytab` se feront sur `bigseg` :

```
sp_placeobject bigseg, mytab
```

`sp_placeobject` ne déplace pas d'objet d'un device de base de données à l'autre. Les pages allouées sur le premier device restent allouées ; les données écrites sur le premier device restent sur le device. `sp_placeobject` affecte uniquement les allocations d'espace à venir.

Remarque Pour déplacer une table entière, vous pouvez supprimer son index clusterisé (le cas échéant), puis créer ou recréer un index clusterisé sur le segment souhaité. Pour transférer totalement un index non clusterisé, supprimez l'index et recréez-le sur un nouveau segment. Pour plus d'informations sur le déplacement d'une table, reportez-vous à la section "Création d'index clusterisés sur les segments", page 748.

Si, après avoir utilisé `sp_placeobject`, vous exécutez `dbcc checkalloc`, le message suivant apparaît pour chaque objet fractionné entre plusieurs segments :

```
Extent not within segment: Object object_name, indid
index_id includes extents on allocation page
page_number which is not in segment segment_name.
```

Vous pouvez ignorer ce message.

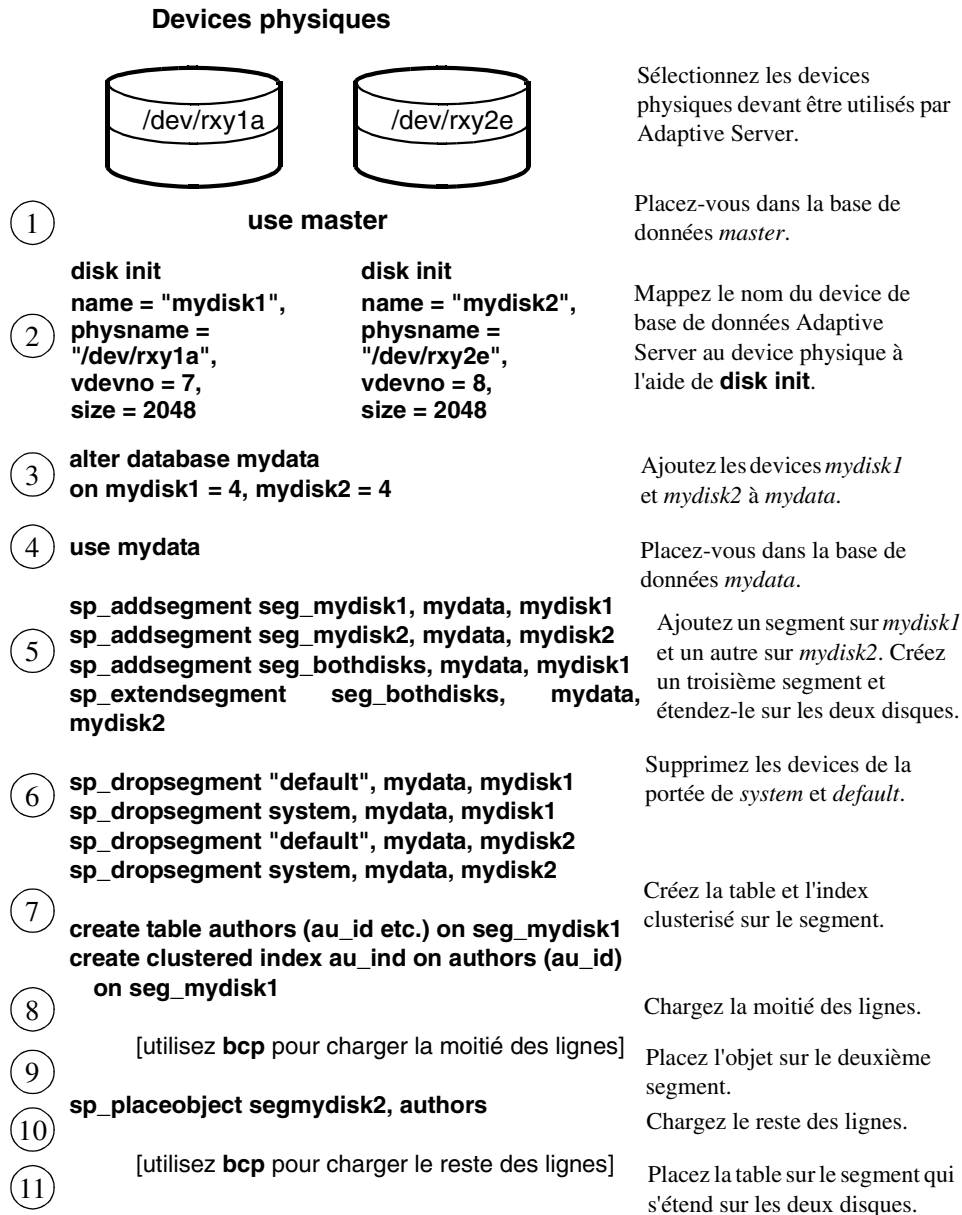
Exemple : fractionnement d'une table et de son index clusterisé entre plusieurs devices

Les performances des applications multi-utilisateur nécessitant beaucoup d'espace peuvent être améliorées si vous fractionnez les tables volumineuses entre des segments situés sur des contrôleurs de disques séparés.

L'ordre des opérations est relativement important à certains stades. En particulier, vous devez créer l'index clusterisé avant de positionner la table sur le deuxième segment.

La figure 23-4 résume la procédure de division d'une table sur deux segments sur un serveur utilisant une taille de page logique de 2 ko :

Figure 23-4 : Division d'une grande table sur deux segments



- 1 Commencez par vous placer dans la base de données master.
- 2 Initialisez les devices avec `disk init` ;
- 3 Attribuez les deux devices à la base de données mydata par la commande `alter database`.
- 4 Placez-vous dans la base de données mydata par la commande `use basededonnées`.
- 5 Créez trois segments. Les deux premiers doivent pointer chacun vers un des nouveaux devices. Etendez la portée du troisième segment pour qu'il désigne les deux devices.
- 6 Supprimez les segments `system` et `default` sur les deux devices.
- 7 Créez la table et son index clusterisé sur le premier segment.
- 8 Chargez la moitié des données de la table sur le premier segment.
- 9 Utilisez `sp_placeobject` pour rediriger toutes les allocations d'espace disque ultérieures vers le deuxième segment.
- 10 Chargez le reste des données sur le deuxième segment.
- 11 Utilisez à nouveau `sp_placeobject` pour placer la table sur le segment qui s'étend sur les deux devices.

L'équilibre d'allocation disque peut évoluer au cours du temps en cas de mises à jour fréquentes de la table. Pour assurer le maintien des avantages de performances, il peut être nécessaire de supprimer la table pour la recréer de temps en temps.

Stockage des pages de texte sur un device distinct

Lorsque vous créez une table comportant des colonnes de type `text` ou `image`, les données sont stockées sur une chaîne distincte de pages de texte. Une table comportant des colonnes `text` ou `image` possède une entrée supplémentaire dans `sysindexes` pour la chaîne de texte ; la colonne de noms `y` est définie avec le nom de la table précédé de la lettre "t" et d'un `indid` de 255. Vous pouvez utiliser `sp_placeobject` pour stocker la chaîne de texte sur un device distinct, en attribuant le nom de la table et le nom de la chaîne de texte à partir de `sysindexes` :

```
sp_placeobject textseg, "mytab.tmytab"
```

Remarque Par défaut, une chaîne de pages de texte est placée sur le même segment que sa table. Une fois que vous avez exécuté `sp_placeobject`, les pages écrites au préalable sur l'ancien device restent allouées, mais toutes les nouvelles allocations d'espace sont effectuées sur le nouveau segment.

Création d'index clusterisés sur les segments

Le niveau inférieur, ou feuille, d'un index clusterisé contient les données de la table. Une table et son index clusterisé se trouvent donc sur le même segment. Si vous créez une table sur un segment et son index clusterisé sur un autre, la table est déplacée avec son index. La table est en effet transférée sur le segment où se trouve l'index clusterisé. Vous pouvez ainsi déplacer rapidement et facilement une table sur d'autres devices de base de données.

La syntaxe correcte pour créer un index clusterisé sur un segment est la suivante :

```
create [unique] clustered index nom_index
on [[basededonnées.]propriétaire.]nom_table (nom_colonne
[, nom_colonne]...)
[with {fillfactor = x, ignore_dup_key, sorted_data,
[ignore_dup_row | allow_dup_row]}]
on nom_segment
```

Pour obtenir un exemple de cette commande, reportez-vous à la section "Segments et index clusterisés", page 757.

Suppression de segments

Lorsque vous utilisez `sp_dropsegment` avec uniquement un nom de segment et le nom de la base de données, le segment nommé est supprimé de la base de données. Cependant, il est impossible de supprimer un segment tant que des objets de base de données lui sont attribués. Vous devez d'abord attribuer les objets à un autre segment ou supprimer les objets, puis le segment.

La syntaxe de suppression d'un segment est la suivante :

```
sp_dropsegment nom_segment, nom_base
```

Il est impossible de supprimer complètement le segment default, system ou log d'une base de données. Une base de données doit contenir au moins un segment default, system et log. Vous pouvez toutefois réduire la portée de ces segments, reportez-vous à la section "Réduction de la portée d'un segment", page 742.

Remarque La suppression d'un segment supprime son nom de la liste de segments de la base de données, mais ne supprime pas les devices de base de données de l'allocation pour cette base, et ne supprime pas les objets des devices.

Si vous supprimez tous les segments d'un device de base de données, l'espace est toujours attribué à la base de données, mais ne peut pas être utilisé pour les objets de base de données. dbcc checkcatalog affiche le message "Missing segment in Sysusages segmap". Pour qu'un device devienne disponible pour une base de données, utilisez sp_extendsegment pour mapper ce device sur le segment par défaut de la base :

```
sp_extendsegment "default", nom_base, nom_device
```

Obtention d'informations relatives aux segments

Il existe quatre procédures système fournissant des informations sur les segments :

- sp_helpsegment répertorie les segments d'une base de données ou affiche des informations concernant un segment spécifique de la base de données ;
- sp_helpdb affiche des informations concernant les relations entre les devices et les segments d'une base de données ;
- sp_help et sp_helpindex fournissent des informations sur les tables et les index, y compris le segment auquel l'objet est attribué.

sp_helpsegment

sp_helpsegment, lorsqu'elle est utilisée sans argument, affiche des informations sur tous les segments de la base de données dans laquelle vous l'exécutez :

```
sp_helpsegment
segment name          status
-----
0 system              0
1 default             1
2 logsegment          0
3 seg1                0
4 seg2                0
```

Pour plus d'informations sur un segment particulier, spécifiez le nom du segment comme argument. Utilisez des guillemets lorsque vous recherchez des informations sur le segment default :

```
sp_helpsegment "default"
```

L'exemple suivant fournit des informations sur seg1 :

```
sp_helpsegment seg1
segment name          status
-----
4 seg1                0

device               size           free_pages
-----
user_data10          15.0MB         6440
user_data11          15.0MB         6440
user_data12          15.0MB         6440

table_name           index_name      indid
-----
customer             customer        0

total_size           total_pages     free_pages       used_pages
-----
45.0MB                23040           19320            3720
```

sp_helpdb

Lorsque vous exécutez `sp_helpdb` dans une base de données et que vous spécifiez le nom de cette dernière, vous voyez des informations concernant les segments de cette base de données.

Par exemple :

```
sp_helpdb mydata
```

```

name          db_size  owner  dbid  created          status
-----
mydata        8.0 MB  sa     4    May 27, 1993    no options set

device_fragments  size          usage          free kbytes
-----
datadev2          4.0 MB       data only      3408
logdev            2.0 MB       log only       2032
seg_mydisk1       2.0 MB       data only      2016

device           segment
-----
datadev2         default
datadev2         system
logdev           logsegment
seg_mydisk1      seg1

```

sp_help et sp_helpindex

Lorsque vous exécutez `sp_help` et `sp_helpindex` dans une base de données et que vous spécifiez un nom de table, vous voyez des informations sur les segments stockant la table ou ses index.

Par exemple :

```
sp_helpindex authors
```

```

index_name      index_description      index_keys
-----
au_index        nonclustered located on seg_mydisk2  au_id

```

Segments et tables système

Il existe trois tables système stockant des informations sur les segments : master..sysusages et deux tables système de la base de données utilisateur, sysindexes et syssegments. sp_helpsegment utilise ces tables. En outre, elle trouve le nom du device de base de données dans sysdevices.

Lorsque vous attribuez un device à une base de données à l'aide de create database ou alter database, Adaptive Server ajoute une ligne à la table master..sysusages. La colonne segmap de la table sysusages fournit des bitmaps des segments de la base de données pour chaque device.

create database crée également la table syssegments dans la base de données utilisateur avec les entrées par défaut suivantes :

segment	name	status
0	system	0
1	default	1
2	logsegment	0

Lorsque vous ajoutez un segment à une base de données à l'aide de sp_addsegment, la procédure :

- ajoute une nouvelle ligne à la table syssegments dans la base de données utilisateur et
- met à jour segmap dans master..sysusages.

Lorsque vous créez une table ou un index, Adaptive Server ajoute une nouvelle ligne à sysindexes. La colonne segment de cette table stocke le numéro du segment, indiquant l'endroit où le serveur allouera de l'espace supplémentaire pour l'objet. Si vous ne précisez pas de nom de segment lorsque vous créez l'objet, celui-ci est placé sur le segment default ; dans le cas contraire, il est placé sur le segment indiqué.

Si vous créez une table contenant des colonnes de type text ou image, une seconde ligne est ajoutée à sysindexes pour la liste liée des pages de texte ; par défaut, la chaîne des pages de texte est placée sur le même segment que la table. Vous trouverez un exemple illustrant l'utilisation de sp_placeobject pour mettre la chaîne de texte sur son propre segment, dans la section "Didacticiel sur les segments", page 753.

La colonne name de la table syssegments est utilisée dans les commandes create table et create index. La colonne status indique le segment par défaut.

Remarque Pour plus d'informations sur la colonne segmap et les tables système gérant le stockage, reportez-vous à la section "Tables système gérant l'allocation d'espace", page 712.

Didacticiel sur les segments

Le didacticiel suivant montre comment créer un segment utilisateur et comment supprimer tous les autres segments du device. Les exemples de cette section prennent le cas d'un serveur utilisant des pages logiques de 2 ko.

Lorsque vous travaillez avec des segments et des devices, n'oubliez pas les caractéristiques suivantes :

- Si vous attribuez de l'espace par fragment, chaque fragment possédera une entrée dans sysusages.
- Lorsqu'un autre fragment de device est attribué à une base de données, tous les segments mappés sur le fragment existant sont mappés sur le nouveau fragment.
- Lorsque vous utilisez alter database pour ajouter de l'espace sur un device qu'une base de données n'utilise pas encore, le segment system et default sont automatiquement mappés sur le nouveau device.

Ce didacticiel commence avec une nouvelle base de données, créée avec un device pour les objets de base de données et un autre pour le journal de transactions :

```
create database mydata on bigdevice = "4M"
log on logdev = "2M"
```

Si vous exécutez alors use mydata, puis sp_helpdb, vous obtenez les résultats suivants :

```
sp_helpdb mydata
```

```

name          db_size  owner    dbid  created        status
-----
mydata        6.0 MB  sa       4    May 27, 1993  no options set

device_fragments  size          usage          free kbytes
-----
bigdevice        4.0 MB       data only      3408
logdev           2.0 MB       log only       2032

device          segment
-----
bigdevice       default
bigdevice       system
logdev          logsegment
```

Comme toutes les nouvelles bases, mydata possède des segments nommés default, system et logsegment. Puisque create database était accompagnée de log on, le segment logsegment est mappé sur son propre device, logdev et les segments default et system sont tous les deux mappés sur bigdevice.

Si vous ajoutez de l'espace sur les mêmes devices de base de données que mydata et que vous exécutez de nouveau sp_helpdb, vous obtenez les entrées suivantes pour les fragments ajoutés :

```
use master
alter database mydata on bigdevice = "2M"
    log on logdev = "1M"
use mydata
sp_helpdb mydata
```

name	db_size	owner	dbid	created	status
mydata	9.0 MB	sa	4	May 27, 1993	no options set

device_fragments	size	usage	free kbytes
bigdevice	2.0 MB	data only	2048
bigdevice	4.0 MB	data only	3408
logdev	1.0 MB	log only	1024
logdev	2.0 MB	log only	2032

device	segment
bigdevice	default
bigdevice	system
logdev	logsegment

Ajoutez toujours de l'espace de journalisation à de l'espace de journalisation et de l'espace de données à de l'espace de données. Si vous essayez d'allouer au journal un segment déjà utilisé pour les données ou inversement, vous devrez utiliser with override. N'oubliez pas que les segments sont mappés sur des devices entiers et pas uniquement sur des fragments d'espace. Si vous modifiez une attribution de segment sur un device, cette modification s'applique à tous les fragments.

L'exemple suivant alloue un nouveau device de base de données non utilisé par mydata :

```
use master
alter database mydata on newdevice = 3
use mydata
sp_helpdb mydata
```

```

name          db_size  owner    dbid  created          status
-----
mydata        12.0 MB sa              4 May 27, 1993  no options set

device_fragments  size          usage          free kbytes
-----
bigdevice        2.0 MB        data only      2048
bigdevice        4.0 MB        data only      3408
logdev           1.0 MB        log only       1024
logdev           2.0 MB        log only       2032
newdevice        3.0 MB        data only      3072

device          segment
-----
bigdevice        default
bigdevice        system
logdev           logsegment
newdevice        default
newdevice        system

```

L'exemple suivant crée un segment appelé `new_space` sur `newdevice` :

```
sp_addsegment new_space, mydata, newdevice
```

Vous trouverez ci-dessous uniquement la partie de `sp_helpdb` concernant le mappage des segments :

```

device          segment
-----
bigdevice        default
bigdevice        system
logdev           logsegment
newdevice        default
newdevice        new_space
newdevice        system

```

Les segments `default` et `system` sont toujours mappés sur `newdevice`. Si vous souhaitez utiliser `new_space` pour stocker une table utilisateur ou un index pour améliorer les performances, en vous assurant que d'autres objets d'utilisateur ne seront pas stockés sur le device par défaut, réduisez la portée des segments `default` et `system` à l'aide de `sp_dropsegment` :

```
sp_dropsegment system, mydata, newdevice
sp_dropsegment "default", mydata, newdevice
```

Le segment `"default"` doit être placé entre guillemets car il s'agit d'un mot réservé Transact-SQL.

Voici un extrait de l'état `sp_helpdb` présentant l'allocation des segments :

```

device                                segment
-----
bigdevice                             default
bigdevice                             system
logdev                                logsegment
newdevice                              new_space

```

Seul `new_space` est mappé sur `newdevice`. Les utilisateurs créant des objets peuvent utiliser la clause `on` avec `new_space` pour placer une table ou un index sur le device correspondant à ce segment. Etant donné que le segment `default` ne pointe pas sur ce device de base de données, les tables et les index créés par les utilisateurs sans la clause `on` ne seront pas placés sur le device.

Si vous utilisez de nouveau `alter database` sur `newdevice`, le nouveau fragment d'espace est représenté sur le même segment que le fragment existant de ce device (c'est-à-dire uniquement sur le segment `new_space`).

A ce stade, si vous utilisez `create table` et que vous indiquez `new_space` comme segment, vous obtiendrez des résultats semblables aux suivants (`sp_help` et `sp_helpsegment`) :

```

create table mytabl (c1 int, c2 datetime)
on new_space
sp_help mytabl

```

```

Name                Owner                Type
-----
mytabl              dbo                  user table

Data_located_on_segment    When_created
-----
May 27 1993  3:21PM
                                new_space

Column_name    Type        Length Nulls Default_name Rule_name
-----
c1              int          4      0 NULL      NULL
c2              datetime     8      0 NULL      NULL

```

Object does not have any indexes.
No defined keys for this object.

```

sp_helpsegment new_space

segment name                                status
-----

```

```

3 new_space 0
device      size      free_pages
-----
newdevice   3.0MB      1528

table_name  index_name  indid
-----
mytabl      mytabl     0

total_size  total_pages free_pages  used_pages
-----
3.0MB      1536      1528      8

```

Segments et index clusterisés

L'exemple suivant crée un index clusterisé sans spécification du nom du segment, en utilisant la table que vous avez créée, dans l'exemple précédent, sur le segment `new_space`. Vérifiez `new_space` après l'exécution de la commande `create index` pour vous assurer qu'aucun objet ne reste sur le segment :

```

/* Don't try this at home */
create clustered index mytabl_cix
  on mytabl(c1)
sp_helpsegment new_space
segment name      status
-----
3 new_space      0

device      size      free_pages
-----
newdevice   3.0MB      1528

total_size  total_pages free_pages  used_pages
-----
3.0MB      1536      1528      8

```

Si vous avez placé une table sur un segment et que vous devez créer un index clusterisé, utilisez la clause `on nom_segment`, sinon la table sera transférée sur le segment default.

Utilisation de la commande *reorg*

Les activités de mise à jour effectuées sur une table peuvent mener à une utilisation inefficace de l'espace et à une réduction des performances. La commande *reorg* permet de réorganiser l'espace des tables pour améliorer les performances.

Ce chapitre aborde les sujets suivants :

Sujet	Page
Sous-commandes de <i>reorg</i>	759
Cas d'utilisation de la commande <i>reorg</i>	761
Utilisation de l'utilitaire <i>optdiag</i> pour évaluer le besoin d'une réorganisation (<i>reorg</i>)	761
Déplacement de lignes redirigées vers leurs pages d'origine	762
Récupération de l'espace inutilisé provenant des suppressions et des mises à jour	763
Récupération de l'espace inutilisé et annulation de la redirection de lignes	764
Reconstruction d'une table	765
Options <i>resume</i> et <i>time</i> de réorganisation des tables de grande taille	767
Utilisation de la commande <i>reorg rebuild</i> sur les index	769

Sous-commandes de *reorg*

La commande *reorg* comporte quatre sous-commandes, qui offrent différents types et niveaux de réorganisation :

- *reorg forwarded_rows* annule la redirection de lignes.
- *reorg reclaim_space* récupère l'espace non utilisé laissé sur une page suite à des suppressions et à des mises à jour qui raccourcissent les lignes.
- *reorg compact* récupère de l'espace et annule la redirection des lignes.

- `reorg rebuild` annule la redirection des lignes et récupère l'espace de page non utilisé, comme `reorg compact`. Par ailleurs, `reorg rebuild` :
 - réécrit toutes les lignes pour les harmoniser avec l'index clusterisé d'une table, le cas échéant ;
 - écrit des lignes dans les pages de données pour les harmoniser avec toute modification apportée aux paramètres de gestion de l'espace à l'aide de `sp_chgattribute` ;
 - supprime et recrée tous les index qui appartiennent à la table.

Les sous-commandes `reclaim_space`, `forwarded_rows` et `compact` :

- causent peu d'interférences avec les autres activités, car elles font appel à de nombreuses transactions de petite taille et de courte durée. Chaque transaction est limitée à huit pages de traitement `reorg`.
- Offrent des options `resume` et `time`, qui permettent de limiter le temps d'exécution d'une commande `reorg` et de reprendre une `reorg` à partir du point où la précédente s'est arrêtée. Cela permet, par exemple, de réorganiser (`reorg`) les tables de grande taille par une série de réorganisations partielles, effectuées aux heures creuses. Pour plus d'informations, reportez-vous à la section "Options `resume` et `time` de réorganisation des tables de grande taille", page 767.

Les considérations ci-dessous s'appliquent à la sous-commande `rebuild` :

- `reorg rebuild` maintient un verrou de table exclusif pendant l'intégralité de sa durée. Sur une table de grande taille, cela peut représenter une période considérable. Cependant, `reorg rebuild` accomplit toutes les opérations effectuées par la suppression et la recréation d'un index clusterisé et prend moins de temps. De plus, `reorg rebuild` reconstruit la table en faisant appel à tous ses paramètres de gestion de l'espace en vigueur. La suppression et la recréation d'un index ne fait pas appel au paramètre de gestion de l'espace `reservepagegap`.
- Dans la plupart des cas, `reorg rebuild` requiert de l'espace disque complémentaire, égal à la taille de la table en cours de reconstruction, plus ses index.

Les restrictions ci-dessous s'appliquent :

- la table spécifiée dans la commande doit utiliser au choix le plan de verrouillage des lignes de données (`datarows`) ou des pages de données (`datapages`) ;

- vous devez être administrateur système ou le propriétaire de l'objet pour exécuter la commande *reorg* ;
- vous ne pouvez pas exécuter *reorg* au sein d'une transaction.

Cas d'utilisation de la commande *reorg*

La commande *reorg* s'avère utile quand :

- un grand nombre de lignes redirigées provoquent des E/S supplémentaires au cours des opérations de lecture ;
- les insertions et les lectures pouvant être effectuées en série sont lentes, car elles s'effectuent sur des pages comportant de l'espace libre non contigu qui doit être récupéré ;
- les opérations avec E/S étendues sont lentes du fait des faibles rapports de clusterisation des pages de données et d'index ;
- *sp_chgattribute* a été utilisé pour modifier un paramètre de gestion d'espace (*reservepagegap*, *fillfactor* ou *exp_row_size*) et la modification doit être appliquée à toutes les lignes et pages existantes d'une table et pas simplement aux mises à jour à venir.

Utilisation de l'utilitaire *optdiag* pour évaluer le besoin d'une réorganisation (*reorg*)

Pour évaluer s'il est nécessaire ou non d'exécuter *reorg*, vous pouvez utiliser les statistiques de la table *systabstats* et l'utilitaire *optdiag*. La table *systabstats* contient les statistiques sur l'utilisation de l'espace dans les tables et l'utilitaire *optdiag* génère des états fondés sur les statistiques des tables *systabstats* et *sysstatistics*.

Pour plus d'informations sur la table *systabstats*, reportez-vous au document *Performances et optimisation*. Pour plus d'informations sur l'utilitaire *optdiag*, reportez-vous au document *Utilitaires pour UNIX*.

Récupération d'espace sans la commande *reorg*

Plusieurs types d'activités récupèrent ou réorganisent l'espace d'une table page par page :

- les insertions, lorsqu'une insertion trouve une page qui disposerait de suffisamment d'espace si l'espace inutilisé était récupéré ;
- la commande `update statistics` (pages d'index seulement) ;
- la recréation d'index clustérisés ;
- le gestionnaire de tâches, si `enable housekeeper GC` prend la valeur 1.

Chacune de ces activités a ses limites et peut s'avérer insuffisante s'il est nécessaire de récupérer ou de réorganiser l'espace sur un nombre élevé de pages. Ainsi, par exemple, les insertions sont susceptibles de s'exécuter plus lentement lorsqu'elles doivent récupérer de l'espace et de ne pas avoir d'effet si de nombreuses pages comportent de l'espace à réorganiser. La récupération d'espace à l'aide du gestionnaire de tâches compacte l'espace inutilisé, mais elle ne s'effectue que lorsque aucune autre tâche ne demande de temps CPU ; et risque donc de ne pas atteindre toutes les pages qui en ont besoin.

Déplacement de lignes redirigées vers leurs pages d'origine

Si une mise à jour rend une ligne trop longue pour qu'elle tienne sur sa page, cette ligne est redirigée vers une autre page. Une référence à la ligne est conservée sur la page initiale, ou la page *d'origine* et tous les accès à la ligne redirigée s'effectuent via cette référence. Par conséquent, il est toujours nécessaire d'accéder à deux pages pour atteindre une page renvoyée. Si un balayage doit lire un grand nombre de pages redirigées, les E/S occasionnées par les accès à la page supplémentaire ralentissent les performances.

La commande `reorg forwarded_rows` annule la redirection de lignes en remplaçant une ligne redirigée dans sa page d'origine ou, si l'espace est suffisant, en supprimant la ligne pour la réinsérer dans une nouvelle page d'origine.

Vous pouvez consulter des statistiques sur le nombre de lignes redirigées d'une table comme suit : lancez une requête sur la table systabstats et faites appel à l'utilitaire optdiag.

Syntaxe de *reorg forwarded_rows*

La syntaxe de *reorg forwarded_rows* est la suivante :

```
reorg forwarded_rows nom_table  
[with {resume, time = nb_minutes}]
```

Pour plus d'informations sur les options *resume* et *time*, reportez-vous à la section "Options *resume* et *time* de réorganisation des tables de grande taille", page 767.

reorg forwarded_rows ne s'applique pas aux index, car ceux-ci ne contiennent aucune ligne redirigée.

Utilisation de *reorg compact* pour supprimer les redirections de lignes

reorg forwarded_rows se fonde sur des indications de la page d'allocation pour rechercher les lignes redirigées. Cela lui confère une grande rapidité, car la recherche ne s'effectue pas sur l'intégralité d'une table, mais signifie également que certaines lignes redirigées risquent de ne pas être trouvées. Après avoir exécuté *reorg forwarded_rows*, vous pouvez évaluer son efficacité en consultant la valeur "Forwarded row count" renvoyée par l'utilitaire *optdiag*. Si la valeur "Forwarded row count" est élevée, vous pouvez alors exécuter *reorg compact*, qui examine une table page par page pour annuler tous les redirections de lignes.

Récupération de l'espace inutilisé provenant des suppressions et des mises à jour

Lorsqu'une tâche effectue une opération de suppression ou de mise à jour qui raccourcit la longueur des lignes, l'espace vide n'est pas récupéré au moment de la transaction, car il est conservé pour le cas où la transaction serait annulée. Si de nombreuses suppressions ou mises à jour qui raccourcissent les lignes sont effectuées sur une table, l'espace non récupéré peut s'accumuler au point de nuire aux performances.

La commande `reorg reclaim_space` récupère l'espace inutilisé laissé par les suppressions ou les mises à jour. Sur chacune des pages qui comportent de l'espace provenant de suppressions validées ou de mises à jour ayant raccourci les lignes, `reorg reclaim_space` ré-écrit les lignes restantes en contigu, de sorte à regrouper tout l'espace inutilisé en fin de page. Si toutes les lignes ont été supprimées et qu'il n'en reste aucune, `reorg reclaim_space` libère la page.

Vous pouvez consulter les statistiques sur le nombre de suppressions de ligne non récupérées dans la table `systabstats` et à l'aide de l'utilitaire `optdiag`. Il n'existe aucune méthode directe d'évaluation de l'espace inutilisé provenant des mises à jour ayant raccourci les lignes.

Syntaxe de `reorg reclaim_space`

La syntaxe de `reorg reclaim_space` est la suivante :

```
reorg reclaim_space nom_table [nom_index]
[with {resume, time = nb_minutes}]
```

Si vous indiquez seulement un nom de table, seules les pages de données de celle-ci seront réorganisées de façon à récupérer l'espace inutilisé. Si vous indiquez le nom d'un index, seules les pages de l'index sont réorganisées.

Pour plus d'informations sur les options `resume` et `time`, reportez-vous à la section "Options `resume` et `time` de réorganisation des tables de grande taille", page 767.

Récupération de l'espace inutilisé et annulation de la redirection de lignes

La commande `reorg compact` combine les fonctions de `reorg reclaim_space` et `reorg forwarded_rows`. Utilisez `reorg compact` quand :

- La reconstruction de l'intégralité d'une table (`reorg rebuild`) ne s'avère pas nécessaire, toutefois la redirection de lignes et l'espace non utilisé provenant des suppressions et des mises à jour risquent de nuire aux performances.
- Il existe une quantité importante de lignes redirigées. Voir "Utilisation de `reorg compact` pour supprimer les redirections de lignes", page 763.

Syntaxe de *reorg compact*

La syntaxe de reorg compact est la suivante :

```
reorg compact nom_table
[with {resume, time = nb_minutes}]
```

Pour plus d'informations sur les options *resume* et *time*, reportez-vous à la section "Options *resume* et *time* de réorganisation des tables de grande taille", page 767.

Reconstruction d'une table

Faites appel à la commande reorg rebuild quand :

- Les E/S étendues ne sont pas sélectionnées pour les requêtes là où elles sont généralement utilisées et l'utilitaire optdiag indique un faible rapport de clusterisation pour les pages de données, les lignes de données ou les pages d'index.
- Vous avez utilisé *sp_chgattribute* pour modifier un ou plusieurs des paramètres de gestion d'espace *exp_row_size*, *reservepagegap* ou *fillfactor* et vous souhaitez que ces modifications s'appliquent non seulement aux données futures mais aussi aux lignes et pages existantes. Pour plus d'informations sur *sp_chgattribute*, reportez-vous au document *Manuel de référence d'Adaptive Server*.

Si une table doit être reconstruite du fait d'un faible rapport de clusterisation, une modification de ses paramètres de gestion de l'espace peut également s'avérer nécessaire (reportez-vous à la section "Modification des paramètres de gestion de l'espace avant d'utiliser reorg rebuild", page 766).

La commande reorg rebuild utilise les paramètres courants de gestion de l'espace d'une table pour récrire les lignes dans la table selon l'index clusterisé de celle-ci, s'il existe. Tous les index de la table sont supprimés, puis recréés avec les valeurs en vigueur de gestion de l'espace pour *reservepagegap* et *fillfactor*. Après l'exécution de rebuild, une table ne comporte plus aucune ligne redirigée, ni d'espace inutilisé provenant de suppressions ou de mises à jour.

Syntaxe de *reorg rebuild*

La syntaxe de reorg rebuild est la suivante :

```
reorg rebuild nom_table
```

Conditions requises pour l'exécution de *reorg rebuild*

Avant d'exécuter *reorg rebuild* sur une table :

- définissez l'option de base de données *select into/bulkcopy/pllsort* sur la valeur *true* et exécutez *checkpoint* dans la base de données ;
- assurez-vous que de l'espace disque supplémentaire, égal à la taille de la table et de ses index, est disponible.

Pour définir *select into/bulkcopy/pllsort* sur la valeur *true* et exécuter *checkpoint* dans la base de données, vous pouvez faire appel aux commandes *isql* ci-dessous :

```
1> use master
2> go
1> sp_dboption pubs2,
    "select into/bulkcopy/pllsort", true
2> go
1> use pubs2
2> go
1> checkpoint
2> go
```

Suite à une exécution de *rebuild* sur une table :

- vous devez sauvegarder la base de données qui contient la table avant de pouvoir sauvegarder le journal de transactions ;
- les statistiques de distribution sont mises à jour ;
- toutes les procédures stockées qui référencent la table seront recompilées à leur exécution suivante.

Modification des paramètres de gestion de l'espace avant d'utiliser *reorg rebuild*

Lorsque la commande *reorg rebuild* reconstruit une table, elle réécrit toutes les lignes de la table et de l'index selon les paramètres courants de la table pour *reservepagegap*, *fillfactor* et *exp_row_size*. Toutes ces propriétés ont un effet sur la rapidité à laquelle les insertions provoquent la fragmentation d'une table, mesurée par un rapport de clusterisation faible.

S'il semble qu'une table se fragmente trop rapidement et nécessite des reconstructions trop fréquentes, il peut s'avérer nécessaire de modifier les paramètres de gestion de l'espace de la table avant d'exécuter *reorg rebuild*.

Pour modifier les paramètres de gestion d'espace, faites appel à la procédure système `sp_chgattribute` (reportez-vous au document *Manuel de référence d'Adaptive Server*). Pour plus d'informations sur les paramètres de gestion de l'espace, reportez-vous au document *Performances et optimisation*.

Options *resume* et *time* de réorganisation des tables de grande taille

Utilisez les options *resume* et *time* de la commande `reorg` lorsque la réorganisation de l'intégralité d'une table représente une tâche trop longue, qui risque de gêner d'autres activités de la base de données. L'option *time* vous permet d'exécuter une `reorg` pendant une durée spécifiée. L'option *resume* permet de lancer une `reorg` à partir du point où la `reorg` précédente s'est arrêtée. Combinées, ces deux options permettent de réorganiser une table de grande taille en plusieurs opérations de réorganisation partielles (par exemple, pendant les heures creuses).

Les options *resume* et *time* ne fonctionnent pas avec `reorg rebuild`.

Syntaxe de *resume* et *time*
dans les commandes
`reorg`

La syntaxe des options *resume* et *time* est la suivante :

```
reorg reclaim_space nom_table [nom_index]
    [with {resume, time = nb_minutes}]

reorg forwarded_rows nom_table
    [with {resume, time = nb_minutes}]

reorg compact nom_table
    [with {resume, time = nb_minutes}]
```

Les considérations ci-dessous doivent être prises en compte :

- Si vous spécifiez seulement l'option *resume*, l'opération de `reorg` commence au point d'arrêt de la précédente et continue jusqu'à la fin de la table.
- Si vous spécifiez seulement l'option *time*, l'opération de `reorg` commence au début de la table et continue pendant le nombre indiqué de minutes.
- Si vous indiquez les deux options, l'opération de `reorg` commence au point d'arrêt de précédente et continue pendant le nombre indiqué de minutes.

Spécification du *nb_minutes* dans l'option *time*

L'argument *nb_minutes* de l'option *time* fait référence au temps écoulé et non au temps CPU. Par exemple, pour exécuter *reorg compact* pendant 30 minutes à partir du moment où une commande *reorg compact* précédente s'est arrêtée, tapez :

```
reorg compact nom_table with resume, time=30
```

Si le processus de *reorg* se met en veille au cours des 30 minutes, cette période est comptée dans le temps écoulé et ne s'ajoute pas à la durée de la *reorg*.

Lorsque la période spécifiée s'est écoulée, *reorg* enregistre dans la table *sysabstats* des statistiques sur la partie de la table ou de l'index qui a été traitée. Ces informations sont utilisées pour redéfinir le point de départ d'une *reorg* avec l'option *resume*. Les points de redémarrage de chacune des trois sous-commandes qui utilisent les options *resume* et *time* sont gérés séparément. Il n'est, par exemple, pas possible de démarrer une réorganisation avec *reorg reclaim_space*, puis de poursuivre avec *reorg compact*.

Si *nb_minutes* est spécifié et que l'opération de *reorg* atteint la fin d'une table ou d'un index avant que le temps ne soit écoulé, elle revient au début de l'objet et se poursuit jusqu'à la fin du délai.

Remarque Les options *resume* et *time* permettent de réorganiser l'intégralité d'une table ou d'un index en plusieurs étapes. Toutefois, si des mises à jour sont apportées entre les exécutions de *reorg*, certaines pages risquent d'être traitées deux fois et d'autres pas du tout.

Utilisation de la commande *reorg rebuild* sur les index

La commande *reorg rebuild* permet de reconstruire les index tout en laissant la table accessible pour les activités de lecture et de mise à jour.

Syntaxe

La syntaxe de reconstruction d'un index est la suivante :

```
reorg rebuild nom_index
```

Commentaires

Pour utiliser *reorg rebuild*, vous devez être le propriétaire de la table de la base de données ou disposer de privilèges d'administrateur système.

Si le nom d'index n'est pas précisé, la reconstruction porte sur toute la table.

Si l'index est précisé, la reconstruction ne porte que sur cet index.

Les conditions d'utilisation de la commande *reorg rebuild* sur un index sont moins contraignantes que pour les tables. Les règles ci-dessous doivent être prises en compte :

- Il n'est pas nécessaire de positionner *select into* pour reconstruire un index.
- La reconstruction d'une table nécessite un espace suffisant pour une copie complète de la table. La reconstruction d'un index s'effectue par petites transactions et désalloue les pages après leur copie ; cette opération ne nécessite donc que l'espace nécessaire pour la copie des pages à chaque transaction.
- Il est possible de reconstruire l'index sur une table alors que des balayages sont actifs au niveau transaction (lecture de données modifiées).

Restrictions

La commande *reorg* s'applique à toutes les tables qui utilisent le verrouillage de lignes de données ou de pages de données. Il est impossible d'utiliser la commande *reorg* pour une table utilisant un verrouillage de toutes les pages.

reorg ne peut pas être utilisée sur un index de texte (nom de la table *sysindexes* associée à une chaîne de texte).

Vous ne pouvez pas exécuter *reorg* dans une transaction.

Il est possible d'utiliser la commande *dump tran* sur une table après reconstruction de son index. En revanche, la commande *dump tran* n'est pas utilisable si la totalité de la table a été reconstruite.

Il est possible de reconstruire l'index de la table *sysabststats*, mais non d'utiliser *reorg rebuild* sur la table elle-même.

Bien que la reconstruction d'un index en ligne soit autorisée sur un index de placement, elle ne reconstruit que les pages d'index. Les pages de données ne sont pas modifiées, donc les lignes de données ne sont ni triées ni réécrites dans des pages nouvelles. Il est possible de reconstruire les pages de données en supprimant un index de placement puis en le recréant.

Méthode de reconstruction des index par *reorg rebuild nom_index*

La reconstruction d'un seul index réécrit toutes les lignes de l'index dans des pages nouvelles. Cela améliore les performances grâce à :

- l'amélioration de la clusterisation au niveau feuille de l'index ;
- l'application à l'index des valeurs stockées pour le facteur de remplissage, qui peut réduire le nombre de ruptures de pages ;
- l'application de la valeur éventuellement stockée pour *intervalle_page_réservée*, qui peut aider à réserver des pages pour des ruptures à venir.

Pour réduire les conflits avec les utilisateurs dont les requêtes nécessitent l'utilisation de l'index, reorg rebuild ne verrouille qu'un petit nombre de pages à la fois. La reconstruction d'un index est une série de transactions indépendantes, dont certaines transactions sont imbriquées. Chaque transaction imbriquée reconstruit environ 32 pages, et chaque transaction externe environ 256 pages. Les verrous d'adresses sont posés sur les pages en cours de modification et libérés à la fin de l'action de niveau supérieur. Les pages qui ne sont plus allouées dans une transaction ne sont pas réutilisables avant le début de la transaction suivante.

Si la commande reorg rebuild s'arrête, les transactions validées ne sont pas annulées. Donc, la partie réorganisée est bien clusterisée avec l'utilisation de l'espace spécifié, mais la partie qui n'a pas été réorganisée est dans le même état qu'avant la commande. L'index reste cohérent au niveau logique.

Remarque La reconstruction de l'index clusterisé n'a aucun impact sur les pages de données de la table. Seules les pages de niveau feuille et les niveaux d'index supérieurs sont affectés. Les pages qui ne sont pas de niveau feuille au-dessus du niveau 1 ne sont pas reconstruites.

Espace nécessaire pour la reconstruction d'un index

Si aucune valeur n'est indiquée pour fill_factor ou reservepagegap, la reconstruction d'un index nécessite un espace supplémentaire d'environ 256 pages ou moins dans le segment de données. L'espace nécessaire pour le journal est supérieur à celui nécessaire pour supprimer l'index et le recréer par la commande create index, mais il ne doit représenter qu'une petite fraction de la taille de l'index. Plus il y a d'espace libre disponible, plus la clusterisation de l'index sera performante.

Remarque La commande reorg rebuild peut ne pas reconstruire les parties de l'index qui sont déjà clusterisées correctement avec utilisation de l'espace appropriée.

Caractéristiques de performances

Les lectures d'index sont plus rapides après l'utilisation de la commande reorg.

L'utilisation de la commande reorg sur une table peut avoir un effet négatif sur les performances des requêtes simultanées.

Messages d'état

L'utilisation de la commande reorg rebuild *nom_index* sur une grande table peut prendre un certain temps. Des messages d'état sont envoyés périodiquement pour indiquer à l'utilisateur la progression de la commande reorg. Les messages de début et de fin sont écrits dans le journal d'erreurs et vers le processus client qui exécute reorg. Les messages de progression ne sont transmis qu'au client.

L'intervalle entre deux signalisations d'état est la plus grande des valeurs suivantes : 10 % des pages à traiter ou 10 000 pages. Un message d'état est émis après traitement de ce nombre de pages. Par conséquent il n'y a jamais plus de 10 messages transmis, quelle que soit la taille de l'index. Les messages d'état des commandes reorg existantes sont envoyés plus souvent.

Contrôle de la cohérence des bases de données

Ce chapitre explique comment vérifier la cohérence d'une base de données et effectuer certaines tâches de maintenance à l'aide des commandes dbcc.

Les sujets traités dans ce chapitre sont les suivants :

Sujet	Page
Présentation de Database Consistency Checker	773
Compréhension des concepts d'allocation de pages et d'objets	774
Contrôles pouvant être effectués avec dbcc	779
Contrôle de la cohérence des bases de données et des tables	780
Contrôle de l'allocation de pages	788
Correction des erreurs d'allocation à l'aide des options fix nofix	791
Génération de rapports avec dbcc tablealloc et dbcc indexalloc	792
Contrôle de cohérence des tables système	793
Stratégies d'utilisation des commandes pour le contrôle de cohérence	793
Contrôle des erreurs à l'aide de la commande dbcc checkverify	802
Suppression d'une base de données endommagée	805
Préparation préliminaire à l'utilisation de dbcc checkstorage	805
Maintenance de dbccdb	818
Génération de rapports à partir de dbccdb	820

Présentation de Database Consistency Checker

dbcc (database consistency checker) fournit les commandes permettant de contrôler la cohérence logique et physique d'une base de données.

Les deux fonctions principales de dbcc sont :

- la vérification des liaisons de pages et des pointeurs de données aux niveaux page et ligne à l'aide de l'option checkstorage ou checktable et de l'option checkdb ;

- la vérification de l'allocation des pages à l'aide des options checkstorage, checkalloc, checkverify, tablealloc et indexalloc.

dbcc checkstorage permet de stocker les résultats des vérifications dans la base de données dbccdb. Vous pouvez éditer des rapports de dbccdb en utilisant les procédures stockées dbcc.

Utilisez les commandes dbcc dans les cas suivants :

- Dans le cadre des activités régulières de maintenance des bases de données (la cohérence de la structure interne d'une base de données est assurée par l'administrateur système ou le propriétaire de la base de données par la mise en oeuvre de contrôles de cohérence réguliers).
- Pour déterminer l'ampleur des dommages éventuellement subis en cas d'erreur système.
- Avant de sauvegarder une base de données, pour augmenter la fiabilité de la sauvegarde.
- Lorsque vous pensez qu'une base de données est endommagée. Par exemple, si vous obtenez le message "Table corrupt" (table altérée) lorsque vous utilisez une table, vous pouvez utiliser dbcc pour déterminer si d'autres tables de la base sont endommagées.

Si vous utilisez Component Integration Services, vous disposez de commandes dbcc supplémentaires spécifiques des bases de données distantes. Pour plus d'informations, reportez-vous au document *Component Integration Services - Guide de l'utilisateur*.

Compréhension des concepts d'allocation de pages et d'objets

Lorsque vous initialisez un device de base de données, la commande disk init divise l'espace en **unités d'allocation**. La taille de l'unité d'allocation dépend de la taille des pages logiques que votre serveur utilise (2, 4, 8 ou 16 Ko). La première page d'une unité d'allocation correspond à la **page d'allocation** ; elle consigne l'utilisation de toutes les pages de l'unité d'allocation. Les pages d'allocation portent l'ID d'objet 99 ; elles ne constituent pas de véritables objets de base de données et n'apparaissent pas dans les tables système, mais les erreurs dbcc qui s'y rapportent indiquent cette valeur.

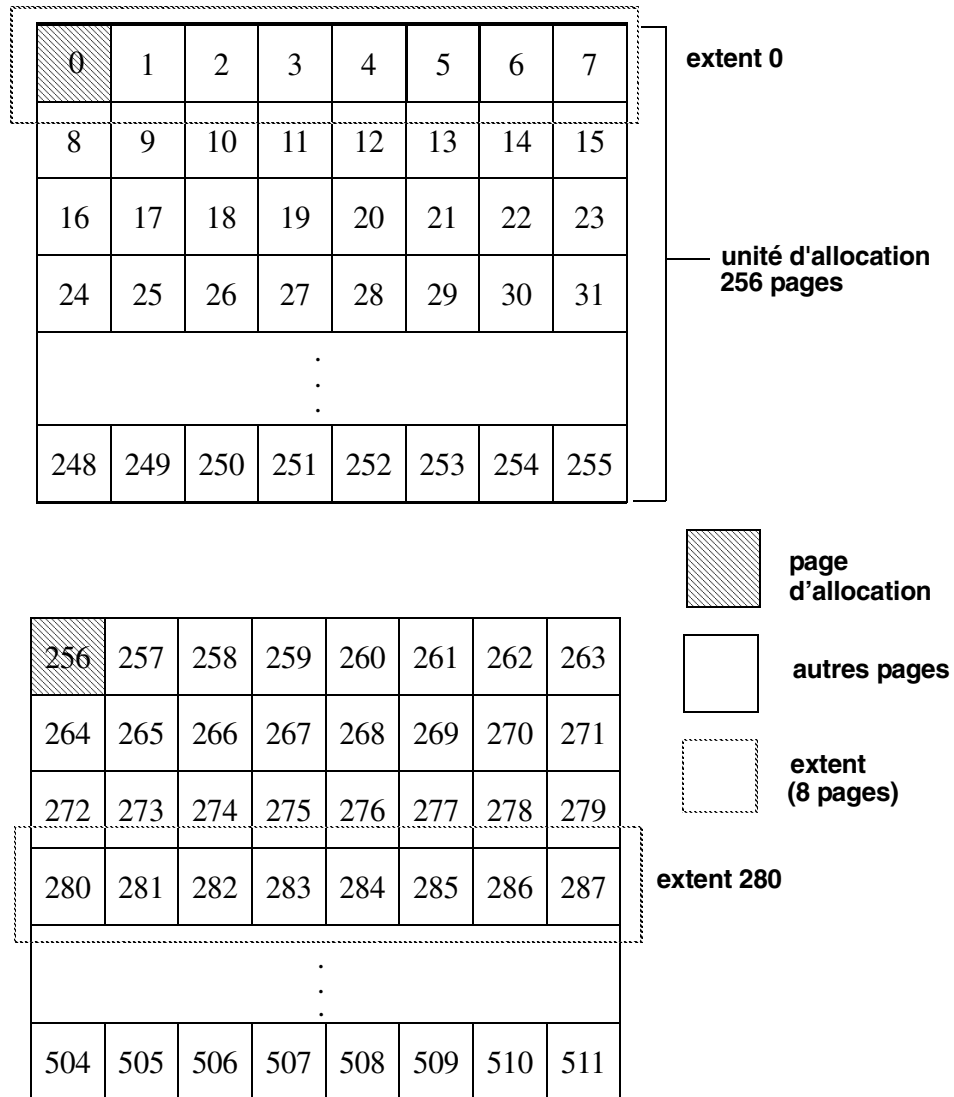
Chaque fois qu'une table ou un index requiert de l'espace, Adaptive Server lui alloue un bloc de 8 pages. Ce bloc de 8 pages s'appelle un *extent*. Chaque unité d'allocation contient 32 extents. La taille de l'extent dépend également de celle des pages logiques du serveur. Adaptive Server utilise les extents comme unités d'allocation et de libération d'espace.

- Lorsque vous créez une table ou un index, Adaptive Server lui alloue un extent.
- Lorsque vous ajoutez des lignes à une table existante, Adaptive Server lui alloue une nouvelle page si les pages existantes sont pleines. Si toutes les pages d'un extent sont pleines, Adaptive Server alloue un extent supplémentaire.
- Lorsque vous supprimez une table ou un index, Adaptive Server libère les extents qu'il occupait.
- Lorsque vous supprimez des lignes d'une table, de telle sorte que celle-ci est réduite d'une page, Adaptive Server libère cette page. Si cette table n'utilise plus un extent, Adaptive Server le libère.

Chaque fois qu'il se produit une allocation ou une libération d'espace sur un extent, Adaptive Server enregistre l'événement correspondant dans la page d'allocation servant au suivi des extents occupés par cet objet. Il s'agit d'un moyen rapide pour suivre les allocations d'espace dans la base de données, puisque les objets peuvent s'agrandir ou rétrécir avec un minimum d'overhead.

La figure 25-1 représente l'organisation des pages de données dans des extents et des unités d'allocation dans les bases de données Adaptive Server.

Figure 25-1 : Gestion des pages avec des extents



dbcc checkalloc contrôle toutes les pages d'allocation (c'est-à-dire la page 0 et toutes les pages divisibles par 256) d'une base de données et affiche les informations trouvées. dbcc indexalloc et dbcc tablealloc servent à contrôler l'allocation pour des objets spécifiques de la base.

Fonctionnement de la table d'allocation d'objets (OAM)

Chaque table et index d'une table possède une *table d'allocation d'objets* (OAM). Celle-ci est stockée sur les pages allouées à la table ou à l'index et est contrôlée lorsqu'une nouvelle page est nécessaire à l'index ou à la table. Chaque page de l'OAM peut contenir la configuration d'allocation de 2 000 à 63 750 pages d'index ou de données.

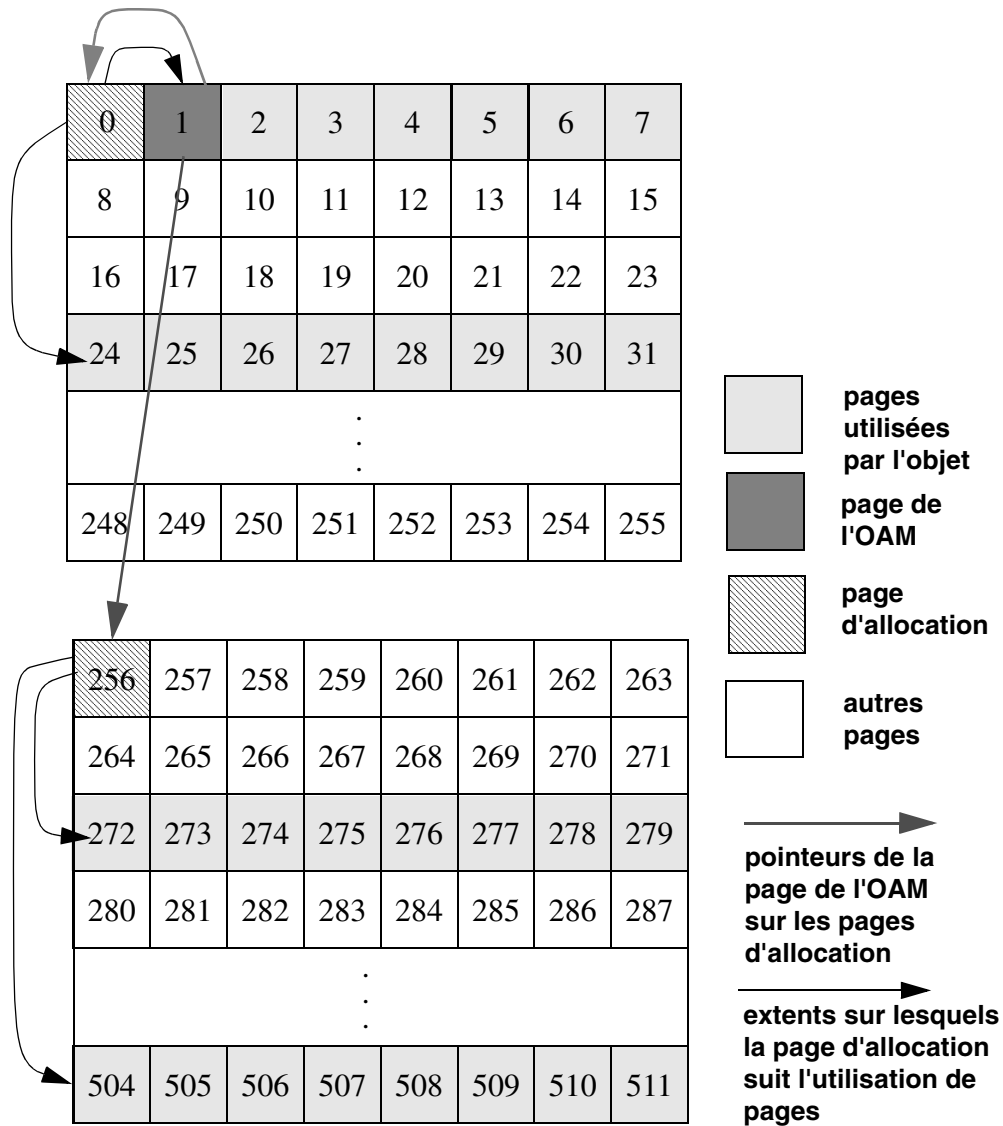
Les pages de l'OAM pointent vers la page d'allocation de chaque unité d'allocation dans laquelle l'objet occupe de l'espace. De leur côté, les pages d'allocation consignent les informations sur l'utilisation des extents et des pages à l'intérieur de l'unité d'allocation. En d'autres termes, si la table titles est stockée sur les extents 24 et 272, la page de l'OAM pour la table titles pointe sur les pages 0 et 256.

La figure 25-2 montre un objet stocké sur 4 extents numérotés 0, 24, 272 et 504 pour un serveur utilisant des pages logiques de 2 Ko. L'OAM est stockée sur la première page du premier segment. Dans ce cas, étant donné que la page 0 est occupée par la page d'allocation, l'OAM se trouve sur la page 1.

L'OAM suivante pointe sur deux pages d'allocation : les pages 0 et 256.

Ces pages d'allocation permettent le suivi des pages utilisées dans chaque extent servant à tous les objets qui occupent de l'espace de stockage dans l'unité d'allocation. Dans cet exemple, l'objet consigne l'allocation et la libération de pages sur les extents 0, 24, 272 et 504.

Figure 25-2 : Pointeurs de pages d'allocation et pointeurs de l'OAM

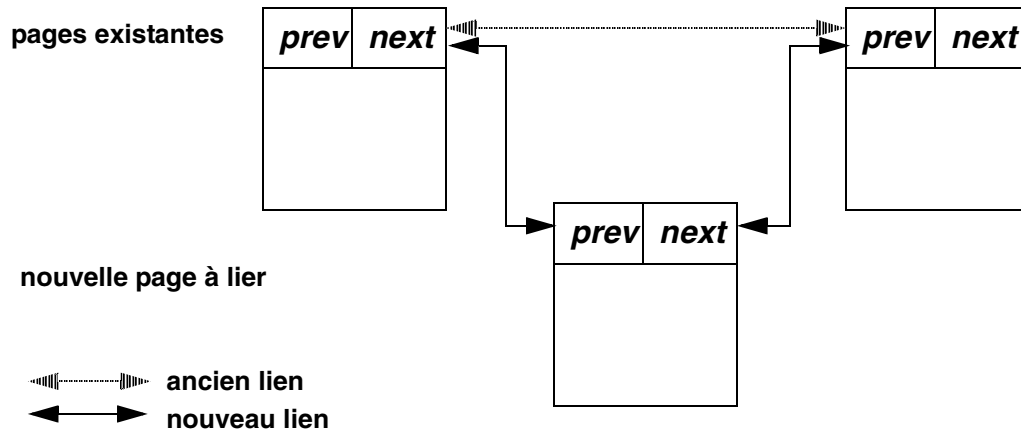


dbcc checkalloc et dbcc tablealloc examinent les informations de page de l'OAM et contrôlent en outre les liens de pages décrits dans la section "Fonctionnement des liens de pages", page 779.

Fonctionnement des liens de pages

Lorsqu'une page est allouée à une table ou à un index, elle est liée aux autres pages utilisées par le même objet. La figure 25-3 illustre ces liens. Chaque page comporte un en-tête contenant le numéro de la page précédente ("prev") et celui de la page suivante ("next"). Lorsqu'une nouvelle page est allouée, les informations d'en-tête des pages voisines sont modifiées de façon à pointer vers cette page. Les commandes dbcc checktable et dbcc checkdb contrôlent les liens de pages. dbcc checktable, tablealloc et indexalloc comparent les liens de pages aux informations de la page d'allocation.

Figure 25-3 : Lien d'une page nouvellement allouée avec les autres pages



Contrôles pouvant être effectués avec dbcc

Le tableau 25-1 présente les contrôles qui peuvent être effectués avec les commandes dbcc. Le tableau 25-2, page 794 compare les différentes commandes dbcc.

Tableau 25-1 : Comparaison des contrôles effectués avec les commandes dbcc

Contrôles effectués	<i>check-storage</i>	<i>check-table</i>	<i>check-db</i>	<i>check-alloc</i>	<i>index-alloc</i>	<i>table-alloc</i>	<i>check-catalog</i>
Contrôle l'allocation des colonnes de type texte	X						
Contrôle la cohérence de l'index		X	X				
Contrôle l'ordre de tri de l'index		X	X				
Contrôle les entrées de pages de l'OAM	X	X	X		X	X	
Contrôle l'allocation des pages	X			X	X	X	
Contrôle la cohérence des pages	X	X	X				
Contrôle la cohérence des pointeurs	X	X	X				
Contrôle les tables système							X
Contrôle les chaînes de colonnes de texte	X	X	X	X			
Contrôle les colonnes de texte	X	X	X				

Remarque Toutes les commandes dbcc peuvent être exécutées avec l'option fix lorsque la base de données est active, à l'exception de dbrepair et checkdb.

Les commandes dbcc avec les mots-clés checktable, fix_text ou reindex ne peuvent être exécutées que par le propriétaire de la table. Les mots-clés checkstorage, checkdb, checkcatalog, checkalloc, indexalloc et tablealloc ne peuvent être utilisés que par le propriétaire de la base de données. Le mot-clé dbrepair ne peut être utilisé que par un administrateur système.

Contrôle de la cohérence des bases de données et des tables

Les commandes dbcc de contrôle de la cohérence des bases de données et des tables sont les suivantes :

- dbcc checkstorage
- dbcc checktable
- dbcc checkdb

dbcc checkstorage

Utilisez `dbcc checkstorage` pour effectuer les contrôles suivants :

- Allocation de colonnes de texte
- Cohérence de l'allocation de pages
- Entrées de pages de l'OAM
- Cohérence du pointeur
- Colonnes de texte et chaînes de colonnes de texte

La syntaxe de `dbcc checkstorage` est la suivante :

```
dbcc checkstorage [(nom_base)]
```

où *nom_base* est le nom de la **base de données cible** (celle qui doit être contrôlée).

Avantages procurés par l'utilisation de *dbcc checkstorage*

La commande `dbcc checkstorage` :

- combine de nombreux contrôles fournis par les autres commandes `dbcc` ;
- ne verrouille pas les tables ni les pages pour les périodes étendues, ce qui permet à `dbcc` de localiser avec précision les erreurs tout en permettant une activité de mise à jour concurrente ;
- modifie l'échelle de manière linéaire à l'aide de la capacité de traitement des E/S d'agrégats ;
- sépare les fonctions de contrôle et de rapport, ce qui permet une évaluation personnalisée et la génération d'un rapport ;
- fournit une description détaillée de l'utilisation de l'espace dans la base de données cible ;
- enregistre l'activité de `dbcc checkstorage` et les résultats dans la base de données `dbccdb`, ce qui permet une analyse de la tendance et fournit une source d'informations précise sur le diagnostic.

Comparaison de commandes *dbcc checkstorage* et des autres commandes *dbcc*

dbcc checkstorage diffère des autres commandes *dbcc* dans la mesure où elle nécessite les éléments suivants :

- Le stockage par la base de données *dbccdb* des informations de configuration et des résultats des contrôles effectués sur la base de données cible. Pour plus d'informations, reportez-vous à la section "Préparation préliminaire à l'utilisation de *dbcc checkstorage*", page 805.
- Au moins deux espaces de travail à utiliser au cours de l'opération de contrôle. Reportez-vous au "Espaces de travail de *dbccdb*", page 94 du *Manuel de référence d'Adaptive Server*.
- Les procédures système et les procédures stockées pour vous aider à configurer votre système pour l'utilisation de *dbcc checkstorage* et pour générer des rapports sur les données stockées dans *dbccdb*. Pour plus d'informations, reportez-vous aux sections "Préparation préliminaire à l'utilisation de *dbcc checkstorage*", page 805, "Maintenance de *dbccdb*", page 818 et "Génération de rapports à partir de *dbccdb*", page 820

dbcc checkstorage ne corrige aucune erreur. Après avoir exécuté *dbcc checkstorage* et généré un rapport pour visualiser les erreurs, vous pouvez exécuter une commande *dbcc* pour les corriger.

Fonctionnement de *dbcc checkstorage*

dbcc checkstorage comporte quatre phases :

- 1 Inspection : *dbcc checkstorage* utilise l'allocation du device et la définition du segment de la base de données en cours de vérification pour déterminer le niveau de traitement parallèle qui peut être utilisé. *dbcc checkstorage* utilise également les paramètres de configuration *max worker processes* et *dbcc named cache* pour limiter le niveau de traitement parallèle qui peut être utilisé.
- 2 Planification : *dbcc checkstorage* génère un plan d'exécution de l'opération qui tire parti du parallélisme mis en évidence à l'étape 1.

- 3 Exécution et optimisation : dbcc checkstorage utilise les processus de production d'Adaptive Server pour effectuer une vérification parallèle et une analyse du stockage de la base de données cible. Cette commande tente d'égaliser le travail effectué par chaque processus de production et consolide le travail des processus de production sous-utilisés. Au cours du contrôle, dbcc checkstorage étend et ajuste le plan généré à l'étape 2 pour tirer parti des informations complémentaires réunies au cours de l'opération de contrôle.
- 4 Rapport et contrôle : au cours du contrôle, la commande dbcc checkstorage enregistre dans dbccdb toutes les erreurs rencontrées dans la base de données cible pour les rapporter et les évaluer ultérieurement. Elle enregistre également les résultats de son analyse de stockage dans dbccdb. Lorsque dbcc checkstorage rencontre une erreur, elle essaie de la corriger et de continuer l'opération ; toutefois, elle met fin aux opérations qui ne peuvent aboutir à cause de cette erreur. Par exemple, un disque défectueux ne provoque pas l'échec de la commande dbcc checkstorage ; toutefois, les opérations de contrôle effectuées sur ce disque ne peuvent pas aboutir, de sorte qu'elles ne sont pas effectuées.

La commande dbcc checkstorage peut échouer dans sa phase d'initialisation si une autre session exécute la commande drop table en même temps. Si cela se produit, réexécutez dbcc checkstorage une fois le processus de suppression de la table terminé.

Performance et évolutivité

dbcc checkstorage modifie l'échelle de manière linéaire à l'aide de la capacité de traitement des E/S d'agrégats pour obtenir une amélioration significative par rapport à dbcc checkalloc. La propriété d'évolutivité de dbcc checkstorage signifie que si la taille de la base de données et la capacité du matériel doublent (capacité de traitement d'E/S réalisable), le temps requis pour un contrôle par dbcc reste inchangé. En général, le fait de doubler la capacité suppose également de doubler le nombre de sous-unités de disque et d'augmenter suffisamment la capacité de la voie d'E/S supplémentaire, du bus système et de la CPU pour obtenir la capacité de traitement du disque souhaitée avec des agrégats complémentaires.

La plupart des contrôles effectués à l'aide de dbcc checkalloc et dbcc checkdb, y compris la vérification des chaînes de colonnes de texte, ne nécessite qu'un seul contrôle lorsque vous utilisez dbcc checkstorage, éliminant ainsi les opérations de contrôle redondantes.

dbcc checkstorage contrôle l'ensemble de la base de données, y compris les pages inutilisées, de sorte que le temps d'exécution est proportionnel à la taille de la base. Par conséquent, lorsque vous utilisez dbcc checkstorage, il n'existe pas de grande différence entre le contrôle d'une base de données presque vide et celui d'une base presque pleine, contrairement à ce qui se produit avec les autres commandes dbcc.

A l'inverse des autres commandes dbcc, les performances de dbcc checkstorage ne dépendent quasiment pas de l'endroit où se trouvent les données. C'est pourquoi elles sont cohérentes à chaque session, même si l'emplacement des données change d'une session à l'autre.

dbcc checkstorage effectue les tâches supplémentaires de définition de l'opération parallèle et d'enregistrement d'un grand nombre de données dans dbccdb, ce qui augmentera la rapidité d'exécution des autres commandes dbcc dans le cas d'une base de données cible de taille réduite.

L'emplacement et l'allocation des espaces de travail utilisés par dbcc checkstorage peuvent modifier les performances et l'évolutivité. Pour plus d'informations sur la manière de configurer les espaces de travail dans le but d'optimiser les performances et l'évolutivité de votre système, reportez-vous à la section "dbccdb Workspaces", page 87 dans le document *Manuel de référence d'Adaptive Server*.

Pour exécuter dbcc checkstorage et l'une des procédures système permettant de générer des rapports avec une seule commande, utilisez sp_dbcc_runcheck. Pour plus d'informations sur les procédures système qui génèrent des rapports, reportez-vous à la section "Génération de rapports à partir de dbccdb", page 820.

dbcc checktable

dbcc checktable contrôle la table spécifiée pour vérifier que :

- l'index et les pages de données sont correctement liés ;
- les index sont correctement triés ;
- les pointeurs sont cohérents ;
- les lignes de données sur chaque page possèdent des entrées dans la table d'offset de ligne ; ces entrées correspondent à l'emplacement des lignes de données sur la page ;

- les lignes de données sur chaque page possèdent des entrées dans la table d'offset de ligne à une page qui correspond à leur position respective dans la page des lignes de données ;
- les statistiques de partition sont correctes pour les tables partitionnées.

La syntaxe de `dbcc checktable` est la suivante :

```
dbcc checktable ({nom_table | id_table}  
[, skip_ncindex])
```

L'option `skip_ncindex` permet d'omettre le contrôle des liens de pages, des pointeurs et de l'ordre de tri des index non clusterisés. Les liens et les pointeurs des pages de données et des index clusterisés sont essentiels à la cohérence des tables. Vous pouvez supprimer et recréer des index non clusterisés si Adaptive Server fait état de problèmes au niveau de leurs liens de pages ou pointeurs.

Lorsque `checkstorage` renvoie une erreur de code de type 100035 et que `checkverify` confirme que l'erreur de bit espace est une erreur grave, vous pouvez utiliser `dbcc checktable` pour corriger l'erreur renvoyée.

La syntaxe est la suivante :

```
dbcc checktable (nom_table, fix_spacebits)
```

nom_table étant le nom de la table à réparer.

Vous pouvez utiliser la commande `dbcc checktable` avec le nom ou l'ID d'objet d'une table. La table `sysobjects` stocke ces informations dans les colonnes `name` et `id`.

L'exemple de rapport ci-dessous correspond à une table non endommagée :

```
dbcc checktable(titles)  
go  
Checking titles  
Le nombre total de pages de données dans cette table  
est 3.  
La table contient 18 lignes de données.  
Exécution de DBCC terminée. Si DBCC a imprimé des  
messages d'erreur, contactez un utilisateur exerçant  
les fonctions d'administrateur système (SA).
```

Si la table est partitionnée, dbcc checktable vérifie les liens de pages de données et les statistiques pour chaque partition. Par exemple :

```
dbcc checktable(historytab)
go
Checking historytab
Le nombre total de pages de données dans la partition
1 est 20.
Le nombre total de pages de données dans la partition
2 est 17.
Le nombre total de pages de données dans la partition
3 est 19.
Le nombre total de pages de données dans la partition
4 est 17.
Le nombre total de pages de données dans la partition
5 est 20.
Le nombre total de pages de données dans la partition
6 est 16.
Le nombre total de pages de données dans la partition
7 est 19.
Le nombre total de pages de données dans la partition
8 est 17.
Le nombre total de pages de données dans la partition
9 est 19.
Le nombre total de pages de données dans la partition
10 est 16.
Le nombre total de pages de données dans cette table
est 190.
La table contient 1536 lignes de données.
Exécution de DBCC terminée. Si DBCC a imprimé des
messages d'erreur, contactez un utilisateur exerçant
les fonctions d'administrateur système (SA).
```

Pour contrôler une table qui ne réside pas dans la base de données courante, vous devez indiquer le nom de la base. Pour contrôler une table détenue par un autre utilisateur, précisez le nom du propriétaire. Tout nom de table qualifié doit être placé entre guillemets. Par exemple :

```
dbcc checktable("pubs2.newuser.testtable")
```

dbcc checktable signale les problèmes suivants :

- Si les liens de pages sont incorrects, dbcc checktable affiche un message d'erreur.

- Si l'ordre de tri (sysindexes.soid) ou le jeu de caractères (sysindexes.csid) d'une table comportant des colonnes avec le type de données char ou varchar est incorrect et que l'ordre de tri de la table est compatible avec l'ordre de tri par défaut d'Adaptive Server, dbcc checktable corrige les valeurs de la table. Seul l'ordre de tri binaire est compatible d'un jeu de caractères à l'autre.

Remarque Si vous modifiez l'ordre de tri, les index utilisateur basés sur des caractères sont marqués en lecture seule et doivent être contrôlés et, le cas échéant, reconstruits. Pour plus d'informations sur le changement d'ordre de tri, reportez-vous au chapitre 7, "Configuration des jeux de caractères, des ordres de tri et des langues".

- S'il existe des lignes de données qui ne sont pas prises en compte dans la première page de l'OAM pour cet objet, dbcc checktable met à jour le nombre de lignes sur cette page. Ce problème n'est pas grave. En effet, la fonction intégrée rowcnt utilise cette valeur pour réaliser des estimations rapides du nombre de lignes, dans des procédures telles que sp_spaceused.

Les performances de dbcc checktable peuvent être améliorées par la lecture de pages enrichie.

dbcc checkdb

dbcc checkdb effectue les mêmes contrôles que dbcc checktable sur chaque table de la base de données spécifiée. Si vous n'indiquez pas de nom de base, dbcc checkdb contrôle la base courante. dbcc checkdb renvoie des messages semblables à ceux de dbcc checktable et apporte le même type de corrections.

La syntaxe de dbcc checkdb est la suivante :

```
dbcc checkdb [(nom_basedonnées [, skip_ncindex]) ]
```

Si vous spécifiez skip_ncindex en mode facultatif, dbcc checkdb ne vérifie aucun index non clusterisé des tables utilisateur de la base.

Contrôle de l'allocation de pages

Les commandes dbcc que vous utilisez pour contrôler l'allocation de pages sont les suivantes :

- dbcc checkalloc
- dbcc indexalloc
- dbcc tablealloc

dbcc checkalloc

dbcc checkalloc garantit que :

- toutes les pages sont correctement allouées,
- les statistiques de partition sont correctes pour les pages d'allocation,
- aucune page non utilisée n'est allouée,
- aucune page non allouée n'est utilisée.

La syntaxe de dbcc checkalloc est la suivante :

```
dbcc checkalloc [(nom_basededonnées [, fix | nofix] )]
```

Si vous ne spécifiez pas de nom de base de données, dbcc checkalloc contrôle la base courante.

L'option fix de dbcc checkalloc permet de corriger toutes les erreurs d'allocation à la place de dbcc tablealloc ainsi que les pages qui restent allouées aux objets qui ont été supprimés d'une base de données. Pour pouvoir utiliser dbcc checkalloc avec l'option fix, vous devez d'abord placer la base de données en mode mono-utilisateur. Pour plus de détails sur l'utilisation des options fix et no fix, reportez-vous à la section "Correction des erreurs d'allocation à l'aide des options fix | nofix", page 791.

Le résultat de dbcc checkalloc se compose d'un bloc de données pour chaque table, y compris les tables système et les index de chaque table. La commande affiche le nombre de pages et d'extents utilisés par chaque table ou index. Les informations sur les tables sont identifiées par INDID=0 ou INDID=1. Les tables sans index clusterisé affichent INDID=0 (par exemple, salesdetail, ci-après). Les tables avec index clusterisé affichent INDID=1 et le rapport correspondant comporte des informations sur le niveau de données et d'index. Reportez-vous aux rapports sur titleauthor et titles, ci-dessous. Les index non clusterisés sont numérotés de manière séquentielle à partir de INDID=2.

L'extrait suivant d'un rapport sur pubs2 porte sur les tables salesdetail, titleauthor et titles :

```

*****
TABLE: salesdetail                OBJID = 144003544
INDID=0  FIRST=297                ROOT=299                SORT=0
        Data level: 0.  3 Data Pages in 1 extents.
INDID=2  FIRST=289                ROOT=290                SORT=1
        Indid      : 2.  3 Index Pages in 1 extents.
INDID=3  FIRST=465                ROOT=466                SORT=1
        Indid      : 3.  3 Index Pages in 1 extents.
TOTAL # of extents = 3
*****
TABLE: titleauthor                OBJID = 176003658
INDID=1  FIRST=433                ROOT=425                SORT=1
        Data level: 1.  1 Data Pages in 1 extents.
        Indid      : 1.  1 Index Pages in 1 extents.
INDID=2  FIRST=305                ROOT=305                SORT=1
        Indid      : 2.  1 Index Pages in 1 extents.
INDID=3  FIRST=441                ROOT=441                SORT=1
        Indid      : 3.  1 Index Pages in 1 extents.
TOTAL # of extents = 4
*****
TABLE: titles                      OBJID = 208003772
INDID=1  FIRST=417                ROOT=409                SORT=1
        Data level: 1.  3 Data Pages in 1 extents.
        Indid      : 1.  1 Index Pages in 1 extents.
INDID=2  FIRST=313                ROOT=313                SORT=1
        Indid      : 2.  1 Index Pages in 1 extents.
TOTAL # of extents = 3
*****

```

dbcc indexalloc

dbcc indexalloc contrôle l'index spécifié en vérifiant que :

- toutes les pages sont correctement allouées,
- aucune page non utilisée n'est allouée,
- aucune page non allouée n'est utilisée.

dbcc indexalloc est une version de dbcc checkalloc au niveau des index qui permet de réaliser les mêmes contrôles de cohérence sur des index individuels. Vous pouvez spécifier soit le nom, soit l'ID d'objet (colonne id de sysobjects) de la table et l'indid de l'index tel qu'il figure dans sysindexes. dbcc checkalloc et dbcc indexalloc indiquent les ID d'index dans leur résultat.

La syntaxe de dbcc indexalloc est la suivante :

```
dbcc indexalloc ({nom_table | id_table}, id_index  
[, {full | optimized | fast | null}  
[, fix | nofix]])
```

Si vous souhaitez utiliser les options fix et nofix avec dbcc indexalloc, vous devez également spécifier l'une des options de rapport (full, optimized, fast ou null). Pour plus de détails sur l'utilisation des options fix et no fix, reportez-vous à la section "Correction des erreurs d'allocation à l'aide des options fix | nofix", page 791. Pour plus de détails sur les rapports, reportez-vous à la section "Génération de rapports avec dbcc tablealloc et dbcc indexalloc", page 792.

Vous pouvez exécuter sp_indsuspect pour contrôler la cohérence de l'ordre de tri des index et dbcc reindex pour réparer les incohérences. Pour plus de détails, reportez-vous aux sections "Utilisation de la procédure sp_indsuspect pour rechercher les index altérés", page 308 et "Reconstruction des index après changement de l'ordre de tri", page 308.

dbcc tablealloc

dbcc tablealloc contrôle la table utilisateur spécifiée en s'assurant que :

- toutes les pages sont correctement allouées,
- les statistiques de partition sont correctes pour les pages d'allocation,
- aucune page non utilisée n'est allouée,
- aucune page non allouée n'est utilisée.

La syntaxe de `dbcc tablealloc` est la suivante :

```
dbcc tablealloc ({nom_table | id_table}  
                [, {full | optimized | fast | null}  
                [, fix | nofix]])
```

Vous pouvez spécifier le nom ou l'ID d'objet de la table dans la colonne `id` de la table `sysobjects`.

Si vous souhaitez utiliser les options `fix` et `nofix` avec `dbcc tablealloc`, vous devez également spécifier l'une des options de rapport (`full`, `optimized`, `fast` ou `null`). Pour plus de détails sur l'utilisation des options `fix` et `nofix`, reportez-vous à la section "Correction des erreurs d'allocation à l'aide des options `fix` | `nofix`", page 791. Pour plus de détails sur les rapports, reportez-vous à la section "Génération de rapports avec `dbcc tablealloc` et `dbcc indexalloc`", page 792.

Correction des erreurs d'allocation à l'aide des options *fix* | *nofix*

Les options `fix` | `nofix` s'utilisent avec les commandes `dbcc checkalloc`, `dbcc tablealloc` et `dbcc indexalloc`. Elles indiquent si la commande corrige ou non l'allocation des erreurs dans les tables. La valeur par défaut pour toutes les tables utilisateur est `fix`. La valeur par défaut pour toutes les tables système est `nofix`.

Avant d'utiliser l'option `fix` sur les tables système, vous devez placer la base de données en mode mono-utilisateur :

```
sp_dboption nom_base, "single user", true
```

Pour utiliser cette commande, la base de données ne doit pas être en cours d'utilisation. Lorsque la commande est exécutée, seul l'utilisateur qui l'a lancée peut accéder à la base. Il est recommandé d'exécuter `dbcc checkalloc` avec l'option `nofix` pour permettre l'accès de la base de données à d'autres utilisateurs, puis d'utiliser la commande `dbcc tablealloc` ou `dbcc indexalloc` avec l'option `fix` pour corriger les erreurs dans les tables ou les index individuellement.

Le résultat de `dbcc tablealloc` avec l'option `fix` affiche les erreurs d'allocation et toutes les corrections qui ont été effectuées. L'exemple suivant présente un message d'erreur qui s'affiche que l'option `fix` soit utilisée ou non :

```
Msg 7939, Level 22, State 1:  
Line 2:  
Table Corrupt: The entry is missing from the OAM for  
object id 144003544 indid 0 for allocation page 2560.
```

Lorsque l'option *fix* est utilisée, le message ci-dessous indique que l'entrée manquante a été restaurée :

```
La rubrique OAM manquante a été insérée.
```

Les options *fix* et *nofix* fonctionnent de manière identique avec les commandes *dbcc indexalloc* et *dbcc tablealloc*.

Génération de rapports avec *dbcc tablealloc* et *dbcc indexalloc*

Les commandes *dbcc tablealloc* et *dbcc indexalloc* permettent d'éditer trois types de rapport :

- *full* : génère un rapport contenant tous les types d'erreurs d'allocation existants. L'utilisation de l'option *full* avec *dbcc tablealloc* aboutit au même résultat que l'utilisation de *dbcc checkalloc* au niveau d'une table.
- *optimized* : génère un rapport basé sur les pages d'allocation répertoriées dans les pages OAM pour la table. Lorsque vous utilisez l'option *optimized*, *dbcc tablealloc* n'édite pas de rapport et ne peut pas réparer les extents non référencés sur les pages d'allocation qui ne sont pas répertoriées dans les pages OAM. Si le type de rapport n'est pas spécifié, ou si vous spécifiez la valeur *NULL*, le rapport *optimized* est utilisé par défaut.
- *fast* : génère un rapport d'exception des pages référencées mais non allouées dans l'extent (erreurs de niveau 2521). Aucun rapport d'allocation n'est généré.

Le tableau 25-2, page 794 établit une comparaison de la vitesse, de l'exhaustivité, du verrouillage et des performances pour ces options et pour les autres commandes *dbcc*.

Contrôle de cohérence des tables système

dbcc checkcatalog permet de contrôler la cohérence au sein des tables système d'une base de données et entre ces tables. Par exemple, elle vérifie que :

- chaque type de syscolumns possède une entrée correspondante dans systypes ;
- chaque table et chaque vue de sysobjects possède au moins une colonne dans syscolumns ;
- le dernier point de reprise de syslogs est correct.

De plus, cette commande répertorie les segments définis pour être utilisés par la base de données.

La syntaxe de dbcc checkcatalog est la suivante :

```
dbcc checkcatalog [(nom_basededonnées)]
```

Si vous ne spécifiez aucun nom de base de données, dbcc checkcatalog contrôle la base de données courante.

```
dbcc checkcatalog (testdb)
```

```
Checking testdb
The following segments have been defined for database 5 (database name testdb).
virtual start addr      size      segments
-----
33554432                4096      0
                                     1
16777216                 102      2
Exécution de DBCC achevée. Si DBCC a imprimé des messages d'erreur, consultez
votre administrateur système.
```

Stratégies d'utilisation des commandes pour le contrôle de cohérence

Les sections suivantes comparent les performances des différentes commandes dbcc, émettent des suggestions concernant la planification et les différentes stratégies pour éviter des conséquences graves sur les performances et fournissent des informations sur les résultats de dbcc.

Comparaison des performances des commandes *dbcc*

Le tableau 25-2 compare la vitesse, l'exhaustivité, le niveau de contrôle et de verrouillage et des performances induites par les commandes *dbcc*. N'oubliez pas que les commandes *dbcc checkdb*, *dbcc checktable* et *dbcc checkcatalog* effectuent des types de contrôle de cohérence différents des commandes *dbcc checkalloc*, *dbcc tablealloc* et *dbcc indexalloc*. *dbcc checkstorage* effectue une combinaison de plusieurs contrôles effectués par les autres commandes. Le tableau 25-1, page 780 présente les contrôles effectués par chaque commande.

Tableau 25-2 : Comparaison des performances des commandes *dbcc*

Commande et option	Niveau	Verrouillage et performances	Vitesse	Profondeur
<i>checkstorage</i>	Chaînes de pages et lignes de données pour tous les index, pages d'allocation, pages OAM, devices et statistiques de partition	Pas de verrou ; exécute des E/S étendues et peut saturer les E/S du système ; peut utiliser le cache dédié avec un impact minimal sur les autres caches	Rapide	Elevée
<i>checktable</i> <i>checkdb</i>	Chaînes de pages, ordre de tri, lignes de données et statistiques de partition pour tous les index	Verrou de table partagé ; <i>dbcc checkdb</i> verrouille une table à la fois et libère le verrou à l'issue du contrôle de cette table	Lente	Elevée
<i>checktable</i> <i>checkdb</i> avec <i>skip_ncindex</i>	Chaînes de pages, ordre de tri et lignes de données pour les tables et les index clusterisés	Verrou de table partagé ; <i>dbcc checkdb</i> verrouille une table à la fois et libère le verrou à l'issue du contrôle de cette table	Jusqu'à 40% de rapidité en plus avec <i>skip_ncindex</i>	Moyenne
<i>checkalloc</i>	Chaînes de pages et statistiques de partition	Pas de verrou ; exécute des E/S étendues et peut saturer les appels d'E/S ; seules les pages d'allocation se trouvent dans le cache	Lente	Elevée
<i>tablealloc full</i> <i>indexalloc full</i> avec <i>full</i>	Chaînes de pages	Verrou de table partagé ; exécute de nombreuses E/S ; seules les pages d'allocation se trouvent dans le cache	Lente	Elevée
<i>tablealloc</i> <i>indexalloc</i> avec <i>optimized</i>	page d'allocation	Verrou de table partagé ; exécute de nombreuses E/S ; seules les pages d'allocation se trouvent dans le cache	Modérée	Moyenne
<i>tablealloc</i> <i>indexalloc</i> avec <i>fast</i>	pages OAM	Verrou de table partagé	Rapide	Faible
<i>checkcatalog</i>	Lignes dans les tables système	Verrous de table partagés sur les catalogues système ; libère le verrou après le contrôle de chaque page ; très peu de pages se trouvent dans le cache	Modérée	Moyenne

Utilisation des E/S étendues et prélecture asynchrone

Certaines commandes dbcc peuvent utiliser des E/S étendues et une prélecture asynchrone lorsque celles-ci sont configurées pour les caches utilisés par les bases de données ou les objets à contrôler.

dbcc checkdb et dbcc checktable utilisent des zones d'E/S étendues pour les contrôles des chaînes de pages sur les tables lorsque ces dernières utilisent un cache avec des E/S étendues configurées. La taille d'E/S utilisée est la plus grande possible. Pour le contrôle des index, dbcc n'utilise que des buffers de 2 Ko.

Les commandes de contrôle d'allocation dbcc checkdb et dbcc checktable et celles de dbcc (checkalloc, tablealloc et indexalloc) utilisent la prélecture asynchrone lorsqu'elle est disponible pour la zone en cours d'utilisation. Pour plus d'informations, reportez-vous à la section "Limites de dbcc", page 671 dans le manuel *Performances et optimisation*.

Les commandes de liaison avec le cache ainsi que les commandes de modification de taille et de pourcentages de prélecture asynchrone pour les zones sont dynamiques. Si vous utilisez ces commandes dbcc pendant les heures creuses, c'est-à-dire lorsque les applications utilisateur ont peu d'impact, vous pouvez modifier ces valeurs pour rendre l'exécution des performances de dbcc plus rapides et restaurer les valeurs standard à l'issue des contrôles. Pour plus d'informations sur ces commandes, reportez-vous au chapitre 19, "Configuration des caches de données".

Planification de la maintenance de la base de données sur votre site

Il existe plusieurs facteurs qui déterminent la fréquence à laquelle vous pouvez exécuter les commandes dbcc et quelles commandes vous devez utiliser.

Utilisation des bases de données

Si votre Adaptive Server est utilisé surtout entre 8:00 et 17:00, du lundi au vendredi, vous pouvez effectuer les contrôles par dbcc la nuit et les week-ends pour éviter un impact significatif sur vos utilisateurs. Si vos tables ne sont pas très volumineuses, vous pouvez exécuter le jeu de commandes dbcc assez fréquemment.

dbcc checkstorage et dbcc checkcatalog fournissent la meilleure couverture au coût le plus bas et assurent la restauration à partir des sauvegardes. Vous pouvez exécuter dbcc checkdb ou dbcc checktable moins souvent pour contrôler l'ordre de tri et la cohérence des index. Ce contrôle ne nécessite pas de coordination avec une autre activité de maintenance de base de données. Réservez l'utilisation des contrôles dbcc au niveau objet et ceux qui utilisent l'option fix pour un diagnostic plus poussé et la correction des erreurs trouvées par dbcc checkstorage.

Si votre Adaptive Server est utilisé 24 heures sur 24, 7 jours sur 7, vous avez la possibilité de limiter l'utilisation des ressources de dbcc checkstorage en réduisant le nombre de processus de production ou en utilisant des files d'attente pour les applications. Si vous décidez de ne pas utiliser dbcc checkstorage, vous avez la possibilité de planifier un cycle de contrôles sur des tables et des index individuels à l'aide de dbcc checktable, dbcc tablealloc et dbcc indexalloc. A la fin du cycle, une fois toutes les tables contrôlées, vous pouvez exécuter dbcc checkcatalog et sauvegarder la base de données. Pour plus d'informations sur l'utilisation des files d'attente pour les applications, reportez-vous au chapitre 4, "Répartition des ressources des moteurs", dans le manuel *Performances et optimisation*.

Certains sites avec des exigences de hautes performances 24 heures sur 24 exécutent les contrôles dbcc dans l'ordre suivant :

- ils sauvegardent la base de données sur bande ;
- ils chargent la sauvegarde de la base de données dans un Adaptive Server distinct pour créer une base de données dupliquée ;
- ils exécutent les commandes dbcc sur la base de données dupliquée ;
- ils exécutent les commandes dbcc avec les options fix sur les objets appropriés dans la base de données d'origine, dans le cas où des erreurs détectées peuvent être réparées avec ces options.

La sauvegarde correspond à une copie logique des pages de la base de données ; par conséquent, les problèmes rencontrés dans la base de données d'origine se retrouvent dans la base de données dupliquée. Cette stratégie permet d'utiliser les sauvegardes pour créer une base de données dupliquée destinée à l'édition de rapports, par exemple.

Prévoyez d'utiliser les commandes dbcc de verrouillage des objets pour éviter les interférences avec les activités sur les bases de données. Par exemple, dbcc checkdb verrouille tous les objets de la base de données lorsqu'elle effectue un contrôle. Vous ne pouvez pas maîtriser l'ordre dans lequel il contrôle les objets. Si vous lancez une application qui utilise table4, table5 et table6, et que l'exécution de dbcc checkdb prend 20 minutes, l'application ne sera pas autorisée à accéder à ces tables pendant ce délai, même lorsque la commande n'est pas en train de les contrôler.

Planification des sauvegardes

Plus vous sauvegardez vos bases de données et vos journaux de transactions souvent, plus vous récupérerez de données en cas de panne. Vous devez déterminer le volume des données que vous êtes prêts à perdre en cas de panne et mettre au point en conséquence un calendrier des sauvegardes.

Après avoir planifié vos sauvegardes, choisissez comment introduire les commandes dbcc dans ce calendrier. Vous n'êtes pas tenu d'effectuer un contrôle dbcc avant chaque sauvegarde. Toutefois, si une altération se produit au cours d'une sauvegarde, vous perdrez peut-être les dernières données.

Le moment idéal pour sauvegarder une base de données se situe après l'exécution d'un contrôle complet de cette base à l'aide de dbcc checkstorage et dbcc checkcatalog. Si ces commandes ne détectent aucune erreur dans la base, vous pouvez en déduire que votre sauvegarde correspond à une base de données non altérée. Les problèmes qui surviennent après le chargement d'une sauvegarde peuvent être corrigés via une nouvelle indexation. Utilisez dbcc tablealloc ou indexalloc sur les tables et les index individuels pour corriger les erreurs d'allocation rapportés par dbcc checkalloc.

Taille des tables et volume des données

Répondez aux questions suivantes concernant vos données :

- Combien de tables contiennent des données indispensables ?
- Quelle est la fréquence de modification de ces données ?
- Quelle est la taille de ces tables ?

dbcc checkstorage s'effectue au niveau de la base de données. Si vos données indispensables et vos données souvent modifiées sont concentrées dans un petit nombre de tables, vous préférerez peut-être exécuter plus souvent sur ces tables les commandes dbcc consacrées aux tables et aux index que vous n'exécuterez dbcc checkstorage sur l'ensemble de la base de données.

Fonctionnement du résultat des commandes *dbcc*

dbcc checkstorage stocke les résultats dans la base de données dbccdb. Vous pouvez imprimer tout un ensemble de rapports à partir de cette base de données. Pour plus de détails, reportez-vous à la section "dbcc checkstorage", page 781.

Le résultat de la plupart des autres commandes dbcc comporte des informations d'identification des objets en cours de contrôle et des messages d'erreur indiquant les problèmes éventuellement rencontrés par la commande dans l'objet. Lorsque dbcc tablealloc et dbcc indexalloc sont exécutées avec l'option fix, le résultat indique également les réparations effectuées par la commande.

L'exemple suivant présente le résultat de dbcc tablealloc pour une table comportant une erreur d'allocation :

```
dbcc tablealloc(table5)
```

Informations de sysindexes sur l'objet en cours de vérification :

```
TABLE: table5          OBJID = 144003544  
INDID=0  FIRST=337    ROOT=2587      SORT=0
```

Message d'erreur :

```
Msg 7939, Level 22, State 1:  
Line 2:  
Table Corrupt: The entry is missing from the OAM for object id  
144003544 indid 0 for allocation page 2560.
```

Message indiquant que l'erreur a été corrigée :

```
La rubrique OAM manquante a été insérée.  
Data level: 0. 67 Data Pages in 9 extents.
```

Rapport dbcc sur l'allocation de pages :

```
TOTAL # of extents = 9  
Alloc page 256 (# of extent=1 used pages=8 ref pages=8)  
EXTID:560 (Alloc page: 512) is initialized. Extent follows:
```

```
NEXT=0 PREV=0 OBJID=144003544 ALLOC=0xff DEALL=0x0 INDID=0 STATUS=0x0
Alloc page 512 (# of extent=2 used pages=8 ref pages=8)
Page 864 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x1)
Page 865 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x3)
Page 866 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x7)
Page 867 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0xf)
Page 868 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x1f)
Page 869 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x3f)
Page 870 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0x7f)
Page 871 allocated (Alloc page: 768 Extent ID: 864 Alloc mask: 0xff)
Alloc page 768 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1024 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1280 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1536 (# of extent=1 used pages=8 ref pages=8)
Alloc page 1792 (# of extent=1 used pages=8 ref pages=8)
Alloc page 2048 (# of extent=1 used pages=8 ref pages=8)
(Autres résultats effacés)
```

Informations sur les ressources utilisées :

Statistical information for this run follows:

```
Total # of pages read = 68
Total # of pages found cache = 68
Total # of physical reads = 0
Total # of saved I/O = 0
```

Message imprimé après exécution de la commande dbcc :

Exécution de DBCC terminée. Si DBCC a imprimé des messages d'erreur, contactez un utilisateur exerçant les fonctions d'administrateur système (SA).

Erreurs provoquées par des problèmes de cohérence dans la base de données

Les erreurs provoquées par des problèmes de cohérence dans la base de données et détectées par dbcc checkstorage sont commentées dans la table dbcc_types. La plupart d'entre elles se situent dans les intervalles 5010 à 5019 et 100 000 à 100 032. Pour plus d'informations sur les erreurs spécifiques, reportez-vous à la section "dbcc_types", page 82 du *Manuel de référence d'Adaptive Server*. dbcc checkstorage enregistre deux types d'erreurs : légères et graves. Pour plus d'informations, reportez-vous à la section "Comparaison des erreurs légères et des erreurs graves", page 800.

Les erreurs provoquées par des problèmes de cohérence dans la base de données et détectées par les commandes dbcc autres que dbcc checkstorage portent généralement un numéro d'erreur compris entre 2500 et 2599 ou entre 7900 et 7999. Ces messages, ainsi que d'autres qui peuvent résulter de problèmes de cohérence dans la base de données (tels que l'erreur 605), peuvent comporter une remarque telle que "Table Corrupt" (Table altérée) ou "Extent not within segment" (Extent en dehors du segment).

Certains messages signalent des problèmes graves de cohérence dans la base de données ; d'autres ont un caractère moins urgent. Quelques-uns enfin peuvent nécessiter l'aide du Support Technique de Sybase, mais la plupart peuvent être résolus en procédant comme suit :

- Exécutez les commandes dbcc qui utilisent l'option fix.
- Suivez les instructions contenues dans le manuel *Guide de dépannage*, qui décrivent les étapes pas à pas pour corriger un grand nombre d'erreurs détectées par dbcc dans la base de données.

Quelles que soient les techniques utilisées pour résoudre les problèmes, les solutions sont beaucoup plus simples lorsque vous détectez l'erreur peu après l'occurrence de l'altération ou de l'incohérence. Des problèmes de cohérence peuvent se produire sur les pages de données peu utilisées, par exemple une table mise à jour une fois par mois. dbcc peut détecter ces problèmes et souvent les corriger pour vous.

Comparaison des erreurs légères et des erreurs graves

Lorsque la commande dbcc checkstorage détecte une erreur dans la base de données cible, elle l'enregistre dans la table dbcc_faults en tant qu'**erreur légère** ou **erreur grave**. Les sections suivantes décrivent ces deux types d'erreurs. Pour plus d'informations, reportez-vous à la section "Contrôle des erreurs à l'aide de la commande dbcc checkverify", page 802.

Erreurs légères

Une *erreur légère* correspond à une incohérence passagère dans Adaptive Server. La plupart des erreurs légères sont provoquées par des incohérences temporaires dans la base de données cible, due à la mise à jour de cette base par un utilisateur au cours de l'opération `dbcc checkstorage` ou à la détection de commandes DDL (Data Definition Language) par `dbcc checkstorage`. Ces erreurs ne sont pas répétées lorsque vous exécutez la commande une deuxième fois. Les erreurs légères peuvent être reclassifiées en comparant les résultats des deux exécutions de `dbcc checkstorage` ou en exécutant `dbcc tablealloc` et `dbcc checktable` après la détection des erreurs légères par `dbcc checkstorage`.

Si les mêmes erreurs légères se reproduisent lors d'exécutions ultérieures de la commande `dbcc checkstorage`, ce sont des erreurs légères "durables" qui peuvent indiquer une altération. Si `dbcc checkstorage` est exécutée en mode mono-utilisateur, les erreurs légères rapportées sont des erreurs légères "durables". Ces erreurs peuvent être corrigées à l'aide de `sp_dbcc_differentialreport` ou en exécutant `dbcc tablealloc` et `dbcc checktable`. Si vous utilisez ces deux dernières commandes, vous n'êtes tenu de contrôler que les tables ou les index qui comportent ces erreurs.

Erreurs graves

Une *erreur grave* correspond à une altération durable d'Adaptive Server qui ne peut pas être corrigée en redémarrant Adaptive Server. Toutes les erreurs graves ne sont pas aussi critiques. Par exemple, chacune des situations décrites ci-dessous provoque une erreur grave, mais les résultats sont différents :

- Une page allouée à une table inexistante ne réduit quasiment pas l'espace de stockage disque disponible.
- Une table dont certaines lignes ne sont pas accessibles par un balayage peut renvoyer des résultats incorrects.
- Une table liée à une autre table déclenche l'arrêt d'une requête.

Certaines erreurs graves peuvent être corrigées par des actions aussi simples que la troncature de la table concernée. D'autres peuvent être corrigées simplement en restaurant la base de données à partir d'une sauvegarde.

Contrôle des erreurs à l'aide de la commande *dbcc checkverify*

dbcc checkverify examine les résultats de la plus récente opération *checkstorage* et reclassifie chaque erreur légère sous la forme d'une erreur grave ou d'une erreur insignifiante. *checkverify* agit comme un deuxième filtre pour supprimer les erreurs parasites des résultats de *checkstorage*.

Fonctionnement de la commande *dbcc checkverify*

checkverify lit les erreurs enregistrées à partir de *dbcc_faults* et corrige chaque erreur légère à l'aide d'une procédure similaire à celle utilisée par l'opération *checkstorage*.

Remarque *checkverify* verrouille la table pour empêcher toute mise à jour concurrente, garantissant ainsi une reclassification correcte des erreurs légères. *checkverify* ne détecte pas les erreurs survenues depuis la dernière exécution de *checkstorage*.

checkverify enregistre les informations dans les tables *dbcc_operation_log* et *dbcc_operation_results* à l'instar de *checkstorage*. La valeur enregistrée correspondant à l'*opid* est identique à celle correspondant à l'*opid* de la dernière opération *checkstorage*. *checkverify* effectue une mise à jour de la colonne état de la table *dbcc_faults* et insère une ligne dans la table *dbcc_fault_params* pour les erreurs traitées.

checkverify n'utilise pas les espaces de travail *scan* ou *text*.

Chaque erreur détectée par *checkstorage* est vérifiée par *checkverify* comme appartenant à l'une des catégories suivantes :

- Erreur grave, répertoriée en tant que telle par *checkstorage*.
- Erreur légère reclassée en erreur grave par *checkverify*, l'activité concurrente ayant été écartée en tant que cause.
- Erreur légère confirmée par *checkverify*. Certaines erreurs légères apparues en l'absence d'activité concurrente dans la base de données ne représentent pas de danger significatif et ne font pas l'objet d'une reclassification en erreur grave. Une erreur légère ne fait l'objet d'aucune reclassification si elle apparaît uniquement à titre d'information et ne constitue donc pas une altération.

- Erreur légère répertoriée en tant qu'erreur insignifiante car pouvant être attribuée à une activité concurrente ou parce que l'activité suivante a masqué l'incohérence de départ.

Une erreur pour laquelle `checkstorage` affecte le code 100011 (erreur de pointeur de texte) est considérée comme une erreur grave lorsque la colonne de texte contient une erreur grave. Si tel n'est pas le cas, elle est répertoriée en tant qu'erreur légère.

Une erreur pour laquelle `checkstorage` affecte le code 100016 (page allouée mais non reliée) est considérée comme une erreur grave lorsque la même erreur survient au cours de deux opérations `checkstorage` successives. Si tel n'est pas le cas, elle est répertoriée en tant qu'erreur légère.

Lorsqu'une erreur pour laquelle `checkstorage` affecte le code 100035 (bit espaces incorrects) est considérée comme une erreur grave, vous pouvez la corriger à l'aide de `dbcc checktable`.

En cas de confirmation par `checkverify` de fautes graves dans la base de données, suivez les mêmes procédures que celles utilisées dans les versions précédentes de Adaptive Server pour la correction des erreurs.

`checkverify` répertorie les codes d'erreur suivants comme se rapportant à des erreurs légères :

- 100020 – annulation de la vérification
- 100025 – erreur de décompte de lignes
- 100028 – allocation de page en dehors du segment courant

Quand utiliser `dbcc checkverify`

Pour vérifier les erreurs permanentes, il vous suffit d'exécuter `checkverify` après chaque exécution de `checkstorage`, y compris après une période prolongée de plusieurs jours ou de plusieurs heures. Toutefois, lors de la planification de votre échéancier, n'oubliez pas que l'état de la base de données se modifie avec le temps et que ces modifications peuvent masquer à la fois les erreurs légères et les erreurs graves.

Ainsi, une page reliée à une table mais ne faisant l'objet d'aucune allocation constitue une erreur grave. En cas de suppression de la table, l'erreur est corrigée et masquée. En cas d'allocation de la page à une autre table, l'erreur persiste mais sa signature est modifiée. La page apparaît alors comme étant reliée à deux tables différentes. En cas de réallocation de la page à la même table, l'erreur apparaît sous la forme d'une chaîne de pages altérée.

Les erreurs permanentes corrigées par une modification ultérieure de la base de données ne posent généralement pas de problème opérationnel. Toutefois, la détection et la vérification rapide de ces erreurs peuvent permettre de localiser une source d'altération avant la survenue de problèmes plus sérieux ou avant la signature des modifications de l'erreur de départ. C'est pour cette raison que Sybase vous recommande d'exécuter *checkverify* dès que possible après l'exécution de *dbcc checkstorage*.

Remarque Lors de l'exécution de *checkstorage* avec la base de données cible en mode mono-utilisateur, aucune erreur légère n'est susceptible de se produire. Il n'est donc pas nécessaire d'exécuter *checkverify*.

checkverify ne s'exécute qu'une seule fois pour chaque exécution de *checkstorage*. Toutefois, en cas d'interruption de *checkverify* en cours d'exécution, vous pouvez relancer son exécution. L'opération reprend alors là où elle s'était interrompue.

Mode d'utilisation de *dbcc checkverify*

La syntaxe est la suivante :

```
dbcc checkverify(nom_basededonnées)
```

nom_basededonnées étant la base de données pour laquelle vous souhaitez vérifier les résultats de *checkstorage*.

checkverify agit sur les résultats de la dernière opération *checkstorage* menée à terme, uniquement pour la base de données indiquée.

Une fois l'opération *checkverify* achevée, Adaptive Server renvoie le message suivant :

```
DBCC checkverify for database name, sequence  
n completed at date time. n suspect conditions  
resolved as faults and n resolved as innocuous.  
n checks were aborted.
```

Vous pouvez exécuter *checkverify* automatiquement après avoir exécuté *checkstorage* à l'aide de *sp_dbcc_runcheck*.

Suppression d'une base de données endommagée

dbcc dbrepair dropdb vous permet de supprimer une base de données endommagée à partir de la base de données master. Aucun utilisateur, y compris celui qui exécute la commande dbrepair, ne peut utiliser la base de données une fois supprimée.

La syntaxe de dbcc dbrepair est la suivante :

```
dbcc dbrepair (nom_basededonnées, dropdb )
```

La commande Transact-SQL drop database ne fonctionne pas sur une base de données ne pouvant pas être restaurée ou utilisée.

Préparation préliminaire à l'utilisation de *dbcc checkstorage*

Pour pouvoir utiliser dbcc checkstorage, vous devez préalablement configurer Adaptive Server et la base de données dbccdb. Le tableau 25-3 répertorie les étapes et les commandes dans l'ordre dans lequel vous devez les utiliser. Chaque action est décrite en détail dans les sections qui suivent.

Remarque Les exemples de cette section prennent le cas d'un serveur utilisant des pages logiques de 2 Ko.

Avertissement ! Ne vous avisez pas d'effectuer les actions ni d'utiliser les commandes décrites dans le tableau 25-3 avant d'avoir lu les informations de la section correspondante. Vous devez comprendre les conséquences de chaque action avant d'effectuer toute modification.

Tableau 25-3 : Tâches préliminaires à l'utilisation de dbcc checkstorage

Pour cette action	Voir	Utilisez cette commande
1. Lisez les recommandations sur la taille de la base de données, les devices (si dbccdb n'existe pas), la taille des espaces de travail, la taille du cache et le nombre de processus de production pour la base de données cible.	"Planification des ressources", page 807 "Planification de la taille de l'espace de travail", page 809	use master sp_plan_dbccdb
2. Si nécessaire, ajustez le nombre de processus de production utilisés par Adaptive Server.	"Configuration des processus de production", page 810	sp_configure number of worker processes memory per worker process
3. Créez un cache nommé named cache for dbcc (facultatif).	"Configuration d'un cache nommé pour dbcc", page 812	sp_cacheconfig
4. Configurez une zone de buffers.	"Configuration d'une zone de buffers d'E/S de 8 pages", page 813	sp_poolconfig
5. Si dbccdb existe déjà, supprimez-la ainsi que tous les devices associés et créez une autre base de données dbccdb.		drop database
6. Initialisez les devices de disque pour les données et le journal de dbccdb.	"Allocation d'espace disque pour dbccdb", page 814	disk init
7. Créez dbccdb sur le device de disque comportant les données.		create database
8. Ajoutez des segments sur le disque (facultatif).	"Segments pour les espaces de travail", page 814	use dbccdb sp_addsegment
9. Remplissez la base de données dbccdb et installez les procédures stockées dbcc.		isql -Usa -P -i \$SYBASE/scripts/installdbccdb
10. Créez les espaces de travail.	"Espaces de travail de dbccdb", page 94	sp_dbcc_createws
11. Mettez à jour les valeurs dans la configuration.	"Mise à jour de la table dbcc_config", page 817	sp_dbcc_updateconfig max worker processes dbcc named cache scan workspace text workspace OAM count threshold IO error abort linkage error abort

Planification des ressources

La sélection du device et de la taille appropriés pour dbccdb s'avère essentielle pour les performances des opérations dbcc checkstorage. `sp_plan_dbccdb` fournit des conseils sur la configuration pour la base de données cible spécifiée, selon que dbccdb existe ou non. Utilisez ces informations pour configurer Adaptive Server et la base de données dbccdb.

Exemples de résultat `sp_plan_dbccdb`

Si dbccdb n'existe pas, `sp_plan_dbccdb` renvoie :

- la taille minimale pour dbccdb,
- les devices utilisables avec dbccdb,
- la taille minimale pour les espaces de travail scan et text,
- la taille minimale pour le cache,
- le nombre de processus de production.

Les valeurs recommandées pour la taille du cache sont approximatives parce que la taille maximale du cache pour dbccdb dépend du masque de l'allocation de pages dans la base de données cible. L'exemple suivant met en œuvre un serveur utilisant des pages logiques de 2 Ko et présente le résultat de `sp_plan_dbccdb` pour la base de données pubs2 lorsque dbccdb n'existe pas :

```
sp_plan_dbccdb pubs2
```

La taille recommandée pour dbccdb est 4 Mo.

Les devices recommandés pour dbccdb sont :

Logical Device Name	Device Size	Physical Device Name
sprocdev	28672	/remote/SERV/sprocs_dat
tun_dat	8192	/remote/SERV/tun_dat
tun_log	4096	/remote/SERV/tun_log

Les valeurs recommandées pour la taille de l'espace de travail, la taille du cache et le nombre de processus sont :

dbname	scan ws	text ws	cache	process count
pubs2	64K	64K	640K	1

Si dbccdb existe déjà, sp_plan_dbccdb renvoie :

- la taille minimale pour dbccdb,
- la taille de la base de données dbccdb existante,
- la taille minimale pour les espaces de travail scan et text,
- la taille minimale pour le cache,
- le nombre de processus de production.

L'exemple suivant présente le résultat de sp_plan_dbccdb pour la base de données pubs2 lorsque dbccdb existe déjà :

```
sp_plan_dbccdb pubs2
Recommended size for dbccdb database is 23MB (data = 21MB, log = 2MB).
dbccdb database already exists with size 8MB.
```

Les valeurs recommandées pour la taille de l'espace de travail, la taille du cache et le nombre de processus sont :

dbname	scan ws	text ws	cache	process count
pubs2	64K	48K	640K	1

Si vous prévoyez de contrôler plusieurs bases de données, utilisez le nom de la base la plus grande pour la base de données cible. Si vous ne fournissez pas de nom pour la base de données cible, sp_plan_dbccdb renvoie les valeurs de la configuration pour toutes les bases de données master..sysdatabases, comme le montre l'exemple suivant :

```
sp_plan_dbccdb
Recommended size for dbccdb is 4MB.
dbccdb database already exists with size 8MB.
```

Les valeurs recommandées pour la taille de l'espace de travail, la taille du cache et le nombre de processus sont :

dbname	scan ws	text ws	cache	process count
master	64K	64K	640K	1
tempdb	64K	64K	640K	1
model	64K	64K	640K	1
sybsystemprocs	384K	112K	1280K	2
pubs2	64K	64K	640K	1

pubs3	64K	64K	640K	1
pubtune	160K	96K	1280K	2
sybsecurity	96K	96K	1280K	2
dbccdb	112K	96K	1280K	2

Pour plus d'informations, reportez-vous à la section sur la procédure système `sp_plan_dbccdb` dans le *Manuel de référence d'Adaptive Server*.

Planification de la taille de l'espace de travail

Deux espaces de travail sont nécessaires pour `dbccdb` : `scan` et `text`. Les besoins en espace pour les espaces de travail dépendent de la taille de la base de données la plus volumineuse contrôlée. Si vous souhaitez effectuer des opérations `dbcc checkstorage` concurrentes, il vous faudra des espaces de travail supplémentaires.

Détermination de la taille de la plus grande base de donnée contrôlée

Plusieurs bases de données peuvent utiliser les mêmes espaces de travail. Par conséquent, les espaces de travail doivent être de taille suffisamment importante pour contenir la base de données la plus volumineuse avec laquelle ils vont être utilisés.

Remarque `sp_plan_dbccdb` suggère des tailles pour les espaces de travail. Les détails suivants concernant la détermination de la taille d'un espace de travail sont juste fournis à titre d'informations.

Chaque page dans la base de données cible est représentée par une ligne de 18 octets dans l'espace de travail `scan`. La taille de cet espace de travail doit correspondre à environ 1,1% de la taille de la base de données cible.

Chaque colonne de type `text` non `NULL` contenue dans la base de données cible insère une ligne de 22 octets dans l'espace de travail `text`. Si la base de données cible contient n colonnes de type `text` de valeur non `NULL`, la taille de l'espace de travail `text` doit être au moins de $(22 * n)$ octets. Le nombre de colonnes de type `text` de valeur non `NULL` est dynamique, aussi allouez suffisamment d'espace pour permettre à l'espace de travail `text` de répondre aux demandes futures. L'espace de travail `text` doit disposer d'au moins 24 pages.

Nombre d'espaces de travail utilisables de façon concurrente

Vous pouvez configurer dbccdb afin d'exécuter dbcc checkstorage de façon concurrente sur plusieurs bases de données. Cela n'est possible que si la seconde opération dbcc checkstorage et les opérations suivantes possèdent leurs propres ressources dédiées. Pour effectuer des opérations dbcc checkstorage concurrentes, chaque opération doit disposer de ses propres espaces de travail scan et text, processus de production et cache réservé.

L'espace total requis pour les espaces de travail dépend de la manière dont les bases de données utilisateur sont contrôlées : en série ou de façon concurrente. Si les opérations dbcc checkstorage sont effectuées en série, les plus grands espaces de travail scan et text peuvent être utilisés pour toutes les bases de données utilisateur. Si les opérations dbcc checkstorage sont effectuées de façon concurrente, alors dbccdb doit être définie de façon à contenir les espaces de travail qui seront utilisés de façon concurrente. Vous pouvez déterminer la taille des espaces de travail en ajoutant la taille des plus grandes bases de données qui seront contrôlées de façon concurrente.

Pour plus d'informations, reportez-vous à la section “dbccdb Workspaces”, page 87.

Configuration d'Adaptive Server pour *dbcc checkstorage*

La présente section contient des informations sur la configuration d'Adaptive Server pour dbcc checkstorage.

Configuration des processus de production

Les paramètres suivants s'appliquent à dbcc checkstorage :

- max worker processes : définissez ce paramètre avec sp_dbcc_updateconfig. Il met à jour la valeur de max worker processes dans la table dbcc_config pour chaque base de données cible.
- number of worker processes : définissez ce paramètre de configuration avec sp_configure. Il met à jour le fichier *server_name.cfg*.
- memory per worker process : définissez ce paramètre de configuration avec sp_configure. Il met à jour le fichier *server_name.cfg*.

Après avoir modifié la valeur des paramètres de `sp_configure`, vous devez redémarrer Adaptive Server pour que la modification soit prise en compte. Pour plus d'informations, reportez-vous au chapitre 5, "Définition des paramètres de configuration".

`max worker processes` spécifie le nombre maximal de processus de production utilisés par `dbcc checkstorage` pour chaque base de données cible, tandis que `number of worker processes` spécifie le nombre total de processus de production gérés par Adaptive Server. Les processus de production ne sont pas destinés à effectuer les opérations `dbcc checkstorage`.

Définissez une valeur suffisamment élevée pour `number of worker processes` pour permettre de calculer le nombre de processus de production spécifié par `max worker processes`. Un petit nombre de processus de production réduit les performances et l'utilisation des ressources de `dbcc checkstorage`. `dbcc checkstorage` n'utilisera pas plus de processus de production que le nombre de devices de base de données utilisé par cette base. La taille du cache, les performances de la CPU et la taille des devices peuvent suggérer un nombre de processus de production plus faible. Si le nombre de processus de production configuré est insuffisant dans Adaptive Server, la commande `dbcc checkstorage` ne fonctionnera pas.

`maximum parallel degree` et `maximum scan parallel degree` n'ont aucune incidence sur les fonctions parallèles de `dbcc checkstorage`. Lorsque `maximum parallel degree` a la valeur 1, le parallélisme n'est pas désactivé dans `dbcc checkstorage`.

`dbcc checkstorage` requiert plusieurs processus, de sorte que `number of worker processes` doit avoir au moins la valeur 1 pour permettre l'existence conjointe d'un processus parent et d'un processus de production.

`sp_plan_dbccdb` recommande certaines valeurs pour le nombre de processus de production en fonction de la taille de la base de données, du nombre de devices et d'autres facteurs. Vous pouvez utiliser des valeurs plus petites pour réduire le chargement sur votre système. `dbcc checkstorage` utilise peut-être un nombre plus faible de processus de production que ne le recommande `sp_plan_dbccdb` ou que vous n'en avez configurés.

L'augmentation du nombre de processus de production utilisés ne garantit pas une augmentation des performances. Le scénario suivant décrit les conséquences de deux différentes configurations.

Une base de données de 8 Go possède 4 Go de données sur le disque A et 0,5 Go de données sur chacun des disques B, C, D, E, F, G, H et I.

Avec 9 processus de production actifs, le temps d'exécution de dbcc checkstorage est de 2 heures, ce qui correspond au temps nécessaire pour contrôler le disque A. Chacun des 8 autres processus de production nécessite 15 minutes et attend que le processus de production du disque A soit terminé.

Avec 2 processus de production actifs, le temps d'exécution de dbcc checkstorage est toujours de 2 heures. Le premier processus de production traite le disque A tandis que l'autre processus de production traite les disques B, C, D, E, F, G, H et I. Dans ce cas, il n'y a pas d'attente et les ressources sont utilisées de manière plus efficace.

memory per worker process spécifie l'allocation totale de la mémoire pour la maintenance des processus de production dans Adaptive Server. La valeur par défaut est celle qui s'applique à dbcc checkstorage.

Configuration d'un cache nommé pour *dbcc*

Si vous utilisez un cache nommé pour dbcc checkstorage, vous devrez peut-être ajuster les paramètres de configuration d'Adaptive Server.

Au cours d'une opération dbcc checkstorage, les espaces de travail sont temporairement liés à un cache qui est également utilisé pour la lecture de la base de données cible. L'utilisation d'un cache nommé alloué à dbcc réduit au maximum l'impact du contrôle de la base de données sur les autres utilisateurs et améliore les performances. Vous pouvez créer un cache distinct pour chaque opération dbcc checkstorage à effectuer de façon concurrente ou créer un cache suffisamment important pour répondre à toutes les exigences des opérations concurrentes. La taille requise pour obtenir les performances les meilleures possibles est fonction de la taille de la base de données cible et de la répartition des données dans cette base. dbcc checkstorage requiert au moins 640 Ko de buffers de 16 Ko chacun par processus de production dans le cache nommé.

Pour améliorer les performances, allouez la plupart du cache dédié à la zone de buffer et ne segmentez pas le cache. La taille recommandée pour le cache correspond à la taille minimum pour la zone de buffers. Ajoutez à cette valeur la taille d'une zone de pages.

Si vous dédiez un cache pour dbcc checkstorage, la zone la plus petite de buffers de pages suffit. Si le cache est partagé, vous pouvez améliorer les performances de dbcc checkstorage en augmentant la taille de la zone de buffers avant d'effectuer l'opération, et en réduisant cette même taille à l'issue de l'opération. Les conditions requises pour la zone de buffers sont les mêmes que celles requises pour le cache partagé. Toutefois, tandis qu'un cache partagé peut répondre aux conditions de taille, d'autres conditions requises pour le cache peuvent limiter la disponibilité du buffer à dbcc checkstorage et engendrer un impact de taille sur les performances conjointes de checkstorage et d'Adaptive Server.

Avertissement ! N'utilisez pas de segmentations dans un cache en cours d'utilisation pour dbcc checkstorage.

Pour configurer Adaptive Server avec un cache nommé pour les opérations de la commande dbcc checkstorage, utilisez sp_cacheconfig et sp_poolconfig. Pour plus d'informations, voir le chapitre 19, "Configuration des caches de données".

Configuration d'une zone de buffers d'E/S de 8 pages

dbcc checkstorage requiert une zone de buffers d'E/S d'un extent. La procédure sp_poolconfig permet de configurer la taille de cette zone et de vérifier que la configuration a réussi. Les informations concernant la taille de la zone sont stockées dans la table dbcc_config.

L'exemple suivant explique comment utiliser sp_poolconfig pour définir une zone de buffers de 16 Ko pour "master_cache" sur un serveur configuré pour des pages logiques de 2 Ko. Le cache nommé est créé pour la base de données master.

```
1> sp_poolconfig "master_cache", "1024K", "16K"
2> go
(return status = 0)
```

L'exemple suivant montre que la zone de buffers est définie pour le cache privé "master_cache" :

```
1> sp_poolconfig "master_cache"
2> go
```

Cache Name	Status	Type	Config Value	Run Value
master_cache	Active	Mixed	2.00 Mb	2.00 Mb

```
-----
Total                2.00 Mb    2.00 Mb
=====
Cache: master_cache,  Status: Active,  Type: Mixed
      Config Size: 2.00 Mb,  Run Size: 2.00 Mb
      Config Replacement: strict LRU,  Run Replacement: strict LRU
IO Size  Wash Size  Config Size  Run Size  APF Percent
-----  -
2 Kb     512 Kb     0.00 Mb     1.00 Mb     10
16 Kb    192 Kb    0.00 Mb     1.00 Mb     10
(return status = 0)
```

Pour plus d'informations sur sp_poolconfig, reportez-vous au Manuel de référence d'Adaptive Server.

Allocation d'espace disque pour *dbccdb*

La base de données dbccdb nécessite de l'espace de stockage supplémentaire sur le disque. Du fait que dbcc checkstorage utilise beaucoup dbccdb, vous devez placer dbccdb sur un device différent de ceux des autres bases de données.

Remarque Ne créez pas dbccdb sur le device master. Assurez-vous que les devices de dbccdb pour les données et ceux pour les journaux sont distincts.

Segments pour les espaces de travail

En allouant des segments aux espaces de travail, vous pouvez contrôler la position de ces derniers et améliorer les performances de dbcc checkstorage. Lorsque vous allouez de nouveaux segments pour leur utilisation exclusive par les espaces de travail, assurez-vous de supprimer le mappage entre les devices liés à ces segments et le segment par défaut à l'aide de sp_dropsegment.

Création de la base de données *dbccdb*

Pour créer la base de données *dbccdb*, procédez comme suit :

- 1 Exécutez `sp_plan_dbccdb` dans la base de données *master* pour obtenir des recommandations sur la taille de la base, sur les devices, sur la taille des espaces de travail et du cache et sur le nombre de processus de production pour la base de données cible. Par exemple, supposons que vous exécutez `sp_plan_dbccdb` avec *pubs2* comme base de données cible alors que *dbccdb* n'existe pas :

```
use master
go
sp_plan_dbccdb pubs2
go
```

Le résultat suivant apparaît :

Recommended size for *dbccdb* is 4MB.

Les devices recommandés pour *dbccdb* sont :

Logical Device Name	Device Size	Physical Device Name
<i>sprocdev</i>	28672	/remote/SERV/sprocs_dat
<i>tun_dat</i>	8192	/remote/SERV/tun_dat
<i>tun_log</i>	4096	/remote/SERV/tun_log

Les valeurs recommandées pour la taille de l'espace de travail, la taille du cache et le nombre de processus sont :

dbname	scan ws	text ws	cache	process count
<i>pubs2</i>	64K	64K	640K	1

Pour plus de détails sur les informations fournies par `sp_plan_dbccdb`, reportez-vous à la section "Planification des ressources", page 807.

- 2 Si la base de données *dbccdb* existe déjà, supprimez-la, ainsi que tous ses devices associés et créez une autre base *dbccdb* :

```
use master
go
if exists (select * from master.dbo.sysdatabases
where name = "dbccdb")
begin
print "+++ Suppression de la base dbccdb"
drop database dbccdb
end
go
```

- 3 Utilisez disk init pour initialiser les devices de type disque contenant les données et le journal de dbccdb :

```
use master
go
disk init
    name = "dbccdb_dat",
    physname = "/remote/disks/masters/",
    size = "4.096K"
go
disk init
    name = "dbccdb_log",
    physname = "/remote/disks/masters/",
    size = "1024K"
go
```

- 4 Utilisez create database pour créer dbccdb sur le device de données que vous avez initialisé à l'étape 3 :

```
use master
go
create database dbccdb
    on dbccdb_dat = 6
    log on dbccdb_log = 2
go
```

- 5 (Facultatif) Ajoutez des segments au device de données de dbccdb pour les espaces de travail scan et text :

```
use dbccdb
go
sp_addsegment scanseg, dbccdb, dbccdb_dat
go
sp_addsegment textseg, dbccdb, dbccdb_dat
go
```

- 6 Créez les tables pour dbccdb et initialisez la table dbcc_types :

```
isql -Ujms -P***** -iinstalldbccdb
```

Le script installdbccdb vérifie l'existence de la base de données avant toute tentative de création des tables. Il ne crée que celles qui n'existent pas déjà dans dbccdb. Si une table de dbccdb vient à être altérée, supprimez-la à l'aide de drop table, puis utilisez installdbccdb pour la recréer.

7 Créez et initialisez les espaces de travail scan et text :

```
use dbccdb
go
sp_dbcc_createews dbccdb, scanseg, scan_pubs2,
scan, "10M"
go
sp_dbcc_createews dbccdb, textseg, text_pubs2,
text, "10M"
go
```

Après avoir installé dbccdb, vous devez mettre à jour la table dbcc_config.

Mise à jour de la table *dbcc_config*

Utilisez `sp_dbcc_updateconfig` pour initialiser la table `dbcc_config` de la *base de données cible*. Vous devez mettre à jour chaque paramètre dbcc séparément pour chaque base de données cible, comme le montre l'exemple suivant :

```
use dbccdb
go

sp_dbcc_updateconfig pubs2, "max worker processes", "4"
go

sp_dbcc_updateconfig pubs2, "dbcc named cache", pubs2_cache, "10K"
go

sp_dbcc_updateconfig pubs2, "scan workspace", scan_pubs2
go

sp_dbcc_updateconfig pubs2, "text workspace", text_pubs2
go

sp_dbcc_updateconfig pubs2, "OAM count threshold", "5"
go

sp_dbcc_updateconfig pubs2, "IO error abort", "3"
go

sp_dbcc_updateconfig pubs2, "linkage error abort", "8"
go
```

Vous pouvez à présent utiliser `dbcc checkstorage` pour contrôler vos bases de données. Pour plus d'informations sur la description des paramètres dbcc, reportez-vous aux valeurs de type code comprises entre 1 et 9 dans la section "dbcc_types", page 82.

Maintenance de *dbccdb*

De temps en temps, vous devrez effectuer les tâches de maintenance suivantes sur dbccdb.

- Réévaluez et mettez à jour la configuration en utilisant :
 - `sp_dbcc_evaluatedb` : recommande des valeurs pour les paramètres de configuration en utilisant les résultats des opérations dbcc checkstorage précédentes.
 - `sp_dbcc_updateconfig` : met à jour les paramètres de configuration pour la base de données spécifiée.
- Supprimez les données obsolètes dans dbccdb :
 - `sp_dbcc_deletedb` : supprime de dbccdb toutes les informations sur la base de données spécifiée.
 - `sp_dbcc_deletehistory` : supprime de dbccdb les résultats des opérations dbcc checkstorage sur la base de données spécifiée.
- Supprimez les espaces de travail superflus.
- Effectuez des contrôles de cohérence sur dbccdb.

Les sections suivantes décrivent en détail les tâches de maintenance.

Réévaluation et mise à jour de la configuration de *dbccdb*

Si les caractéristiques des bases de données utilisateur changent, utilisez `sp_dbcc_evaluatedb` pour réévaluer la configuration courante de dbccdb et recommander des valeurs plus appropriées.

Les modifications apportées aux bases de données utilisateur peuvent avoir des conséquences sur la configuration de dbccdb :

- La création, la suppression ou la modification d'une base de données utilisateur peut avoir des conséquences sur la taille des espaces de travail et du cache nommé ou sur le nombre de tâches de production stockées dans la table dbcc_config.
- Les modifications apportées à la taille du cache nommé ou au nombre de tâches de production pour dbcc_checkstorage peuvent nécessiter la reconfiguration du cache, du buffer et des processus de production.

Si les résultats des opérations dbcc checkstorage sont disponibles pour la base de données cible, utilisez sp_dbcc_evaluatedb pour déterminer les nouvelles valeurs de configuration. sp_dbcc_configreport édite également un rapport sur les paramètres de configuration relatifs à la base de données indiquée.

Utilisez sp_dbcc_updateconfig pour ajouter de nouvelles bases de données à la table dbcc_config et pour modifier les valeurs de la configuration dans dbcc_config en faveur des valeurs recommandées par sp_dbcc_evaluatedb.

Suppression des données obsolètes dans *dbccdb*

Adaptive Server stocke les données générées par dbcc checkstorage dans dbccdb. Vous devez utiliser régulièrement sp_dbcc_deletehistory pour supprimer de dbccdb les données de la base de données cible créée avant la date que vous spécifiez.

Lorsqu'une base de données est supprimée, toutes les informations sur sa configuration ainsi que les résultats de dbcc checkstorage relatifs à cette base doivent être supprimés de dbccdb. Utilisez sp_dbcc_deletedb pour supprimer de dbccdb toutes les informations d'une base de données supprimée.

Suppression des espaces de travail

Il se peut que vous deviez supprimer les espaces de travail inutiles. Pour ce faire, dans dbccdb, émettez :

```
drop table nom_espacedetravail
```

Réalisation de contrôles de cohérence sur *dbccdb*

L'activité de mise à jour étant limitée pour les tables dbccdb, les risques d'altération devraient être moins fréquents. Deux signes annoncent une altération dans dbccdb :

- l'échec de dbcc checkstorage pendant la phase d'initialisation, lors de l'évaluation du travail à effectuer, ou pendant la phase finale, lors de l'enregistrement des résultats ;

- la perte d'informations trouvée par *dbcc checkstorage* sur des erreurs résultant d'une altération des erreurs enregistrées.

Une altération grave de *dbccdb* peut provoquer l'échec de *dbcc checkstorage*. Pour que *dbcc checkstorage* localise les altérations graves de *dbccdb*, vous pouvez créer une base de données identique que vous appellerez *dbccalt* et qui servira uniquement à contrôler *dbccdb*. La création de *dbccalt* s'effectue de la même façon que pour *dbccdb*, comme le décrit la section "Préparation préliminaire à l'utilisation de *dbcc checkstorage*", page 805.

Si aucun device n'est disponible pour *dbccalt*, vous pouvez utiliser tout device qui n'est pas lié aux bases de données master et *dbccdb*.

dbcc checkstorage et les procédures système *dbcc* fonctionnent de la même manière avec *dbccalt* qu'avec *dbccdb*. Lorsque *dbccdb* est la base de données cible, *dbcc checkstorage* utilise *dbccalt*, si cette base existe. Si *dbccalt* n'existe pas, *dbccdb* peut servir de base de données de gestion lors de son propre contrôle. Lorsque *dbccdb* est la base de données cible et que *dbccalt* existe, les résultats des opérations *dbcc checkstorage* sur *dbccdb* sont stockés dans *dbccalt*. Si *dbccalt* n'existe pas, ces résultats sont stockés dans *dbccdb*.

dbcc checkalloc et *dbcc checktable* peuvent être utilisés indifféremment pour contrôler *dbccdb*.

Si *dbccdb* vient à être altérée, vous pouvez la supprimer et la recréer, ou bien charger une version plus ancienne à partir d'une sauvegarde. Si vous la supprimez, certaines données de son historique des diagnostics seront perdues.

Génération de rapports à partir de *dbccdb*

Plusieurs procédures stockées *dbcc* sont fournies avec *dbccdb* pour vous permettre de générer des rapports à partir des données de *dbccdb*.

Rapport de synthèse des opérations *dbcc checkstorage*

sp_dbcc_summaryreport édite les rapports de toutes les opérations *dbcc checkstorage* effectuées pour la base de données spécifiée jusqu'à la date spécifiée. L'exemple suivant présente le résultat pour cette commande :

```
sp_dbcc_summaryreport
DBCC Operation : checkstorage

Database Name      Start time          End Time           Operation ID
```

User Name	Hard Faults	Soft Faults	Text Columns	Abort	Count
-----	-----	-----	-----	-----	-----
sybssystemprocs	05/12/1997	10:54:45	10:54:53		1
0	0	0	0		
sa					
sybssystemprocs	05/12/1997	11:14:10	11:14:19		2
0	0	0	0		
sa					

Pour plus de détails, reportez-vous au chapitre 27, "Procédures stockées dbcc", du *Manuel de référence d'Adaptive Server*.

Rapport sur la configuration, les statistiques et les informations

sp_dbcc_fullreport édite les rapports suivants dans l'ordre :

- sp_dbcc_summaryreport : pour consulter un exemple de résultat pour ce rapport, reportez-vous à la section "Rapport de synthèse des opérations dbcc checkstorage", page 820.
- sp_dbcc_configreport : pour consulter un exemple, reportez-vous à la section "Visualisation des informations de configuration pour une base de données cible", page 821.
- sp_dbcc_statisticsreport : pour consulter un exemple, reportez-vous à la section "Rapport d'informations sur les statistiques de dbcc_counter", page 823.
- sp_dbcc_faultreport short : pour consulter un exemple, reportez-vous à la section "Rapport des erreurs rencontrées dans une base de données objet", page 822.

Visualisation des informations de configuration pour une base de données cible

Utilisez sp_dbcc_configreport pour générer un rapport sur les informations de configuration pour une base de données cible. L'exemple suivant présente le résultat pour cette commande :

```
sp_dbcc_configreport
```

Reporting configuration information of database sybsystemprocs.

Parameter Name	Value	Size
database name	sybsystemprocs	51200K
dbcc named cache	default data cache	1024K
text workspace	textws_001 (id = 544004969)	128K
scan workspace	scanws_001 (id = 512004855)	1024K
max worker processes	1	
operation sequence number	2	

Comparaison des résultats des opérations *dbcc checkstorage*

`sp_dbcc_differentialreport` permet de comparer les résultats des opérations `dbcc checkstorage` effectuées pour la base de données objet spécifiée à des dates spécifiées. L'exemple suivant présente le résultat pour cette commande :

```
sp_dbcc_differentialreport master, sysprocedures,  
checkstorage, "01/01/96", "01/02/96"
```

The following changes in dbcc counter values for the object "sysprocedures" in database master have been noticed between 01/01/96 and 01/02/96.

Description	Date1	Date2
pages used	999	1020
pages reserved	1000	1024
page extent gaps	64	67

Rapport des erreurs rencontrées dans une base de données objet

`sp_dbcc_faultreport` édite le rapport des erreurs survenues jusqu'à la date indiquée dans la base de données objet spécifiée. Vous pouvez générer un rapport court ou long. L'exemple suivant présente un rapport court :

```
sp_dbcc_faultreport 'short'
```

Database Name : sybsystemprocs

Table Name	Index	Type	Code	Description	Page Number
------------	-------	------	------	-------------	-------------

```

-----
sysprocedures      0    100031 page not allocated      5702
sysprocedures      1    100031 page not allocated      14151
syslogs            0    100022 chain start error          24315
syslogs            0    100031 page not allocated      24315

```

L'exemple suivant présente une partie du résultat d'un rapport long pour la base de données sybssystemprocs. Le rapport complet fournit ces mêmes informations pour chaque objet de la base de données cible.

```
sp_dbcc_faultreport 'long'
```

Generating 'Fault Report' for object sysprocedures in database sybssystemprocs.

```

Type Code: 100031; Soft fault, possibly spurious
Page reached by the chain is not allocated.
page id: 14151
page header:
0x00003747000037880000374600000005000648B803EF0001000103FE0080000F
Header for 14151, next 14216, previous 14150, id = 5:1
  time stamp = 0x0001000648B8, next row = 1007, level = 0
  free offset = 1022, minlen = 15, status = 128(0x0080)
.

```

Rapport d'informations sur les statistiques de *dbcc_counter*

sp_dbcc_statisticsreport édite le rapport des informations statistiques de la table dbcc_counter générées par dbcc checkstorage jusqu'à la date spécifiée. L'exemple suivant présente le résultat pour cette commande :

```
sp_dbcc_statisticsreport 'sybssystemprocs',
'sysobjects'
```

Statistics Report on object sysobjects in database sybssystemprocs

Parameter Name	Index Id	Value
count	0	160.0
max size	0	99.0
max count	0	16.0
bytes data	0	12829.0
bytes used	0	15228.0
count	1	16.0
max size	1	9.0

Génération de rapports à partir de dbccdb

```

max level      1      0.0
max count     1      16.0
bytes data    1      64.0
bytes used    1     176.0
count        2     166.0
max level     2      1.0
max size      2     39.0
max count     2     48.0
bytes data    2    3092.0
bytes used    2    4988.0
    
```

Parameter Name	Index Id	Partition	Value	Dev_name
page gaps	0	1	16.0	master
pages used	0	1	17.0	master
extents used	0	1	3.0	master
overflow pages	0	1	0.0	master
pages overhead	0	1	1.0	master
pages reserved	0	1	6.0	master
page extent gaps	0	1	7.0	master
ws buffer crosses	0	1	7.0	master
page extent crosses	0	1	7.0	master
page gaps	1	1	1.0	master
pages used	1	1	2.0	master
extents used	1	1	1.0	master
overflow pages	1	1	0.0	master
pages overhead	1	1	1.0	master
pages reserved	1	1	6.0	master
page extent gaps	1	1	0.0	master
ws buffer crosses	1	1	0.0	master
page extent crosses	1	1	0.0	master
page gaps	2	1	5.0	master
pages used	2	1	8.0	master
extents used	2	1	1.0	master
overflow pages	2	1	0.0	master
pages overhead	2	1	1.0	master
pages reserved	2	1	0.0	master
page extent gaps	2	1	0.0	master
ws buffer crosses	2	1	0.0	master
page extent crosses	2	1	0.0	master

Elaboration d'un plan de sauvegarde et de reprise

Adaptive Server est doté de procédures de **reprise automatique** qui protègent vos données des coupures de courant et des pannes d'ordinateur. Pour vous protéger des pannes de disque, veillez à effectuer des sauvegardes régulières et fréquentes de vos bases de données.

Ce chapitre donne des informations qui vous aident à développer un plan de sauvegarde et de reprise. Sa première partie présente les processus de sauvegarde et de reprise d'Adaptive Server. Sa seconde partie traite des problèmes de sauvegarde et de reprise à traiter avant d'utiliser le système pour la production.

Sujet	Page
Suivi des modifications de la base de données	826
Synchronisation d'une base de données avec son journal : points de reprise	827
Reprise automatique après panne ou arrêt du système	831
Ordre de reprise des bases de données défini par l'utilisateur	832
Isolement de panne lors de la reprise	834
Utilisation des commandes de sauvegarde et chargement	846
Pause et reprise de la mise à jour des bases de données	855
Désignation du responsable des sauvegardes	867
Utilisation du Backup Server pour la sauvegarde et la reprise	868
Démarrage et arrêt du Backup Server	873
Configuration du serveur pour l'accès à distance	873
Choix d'un support de sauvegarde	874
Création de noms de devices logiques pour les devices de sauvegarde locaux	875
Planification des sauvegardes des bases de données utilisateur	877
Planification des sauvegardes de master	880
Planification des sauvegardes de la base de données model	882
Planification des sauvegardes de la base de données sybsystemprocs	882
Configuration d'Adaptive Server pour les chargements simultanés	883
Collecte des statistiques de sauvegarde	884

Suivi des modifications de la base de données

Adaptive Server se sert des transactions pour garder trace de toutes les modifications de la base de données. Les transactions sont les unités de travail d'Adaptive Server. Une transaction est constituée d'une ou de plusieurs instructions Transact-SQL qui aboutissent ou échouent, en tant qu'unité.

Toute instruction SQL qui modifie des données est considérée comme une **transaction**. Les utilisateurs peuvent également définir des transactions en plaçant une série d'instructions au sein d'un bloc `begin transaction...end transaction`. Pour plus d'informations sur les transactions, reportez-vous au chapitre 18, "Transactions : Maintien de la cohérence des données et restauration", du *Guide de l'utilisateur Transact-SQL*.

A chaque base de données est associé un **journal de transactions**, la table système `syslogs`. Le journal de transactions enregistre automatiquement toutes les transactions émises par chaque utilisateur de la base de données. Vous ne pouvez pas désactiver la consignation des transactions dans le journal.

Le journal de transactions est un *journal de préécriture*. Lorsqu'un utilisateur émet une instruction qui va modifier la base de données, Adaptive Server inscrit la modification dans le journal. Une fois que toutes les modifications générées par une instruction sont enregistrées dans le journal, une copie de la page de données est placée en mémoire cache. La page de données reste dans le cache jusqu'à ce qu'une autre page ait besoin de mémoire. Elle est alors écrite sur le disque.

Si une instruction d'une transaction n'aboutit pas, Adaptive Server annule toutes les modifications effectuées par la transaction. Adaptive Server inscrit un enregistrement "end transaction" dans le journal à la fin de chaque transaction, indiquant l'état (succès ou échec) de la transaction.

Informations sur le journal de transactions

Le journal de transactions contient suffisamment d'informations sur chaque transaction pour permettre sa reprise. Utilisez la commande `dump transaction` pour copier les informations du journal sur une bande ou un disque. Exécutez `sp_spaceused syslogs` pour vérifier la taille du journal ou `sp_helpsegment logsegment` pour vérifier l'espace disponible en prévision de l'augmentation de la taille du journal.

Avertissement ! N'utilisez jamais les commandes `insert`, `update` ou `delete` pour modifier `syslogs`.

Synchronisation d'une base de données avec son journal : points de reprise

Les points de reprise permettent d'écrire toutes les pages modifiées (pages modifiées dans la mémoire, mais pas encore écrites sur le disque, depuis le dernier point de reprise) sur le device de base de données. Le mécanisme de **point de reprise** automatique d'Adaptive Server garantit que toutes les pages de données modifiées par des transactions terminées sont régulièrement écrites du cache dans le device de base de données. La synchronisation de la base de données et de son journal de transactions réduit le délai de reprise de la base après une panne système.

Définition de l'intervalle de reprise

Généralement, la reprise automatique demande entre quelques secondes et quelques minutes par base de données. Le délai dépend de la taille de la base de données, de celle du journal de transactions et du nombre de transactions à valider ou à annuler.

Utilisez `sp_configure` avec le paramètre `recovery interval in minutes` pour spécifier le délai de reprise maximal admissible. Adaptive Server effectue des points de reprise automatiques assez souvent pour restaurer la base de données dans ce délai.

```
sp_configure "recovery interval in minutes"
```

La valeur par défaut, 5, définit un délai de reprise de 5 minutes par base de données. Pour ramener ce délai à 3 minutes, exécutez :

```
sp_configure "recovery interval in minutes", 3
```

Remarque L'intervalle de reprise n'a aucune incidence sur les transactions longues et peu journalisées (telles que `create index`) qui sont actives lorsque Adaptive Server est victime de la panne. En effet, l'annulation de ces transactions peut prendre autant de temps que leur exécution. Pour gagner du temps, sauvegardez chaque base de données dès que vous avez créé un index sur une de ses tables.

Procédure de point de reprise automatique

Toutes les minutes environ, la tâche de point de reprise contrôle toutes les bases de données sur le serveur, pour déterminer le nombre d'enregistrements ajoutés au journal de transactions depuis le dernier point de reprise. Si Adaptive Server estime que le délai requis pour restaurer ces transactions est supérieur au délai de reprise de la base de données, il définit un point de reprise.

Les pages modifiées sont écrites du cache dans les devices de base de données et le point de reprise est enregistré dans le journal de transactions. La tâche de point de reprise reste ensuite en veille pendant une minute.

Pour afficher la tâche correspondant au point de reprise, exécutez `sp_who`. Cette tâche est généralement affichée sous "CHECKPOINT SLEEP" dans la colonne "cmd" :

spid	status	loginame	hostname	blk	dbname	cmd
1	running	sa	mars	0	master	SELECT
2	sleeping	NULL		0	master	NETWORK HANDLER
3	sleeping	NULL		0	master	MIRROR HANDLER
4	sleeping	NULL		0	master	HOUSEKEEPER
5	sleeping	NULL		0	master	CHECKPOINT SLEEP

Point de reprise après mise à niveau de la base utilisateur

Adaptive Server insère un enregistrement de point de reprise immédiatement après la mise à niveau d'une base de données utilisateur. Adaptive Server se sert de cet enregistrement pour vérifier que la commande `dump database` est exécutée avant la commande `dump transaction` sur la base de données mise à niveau.

Troncature du journal après des points de reprise automatiques

Les administrateurs système peuvent tronquer le journal de transactions lorsqu'Adaptive Server effectue un point de reprise automatique.

Pour définir l'option de base de données `trunc log on chkpt` qui tronque le journal de transactions s'il contient plus de 50 lignes au moment de l'exécution d'un point de reprise automatique, exécutez la commande suivante à partir de la base de données master :

```
sp_dboption nom_basededonnées, "trunc log on chkpt", true
```

Cette option est inadaptée aux environnements de production car elle ne copie pas le journal de transactions avant de le tronquer. N'utilisez la commande `trunc log on chkpt` que pour :

- les bases de données dont le journal de transactions ne peut être sauvegardé car il ne se trouve pas dans un segment distinct ;
- les bases de données test pour lesquelles les sauvegardes courantes sont sans importance.

Remarque Si vous laissez l'option `trunc log on chkpt` définie à `off` (condition par défaut), le journal de transactions continue à croître jusqu'à ce que vous le tronquiez via la commande `dump transaction`.

Pour éviter que le journal n'arrive à saturation, vous devez définir une procédure de seuil ultime de façon à sauvegarder le journal de transactions. Pour plus d'informations sur la création de procédures associées aux seuils, reportez-vous au chapitre 29, "Gestion de l'espace libre avec des seuils".

Points de reprise libres

Lorsqu'Adaptive Server n'a pas de tâche utilisateur à traiter, une tâche du gestionnaire commence automatiquement à écrire les buffers modifiés sur le disque. Si elle peut vider tous les groupes de buffers actifs dans les caches configurés, elle active la tâche de point de reprise. Cette dernière détermine si elle doit exécuter un point de reprise sur la base de données.

Les points de reprise qui se produisent suite à la tâche de gestion interne sont appelés *points de reprise libres*. Ils ne supposent pas l'écriture de nombreuses pages modifiées dans le device de base de données, dans la mesure où la tâche de gestion interne a déjà effectué ce travail. Ils peuvent entraîner une réduction du délai de reprise de la base de données.

Pour en savoir plus sur l'optimisation de la tâche de gestion interne, reportez-vous au chapitre 3, "Utilisation des moteurs et des CPU", du document *Performances et optimisation*.

Requête manuelle d'un point de reprise

Les propriétaires de base de données peuvent émettre la commande checkpoint pour forcer l'écriture sur disque de toutes les pages modifiées se trouvant en mémoire. Les points de reprise manuels ne tronquent pas le journal, même si l'option `trunc log on chkpt` de `sp_dboption` est activée.

Utilisez la commande checkpoint :

- Par mesure de précaution, dans certaines circonstances : par exemple, juste avant un shutdown with nowait planifié, de sorte que le mécanisme de reprise d'Adaptive Server ait lieu pendant l'intervalle de reprise. (Un shutdown ordinaire effectue un point de reprise.)
- Pour valider une modification des options de base de données après l'exécution de la procédure système `sp_dboption`. (Après l'exécution de `sp_dboption`, un message d'information vous rappelle que vous devez exécuter checkpoint.)

Reprise automatique après panne ou arrêt du système

Chaque fois que vous relancez Adaptive Server (après une coupure de courant, une panne du système d'exploitation ou l'exécution de la commande shutdown, par exemple), il effectue automatiquement un ensemble de procédures de reprise sur chaque base de données.

Le mécanisme de reprise compare chaque base de données à son journal de transactions. Si l'enregistrement du journal pour une modification est plus récent que la page de données, le mécanisme de reprise réapplique la modification à partir du journal de transactions. Si une transaction était en cours au moment de la panne, le mécanisme de reprise annule toutes les modifications effectuées par la transaction.

Lorsque vous amorcez Adaptive Server, le processus de reprise de la base de données se déroule comme suit :

- 1 reprise de la base master,
- 2 reprise de la base sybsystemprocs,
- 3 reprise de la base model,
- 4 création de la base tempdb (par copie de model),
- 5 reprise de la base sybsystemdb,
- 6 reprise de la base sybsecurity,
- 7 reprise des bases de données utilisateur, dans l'ordre de sysdatabases.dbid ou selon l'ordre spécifié par sp_dbrecovery_order. Pour en savoir plus sur sp_dbrecovery_order, reportez-vous aux sections ci-après.

Les utilisateurs peuvent se connecter à Adaptive Server dès que les bases de données système ont été restaurées, mais ne peuvent pas accéder aux autres bases de données tant qu'elles n'ont pas été restaurées.

Choix des messages à afficher pendant une reprise

La variable de configuration print recovery information détermine si Adaptive Server affiche des messages détaillés relatifs à chaque transaction sur l'écran de la console pendant la reprise. Par défaut, ces messages ne sont pas affichés. Pour afficher les messages, utilisez :

```
sp_configure "print recovery information", 1
```

Ordre de reprise des bases de données défini par l'utilisateur

`sp_dbrecovery_order` permet de définir l'ordre de reprise des bases de données utilisateur. Il est ainsi possible d'établir un ordre de reprise selon lequel, par exemple, les bases de données stratégiques seront restaurées avant les bases de données de priorité moindre.

Les caractéristiques importantes de l'ordre de reprise sont les suivantes :

- Les bases de données système sont restaurées en premier, dans l'ordre suivant :
 - master
 - model
 - tempdb
 - sybsystemdb
 - sybsecurity
 - sybsystemprocs

Toutes les autres bases de données sont considérées comme des bases utilisateur et vous pouvez indiquer l'ordre de leur reprise.

- Vous pouvez utiliser `sp_dbrecovery_order` pour spécifier l'ordre de reprise des bases de données utilisateur et indiquer l'ordre de reprise défini par l'utilisateur pour une base de données ou pour toutes les bases de données.
- Les bases de données utilisateur pour lesquelles aucun ordre de reprise n'a été défini via `sp_dbrecovery_order` sont restaurées en fonction de leur ID de base de données après reprise de toutes les bases dotées d'un ordre de reprise défini par l'utilisateur.
- Si vous n'utilisez pas `sp_dbrecovery_order` pour définir un ordre de reprise des bases de données, les bases de données utilisateur sont restaurées dans l'ordre de leur ID.

Utilisation de *sp_dbrecovery_order*

Pour utiliser *sp_dbrecovery_order* afin de définir ou de modifier un ordre de reprise défini par l'utilisateur, vous devez vous trouver dans la base de données master et détenir les droits de l'administrateur système. Tout utilisateur, dans n'importe quelle base de données, peut exécuter *sp_dbrecovery_order* pour afficher l'ordre de reprise des bases de données défini par l'utilisateur.

La syntaxe de *sp_dbrecovery_order* est la suivante :

```
sp_dbrecovery_order  
  [nom_basededonnées [, ordre_rest [, force]]]
```

où *nom_basededonnées* est le nom de la base de données à laquelle vous souhaitez affecter un ordre de reprise et *ordre_rest*, l'ordre de reprise de la base.

L'ordre de reprise doit être consécutif, en commençant par 1. Vous ne pouvez pas affecter la séquence 1, 2, 4, en pensant affecter 3 à une autre base de données ultérieurement.

Pour insérer une base de données dans une séquence de reprise définie par l'utilisateur sans la mettre à la fin, entrez *ordre_rest* et spécifiez *force*. Par exemple, si les bases A, B et C sont affectées de la séquence de reprise utilisateur 1, 2, 3 et que vous souhaitez insérer la base de données pubs2 en deuxième position, entrez :

```
sp_dbrecovery_order pubs2, 2, force
```

Cette commande affecte le numéro 3 dans l'ordre de reprise à la base de données B et le numéro 4 à la base de données C.

Modification ou suppression de la position de reprise d'une base de données

Pour modifier la position d'une base de données dans une séquence de reprise définie par l'utilisateur, supprimez la base de la séquence de reprise et réinsérez-la à la position souhaitée. Si la nouvelle position n'est pas à la fin de la séquence, utilisez l'option *force*.

Pour supprimer une base de données d'une séquence de reprise, indiquez -1 comme ordre de reprise.

Par exemple, pour déplacer la base de données pubs2 de la position de reprise 2 à la position 1, supprimez-la de la séquence de reprise, puis attribuez-lui de nouveau un ordre de reprise, comme suit :

```
sp_dbrecovery_order pubs2, -1
sp_dbrecovery_order pubs2, 1, "force"
```

Affichage de l'ordre de reprise des bases de données défini par l'utilisateur

Pour afficher l'ordre de reprise de toutes les bases de données, utilisez :

```
sp_dbrecovery_order
```

Cette commande génère un résultat semblable à celui-ci :

The following databases have user specified recovery order:

Recovery Order	Database Name	Database Id
1	dbccdb	8
2	pubs2	5
3	pubs3	6
4	pubtune	7

The rest of the databases will be recovered in default database id order.

Pour afficher l'ordre de reprise d'une base de données spécifique, indiquez son nom :

```
1> sp_dbrecovery_order pubs2
2> go
Database Name Database id Recovery Order
-----
pubs2          5          2
```

Isolement de panne lors de la reprise

Les procédures de reprise, appelées plus simplement "reprise", reconstruisent les bases de données du serveur à partir des journaux de transactions. Elles sont déclenchées dans les situations suivantes :

- Démarrage d'Adaptive Server
- Utilisation de la commande load database
- Utilisation de la commande load transaction

La définition du mode d'isolement des pannes contrôle le comportement de la procédure de reprise si des données corrompues sont détectées pendant l'annulation ou la réapplication d'une transaction dans une base de données.

Si un index est signalé comme suspect, l'administrateur système peut réagir en le supprimant, puis en le recréant.

L'isolement des pannes en cours de reprise permet :

- d'interdire l'accès à la base de données dans son intégralité ou seulement aux pages suspectes lorsque la reprise détecte une corruption de données ;
- de définir si une base de données dans son intégralité comportant des pages suspectes n'est accessible en ligne qu'en mode read_only (lecture seule) ou si les pages en ligne sont accessibles en mode modification ;
- d'afficher la liste des bases de données comportant des pages suspectes ;
- d'afficher la liste des pages suspectes d'une base de données spécifiée, par ID de page, ID d'index et nom d'objet ;
- de mettre en ligne les pages suspectes pendant leur réparation à l'intention de l'administrateur système ;
- de mettre en ligne les pages suspectes pour tous les utilisateurs de la base après qu'elles ont été réparées.

La possibilité d'isoler uniquement les pages suspectes tout en rendant accessible le reste de la base de données facilite la gestion en cas de corruption des données. Vous pouvez diagnostiquer les pannes et y remédier éventuellement, tout en laissant l'essentiel des données de la base accessible aux utilisateurs. Vous pouvez estimer l'étendue des dommages et programmer des réparations d'urgence ou remettre l'opération à plus tard.

L'isolement des pannes lors de la reprise ne s'applique qu'aux bases de données utilisateur. La reprise d'une base de données système comportant une page corrompue entraîne la mise hors ligne de l'ensemble de la base. Vous ne pouvez pas restaurer une base de données système tant que vous n'avez pas réparé ou supprimé toutes ses pages corrompues.

Persistance des pages hors ligne

Les pages suspectes mises hors ligne le restent jusqu'à ce que vous réamorciez le serveur. Les informations sur les pages hors ligne sont enregistrées dans `master.dbo.sysattributes`.

Utilisez les commandes `drop database` et `load database` pour effacer les entrées concernant les pages suspectes dans `master.dbo.sysattributes`.

Configuration de l'isolement des pannes lors de la reprise

A l'installation d'Adaptive Server, le mode d'isolement des pannes lors de la reprise est "databases", qui marque les bases suspectes et met hors ligne l'intégralité d'une base s'il détecte une page corrompue.

Isolement des pages suspectes

Pour isoler les pages suspectes afin qu'elles soient seules mise hors ligne et que le reste de la base de données demeure accessible aux utilisateurs, exécutez la commande `sp_setsuspect_granularity` pour définir le mode d'isolement de reprise à "page". Ce mode entre en vigueur à la prochaine exécution d'une reprise de la base de données.

La syntaxe de `sp_setsuspect_granularity` est la suivante :

```
sp_setsuspect_granularity
    [dbname [{"database" | "page"} [, "read_only"]]]
```

Avec *nom_base* et `database` ou `page` comme second argument, `sp_setsuspect_granularity` définit le mode d'isolement lors de la reprise.

Sans l'argument `database` ou `page`, `sp_setsuspect_granularity` affiche les valeurs courantes et configurées du mode d'isolement lors de la reprise pour la base de données spécifiée. Sans aucun argument, elle affiche ces paramètres pour la base de données courante.

S'il est impossible de circonscrire la corruption à une page, la procédure de reprise marque l'intégralité de la base de données comme suspecte, même si vous avez indiqué "page" comme mode d'isolement lors de la reprise. Cela se produit, par exemple, en cas de corruption du journal de transactions ou d'indisponibilité d'une ressource globale.

Lorsque la procédure de reprise marque certaines pages comme suspectes, par défaut, la base de données reste accessible en lecture et en écriture, les pages suspectes étant mises hors ligne et donc inaccessibles. Si, toutefois, vous spécifiez l'option `read_only` de `sp_setsuspect_granularity` et que la reprise marque des pages comme suspectes, la base de données est mise en ligne en mode `read_only` et ne peut pas être modifiée. Si, tout en choisissant l'option `read_only`, vous souhaitez autoriser les utilisateurs à modifier les pages non suspectes, vous pouvez rendre inscriptible la partie en ligne de la base de données via `sp_dboption` :

```
sp_dboption pubs2, "read only", false
```

Dans ce cas, les pages suspectes restent hors ligne jusqu'à ce que vous les réparez ou les forciez, comme décrit à la section "Mise en ligne des pages hors ligne", page 839.

Augmentation du nombre des pages suspectes admises

Le seuil de suspicion est le nombre de pages suspectes à partir duquel la reprise marque l'intégralité de la base de données comme suspecte, même si le mode d'isolement lors de la reprise est "page". Par défaut, sa valeur est de 20 pages par base de données. Pour modifier le seuil de suspicion, utilisez `sp_setsuspect_threshold`.

La syntaxe de `sp_setsuspect_threshold` est la suivante :

```
sp_setsuspect_threshold [dbname [,seuil]]
```

Avec les arguments `nom_base` et `seuil`, `sp_setsuspect_threshold` affiche le seuil de suspicion courant et configuré pour la base de données spécifiée. Sans aucun argument, elle affiche ces paramètres pour la base de données courante.

Vous pouvez configurer le mode d'isolement lors de la reprise et le seuil de suspicion au niveau de la base de données.

Vous ne pouvez pas exécuter `sp_setsuspect_granularity` ou `sp_setsuspect_threshold` au sein d'une transaction.

Vous devez détenir le rôle `sa_role` et vous trouver dans la base de données master pour définir des valeurs via `sp_setsuspect_granularity` et `sp_setsuspect_threshold`. Tout utilisateur peut exécuter ces procédures avec le seul nom de la base de données comme argument, afin d'afficher les valeurs configurées pour cette base de données, comme illustré ci-après :

Isolement de panne lors de la reprise

```
sp_setsuspect_granularity pubs2
DB Name  Cur. Suspect Gran.  Cfg. Suspect Gran.  Online mode
-----  -
pubs2    page                page                read/write
sp_setsuspect_threshold pubs2
DB Name  Cur. Suspect threshold  Cfg. Suspect threshold
-----  -
pubs2    20                      30
```

Dans cet exemple, le mode d'isolement lors de la reprise de la base de données pubs2 était "page" et le seuil de suspicion était égal à 20 lors de la dernière exécution d'une reprise sur cette base de données (valeurs du seuil de suspicion courant). A la reprise suivante sur cette base de données, le mode d'isolement lors de la reprise sera "page" et le seuil de suspicion sera égal à 30 (valeurs configurées).

Sans argument, `sp_setsuspect_granularity` et `sp_setsuspect_threshold` affichent les valeurs courantes et configurées pour la base de données courante, s'il s'agit d'une base utilisateur.

Informations relatives aux bases de données et aux pages hors ligne

Pour connaître les bases de données comportant des pages hors ligne, exécutez `sp_listsuspect_db`. La syntaxe est la suivante :

```
sp_listsuspect_db
```

L'exemple suivant affiche des informations générales sur les pages suspectes :

```
sp_listsuspect_db
The database 'dbt1' has 3 suspect pages belonging to 2 objects.
(return status = 0)
```

Pour obtenir des informations détaillées sur chacune des pages hors ligne, utilisez `sp_listsuspect_page`. La syntaxe est la suivante :

```
sp_listsuspect_page [dbname]
```

Si vous ne spécifiez pas `dbname`, la commande s'applique par défaut à la base de données courante. L'exemple suivant montre le résultat détaillé au niveau page de `sp_listsuspect_page` dans la base de données dbt1.

```
sp_listsuspect_page dbt1
DBName  Pageid    Object    Index    Access
-----
dbt1    384       tab1      0        BLOCK_ALL
dbt1    390       tab1      0        BLOCK_ALL
dbt1    416       tab1      1        SA_ONLY
```

(3 rows affected, return status = 0)

Si la valeur de la colonne "Access" est SA_ONLY, la page suspecte est 1 et elle est accessible aux utilisateurs détenteurs du rôle sa_role. Si la valeur est BLOCK_ALL, personne ne peut accéder à cette page.

Tout utilisateur peut exécuter sp_listsuspect_db et sp_listsuspect_page à partir de n'importe quelle base de données.

Mise en ligne des pages hors ligne

Pour rendre accessibles toutes les pages hors ligne d'une base de données, exécutez sp_forceonline_db. La syntaxe est la suivante :

```
sp_forceonline_db dbname,
{"sa_on" | "sa_off" | "all_users"}
```

Pour rendre accessible une page hors ligne, exécutez sp_forceonline_page. La syntaxe est la suivante :

```
sp_forceonline_page dbname, pgid
{"sa_on" | "sa_off" | "all_users"}
```

Avec ces deux procédures, vous spécifiez le type d'accès.

- "sa_on" rend la page ou la base de données suspecte accessible aux seuls utilisateurs détenteurs du rôle sa_role. Cela est très utile pour réparer les pages suspectes et tester les réparations avec la base de données active, sans autoriser pour autant l'accès des utilisateurs normaux aux pages suspectes. Vous pouvez également utiliser cette méthode pour effectuer un dump database ou un dump transaction with no_log sur une base de données comportant des pages suspectes, ce qui serait impossible si les pages restaient hors ligne.
- "sa_off" bloque l'accès de tous les utilisateurs, administrateurs système compris. Cette commande annule une procédure antérieure sp_forceonline_db ou un sp_forceonline_page avec "sa_on".
- "all_users" met en ligne et rend accessibles à tous les utilisateurs les pages hors ligne une fois qu'elles ont été corrigées.

Contrairement à la mise en ligne des pages suspectes avec "sa_on" qui permet leur remise hors ligne via "sa_off", l'exécution de sp_forceonline_page ou sp_forceonline_db pour mettre en ligne des pages pour "all users" est une action irréversible. Il n'y a aucun moyen de remettre les pages hors ligne.

Avertissement ! Adaptive Server n'effectue aucun contrôle sur les pages remises en ligne. Il vous incombe de vérifier que les pages remises en ligne ont été réparées.

Vous ne pouvez pas exécuter sp_forceonline_db ou sp_forceonline_page au sein d'une transaction.

Pour exécuter sp_forceonline_db et sp_forceonline_page, vous devez détenir le rôle sa_role et vous trouver dans la base de données master.

Isolement des pannes au niveau index pour les tables verrouillées au niveau des données seules

Lorsque les pages d'un index d'une table verrouillée au niveau des données seules sont marquées comme suspectes pendant une reprise, l'intégralité de l'index est mis hors ligne. Deux procédures système gèrent les index hors ligne :

- sp_listsuspect_object
- sp_forceonline_object

Le plus souvent, l'administrateur système utilise sp_forceonline_object pour rendre un index suspect accessible aux seuls détenteurs du rôle sa_role. Si l'index suspect est associé à une table utilisateur, vous pouvez le supprimer, puis le recréer.

Reportez-vous au Manuel de référence d'Adaptive Server pour en savoir plus sur sp_listsuspect_objec et sur sp_forceonline_object.

Incidences des pages hors ligne

Les restrictions suivantes s'appliquent aux bases de données comportant des pages hors ligne :

- Les transactions qui requièrent des données hors ligne, directement ou indirectement (par exemple, du fait de contraintes d'intégrité référentielle), échouent en affichant un message.
- Vous ne pouvez pas exécuter dump database si une partie de la base de données est hors ligne.

L'administrateur système peut forcer la mise en ligne de pages hors ligne via la commande `sp_forceonline_db` avec "sa_on", sauvegarder la base de données, puis exécuter `sp_forceonline_db` avec "sa_off" une fois la sauvegarde achevée.

- Vous ne pouvez pas exécuter dump transaction with no_log ou dump transaction with truncate_only si une partie de la base de données est hors ligne.

L'administrateur système peut forcer la mise en ligne de pages hors ligne via la commande `sp_forceonline_db` avec "sa_on", sauvegarder le journal de transactions via `with no_log`, puis exécuter `sp_forceonline_db` avec "sa_off" une fois la sauvegarde achevée.

- Pour supprimer une table ou un index contenant des pages hors ligne, vous devez effectuer une transaction dans la base de données master. Faute de quoi, la suppression échoue car elle doit supprimer des entrées pour les pages suspectes à partir de `master.dbo.sysattributes`. Dans l'exemple suivant, l'objet est supprimé et les informations concernant ses pages hors ligne sont effacées à partir de `master.dbo.sysattributes`.

Pour supprimer l'index `authors_au_id_ind` contenant des pages suspectes de la base de données `pubs2`, supprimez-le au sein d'une transaction base de données master, en procédant comme suit :

```
use master
go
sp_dboption pubs2, "ddl in tran", true
go
use pubs2
go
checkpoint
go
begin transaction
drop index authors.au_id_ind
commit
go
use master
go
```

```
sp_dboption pubs2, "ddl in tran", false
go
use pubs2
go
checkpoint
go
```

Stratégies de reprise exploitant l'isolement des pannes lors de la reprise

Il existe deux stratégies principales pour ramener une base de données comportant des pages suspectes à un état cohérent pendant que des utilisateurs y accèdent : le rechargement et la réparation.

Les deux méthodes requièrent :

- une sauvegarde correcte de la base de données,
- une série de sauvegardes fiables du journal de transactions jusqu'au point où la base de données est restaurée avec des pages suspectes,
- une sauvegarde du journal de transactions sur un device immédiatement après reprise de la base de données, pour capturer les modifications effectuées sur les pages hors ligne,
- des sauvegardes en continu du journal de transactions sur des devices pendant que des utilisateurs travaillent sur la base de données partiellement hors ligne.

Stratégie de rechargement

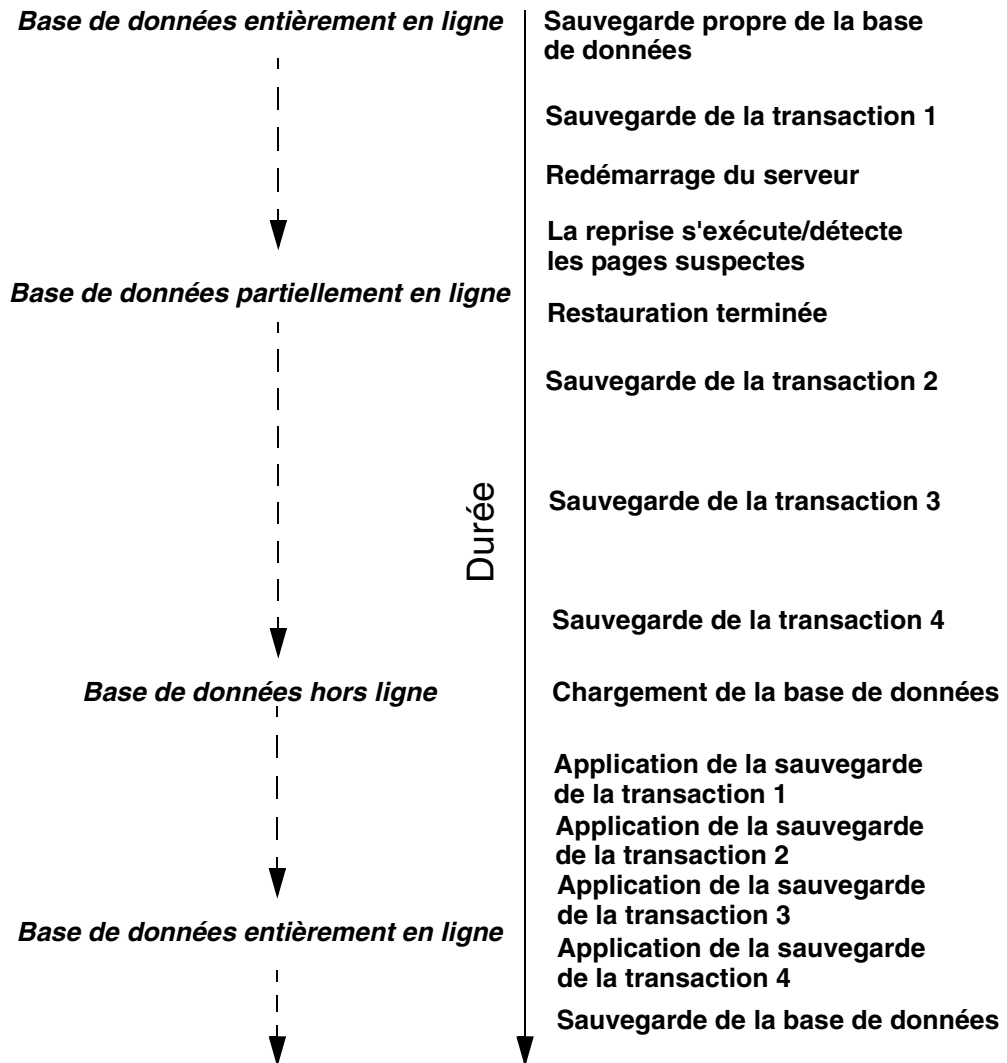
Le rechargement consiste à restaurer une base de données "propre" à partir de sauvegardes. A un moment opportun, chargez la sauvegarde la plus récente de la base de données et appliquez les journaux de transactions pour restaurer la base.

load database efface les informations concernant la page suspecte des tables système master.dbo.sysdatabases et master.dbo.sysattributes.

Dès que la base de données restaurée est en ligne, sauvegardez-la immédiatement.

La figure 26-1 présente la stratégie de rechargement des bases de données.

Figure 26-1 : Stratégie de rechargement



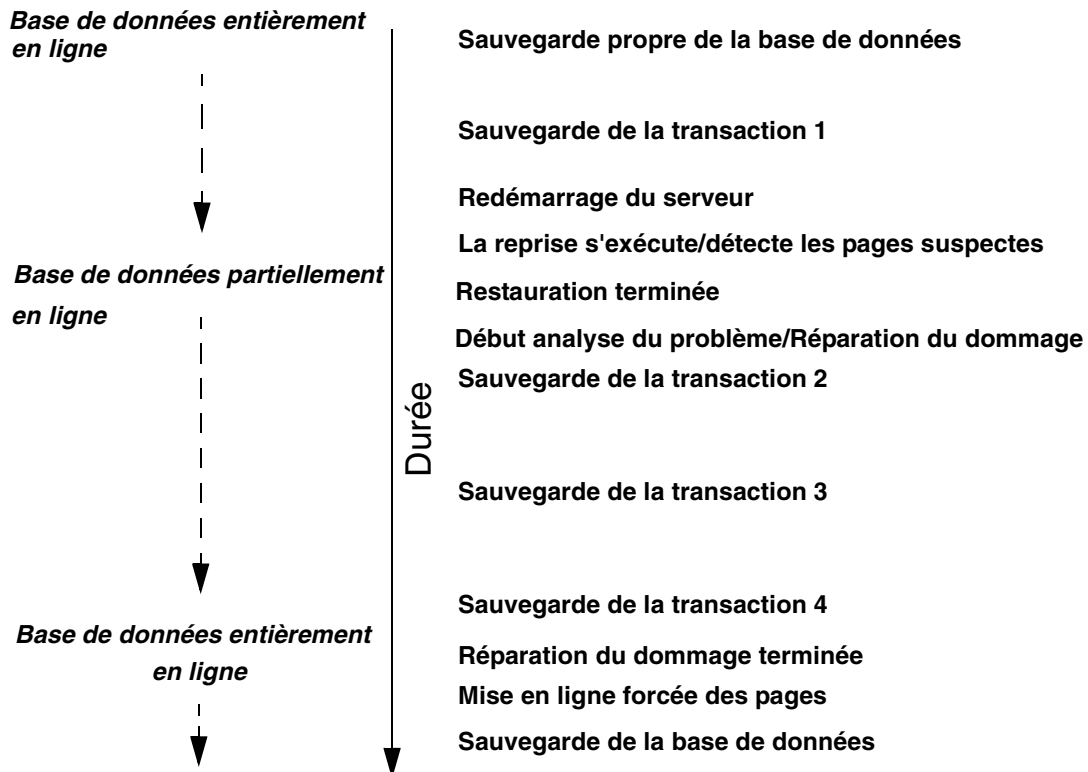
Stratégie de réparation

La stratégie de réparation suppose de réparer les pages corrompues pendant que la base de données est partiellement hors ligne. Pour faire le diagnostic de pannes et y remédier, utilisez des méthodes connues, comme les commandes dbcc, lancez des requêtes avec les résultats connus sur les pages suspectes et, au besoin, prenez contact avec le Support Technique de Sybase. La réparation des dommages peut également amener à supprimer et recréer des objets contenant des pages suspectes.

Vous pouvez utiliser soit `sp_forceonline_page` pour mettre en ligne les pages individuellement, au fur et à mesure qu'elles sont réparées, soit attendre que toutes les pages hors ligne soient réparées et les mettre en ligne toutes à la fois avec `sp_forceonline_db`.

La stratégie de réparation ne requiert pas de mettre hors ligne l'intégralité de la base de données. La figure 26-2 montre la stratégie utilisée pour réparer les pages corrompues.

Figure 26-2 : Stratégie de réparation



Estimation de l'étendue de la corruption

Vous pouvez parfois utiliser la procédure d'isolement des pannes lors de la reprise pour estimer l'étendue de la corruption, en forçant l'exécution de la reprise et en examinant le nombre de pages marquées suspectes ainsi que les objets auxquels elles appartiennent.

Par exemple, si des utilisateurs signalent des pannes sur une base de données particulière, définissez le mode d'isolement lors de la reprise à "page" et forcez la reprise en redémarrant Adaptive Server. La reprise terminée, exécutez `sp_listsuspect_db` ou `sp_listsuspect_page` pour déterminer le nombre de pages suspectes et les objets de base de données concernés.

Si l'intégralité de la base de données est marquée suspecte et que vous recevez le message :

```
Reached suspect threshold '%d' for database '%.*s'.  
Increase suspect threshold using  
sp_setsuspect_threshold.
```

Lancez `sp_setsuspect_threshold` pour augmenter le seuil de suspicion et forcez de nouveau l'exécution d'une reprise. Chaque fois que vous recevez ce message, vous pouvez augmenter le seuil et relancer la reprise jusqu'à ce que la base de données soit de nouveau en ligne. Si vous ne recevez pas ce message, la corruption n'est pas localisée dans des pages spécifiques, auquel cas cette stratégie n'est pas applicable.

Utilisation des commandes de sauvegarde et chargement

Dans le cas d'une panne de disque, un crash, par exemple, vous pouvez restaurer vos bases de données si et seulement si, vous disposez de sauvegardes régulières de ces bases et de leurs journaux de transactions. La reprise sera d'autant plus complète que vous aurez régulièrement exécuté les commandes `dump database` et `dump transaction` pour sauvegarder vos bases et les commandes `load database` et `load transaction` pour les restaurer. Ces commandes sont brièvement décrites ci-après et plus en détail dans le chapitre 27, "Sauvegarde et restauration de bases de données utilisateur", et le chapitre 28, "Restauration des bases de données système".

Avertissement ! N'utilisez jamais les commandes de copie du système d'exploitation pour copier un device de base de données. Le fait de charger la copie dans Adaptive Server provoquerait une corruption générale de la base de données.

Les commandes de sauvegarde peuvent aboutir même si la base de données est corrompue. Avant de sauvegarder une base de données, exécutez la commande `dbcc` pour vérifier sa cohérence. Pour plus d'informations, reportez-vous à la section chapitre 25, "Contrôle de la cohérence des bases de données",.

Avertissement ! Si vous sauvegardez directement sur bande, n'y stockez pas d'autres types de fichiers (sauvegardes UNIX, fichiers tar, etc.). Cela risquerait d'endommager les fichiers de sauvegarde Sybase. Par contre, si vous effectuez la sauvegarde dans un système de fichiers UNIX, vous pouvez archiver sur bande les fichiers créés de cette manière.

Sauvegarde systématique de la base de données : *dump database*

La commande `dump database` copie l'intégralité de la base de données, avec toutes les données et le journal de transactions. `dump database` *ne tronque pas* le journal.

`dump database` autorise les **sauvegardes dynamiques**. Les utilisateurs peuvent continuer à modifier la base de données pendant la sauvegarde. Il est ainsi peu contraignant d'effectuer des sauvegardes régulièrement.

dump database s'exécute en trois étapes. Un message de progression vous informe de la fin de chaque étape. Lorsque la sauvegarde est terminée, elle reflète toutes les modifications effectuées pendant son exécution, excepté celles commencées au cours de l'étape 3.

Sauvegarde systématique du journal de transactions : *dump transaction*

Exécutez la commande `dump transaction` (ou sa version abrégée, `dump tran`) pour effectuer des sauvegardes systématiques du journal de transactions. `dump transaction` est semblable aux sauvegardes incrémentielles fournies par nombre de systèmes d'exploitation. Elle copie le journal de transactions, fournissant, le cas échéant, un enregistrement des modifications effectuées depuis la dernière sauvegarde de la base ou du journal de transactions. Une fois le journal copié, `dump transaction` tronque la partie inactive.

`dump transaction` est plus rapide et occupe moins d'espace de stockage qu'une sauvegarde complète de la base de données et elle est généralement exécutée plus souvent. Les utilisateurs peuvent continuer à modifier la base de données pendant la sauvegarde. Vous ne pouvez exécuter `dump transaction` que si le journal de la base se trouve dans un segment distinct.

Après une panne de disque, spécifiez l'option `with no_truncate` de `dump transaction` pour sauvegarder le journal de transactions. Vous disposez ainsi d'un enregistrement du journal de transactions jusqu'au moment de la panne.

Copie du journal après une panne de device : *dump tran with no_truncate*

Si votre device de données tombe en panne et que la base de données est inaccessible, utilisez l'option `with no_truncate` de `dump transaction` pour obtenir une copie courante du journal. Cette option ne tronque pas le journal. Vous ne pouvez l'utiliser que si le journal de transactions se trouve sur un segment distinct et que la base de données master est accessible.

Restauration de l'intégralité de la base de données : *load database*

Exécutez la commande `load database` pour charger la sauvegarde créée avec `dump database`. Vous pouvez charger la sauvegarde dans une base de données préexistante ou en créer une nouvelle via l'option `for load`. Si vous créez une base de données, affectez-lui au moins autant d'espace qu'à la base d'origine.

Avertissement ! Vous ne pouvez pas charger de sauvegarde effectuée sur une autre plate-forme ou générée dans une version antérieure à la version 10.0 de SQL Server. Si la base de données que vous chargez comporte des tables contenant les clés primaires pour des tables d'autres bases de données, vous devez charger la sauvegarde dans une base de données portant le même nom que la base sauvegardée.

La commande `load database` définit l'état de la base à "offline", ce qui signifie que vous ne pouvez pas utiliser les options `no chkpt` ou `recovery`, `dbo use only` et `read only` de `sp_dboption` avant de charger une base de données. Toutefois, personne ne peut exploiter une base de données pendant son chargement et celui du journal de transactions. Pour rendre la base de données accessible aux utilisateurs, lancez la commande `online database`.

Une fois la base de données chargée, Adaptive Server peut avoir besoin de :

- mettre à "zéro" toutes les pages inutilisées, si la base est chargée dans une base plus grande que la base sauvegardée,
- finaliser la reprise, en appliquant aux données les modifications du journal de transactions.

Selon le nombre de pages non allouées ou de transactions longues, cette opération peut prendre quelques secondes ou plusieurs heures pour une base de données très volumineuse. Adaptive Server émet des messages indiquant qu'il met les pages "à zéro" ou qu'il a commencé la reprise. Ces messages sont normalement mis en buffer. Pour les afficher, exécutez :

```
set flushmessage on
```


Application des modifications à la base de données : *load transaction*

Une fois la base de données chargée, exécutez la commande `load transaction` (ou sa version abrégée, `load tran`) pour charger les sauvegardes des journaux de transactions *dans l'ordre de réalisation des sauvegardes*. Ce processus reconstruit la base de données en réappliquant les modifications enregistrées dans le journal de transactions. Si besoin est, vous pouvez restaurer une base de données jusqu'à un point déterminé du journal de transactions, via l'option `until_time` de `load transaction`.

Les utilisateurs ne peuvent pas modifier la base de données entre les commandes `load database` et `load transaction`, du fait de l'état "offline" imposé par la commande `load database`.

Vous ne pouvez charger les sauvegardes du journal de transactions que si elles sont au même niveau de version que la base de données associée.

Lorsque l'intégralité de la séquence des sauvegardes du journal de transactions a été chargée, la base de données reflète toutes les transactions validées au moment de la dernière sauvegarde du journal de transactions.

Mise à disposition des utilisateurs de la base de données : *online database*

A la fin de la séquence de chargement, passez la base de données à l'état "online" pour la mettre à disposition des utilisateurs. Une base de données chargée via `load database` reste inaccessible tant que vous n'exécutez pas la commande `online database`.

Avant de lancer cette commande, vérifiez que vous avez bien chargé tous les journaux de transactions requis.

Déplacement d'une base de données vers un autre Adaptive Server

Vous pouvez utiliser `dump database` et `load database` pour déplacer une base de données d'un Adaptive Server à un autre, sous réserve que les deux Adaptive Server soient actifs sur la même plate-forme matérielle et logicielle. Vous devez toutefois vérifier que les affectations de devices sur l'Adaptive Server de destination correspondent à celles de l'original. Faute de quoi, les segments système et définis par l'utilisateur dans la nouvelle base ne correspondront pas à ceux de la base d'origine.

Pour préserver l'affectation des devices lorsque vous chargez une sauvegarde de base de données sur un nouvel Adaptive Server, respectez les mêmes instructions que pour restaurer une base de données utilisateur à partir d'un disque en panne. Pour plus d'informations, reportez-vous à la section "Contrôle de l'utilisation de l'espace", page 945.

Suivez également les conseils ci-après lorsque vous déplacez des bases de données système vers d'autres devices :

- Avant de déplacer la base de données master, annulez toujours la mise en miroir du device maître. A défaut, Adaptive Server tentera d'utiliser le fichier de l'ancien device miroir lorsque vous démarrerez Adaptive Server avec le nouveau device.
- Lorsque vous déplacez la base de données master, choisissez un device de même taille que l'original pour éviter des erreurs d'allocation dans sysdevices.
- Pour déplacer la base de données sybsecurity, faites passer la nouvelle base en mode mono-utilisateur avant d'y charger les anciennes données.

Mise à niveau d'une base de données utilisateur

Vous pouvez charger des sauvegardes dans la version courante d'Adaptive Server à partir de n'importe quelle version d'Adaptive Server version 10.0 et ultérieures. La base de données chargée n'est pas mise à niveau tant que vous n'exécutez pas online database.

Les étapes de mise à niveau d'une base de données utilisateur sont les mêmes que pour une base de données système :

- 1 Exécutez load database pour charger la sauvegarde d'une base de données version 10.0 ou ultérieures d'Adaptive Server. load database met la base de données à l'état "offline".
- 2 Exécutez load transaction pour charger, *dans l'ordre*, tous les journaux de transactions générés depuis la dernière sauvegarde de la base de données. Vérifiez que tous les journaux de transactions sont chargés avant de passer à l'étape 3.

- 3 Exécutez online database pour mettre la base de données à niveau. La commande online database met la base à niveau car son état actuel est incompatible avec la version actuelle d'Adaptive Server. La mise à niveau achevée, la base de données passe à l'état "online", ce qui la met à disposition du public.
- 4 Effectuez une sauvegarde de la base de données mise à niveau. Vous devez exécuter la commande dump database pour pouvoir exécuter la commande dump transaction.

Pour en savoir plus sur load database, load transaction et online database, reportez-vous au *Manuel de référence d'Adaptive Server*.

Options spéciales de *dump transaction*

Dans certaines circonstances, le modèle simple décrit ci-dessus ne s'applique pas. Le tableau 26-1 indique quand il convient d'utiliser les options spéciales with no_log et with truncate_only au lieu de la commande standard dump transaction.

Avertissement ! Conformez-vous *strictement* aux indications du tableau 26-1 pour l'utilisation des commandes spéciales dump transaction. Notamment, n'utilisez dump transaction with no_log qu'en dernier recours et uniquement si dump transaction with no_truncate a échoué. La commande dump transaction with no_log libère extrêmement peu d'espace dans le journal de transactions. Si vous continuez à charger des données après l'exécution de dump transaction with no_log, le journal peut être saturé, entraînant l'échec de toute commande dump transaction ultérieure. Utilisez la commande alter database pour allouer de l'espace supplémentaire à la base de données.

Tableau 26-1 : Utilisation de dump transaction avec truncate_only ou with no_log

Lorsque	Utiliser
Le journal se trouve sur le même segment que les données.	dump transaction with truncate_only pour tronquer le journal dump database pour copier l'intégralité de la base de données, journal compris

Lorsque	Utiliser
La reprise des transactions récentes ne vous intéresse pas (par exemple, si elles ont été effectuées dans un autre environnement de développement).	<code>dump transaction with truncate_only</code> pour tronquer le journal <code>dump database</code> pour copier l'intégralité de la base de données
Votre méthode habituelle pour sauvegarder le journal de transactions (commande standard <code>dump transaction</code> ou <code>dump transaction with truncate_only</code>) a échoué par manque d'espace dans le journal.	<code>dump transaction with no_log</code> pour tronquer le journal sans enregistrer l'événement <code>dump database</code> immédiatement après pour copier l'intégralité de la base de données, journal compris

Options de chargement spéciales pour identifier les fichiers sauvegardés

Utilisez l'option `with headeronly` pour obtenir des informations d'en-tête pour un fichier particulier ou pour le premier fichier d'une bande. Utilisez l'option `with listonly` pour obtenir des informations sur tous les fichiers d'une bande. Ces options ne chargent pas réellement les bases de données ou les journaux de transactions sur la bande.

Remarque Ces options s'excluent mutuellement. Si vous spécifiez les deux, `with listonly prime`.

Restauration d'une base de données à partir de sauvegardes

La figure 26-3 illustre le processus de reprise d'une base de données créée à 16 h 30 le lundi et sauvegardée immédiatement après sa création. Des sauvegardes complètes des bases de données sont effectuées tous les soirs à 17 h. Les sauvegardes des journaux de transactions ont lieu tous les jours à 10 h, 12 h, 14 h et 16 h :

Figure 26-3 : Restauration d'une base de données, exemple de scénario

Exécution des sauvegardes systématiques		Restauration de la base de données à partir des sauvegardes	
Lundi 16 h 30	create database	Mardi 18 h 15 Bande 7	dump transaction with no_truncate
Lundi 17 h Bande 1 (180 Mo)	dump database	Mardi 18 h 20 Bande 6	load database
Mardi 10 h Bande 2 (45 Mo)	dump transaction	Mardi 18 h 35 Bande 7	load transaction
Mardi midi Bande 3 (45 Mo)	dump transaction	Mardi 18 h 50	online database
Mardi 14 h Bande 4 (45 Mo)	dump transaction		
Mardi 16 h Bande 5 (45 Mo)	dump transaction		
Mardi 17 h Bande 6 (180 Mo)	dump database		
Mardi 18 h	Panne de device !		

Si le disque qui stocke les données tombe en panne mardi à 18 h, procédez comme suit pour restaurer la base de données :

- 1 exécutez `dump transaction with no_truncate` pour obtenir une sauvegarde courante du journal de transactions ;
- 2 exécutez `load database` pour charger la sauvegarde la plus récente de la base de données (bande 6). `load database` met la base à l'état "offline" ;
- 3 exécutez `load transaction` pour appliquer la sauvegarde la plus récente du journal de transactions (bande 7) ;
- 4 exécutez `online database` pour remettre la base de données "online".

La figure 26-4 montre comment restaurer la base de données lorsque le device de données est tombé en panne mardi à 16 h 59, c'est-à-dire juste avant le début des sauvegardes programmées tous les soirs :

Figure 26-4 : Restauration d'une base de données, deuxième exemple de scénario

Exécution des sauvegardes systématiques		Restauration de la base de données à partir des sauvegardes	
Lundi 16 h 30	create database	Mardi 17 h 15 Bande 6	dump transaction with no_truncate
Lundi 17 h Bande 1 (180 Mo)	dump database	Mardi 17 h 20 Bande 1	load database
Mardi 10 h Bande 2 (45 Mo)	dump transaction	Mardi 17 h 35 Bande 2	load transaction
Mardi midi Bande 3 (45 Mo)	dump transaction	Mardi 17 h 40 Bande 3	load transaction
Mardi 14 h Bande 4 (45 Mo)	dump transaction	Mardi 17 h 45 Bande 4	load transaction
Mardi 16 h Bande 5 (45 Mo)	dump transaction	Mardi 17 h 50 Bande 5	load transaction
Mardi 16 h 59	Panne de device !	Mardi 17 h 55 Bande 6	load transaction
Mardi 17 h Bande 6	dump database	Mardi 18 h	online database

Pour restaurer la base de données, procédez comme suit :

- 1 Exécutez `dump transaction with no_truncate` pour obtenir une sauvegarde courante du journal de transactions sur la bande 6 (la bande qui aurait dû être utilisée pour la sauvegarde de routine de la base de données).
- 2 Exécutez `load database` pour charger la sauvegarde la plus récente de la base de données (bande 1). `load database` met la base à l'état "offline".

- 3 Exécutez load transaction pour charger les bandes 2, 3, 4 et 5 et la sauvegarde la plus récente du journal de transactions, la bande 6.
- 4 Exécutez online database pour remettre la base de données "online".

Pause et reprise de la mise à jour des bases de données

quiesce database hold vous permet de bloquer la mise à jour d'une ou plusieurs bases de données lors de l'annulation de la mise en miroir d'un disque ou de la copie externe des devices de la base. Parce qu'aucune écriture n'a lieu pendant ce temps, la copie externe (l'image secondaire) de la base de données est identique à l'image primaire. Quand la base de données se trouve à l'état de repos, seules les requêtes en lecture sont autorisées. Pour reprendre la mise à jour de la base de données, exécutez quiesce database release à la fin de la copie externe. Vous pouvez charger la copie externe de la base de données sur un serveur secondaire en veillant à posséder une copie de votre image primaire cohérente du point de vue transactionnel. Vous pouvez exécuter quiesce database hold à partir d'une connexion isql puis vous connecter avec une autre connexion isql et exécuter quiesce database release.

La syntaxe de la commande quiesce database est la suivante :

```
quiesce database nom_étiquette hold nom_basededonnées [, nom_basededonnées]  
[for external dump]
```

ou

```
quiesce database nom_étiquette release
```

où *nom_étiquette* est une étiquette définie par l'utilisateur pour la liste de bases de données à mettre en pause ou à relancer et *nom_basededonnées* le nom de la base de données dont vous suspendez les mises à jour.

Remarque *nom_étiquette* doit respecter les règles appliquées aux identifiants. Reportez-vous au *Manuel de référence d'Adaptive Server* pour la liste des types de requêtes. Vous devez utiliser le même *nom_étiquette* pour quiesce database...hold et quiesce database...release.

Ainsi, pour accéder à la base de données pubs2, entrez :

```
quiesce database pubs_tag hold pubs2
```

Adaptive Server écrit des messages identiques à ce qui suit dans le journal d'erreurs :

```
QUIESCE DATABASE command with tag pubs_tag is being executed by process 9.
```

```
Process 9 successfully executed QUIESCE DATABASE with HOLD option for tag  
pubs_tag. Processes trying to issue IO operation on the quiesced database(s)  
will be suspended until user executes Quiesce Database command with RELEASE  
option.
```

Les mises à jour de pubs2 ne pourront se terminer que lorsque la base de données aura été libérée. Pour libérer les tables de la base de données pub2, entrez :

```
quiesce database pubs_tag release
```

Une fois la base libérée, vous pouvez appeler le serveur secondaire à l'aide du paramètre -q si vous avez exécuté la clause for external dump. La reprise assure la cohérence transactionnelle des bases de données mais vous pouvez attendre d'amener la base en ligne avant d'appliquer le journal de transactions.

Consignes d'utilisation de quiesce database

L'utilisation la plus simple de quiesce database est la copie d'une installation complète. Elle assure la cohérence des mappages du système. Ces mappages sont effectués sur l'installation secondaire quand les bases de données système qui les contiennent sont copiées physiquement dans le cadre de l'exécution de quiesce database hold pour plusieurs bases de données. Ils sont terminés quand toutes les bases de données utilisateur de l'installation source sont copiées ensemble. quiesce database autorise huit noms de bases de données pendant une même opération. Si une installation source contient plus de huit bases de données, vous pouvez exécuter plusieurs instances de quiesce database hold pour créer plusieurs états de repos simultanés pour plusieurs groupes de bases de données.

Pour créer une installation source sans éléments existants, utilisez des scripts quasiment identiques afin de créer les installations primaire et secondaire. Le script de l'installation secondaire peut varier en fonction du nom des devices physiques transmis à la commande `disk init`. Cette approche requiert d'appliquer au serveur secondaire les mêmes mises à jour des devices système qu'au serveur primaire. Par exemple, si vous exécutez une commande `alter database` sur le serveur primaire, vous devez l'exécuter également sur le serveur secondaire avec des paramètres identiques. Cela nécessite que les devices de base de données soient pris en charge par un gestionnaire de volume à même de fournir aux serveurs primaire et secondaire les mêmes noms de devices physiques pour des devices physiquement distincts.

Votre site peut élaborer ses propres procédures de copie externe de devices de base de données. Mais Sybase recommande ce qui suit :

- incluez la base master dans la liste de bases de quiesce database ;
- attribuez un nom aux devices utilisant des chaînes identiques sur les serveurs primaire et secondaire ;
- assurez un environnement identique aux bases de données système master, model et sybsystemprocs dans les installations primaire et secondaire. Il faut en particulier que les mappages de sysusages et les identifiants de base de données des bases copiées soient identiques sur les serveurs primaire et secondaire et que l'identifiant des bases des deux serveurs soit identique dans sysdatabases.
- Le mappage entre syslogins.suid et sysusers.suid doit demeurer cohérent sur le serveur secondaire.
- Si les serveurs primaire et secondaire partagent une copie de master et si l'entrée sysdevices de chaque device copié utilise des chaînes identiques, les valeurs de *physname* sur les deux serveurs doivent être physiquement distinctes.
- La copie externe d'une base de données s'accompagne des restrictions suivantes :
 - La copie ne peut commencer qu'à la fin de l'exécution de quiesce database hold.
 - Tous les devices de l'ensemble des bases figurant dans la liste de quiesce database doivent être copiés.
 - Il faut attendre la fin de la copie externe avant d'invoquer quiesce database release.

- Pendant l'intervalle accordé par quiesce database pour la copie externe, il est impossible de procéder à des mises à jour sur l'un des disques de l'une quelconque des bases de données figurant dans la liste de quiesce database. Cet espace est défini dans sysusages. Néanmoins, si l'espace d'un device est partagé par une base figurant dans la liste de quiesce database et une base qui n'y figure pas, il se peut que le device partagé soit mis à jour pendant la copie externe. Lorsque vous décidez de l'endroit où vous localisez les bases de données dans un système sur lequel vous envisagez d'effectuer des copies externes, deux possibilités s'offrent à vous :
 - séparer les bases de données afin qu'elles ne partagent pas de devices dans l'environnement où vous exécuterez quiesce database ou bien
 - prévoir de copier toutes les bases de données sur le device (après avoir suivi la recommandation énoncée ci-dessus d'effectuer une copie de toute l'installation).
- Exécutez quiesce database uniquement lorsque l'activité de mise à jour des bases de données est peu importante (de préférence pendant une période d'activité lecture seulement). Lorsque vous mettez la base de données au repos pendant une période calme, vous incommodez un nombre réduit d'utilisateurs et, selon le sous-système d'E/S tiers chargé d'effectuer la copie externe, vous diminuez le temps requis par la synchronisation des devices participant à la copie.

Rôles serveur dans une relation primaire et secondaire

Si votre site comprend deux Adaptive Servers, dont l'un joue le rôle de serveur primaire et l'autre celui de serveur secondaire recevant des copies externes des bases de données du serveur primaire, vous ne devez jamais mélanger leurs rôles. En d'autres termes, chaque serveur peut changer de rôle (le serveur primaire peut devenir serveur secondaire et réciproquement), mais un même serveur ne peut pas remplir les deux rôles simultanément.

Démarrage du serveur secondaire à l'aide de l'option -q

L'option -q de `dataserver` identifie le serveur secondaire. N'utilisez pas l'option -q pour démarrer le serveur primaire. Avec l'option -q, les bases de données utilisateur copiées lors de l'exécution de `quiesce database for external dump` demeurent hors ligne jusqu'à ce que l'une des séquences suivantes soit exécutée :

- enregistrement du journal de transactions d'une base de données du serveur primaire avec accès en attente (c'est-à-dire `dump tran with standby_access`) suivi de `load tran with standby_access` pour copier cette base sur le serveur secondaire, puis exécution de `online database for standby access` sur cette même base.
- forçage de la base de données en ligne pour accès en lecture et écriture en exécutant `online database`. Si vous procédez comme cela cependant, la reprise de la base de données écrit des enregistrements de journal de compensation et vous ne pouvez pas charger le journal de transactions sans, soit charger la base, soit effectuer une nouvelle copie des devices primaires en exécutant `quiesce database`.

Les bases de données système dérogent à l'option -q, montent en ligne et écrivent des enregistrements de journal de compensation pour toutes les transactions annulées.

Valeur de l'enregistrement du journal de la base de données "au repos" mise à jour

Si vous démarrez le serveur secondaire en exécutant l'option -q de `dataserver`, Adaptive Server émet un message au démarrage indiquant que la base de données est "in quiesce" pour toutes les bases de données marquées en interne comme "au repos".

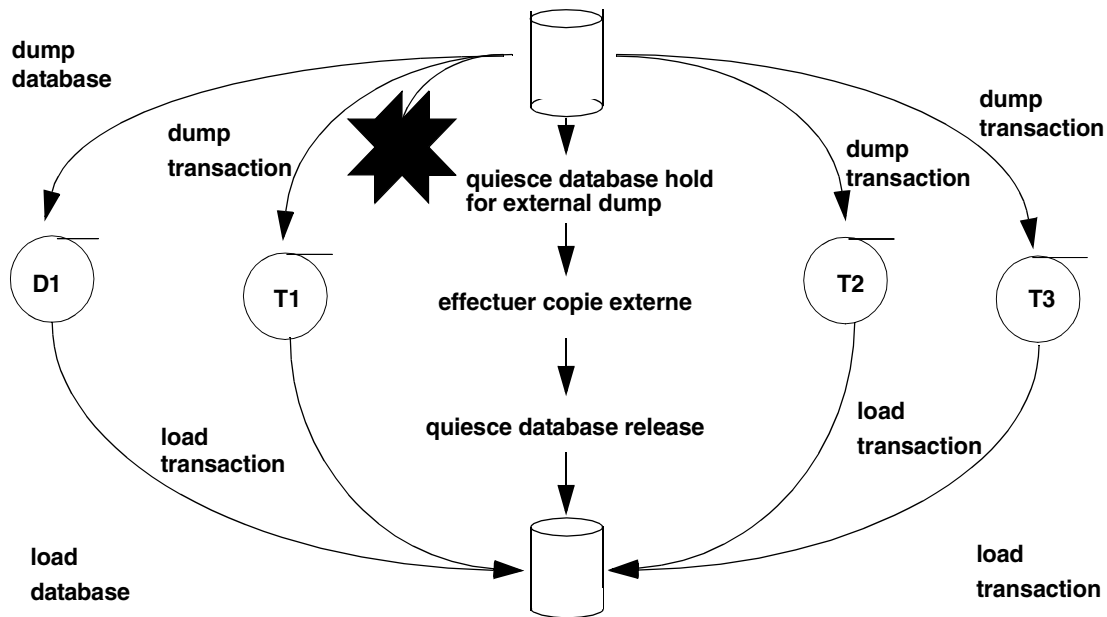
La reprise à l'aide de -q des bases de données copiées avec `quiesce database for external dump` fonctionne dans une large mesure comme la reprise effectuée par `load database`. Comme dans le cas de `load database`, elle enregistre en interne l'adresse du dernier enregistrement de journal courant, afin qu'une nouvelle commande `load transaction` puisse la comparer à celle du dernier enregistrement de journal courant précédent. Une différence entre ces deux valeurs signifie que la base de données secondaire a fait l'objet d'une activité et Adaptive Server indique l'erreur numéro 4306.

Mise à jour du numéro de séquence de sauvegarde

Tout comme dump database, quiesce database met à jour les numéros de séquence de sauvegarde en cas d'écritures non consignées dans le journal. Ceci évite que vous utilisiez une sauvegarde antérieure de la base de données ou une copie externe comme base d'une séquence de sauvegarde.

Par exemple, dans la méthode de reprise semi-automatique décrite à la figure 26-5, dump database (D1), dump transaction (T1), quiesce database, dump transaction (T2) et dump transaction (T3) produisent des archivages :

Figure 26-5 : Séquence de sauvegarde en reprise semi-automatique



En général, dans un environnement comportant des mises à jour consignées dans le journal mais pas dump tran with truncate_only, vous pouvez charger D1, T1, T2 et T3 successivement en évitant les quiesce database hold. Cette approche s'utilise en cas de reprise semi-automatique, quand les sauvegardes de bases de données réussies sur le serveur primaire simplifient les scénarios de reprise après panne de disque. Sur le serveur secondaire ou serveur en attente, utilisé pour les systèmes d'aide à la décision, vous pourrez préférer des applications progressives continues de load transaction plutôt que des interruptions provoquées par la copie externe.

Cependant, en cas d'opération non consignée dans le journal (par exemple, un `select into` tel que décrit à la figure 26-5) après le `dump transaction` générant T1, `dump transaction to archive` n'est pas autorisée et vous devez, soit créer une autre sauvegarde de la base, soit émettre `quiesce database for external copy` puis procéder à une nouvelle copie externe de la base. L'émission de l'une ou l'autre de ces commandes met à jour le numéro de séquence de sauvegarde et supprime la marque bloquant `dump transaction to archive`.

Le recours éventuel à la clause `for external dump` dépend du traitement que la reprise appliquera à la base de données au repos marquée `in quiesce`.

`quiesce database hold`

Si vous émettez `quiesce database` sans la clause `for external dump`, le serveur secondaire ne fonctionne pas pendant la copie externe créant le jeu secondaire de bases de données et la reprise avec `-q` ne voit pas de bases de données copiées marquées "au repos." Elle effectue la reprise de chaque serveur normalement lors de la reprise au démarrage ; elle n'en effectue pas la reprise comme décrit précédemment pour `for load database`. Toute tentative subséquente d'exécuter un `load tran` sur l'une de ces bases de données génère l'erreur 4306, "There was activity on database since last load ..." (il y a eu une activité sur la base de données depuis le dernier chargement) ou 4305, "Specified file '%.*s' is out of sequence ..." (le fichier spécifié '%.*s' est hors séquence...).

Qu'il se soit déroulé ou non une activité non consignée dans la base de données primaires, le numéro de séquence de sauvegarde n'est pas incrémenté par `quiesce database hold` et les bits d'écritures non consignées ne sont pas supprimés par `quiesce database release`.

`quiesce database hold for external dump`

Lors de l'exécution de `quiesce database for external dump`, la copie externe de la base de données "se souvient" qu'elle a été effectuée pendant un intervalle de repos, ce qui permet à `-q` de procéder à sa reprise comme avec `load database`. `quiesce database release` supprime ces informations de la base de données primaire. Si des écritures non consignées avaient empêché l'exécution de `dump tran to archive` sur le serveur primaire, `dump tran to archive` est maintenant actif.

En cas d'écritures non consignées depuis la dernière exécution de `dump database` ou `quiesce database hold for external dump` dans l'une quelconque des bases de données figurant dans la liste de `quiesce database`, le numéro de séquence de sauvegarde est mis à jour par `quiesce database hold for external dump` et les informations des écritures non consignées sont supprimées par `quiesce database release`. La mise à jour du numéro de séquence provoque l'échec de `load tran` si cette commande est appliquée à une cible autre que la copie externe créée par la `quiesce database` qui l'a mise à jour. Ceci ressemble au comportement `dump database` d'une base de données comportant des écritures non consignées.

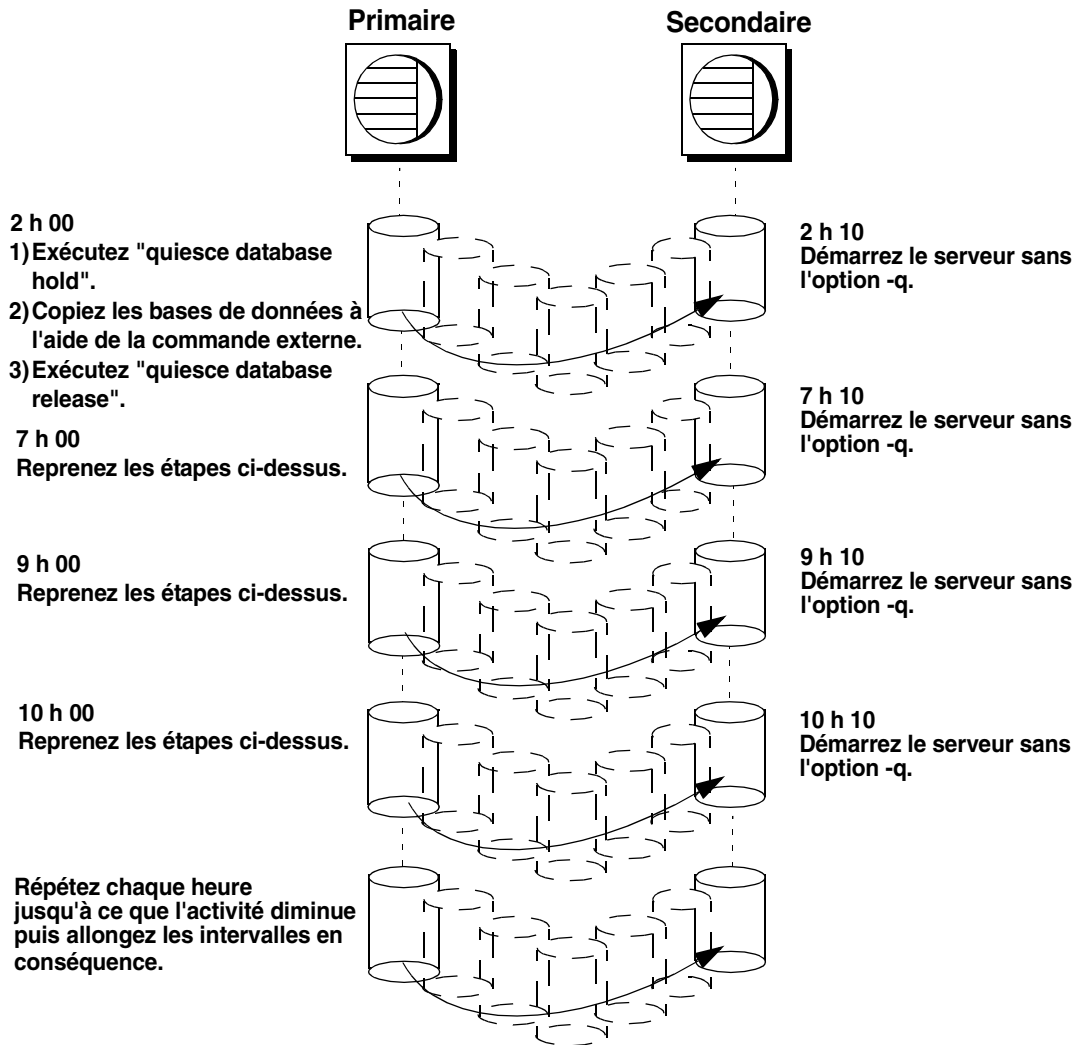
Avvertissement ! `quiesce database for external dump` supprime la balise interne qui vous empêche d'exécuter `dump transaction to device_archive` que vous ayez ou non effectué une copie externe ou une sauvegarde de la base de données. `quiesce database` n'a aucun moyen de savoir si vous avez procédé à une copie externe. *C'est à vous d'effectuer cette tâche.* Si vous exécutez `quiesce database hold for external dump` pour effectuer une protection provisoire contre l'écriture au lieu d'une copie servant de base à une nouvelle séquence de sauvegarde, alors que votre application comprend des écritures non consignées, Adaptive Server risque d'autoriser la création de sauvegardes de journal de transactions inutilisables. Dans ce cas, `dump transaction to device_archive` réussit dans un premier temps mais les futures commandes de transaction de chargement risquent de refuser ces archivages parce qu'ils sont hors séquence.

Sauvegarde des devices primaires à l'aide de *quiesce database*

En général, les utilisateurs sauvegardent leurs bases de données à l'aide de `quiesce database` en ayant recours à l'une des méthodes décrites ci-après. Toutes deux vous permettent de décharger les applications d'aide à la décision du serveur de processeur de transactions en ligne (OLTP) pendant le service normal :

- Actualisation itérative du device primaire – copie du device primaire sur le device secondaire pendant les intervalles d'actualisation Mettez la base au repos avant chaque actualisation. La figure 26-6 représente un système procédant à des sauvegardes hebdomadaires à l'aide de cette méthode :

Figure 26-6 : Planning de sauvegardes avec la méthode d'actualisation itérative

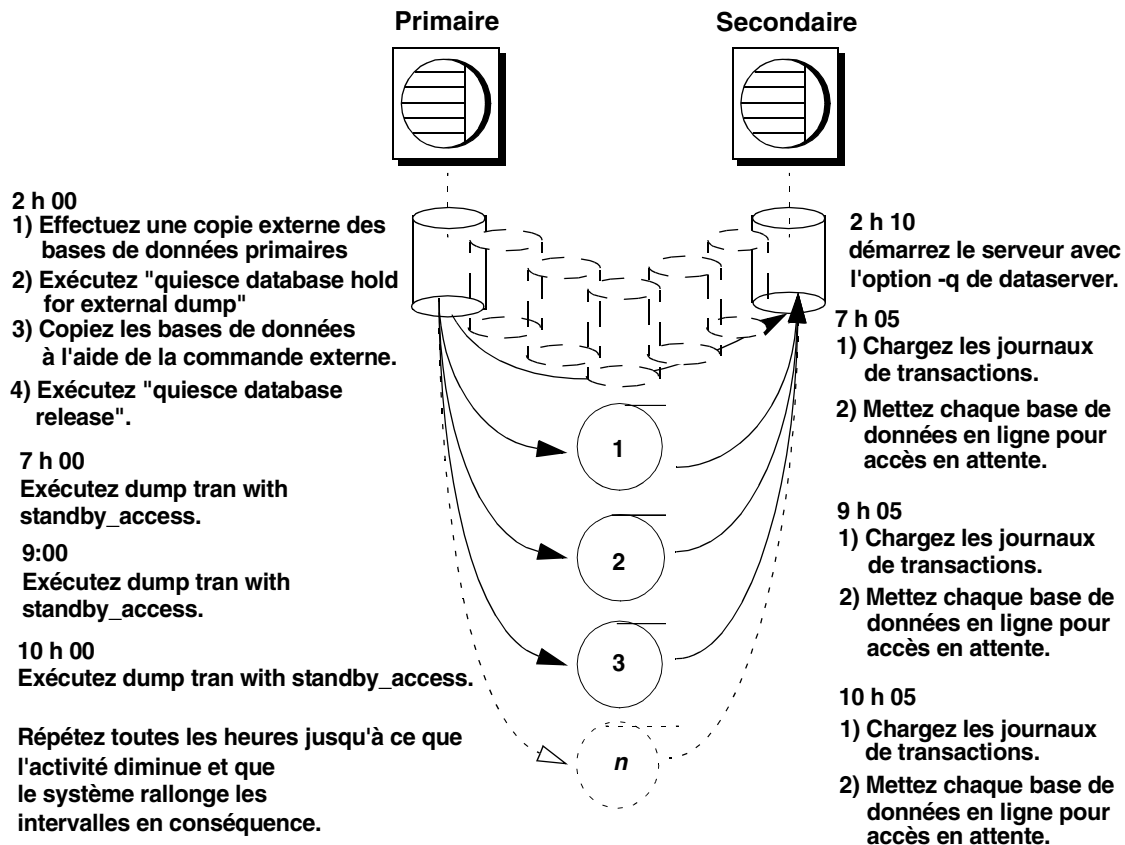


Si vous utilisez la méthode d'actualisation itérative, il n'est pas nécessaire d'exécuter l'option -q pour redémarrer le serveur secondaire (après un blocage ou une maintenance système). Les transactions non terminées génèrent des enregistrements de journal de compensation et les bases de données concernées montent en ligne normalement.

- Méthode de la reprise semi-automatique – autorise une totale concurrence d'accès pour le serveur OLTP parce qu'elles ne bloquent pas les écritures.

Après la copie externe (secondaire) des devices de bases de données primaires à l'aide de la clause `for external dump`, actualisez les bases de données secondaires par application périodique des journaux de transactions avec les sauvegardes du serveur primaire. Cette méthode nécessite de mettre les bases au repos une fois pour effectuer une copie externe du groupe de bases de données puis de les actualiser périodiquement en exécutant la commande `dump tran with standby_access`. La figure 26-7 représente un système ayant recours à la mise à jour quotidienne du device primaire puis à des sauvegardes horaires du journal de transactions.

Figure 26-7 : Planning de sauvegardes avec la méthode de reprise semi-automatique



Reprise de bases de données avec la méthode de reprise semi-automatique

Si vous avez recours à la méthode de la reprise semi-automatique, il faut indiquer à Adaptive Server s'il démarre le serveur primaire ou le serveur secondaire. Exécutez l'option `-q` de la commande `dataserver` pour préciser que vous démarrez le serveur secondaire. Si vous ne le démarrez pas avec l'option `-q` :

- la reprise des bases de données s'effectue normalement, comme avec `load database` ;
- les transactions non consignées au moment de l'émission de `quiesce database` sont annulées.

Pour plus d'informations, reportez-vous à la section "Démarrage du serveur secondaire à l'aide de l'option `-q`", page 859.

La séquence de reprise se déroule différemment si la base de données est marquée `in quiesce`.

Reprise de bases de données non marquées "in quiesce"

Avec l'option `-q`, la reprise d'une base non marquée `in quiesce` se déroule comme sur le serveur primaire. Autrement dit, si des opérations antérieures ne sont pas en cours de chargement, la base fait l'objet d'une reprise totale et monte en ligne. Les transactions non terminées sont annulées et des enregistrements de journal de compensation sont écrits pendant la reprise.

Reprise de bases de données marquées "in quiesce"

- Bases de données utilisateur – Les bases utilisateur marquées `in quiesce` font l'objet d'une reprise identique à celle générée par la commande `load database`. Cela permet à `load tran` de détecter l'activité qui s'est éventuellement déroulée dans la base de données primaire depuis l'arrêt du serveur. Après le démarrage du serveur secondaire à l'aide de l'option `-q`, le processus de reprise rencontre la marque `in quiesce`. Adaptive Serveur émet un message indiquant que la base de données est en cours de chargement et hors ligne. Si vous utilisez la méthode de la reprise semi-automatique, ne mettez pas la base de données en ligne pour lui faire jouer son rôle de système d'aide à la décision avant que ne soit chargée la première sauvegarde de transactions générée par une commande `dump tran with standby_access`. Exécutez ensuite `online database for standby_access`.

- Bases de données système – Les bases systèmes viennent entièrement en ligne immédiatement. La marque `in quiesce` est effacée et ignorée.

Marque des copies archivées pendant l'état repos

`quiesce database hold for external dump` signale votre intention de procéder à des copies externes de vos bases de données pendant l'état repos. Parce que ces copies externes sont effectuées après l'émission de `quiesce database hold`, la cohérence transactionnelle de la base est assurée puisque vous êtes certain qu'il n'y a pas eu d'écritures pendant l'intervalle entre `quiesce database hold` et `quiesce database release` ; la reprise peut donc s'exécuter comme une reprise au démarrage. Cette opération est illustrée par la figure 26-5.

En l'absence de mises à jour non consignées et de commande `dump tran with truncate_only`, vous pouvez charger D1, T1, T2 et T3 successivement en contournant les éventuelles commandes `quiesce database...hold`. En revanche, si une opération non consignée (telle qu'une `select into` représentée à la figure 26-5) se produit après la transaction de sauvegarde à l'origine de T1, il devient impossible d'archiver la transaction de sauvegarde.

L'exécution de la clause `quiesce database hold for external dump` résout ce problème en effaçant les bits d'état qui empêchent l'exécution de la commande `dump transaction to archive` suivante et en modifiant le numéro de séquence de la copie externe afin de créer une base pour la séquence de chargement. Le numéro de séquence ne change pas s'il n'existe pas d'écritures non consignées.

Vous pouvez effectuer des copies externes des bases de données avec ou sans la clause `for external dump`. Mais pour appliquer les sauvegardes de transactions ultérieures du serveur primaire au serveur secondaire, il faut inclure la clause `for external dump`, comme illustré ci-après :

```
quiesce database nom_étiquette hold
nom_basededonnées [, nom_basededonnées] ... [for
external dump]
```

Par exemple :

```
quiesce database pubs_tag hold pubs2 for external
dump
```

Si les mappages de la base de données n'ont pas changé depuis le lancement de l'instance primaire de la base, les étapes de la réalisation d'une sauvegarde externe pour une seule base sont les suivantes :

- 1 Exécutez la commande `quiesce database hold for external dump` :

```
quiesce database pubs_tag hold pubs2 for external
dump
```

- 2 Effectuez une copie externe de la base de données à l'aide de la méthode adéquate pour votre site.

- 3 Exécutez `quiesce database nom_étiquette release` :

```
quiesce database pubs_tag release
```

Avertissement ! La suppression des bits d'état et la mise à jour du numéro de séquence vous permet de procéder à une transaction de sauvegarde avec ou sans copie externe après l'exécution de `quiesce database`. Adaptive Server n'est pas en mesure de savoir si vous avez effectué une copie externe pendant le temps qui sépare `quiesce database... hold for external dump` de `quiesce database... release`. Si vous exécutez la commande `quiesce database hold for external dump` pour effectuer une protection contre l'écriture provisoire au lieu d'une copie pouvant servir de base à une nouvelle séquence de sauvegarde alors que votre application comprend des écritures non consignées, Adaptive Server vous laisse créer des sauvegardes d'enregistrements de transactions inutilisables. `dump transaction to device_archive` réussit mais `load transaction` refuse ces archivages parce qu'ils sont hors séquence.

Désignation du responsable des sauvegardes

Dans nombre d'entreprises, un opérateur est chargé de l'exécution de toutes les opérations de sauvegarde et de reprise. Seul un administrateur système, le propriétaire d'une base de données ou un opérateur sont habilités à exécuter les commandes de sauvegarde et de chargement. Le propriétaire d'une base de données ne peut sauvegarder que sa propre base. L'opérateur et l'administrateur système peuvent sauvegarder et charger n'importe quelle base.

Tout utilisateur peut exécuter `sp_volchanged` pour signaler au Backup Server le changement d'un volume de bande.

Utilisation du Backup Server pour la sauvegarde et la reprise

Les sauvegardes et les reprises sont effectuées par un programme Open Server dénommé Backup Server, exécuté sur la même machine qu'Adaptive Server. Vous pouvez effectuer des sauvegardes sur le réseau avec un Backup Server sur l'ordinateur distant et un autre sur l'ordinateur local.

Remarque Backup Server ne peut pas sauvegarder sur des volumes multi-disques.

Backup Server :

- Crée et charge à partir de "sauvegardes sur devices". La **sauvegarde sur device** permet d'utiliser jusqu'à 32 devices de sauvegarde en parallèle. La base de données est alors divisée en parties presque égales, chaque partie étant sauvegardée sur un device distinct.
- Crée et charge des sauvegardes simples qui occupent plusieurs bandes.
- Sauvegarde et charge via le réseau sur un Backup Server exécuté sur une autre machine.
- Sauvegarde plusieurs bases de données ou journaux de transactions sur une seule bande.
- Charge un seul fichier à partir d'une bande contenant plusieurs bases de données ou journaux de transactions.
- Prend en charge les options de gestion de bandes spécifiques de la plate-forme.
- Achemine les requêtes de gestion de volume vers la session dans laquelle a été émise la commande de sauvegarde ou de chargement ou vers la console de l'opérateur.
- Détecte les caractéristiques physiques des devices de sauvegarde pour déterminer les protocoles, la taille de blocs et autres caractéristiques.

Relations entre Adaptive Server et Backup Server

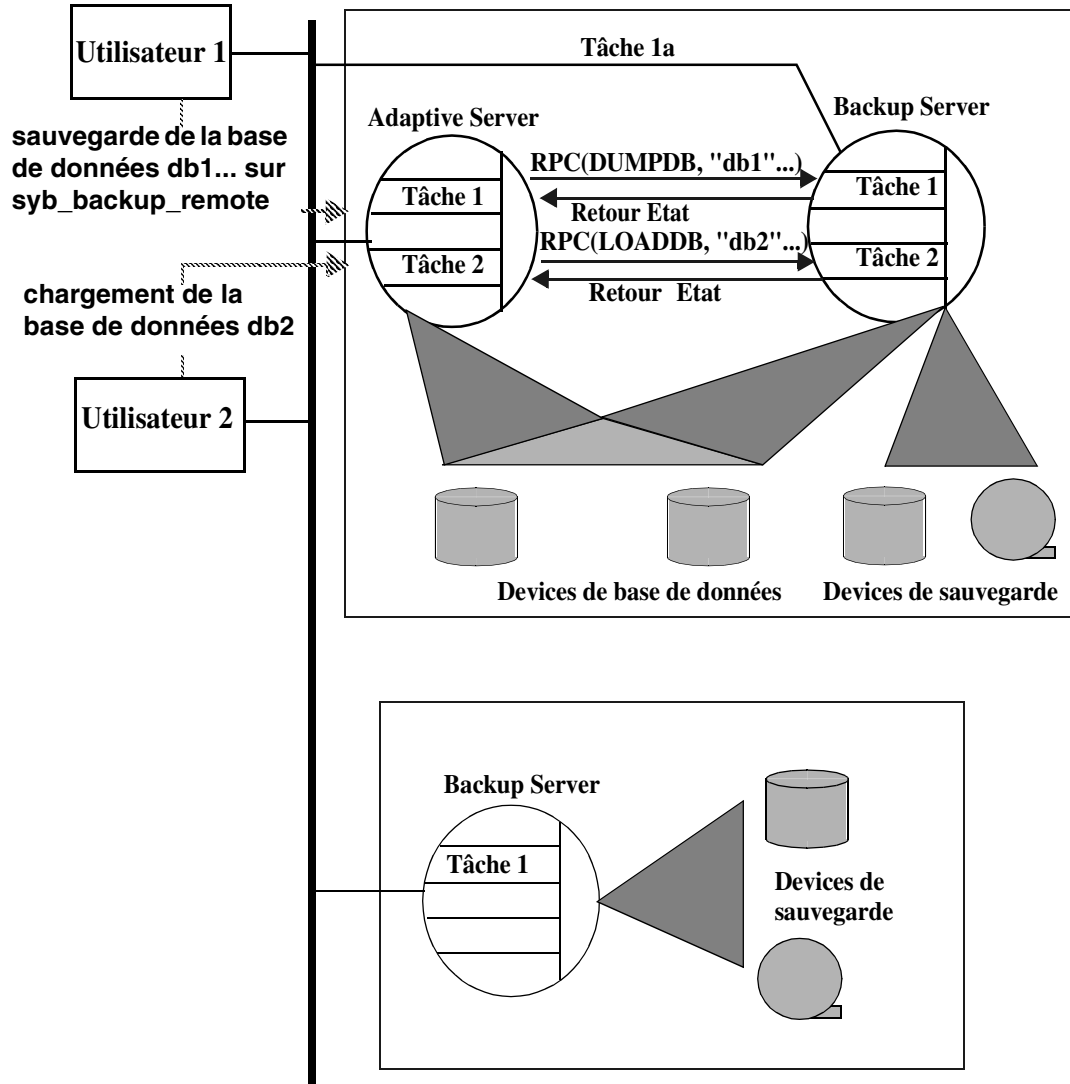
La figure 26-8 montre deux utilisateurs effectuant simultanément des opérations de sauvegarde sur deux bases de données :

- L'utilisateur1 sauvegarde la base de données db1 sur un Backup Server distant.
- L'utilisateur2 charge la base de données db2 à partir du Backup Server local.

Chaque utilisateur lance la commande de sauvegarde ou de chargement appropriée à partir d'une session Adaptive Server. Adaptive Server interprète la commande et envoie des appels de procédure à distance (RPC) au Backup Server. Les appels précisent les pages de la base de données à sauvegarder ou à charger, les devices de sauvegarde à utiliser et d'autres options.

Pendant l'exécution des sauvegardes et des chargements, Adaptive Server et Backup Server utilisent les RPC pour échanger instructions et messages d'état. Le Backup Server et non Adaptive Server, effectue tous les transferts de données requis par les commandes de sauvegarde et de chargement.

Figure 26-8 : Adaptive Server et Backup Server avec Backup Server distant



Lorsque le Backup Server local reçoit les instructions de sauvegarde de l'utilisateur1, il lit les pages spécifiées dans les devices de la base de données et les envoie au Backup Server distant. Ce dernier sauvegarde les données sur un support hors ligne.

Simultanément, le Backup Server local exécute la commande de chargement de l'utilisateur2 en lisant les données à partir des devices de sauvegarde locaux et en les inscrivant sur le device de la base de données.

Communication avec le Backup Server

Pour utiliser les commandes de sauvegarde et de chargement, Adaptive Server doit pouvoir communiquer avec son Backup Server. Voici les conditions requises :

- Le Backup Server doit être exécuté sur la même machine qu'Adaptive Server (ou sur le même cluster pour OpenVMS).
- Le Backup Server doit figurer dans la table master..syssservers. L'entrée du Backup Server, SYB_BACKUP, est créée dans syssservers lors de l'installation d'Adaptive Server. Utilisez la commande sp_helpserver pour vérifier cette information.
- Le Backup Server doit figurer dans le fichier d'interface. L'entrée pour le Backup Server local est créée lors de l'installation d'Adaptive Server. Le nom du Backup Server dans le fichier d'interface doit correspondre au nom de la colonne srvnet pour l'entrée SYB_BACKUP dans master..syssservers. Si vous avez installé un Backup Server distant sur une autre machine, créez le fichier d'interface sur un système de fichiers partagé par les deux machines ou copiez l'entrée dans le fichier d'interface local. Le nom du Backup Server distant doit être identique dans les deux fichiers d'interface.
- L'utilisateur qui démarre le processus Backup Server doit détenir l'autorisation d'écriture sur les devices de sauvegarde. L'utilisateur "sybase" qui démarre habituellement Adaptive Server et le Backup Server peut lire et écrire dans les devices de la base de données.
- Adaptive Server doit être configuré pour l'accès à distance. Par défaut, Adaptive Server est installé avec l'accès à distance activé. Pour plus d'informations, reportez-vous à la section "Configuration du serveur pour l'accès à distance", page 873.

Montage d'un nouveau volume

Remarque Backup Server ne peut pas sauvegarder sur des volumes multi-disques.

Au cours du processus de sauvegarde et de reprise, l'opérateur change les volumes de bande. Si le Backup Server détecte une panne sur le volume actuellement monté, il demande un changement de volume en envoyant des messages au client ou à la console de l'opérateur. Après le montage d'un autre volume, l'opérateur avertit le Backup Server en exécutant `sp_volchanged` sur Adaptive Server.

Sur les systèmes UNIX, le Backup Server demande un changement de volume lorsque la bande est saturée. L'opérateur monte une autre bande, puis exécute `sp_volchanged` (voir le tableau 26-2).

Tableau 26-2 : Changement de volumes de bande sur un système UNIX

Séquence	Opérateur utilisant <i>isql</i>	Adaptive Server	Backup Server
1	Exécute la commande <code>dump database</code> .		
2		Envoie la requête de sauvegarde au Backup Server.	
3			Reçoit le message de requête de sauvegarde d'Adaptive Server. Envoie à l'opérateur le message de montage de bande. Attend la réponse de l'opérateur.
4	Reçoit la requête de changement de volume du Backup Server. Monte les bandes. Exécute <code>sp_volchanged</code> .		
5			Vérifie les bandes. Si les bandes sont correctes, commence la sauvegarde. Lorsque la bande est saturée, envoie une demande de changement de volume à l'opérateur.

Séquence	Opérateur utilisant <i>isql</i>	Adaptive Server	Backup Server
6	Reçoit la demande de changement de volume du Backup Server. Monte les bandes. Exécute <code>sp_volchanged</code> .		
7			Continue la sauvegarde. Lorsque la sauvegarde est terminée, envoie des messages à l'opérateur et à Adaptive Server.
8	Reçoit le message indiquant que la sauvegarde est terminée. Retire et étiquette les bandes.	Reçoit le message indiquant que la sauvegarde est terminée. Libère les verrous. Met fin à la commande <code>dump database</code> .	

Démarrage et arrêt du Backup Server

La plupart des systèmes UNIX se servent de l'utilitaire `startserver` pour démarrer le Backup Server sur la même machine qu'Adaptive Server. Sous Windows NT, vous pouvez démarrer le Backup Server à partir de Sybase Central. Pour des informations sur le démarrage de Backup Server, reportez-vous au Manuel de configuration pour votre plate-forme.

Exécutez la commande `shutdown` pour arrêter un Backup Server. Pour plus d'informations sur cette commande, reportez-vous au chapitre 4, "Diagnostic des problèmes système", et au document *Manuel de référence d'Adaptive Server*.

Configuration du serveur pour l'accès à distance

Le paramètre de configuration `remote access` est défini sur 1 lors de l'installation d'Adaptive Server. Cela permet à Adaptive Server d'exécuter des appels de procédure à distance vers le Backup Server.

Pour des raisons de sécurité, vous pouvez souhaiter désactiver l'accès à distance, excepté lorsque vous devez effectuer des sauvegardes et des chargements. Pour désactiver l'accès à distance, exécutez :

```
sp_configure "allow remote access", 0
```

Avant d'effectuer une sauvegarde ou un chargement, réactivez l'accès à distance via la commande :

```
sp_configure "allow remote access", 1
```

allow remote access est dynamique et ne requiert pas de redémarrer Adaptive Server. Seul un responsable de la sécurité du système (SSO) peut configurer allow remote access.

Choix d'un support de sauvegarde

Les bandes constituent un support de sauvegarde privilégié, dans la mesure où elles permettent de conserver hors ligne une bibliothèque de sauvegardes de bases de données et de journaux de transactions. La sauvegarde de bases de données volumineuses peut occuper plusieurs volumes de bande. Sur les systèmes UNIX, le Backup Server requiert des devices de bande sans rembobinage pour toutes les sauvegardes et tous les chargements.

Pour connaître les devices de sauvegarde acceptés, reportez-vous au Manuel de configuration pour votre plate-forme.

Protection des bandes de sauvegarde contre la réécriture

Le paramètre de configuration `tape retention in days` détermine le délai (en jours) pendant lequel les bandes de sauvegarde sont protégées contre la réécriture. La valeur par défaut de `tape retention in days` est 0, ce qui signifie que les bandes peuvent être immédiatement réécrites.

Exécutez `sp_configure` pour modifier la valeur de `tape retention in days`. La nouvelle valeur prend effet au redémarrage d'Adaptive Server :

```
sp_configure "tape retention in days", 14
```

Les commandes `dump database` et `dump transaction` disposent toutes deux d'une option `retaindays` qui prend le pas sur la valeur de `tape retention in days` pour la sauvegarde concernée.

Sauvegarde dans des fichiers ou sur disque

Les sauvegardes sur fichier ou sur disque sont généralement déconseillées. Si le disque ou l'ordinateur contenant ce fichier tombe en panne, il n'y a aucun moyen de récupérer les sauvegardes. Sur les systèmes UNIX et PC, la sauvegarde de l'intégralité de la base de données master doit tenir dans un seul volume. Sur ces systèmes, une sauvegarde sur fichier ou disque est la seule solution si la base de données master est trop volumineuse pour tenir sur un seul volume de bande, à moins que vous ne disposiez d'un second Adaptive Server pouvant émettre des requêtes `sp_volchanged`.

Les sauvegardes sur fichier ou disque peuvent être copiées sur bande pour un stockage hors ligne, mais ces bandes doivent être recopiées dans un fichier en ligne pour qu'Adaptive Server puisse les lire. Backup Server ne peut pas lire directement une sauvegarde effectuée sur un fichier disque, puis copiée sur une bande.

Création de noms de devices logiques pour les devices de sauvegarde locaux

Si vous sauvegardez vers ou chargez à partir de devices locaux (autrement dit, si vous n'effectuez pas de sauvegardes via le réseau sur un Backup Server distant), vous pouvez spécifier les devices de sauvegarde en indiquant leur emplacement physique ou leur nom logique. Dans ce dernier cas, vous pouvez créer des noms logiques de devices de sauvegarde dans la table système `sysdevices` de la base de données master.

Remarque Si vous procédez à une sauvegarde ou un chargement à partir d'un Backup Server distant, vous devez spécifier le chemin d'accès absolu du device de sauvegarde. Vous ne pouvez pas utiliser le nom logique du device.

La table `sysdevices` contient des informations sur chaque base de données et chaque device de sauvegarde, indiquant notamment son *nom_physique* (le nom réel du fichier ou du device dans le système d'exploitation) et son *nom_device* (ou nom logique, reconnu exclusivement dans Adaptive Server). Sur la plupart des plates-formes, Adaptive Server possède un ou deux alias pour les devices de bandes installés dans `sysdevices`. Les noms physiques de ces devices sont les noms usuels des disques sur la plate-forme, les noms logiques sont `tapedump1` et `tapedump2`.

Lorsque vous créez des scripts de sauvegarde et des procédures de seuil, utilisez de préférence les noms logiques et non les noms physiques des devices et, dans la mesure du possible, modifiez les scripts et les procédures qui font référence aux noms physiques des devices chaque fois que vous remplacez un device de sauvegarde. Si vous utilisez les noms logiques de device, vous pouvez simplement supprimer l'entrée `sysdevices` du device en panne et créer une nouvelle entrée associant le nom logique à un autre device physique.

Liste des noms courants de devices

Pour afficher la liste des devices de sauvegarde de votre système, exécutez :

```
select * from master..sysdevices
      where status = 16 or status = 24
```

Pour afficher la liste des noms physiques et logiques des devices de base de données et de sauvegarde, exécutez `sp_helpdevice` :

```
sp_helpdevice tapedump1
device_name physical_name
description
status cntrltype device_number low      high
-----
tapedump1 /dev/nrmt4
tape, 625 MB, dump device
      16          3          0          0    20000
```

Ajout d'un device de sauvegarde

Pour ajouter un device de sauvegarde, exécutez `sp_addumpdevice` :

```
sp_addumpdevice{ "tape" | "disk" } , nom_logique, nom_physique,  
taille_bande
```

nom_physique étant un chemin d'accès absolu ou relatif. Au cours des sauvegardes et des chargements, le serveur Backup Server résout les noms de chemin relatifs en effectuant une recherche dans le répertoire de travail courant d'Adaptive Server.

taille_bande correspond à la capacité de la bande en mégaoctets. Les autres plates-formes requièrent ce paramètre pour les devices bande, mais l'ignorent pour les devices disque. Le Backup Server utilise le paramètre *taille_bande* si la commande de sauvegarde ne spécifie pas la capacité de la bande.

taille_bande ne doit pas être inférieur à 1 Mo et doit être légèrement en dessous de la capacité estimée du device.

Redéfinition d'un nom de device logique

Pour attribuer un nom de device logique existant à un autre device physique, supprimez le device avec `sp_dropdevice` puis ajoutez-le avec `sp_addumpdevice`. Exemple :

```
sp_dropdevice tapedump2  
sp_addumpdevice "tape", tapedump2, "/dev/nrmt8", 625
```

Planification des sauvegardes des bases de données utilisateur

Une tâche majeure lors du développement d'un plan de sauvegarde est de déterminer la fréquence de sauvegarde des bases de données. Cette fréquence détermine le volume de travail que vous risquez de perdre en cas de panne du disque. Cette section donne quelques indications concernant la sauvegarde des bases de données utilisateur et des journaux de transactions.

Planification des sauvegardes systématiques

Sauvegardez chaque base de données utilisateur juste après sa création pour disposer d'un point de départ, puis selon une planification fixe. Une sauvegarde quotidienne du journal de transactions et une sauvegarde hebdomadaire de la base de données sont le minimum conseillé. Dans un service travaillant sur des bases de données volumineuses faisant l'objet de nombreuses transactions, prévoyez plutôt des sauvegardes quotidiennes des bases de données et des sauvegardes toutes les heures ou demi-heures du journal de transactions.

Les bases de données interdépendantes (avec transactions bases de données croisées, triggers ou intégrité référentielle) doivent être sauvegardées simultanément, à un moment où il ne se produit pas de modifications de données inter-bases. Si une panne se produit sur une de ces bases et que celle-ci doit être rechargée, toutes les bases doivent l'être à partir de leurs sauvegardes simultanées.

Avertissement ! Sauvegardez toujours les deux bases de données immédiatement après avoir ajouté, modifié ou supprimé une contrainte sur plusieurs bases de données ou supprimé une table contenant une contrainte sur plusieurs bases de données.

Autres sauvegardes de bases de données à planifier

Outre les sauvegardes de routine, vous devez sauvegarder une base de données utilisateur chaque fois que vous la mettez à niveau, créez un index, effectuez une opération non consignée ou exécutez la commande `dump transaction with no_log` ou la commande `dump transaction with truncate_only`.

Sauvegarde d'une base de données après mise à niveau

Dès que vous avez mis une base de données utilisateur au niveau de la version courante d'Adaptive Server, sauvegardez la base de données mise à niveau pour créer une sauvegarde compatible avec la version courante. Vous devez exécuter la commande `dump database` sur la base de données mise à niveau avant de pouvoir exécuter la commande `dump transaction`.

Sauvegarde d'une base de données après la création d'un index

Lorsque vous ajoutez un index à une table, `create index` est enregistré dans le journal de transactions. Lorsqu'il remplit d'informations les pages d'index, Adaptive Server ne consigne pas les modifications.

Si le device de base de données tombe en panne après la création d'un index, la reconstruction de l'index avec `load transaction` peut demander davantage de temps que sa création avec `create index`. Pour gagner du temps, sauvegardez chaque base de données dès que vous avez créé un index sur une de ses tables.

Sauvegarde d'une base de données après des opérations non consignées

Adaptive Server inscrit directement sur disque les données pour les commandes suivantes, sans insérer d'entrées (sauf pour `bcp`, les entrées minimales) dans le journal de transactions :

- `writetext` non consigné
- `select into` dans une table permanente
- `bulkcopy` rapide (`bcp`) dans une table sans triggers ni index

Vous ne pouvez restaurer aucune modification effectuée sur la base de données après émission d'une de ces commandes. Pour que ces commandes puissent être restaurées, émettez `dump database` immédiatement après leur exécution.

Sauvegarde d'une base de données après troncature du journal

`dump transaction with truncate_only` et `dump transaction with no_log` suppriment les transactions du journal sans effectuer de copie de sauvegarde. Pour assurer la possibilité de reprise, vous devez sauvegarder la base de données chaque fois que vous exécutez l'une de ces deux commandes à cause d'un manque d'espace disque. Vous ne pouvez pas copier le journal de transactions tant que vous n'avez pas effectué cette opération. Pour plus d'informations, reportez-vous à la section "Options spéciales de `dump transaction`", page 851.

Si l'option de base de données `trunc log on chkpt` est définie sur `true` et que le journal de transactions contient au moins 50 lignes, Adaptive Server tronque le journal au niveau d'un point de reprise automatique. Dans ce cas, vous devez sauvegarder l'intégralité de la base de données (mais non le journal de transactions) pour que la reprise soit possible.

Planification des sauvegardes de *master*

Sauvegardez la base de données master *régulièrement et fréquemment*.

Les sauvegardes de la base de données master sont utilisées dans la procédure de reprise si la panne a une incidence sur cette base master. Si vous ne disposez pas d'une sauvegarde courante de master, vous risquez de devoir reconstruire les tables système vitales à un moment où vous êtes pressé de remettre vos bases de données en état.

Sauvegarde de *master* après chaque changement

Bien que vous puissiez restreindre la création d'objets de base de données dans master, les procédures système telles que `sp_addlogin` et `sp_droplogin`, `sp_password` et `sp_modifylogin` autorisent les utilisateurs à modifier les tables système dans la base de données. Sauvegardez fréquemment la base de données master pour enregistrer ces changements.

Sauvegardez la base de données master après chaque commande ayant une incidence sur les disques, le stockage, les bases de données ou les segments. Sauvegardez toujours master après l'exécution des commandes ou procédures système suivantes :

- `disk init`, `sp_addumpdevice` ou `sp_dropdevice`
- Commandes de mise en miroir de disque
- Procédures système des segments `sp_addsegment`, `sp_dropsegment` ou `sp_extendsegment`
- `create procedure` ou `drop procedure`
- `sp_logdevice`
- `sp_configure`
- `create database` ou `alter database`

Sauvegarde des scripts et des tables système

Pour une protection encore plus efficace, sauvegardez les scripts contenant toutes vos commandes `disk init`, `create database` et `alter database` et effectuez une copie de sauvegarde de vos tables `sysdatabases`, `sysusages` et `sysdevices` chaque fois que vous exécutez une de ces commandes.

Vous ne pouvez pas utiliser `dataserver` pour récupérer automatiquement les modifications résultant de ces commandes. Si vous conservez vos scripts (fichiers contenant les instructions `Transact-SQL`), vous pouvez les exécuter pour recréer les modifications. Sinon, vous devez réémettre chaque commande sur la base de données master reconstruite.

Vous devez également conserver une copie de sauvegarde de `syslogins`. Lorsque vous restaurez master à partir d'une sauvegarde, comparez la copie de sauvegarde à la version courante de la table pour vérifier que les utilisateurs conservent les mêmes ID.

Pour plus d'informations sur les requêtes à exécuter sur les tables système, reportez-vous à "Sauvegarde de master et conservation de copies des tables système", page 26.

Troncature du journal de transactions de la base de données master

Dans la mesure où le journal de transactions de la base de données master se trouve sur les mêmes devices que les données, vous ne pouvez pas sauvegarder distinctement son journal de transactions. Vous ne pouvez pas non plus déplacer le journal de la base master. Vous devez toujours utiliser `dump database` pour sauvegarder la base de données master. Exécutez régulièrement `dump transaction` avec l'option `truncate_only` (par exemple, après chaque sauvegarde de la base de données) pour purger le journal de transactions de la base de données master.

Prévention des changements de volume et des reprises

Lorsque vous sauvegardez la base de données master, vérifiez que l'intégralité de la sauvegarde tient sur un seul volume, à moins que vous ne disposiez d'un second Adaptive Server capable de communiquer avec votre Backup Server. Vous devez lancer Adaptive Server en mode mono-utilisateur avant de charger la base de données master. Cela ne permet pas à une connexion utilisateur distincte de répondre au message de changement de volume du Backup Server pendant le chargement. Dans la mesure où master est généralement de petite taille, sa sauvegarde sur un seul volume de bande est rarement problématique.

Planification des sauvegardes de la base de données *model*

Conservez une sauvegarde courante de la base de données *model*. Effectuez une nouvelle sauvegarde chaque fois que vous modifiez la base de données *model*. Si *model* est endommagée et que vous ne disposez pas de sauvegarde, vous devez réintroduire toutes les modifications effectuées pour la restaurer.

Troncature du journal de transactions de la base de données *model*

Comme la base *master*, *model* stocke son journal de transactions sur les mêmes devices que les données. Vous devez toujours utiliser `dump database` pour sauvegarder la base de données *model* et `dump transaction` avec `truncate_only` pour purger le journal de transactions après chaque sauvegarde de base de données.

Planification des sauvegardes de la base de données *sybsystemprocs*

La base de données *sybsystemprocs* ne stocke que les procédures système. Restaurez cette base de données en exécutant le script `installmaster`, sauf si vous avez modifié la base de données.

Si vous changez les autorisations sur certaines procédures système ou créez vos propres procédures système dans *sybsystemprocs*, vous avez le choix entre deux procédures de reprise :

- Exécutez `installmaster` puis entrez à nouveau toutes vos modifications en recréant vos procédures ou en réexécutant les commandes `grant` et `revoke`.
- Sauvegardez *sybsystemprocs* à chaque modification.

Ces deux options de reprise sont décrites au chapitre 28, "Restauration des bases de données système".

Comme les autres bases de données système, sybssystemprocs stocke son journal de transactions sur les mêmes devices que les données. Vous devez toujours utiliser dump database pour sauvegarder sybssystemprocs. En principe, l'option trunc log on chkpt étant définie par défaut sur true (activée) dans sybssystemprocs, vous n'avez pas à tronquer le journal de transactions. Si vous modifiez cette option, veillez à tronquer le journal lorsque vous sauvegardez la base de données.

Si vous travaillez sur un système UNIX ou un PC et que vous ne disposez que d'un Adaptive Server capable de communiquer avec votre Backup Server, vérifiez que l'intégralité de la sauvegarde de sybssystemprocs tient sur un seul device de sauvegarde. Pour signaler les changements de volume, il faut exécuter sp_volchanged ; cette procédure est inutilisable sur un serveur pendant la reprise de sybssystemprocs.

Configuration d'Adaptive Server pour les chargements simultanés

Adaptive Server peut exécuter simultanément plusieurs commandes de chargement et de sauvegarde. Le chargement de chaque base de données active requiert un buffer de 16 ko. Par défaut, Adaptive Server est configuré pour six chargements simultanés. Pour augmenter ce nombre, l'administrateur système peut augmenter la valeur du paramètre number of large i/o buffers :

```
sp_configure "number of large i/o buffers", 12
```

Ce paramètre requiert de redémarrer Adaptive Server. Pour plus d'informations, reportez-vous à la section "number of large i/o buffers", page 104. Ces buffers ne sont pas utilisés pour les commandes dump ou load transaction.

Collecte des statistiques de sauvegarde

Exécutez dump database pour effectuer plusieurs sauvegardes d'une base de données utilisateur et dump transaction pour sauvegarder un journal de transactions. Restaurez la base de données avec load database et appliquez les sauvegardes successives du journal de transactions avec load transaction.

Conservez les statistiques sur la durée de chaque sauvegarde et sur l'espace qu'elle requiert. Plus vous serez proche des conditions de sauvegarde réelles, plus vos prévisions seront justes.

Après avoir développé et testé vos procédures de sauvegarde, validez-les et conservez-en une trace papier. Définissez un planning de sauvegardes raisonnable et respectez-le. Si vous développez, documentez et testez vos procédures de sauvegarde à l'avance, vous serez mieux préparé à remettre vos bases de données en ligne en cas de panne.

Sauvegarde et restauration de bases de données utilisateur

La seule garantie que vous ayez, face aux dommages occasionnés aux bases de données par une panne des devices de disques, réside dans des sauvegardes fréquentes et périodiques.

Le chapitre traite des sujets suivants :

Sujet	Page
Syntaxe de la commande dump and load	886
Spécification du device de sauvegarde et de base de données	889
Spécification de l'option de compression	894
Spécification d'un Backup Server distant	900
Spécification de la densité de bande, de la taille de bloc et de la capacité	902
Spécification du nom de volume	907
Identification d'une sauvegarde	908
Amélioration des performances de sauvegarde ou de chargement	911
Spécification d'autres devices de sauvegarde : clause stripe on	917
Options d'exploitation de bande	920
Remplacement de la destination par défaut des messages	924
Mise des bases de données en ligne avec with standby_access	927
Obtention d'informations sur les fichiers de sauvegarde	929
Copie du journal après une panne de device	932
Troncature d'un journal ne se trouvant pas sur un segment distinct	934
Troncature du journal en environnement de début de développement	934
Troncature d'un journal sans espace libre	935
Réponse aux requêtes de changement de volume	939
Restauration d'une base de données : instructions détaillées	944
Chargement des sauvegardes de bases de données effectuées avec des versions antérieures	952
Liaisons de cache et chargement de bases de données	956
Contraintes entre bases de données et chargement de bases de données	958

Syntaxe de la commande dump and load

Les commandes dump database, dump transaction, load database et load transaction ont une syntaxe comparable. En effet, les fonctions de sauvegarde et de restauration utilisent le nom d'une base de données et d'au moins un device de sauvegarde. Ces commandes peuvent également comprendre les options suivantes :

- `compress::niveau_compression::` pour compresser les fichiers de sauvegarde
- `at nom_serveur` pour spécifier le Backup Server distant
- `density`, `blocksize` et `capacity` pour spécifier les caractéristiques de stockage sur bande
- `dumpvolume` pour spécifier le nom de volume de l'étiquette de bande ANSI
- `file = nom_fichier` pour spécifier le nom du fichier cible de la sauvegarde ou source de la restauration
- `stripe on device_sauvegarde` pour spécifier des devices de sauvegarde supplémentaires
- `dismount`, `unload`, `init` et `retaindays` pour préciser l'exploitation de la bande
- `notify` pour spécifier si les messages Backup Server sont envoyés au client ayant lancé la sauvegarde ou le chargement ou bien vers la console de l'opérateur `operator_console`

Le tableau 27-1 donne la syntaxe des sauvegardes de base de données et du journal, ainsi que de la sauvegarde du journal après une panne de device. Il indique les informations fournies par chaque élément des instructions dump database ou dump transaction.

Tableau 27-1 : Syntaxe des commandes de sauvegarde courantes et du journal après une panne de device

Informations fournies	Tâche	
	Sauvegarde de la base de données ou du journal	Sauvegarde du journal après une panne de device
Commande	<code>dump {database transaction}</code>	<code>dump transaction</code>
Nom de la base de données	<code>nom_basededonnées</code>	<code>nom_basededonnées</code>
Compression	<code>to [compress:: [niveau_compression::]]</code>	<code>to [compress:: [niveau_compression::]]</code>

Informations fournies	Tâche	
	Sauvegarde de la base de données ou du journal	Sauvegarde du journal après une panne de device
Device de sauvegarde	<i>device_sauvegarde</i>	<i>device_sauvegarde</i>
Backup Server distant	[at <i>nom_serveur</i>]	[at <i>nom_serveur</i>]
Caractéristiques du device de sauvegarde (bande)	[density = <i>densité</i> , blocksize = <i>nombre_octets</i> , capacity = <i>nombre_kilo_octets</i>]	[density = <i>densité</i> , blocksize = <i>nombre_octets</i> , capacity = <i>nombre_kilo_octets</i>]
Nom de volume	[, dumpvolume = <i>nom_volume</i>]	[, dumpvolume = <i>nom_volume</i>]
Nom de fichier	[, file = <i>nom_fichier</i>]	[, file = <i>nom_fichier</i>]
Caractéristiques des devices supplémentaires (31 maximum ; un ensemble d'instructions par device)	[stripe on [compress::[<i>niveau_compression</i> ::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i> , blocksize = <i>nombre_octets</i> , capacity = <i>nombre_kilo_octets</i> , file = <i>nom_fichier</i> , dumpvolume = <i>nom_volume</i>]]...	[stripe on [compress::[<i>niveau_compression</i> ::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i> , capacity = <i>nombre_kilo_octets</i> , file = <i>nom_fichier</i> , dumpvolume = <i>nom_volume</i>]]...
Options s'appliquant à la sauvegarde dans son intégralité	[with { density = <i>densité</i> , blocksize = <i>nombre_octets</i> , capacity = <i>nombre_kilo_octets</i> , file = <i>nom_fichier</i> , [nodismount dismount], [nounload unload], [retaindays = <i>nombre_jours</i>], [noinit init], file = <i>nom_fichier</i> , dumpvolume = <i>nom_volume</i> standby_access	[with { density = <i>densité</i> , blocksize = <i>nombre_octets</i> , capacity = <i>nombre_kilo_octets</i> , file = <i>nom_fichier</i> , [nodismount dismount], [nounload unload], [retaindays = <i>nombre_jours</i>], [noinit init], file = <i>nom_fichier</i> , dumpvolume = <i>nom_volume</i> , standby_access
Ne pas tronquer le journal		no_truncate
Destination des messages	[, notify = {client operator_console}]]]	[, notify = {client operator_console}]]]

Le tableau 27-2 donne la syntaxe de chargement d'une base de données, d'application des transactions du journal et d'extraction des informations relatives aux fichiers et en-têtes de sauvegarde.

Tableau 27-2 : Syntaxe des commandes de chargement

Informations fournies	Tâche	
	Chargement de la base de données ou application des transactions récentes	Renvoi des informations relatives aux en-têtes ou aux fichiers, mais sans chargement de sauvegarde
Commande	load {database transaction}	load {database transaction}
Nom de la base de données	<i>nom_basededonnées</i>	<i>nom_basededonnées</i>
Compression	from [compress::]	from [compress::]
Device de sauvegarde	<i>device_sauvegarde</i>	<i>device_sauvegarde</i>
Backup Server distant	[at <i>nom_serveur</i>]	[at <i>nom_serveur</i>]
Caractéristiques du device de sauvegarde (bande)	[density = <i>densité</i> ,	[density = <i>densité</i> ,
Nom de volume	[, dumpvolume = <i>nom_volume</i>]	[, dumpvolume = <i>nom_volume</i>]
Nom de fichier	[, file = <i>nom_fichier</i>]	[, file = <i>nom_fichier</i>]
Caractéristiques des devices supplémentaires (31 maximum ; un ensemble d'instructions par device)	[stripe on [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i> , file = <i>nom_fichier</i> , dumpvolume = <i>nom_volume</i>]]...	[stripe on [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i> , file = <i>nom_fichier</i> , dumpvolume = <i>nom_volume</i>]]...
Exploitation de la bande	[with{ [density = <i>densité</i> , dumpvolume = <i>nom_volume</i> , file = <i>nom_fichier</i> , [nodismount dismount], [nounload unload]	[with{ [density = <i>densité</i> , dumpvolume = <i>nom_volume</i> , file = <i>nom_fichier</i> , [nodismount dismount], [nounload unload]
Obtenir des informations d'en-tête		[, headeronly]
Liste des fichiers de sauvegarde		[, listonly [= full]]
Destination des messages	[, notify = {client operator_console}]]]	[, notify = {client operator_console}]]]
Non chargement de transactions ouvertes	<i>standby_access</i>	

Le tableau 27-3 donne la syntaxe pour la troncature d'un journal :

- qui ne réside pas sur un segment distinct
- Sans copie de sauvegarde
- Lorsque l'espace libre est insuffisant pour exécuter une commande dump transaction ou dump transaction with truncate_only avec succès

Tableau 27-3 : Options particulières de transactions de sauvegarde

Informations fournies	Tâche		
	Troncature d'un journal résidant sur le même segment que les données	Troncature du journal sans réaliser de sauvegarde	Troncature du journal lorsque l'espace libre est insuffisant
Commande	dump transaction	dump transaction	dump transaction
Nom de la base de données	nom_basededonnées	nom_basededonnées	nom_basededonnées
Ne pas copier le journal	with truncate_only	with truncate_only	with no_log

Vous trouverez dans la suite de ce chapitre de plus amples détails sur les informations fournies dans les commandes de chargement et de sauvegarde, ainsi que dans les messages de changement de volume. Les sauvegardes et chargements sont décrits en premier, suivis des sauvegardes du journal après une panne de device et de la syntaxe particulière pour tronquer le journal sans effectuer de sauvegarde.

Pour plus d'informations sur les autorisations requises pour l'exécution des commandes de sauvegarde et de chargement, reportez-vous à la section "Désignation du responsable des sauvegardes", page 867.

Spécification du device de sauvegarde et de base de données

Toutes les commandes de sauvegarde et de chargement doivent comprendre au moins le nom de la base de données qui est sauvegardée ou chargée. Les commandes de sauvegarde ou de chargement des données (exception faite de la troncature d'un journal de transactions) doivent également inclure le nom du device de sauvegarde.

Le tableau 27-4 donne la syntaxe pour sauvegarder et charger une base de données ou un journal.

Tableau 27-4 : Indication du nom de la base de données et du device de sauvegarde

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
Nom de la base de données	dump {database tran} <i>nom_basededonnées</i>	load {database tran} <i>nom_basededonnées</i>
Device de sauvegarde	<i>device_sauvegarde</i>	<i>device_sauvegarde</i>
	to [compress::[niveau_compression::]]	from [compress::]
Device de sauvegarde	<pre> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::[niveau_compression::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>, [nodismount dismount], [nounload unload], retaindays = <i>nombre_jours</i>, [noinit init], [notify = {client operator_console}}] standby_access}} </pre>	<pre> [at <i>nom_serveur</i>] [density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>, [nodismount dismount], [nounload unload], [notify = {client operator_console}}]] </pre>

Règles de spécification des noms de bases de données

Le nom d'une base de données peut être indiqué sous la forme d'un littéral, d'une variable locale ou d'un paramètre de procédure stockée.

Si vous chargez une base de données à partir d'une sauvegarde :

- Cette base doit exister. Vous pouvez créer une base de données à l'aide de l'option `for load` de `create database` ou effectuer le chargement dans une base existante. Dans tous les cas, la base de données chargée écrase toutes les informations contenues dans la base existante.
- Il n'est pas nécessaire d'attribuer à la base de données le même nom de la base sauvegardée. Par exemple, vous pouvez sauvegarder la base `pubs2`, créer une base de données sous le nom `pubs2_archive` et charger la sauvegarde dans cette nouvelle base.

Avertissement ! Ne modifiez jamais le nom d'une base de données contenant des clés primaires qui la référencent dans d'autres bases de données. Si vous devez charger la sauvegarde d'une telle base de données en modifiant son nom, commencez par supprimer les références qui y sont faites dans d'autres bases.

Règles de spécification des devices de sauvegarde

Lors de la spécification d'un device de sauvegarde :

- vous pouvez spécifier le device de sauvegarde comme littéral, variable locale ou paramètre d'une procédure stockée,
- vous ne pouvez pas utiliser le "device null" pour sauvegarder ou charger une base de données (sous UNIX, `/dev/null` ; sous OpenVMS, tout device commençant par NL - ce type de device n'existe pas sur PC).
- Lorsque vous effectuez une sauvegarde vers ou un chargement depuis un device local, vous pouvez spécifier son nom sous l'une des formes suivantes :
 - nom de chemin d'accès absolu,
 - nom de chemin d'accès relatif,
 - nom d'un device logique provenant de la table système `sysdevices`.

Le Backup Server résout les chemins d'accès relatifs à l'aide du répertoire de travail courant d'Adaptive Server.

- Lorsque vous effectuez une sauvegarde ou un chargement via le réseau :
 - Vous devez indiquer le chemin d'accès absolu du device de sauvegarde. Vous ne pouvez pas utiliser un chemin d'accès relatif ni un nom de device logique figurant dans la table système sysdevices.
 - Vous devez entrer un chemin d'accès reconnu par la machine sur laquelle fonctionne Backup Server.
 - Si le nom comporte des caractères autres que des lettres, des chiffres et des tirets de soulignement (_), vous devez les placer entre guillemets.
- Si vous sauvegardez un journal de transaction à l'aide de `with standby_access`, vous devez charger la sauvegarde avec `with standby_access`.

Exemples

Dans les exemples suivants, un seul device de type bande est utilisé pour les sauvegardes et les chargements. (Il n'est pas obligatoire d'utiliser le même device pour les sauvegardes et les chargements).

Sous UNIX :

```
dump database pubs2 to "/dev/nrmt4"  
load database pubs2 from "/dev/nrmt4"
```

Sous Windows NT :

```
dump database pubs2 to "\\.\tape0"  
load database pubs2 from "\\.\tape0"
```

Vous pouvez également effectuer la sauvegarde dans un fichier du système d'exploitation. L'exemple suivant s'applique à Windows NT :

```
dump database pubs2 to "d:\backups\backup1.dat"  
load database pubs2 from "d:\backupbackup1.dat"
```

Détermination du device de sauvegarde (bande) par Backup Server

Lorsque vous exécutez une commande `dump database` ou `dump transaction`, Backup Server vérifie que le type de device de sauvegarde spécifié est connu d'Adaptive Server (fourni et supporté en interne). S'il est inconnu, Backup Server recherche la configuration du device dans le fichier de configuration des bandes (son emplacement par défaut est `$$SYBASE/backup_tape.cfg`).

S'il la trouve, la commande `dump` est exécutée.

S'il ne la trouve pas dans le fichier de configuration du device, la commande `dump` échoue et le message d'erreur ci-dessous apparaît :

```
Le device ne figure pas dans le fichier de
configuration. Vous devez spécifier INIT pour
configurer le device.
```

Pour configurer le device, exécutez la commande `dump database` ou `dump transaction` avec le paramètre `init`. A l'aide des appels du système d'exploitation, Backup Server tente de définir les caractéristiques du device ; s'il y parvient, il les stocke dans le fichier de configuration de la bande.

S'il échoue, il utilise les caractéristiques par défaut ne permettant d'effectuer qu'une sauvegarde par bande. Le device ne peut pas être utilisé si, lors de la configuration, il n'a pas été possible d'écrire au moins un fichier de sauvegarde.

La configuration de bande par Backup Server ne fonctionne que sur les plates-formes UNIX.

Fichier de configuration du device de sauvegarde (bande)

Format	<p>Le fichier de configuration de device de type bande contient des informations, qui ne sont utilisées que par la commande <code>dump</code>.</p> <p>Le format de ce fichier est une entrée de device de type bande par ligne. Les champs sont séparés par des blancs ou des tabulations.</p>
Création	<p>Ce fichier n'est créé que lorsque Backup Server est prêt à écrire sur celui-ci (<code>dump database</code> ou <code>dump transaction</code> à l'aide d'<code>init</code>). Lorsque Backup Server tente d'écrire pour la première fois dans ce fichier, le message d'avertissement suivant apparaît :</p> <pre>Attention, impossible d'ouvrir le fichier de configuration du device pour lecture. Erreur système. Il n'existe pas de fichier ni de répertoire de ce type.</pre>

	Ignorez ce message. Après avoir émis ce message, Backup Server crée le fichier et y écrit les informations sur la configuration.
Modification manuelle	Le fichier de configuration étant mis à jour par Backup Server, le seul cas dans lequel l'utilisateur doit intervenir sur le fichier est celui de l'affichage du message d'erreur suivant : <pre> Ce type de device n'est pas le même que dans le fichier de configuration. Veuillez supprimer l'entrée de ce device dans le fichier de configuration, puis reconfigurer le device en lançant un DUMP avec le qualificatif INIT. </pre> Il signifie que la configuration matérielle de bande d'un device a changé. Supprimez la ligne comportant l'entrée pour ce device et exécutez la commande dump en suivant les instructions.
Emplacement par défaut	Le nom du chemin d'accès par défaut du fichier de configuration est <i>\$SYBASE/backup_tape.cfg</i> . Vous pouvez modifier l'emplacement par défaut avec les utilitaires d'installation Sybase. Pour plus d'informations, reportez-vous au Guide d'installation pour votre plate-forme.

Spécification de l'option de compression

La commande dump comprend l'option compress qui vous permet de compresser les bases de données et les journaux de transaction à l'aide Backup Server.

Le tableau 27-5 fournit la syntaxe des commandes dump database ... compress et dump transaction ... compress :

Tableau 27-5 : Indication du nom de la base de données et du device de sauvegarde

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	dump {database tran} <i>nom_basededonnées</i> to	load {database tran} <i>nom_basededonnées</i> from
Option compress	[compress::[niveau_compression::]]	[compress::]

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	<pre>device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] [stripe on</pre>	<pre>device_sauvegarde} [at nom_serveur] [density = densité, dumpvolume = nom_volume file = nom_fichier] [stripe on</pre>
Option compress	[compress::[niveau_compression::]]	[compress::]
	<pre>device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], retaindays = nombre_jours, [noinit init], [notify = {client operator_console}] standby_access}]</pre>	<pre>device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], [notify = {client operator_console}}]}</pre>

Syntaxe

La syntaxe partielle des commandes dump database ... compress et dump transaction ... compress est la suivante :

```
dump database nom_basededonnées
to compress::[niveau_compression::]device_sauvegarde
...[stripe on
compress::[niveau_compression::]device_sauvegarde] ...
dump transaction nom_basededonnées
to compress::[niveau_compression::]device_sauvegarde
...[stripe on
compress::[niveau_compression::]device_sauvegarde] ...
```

où *nom_basedonnées* est la base dans laquelle vous chargez et *compress::niveau_compression* un nombre compris en 0 et 9, 0 correspondant à aucune compression et 9 à une compression maximum. Si vous n'indiquez pas de *niveau_compression*, la valeur par défaut est 6. *device_sauvegarde* correspond au chemin du fichier d'archivage de la base de données ou du journal de transactions que vous compressez. Si vous ne précisez pas de chemin pour votre fichier de sauvegarde, Adaptive Server le crée dans son répertoire de démarrage.

Utilisez la clause *stripe on* pour utiliser plusieurs devices de sauvegarde pour une seule sauvegarde. Reportez-vous à la section "Spécification d'autres devices de sauvegarde : clause *stripe on*", page 917 pour obtenir un complément d'informations sur la clause *stripe on*.

Remarque L'option *compress* ne fonctionnant qu'avec les archivages locaux, vous ne pouvez pas utiliser l'option *servername*.

Exemple

```
dump database pubs2 to
    "compress::4::/opt/bin/Sybase/dumps/dmp090100.dmp"
```

```
L'identifiant de la session Backup Server est : 9. Utilisez cette valeur
lors de l'exécution de la procédure système 'sp_volchanged' après la
réalisation d'une requête de changement de volume émise par le Backup Server.
Backup Server: 4.132.1.1: Tentative d'ouverture du device de flux d'octets :
'compress::4::/opt/bin/Sybase/dumps/dmp090100.dmp::00'
Backup Server: 6.28.1.1: Nom dumpfile 'pubs2002580BD27
' Numéro de section 1 monté sur
'compress::4::/opt/bin/Sybase/dumps/dmp090100.dmp::00'
Backup Server: 4.58.1.1: Database pubs2: 394 kilobytes DUMPed.
Backup Server: 4.58.1.1: Database pubs2: 614 kilobytes DUMPed.
Backup Server: 3.43.1.1: Nombre de phases de vidage 1 achevé.
Backup Server: 3.43.1.1: Nombre de phases de vidage 2 achevé.
Backup Server: 3.43.1.1: Nombre de phases de vidage 3 achevé.
Backup Server: 4.58.1.1: Database pubs2: 622 kilobytes DUMPed.
Backup Server: 3.42.1.1: DUMP achevé (database pubs2).
```

Le *niveau_compression* doit être compris entre 0 et 9. L'option *compress* ne reconnaît pas les nombres situés hors de ces limites, considère qu'ils font partie du nom du fichier et compresse les fichiers avec le niveau de compression par défaut. Par exemple, la syntaxe suivante crée un fichier dénommé *99::pubs2.cmp*, qui est compressé à l'aide du niveau de compression par défaut de 1 :

```
dump database pubs2 to "compress::99::pubs2.cmp"
```


En général, plus le nombre de compression est élevé, plus les archivages sont petits. Le résultat de la compression dépend néanmoins du contenu des fichiers.

Le tableau 27-6 fournit les niveaux de compression de la base de données pubs2. Ces nombres ne sont donnés qu'à titre indicatif – en fonction du niveau du système d'exploitation et de la configuration, ceux de votre site seront peut-être différents.

Tableau 27-6 : Niveaux de compression et taille des fichiers compressés de pub2

Niveaux de compression	Taille des fichiers compressés
Aucune compression/Niveau 0	630 ko
Niveau 1	128 ko
Niveau 2	124 ko
Niveau 3	121 ko
Niveau 4	116 ko
Niveau 5	113 ko
Niveau 6 (niveau de compression par défaut)	112 ko
Niveau 7	111 ko
Niveau 8	110 ko
Niveau 9	109 ko

Plus le niveau de compression est élevé, plus le processus utilise l'unité centrale.

Par exemple, avant d'archiver vos fichiers à un niveau de compression 9, évaluez le rapport entre l'effort nécessaire au traitement et la taille de l'archive. Le niveau de compression par défaut (6) fait un usage optimal de l'unité centrale et produit une archive 60 à 80 % plus petite qu'une archive normale non compressée. Sybase conseille de commencer par recourir au niveau de compression par défaut, puis de l'augmenter ou de le diminuer en fonction des performances souhaitées.

La syntaxe complète de dump database et de dump transaction figure dans le *Manuel de référence*.

Fichiers de sauvegarde et sauvegardes compressées de Backup Server

Lorsque vous exécutez `dump database` ou `dump transaction` à l'aide d'un fichier d'archive existant, Backup Server contrôle automatiquement l'en-tête de ce dernier. S'il est illisible, Backup Server suppose qu'il s'agit d'un fichier valable qui n'est pas une archive et vous invite à modifier le fichier de sauvegarde avec le message suivant :

```
Backup Server: 6.52.1.1: OPERATOR: Volume to be overwritten on
'/opt/SYBASE/DUMPS/model.dmp' has unrecognized label data (le volume à écraser
sur '/opt/SYBASE/DUMPS/model.dmp' possède des données d'étiquettes inconnues).
Backup Server: 6.78.1.1: EXECUTE sp_volchanged
    @session_id = 5,
    @devname = '/opt/SYBASE/DUMPS/model.dmp',
    @action = { 'PROCEED' | 'RETRY' |
'ABORT' },
    @vname = <nom_nouveau_volume>
```

C'est pourquoi, si vous exécutez `dump database` ou `dump transaction` vers un fichier sans l'option `compress::` dans un fichier de sauvegarde compressé, Backup Server ne reconnaît pas les informations de l'en-tête de l'archive en raison de la compression.

Exemple

```
dump database model to 'compress::model.cmp'
go
dump database model to 'model.cmp'
go
```

La seconde commande `dump database` signale une erreur et vous propose `sp_volchanged`.

Pour éviter cela, incluez,

- soit l'option `with init` dans vos commandes `dump database` et `dump transaction` suivantes :

```
dump database model to 'compress::model.cmp'
go
dump database model to 'model.cmp'
    with init
go
```

- soit l'option `compress::` dans les commandes `dump database` et `dump transaction` suivantes :

```
dump database model to 'compress::model.cmp'
go
dump database model to 'compress::model.cmp'
go
```

Le recours à l'option `compress::` dans des archives de sauvegardes non compressées, comme dans l'exemple suivant, ne génère pas d'erreurs :

```
dump database model to 'model.cmp'  
go  
dump database model to 'compress::model.cmp'  
go
```

Chargement de bases de données et de journaux de transactions sauvegardés avec l'option `compress`

Si vous exécutez `dump ... compress` pour sauvegarder une base de données ou un journal de transactions, vous devez charger la sauvegarde ainsi obtenue à l'aide de l'option `load ... compress`.

La syntaxe partielle de `load database .. compress` et de `load transaction .. compress` est la suivante :

```
load database nom_basededonnées  
from compress::device_sauvegarde  
...[stripe on compress::device_sauvegarde]...  
  
load transaction nom_basededonnées  
from compress::device_sauvegarde  
...[stripe on compress::device_sauvegarde]...
```

Dans cette syntaxe, le *nom_basededonnées* correspond à la base de données archivée et `compress::` invoque la décompression de la base de données ou du journal de transactions archivé. *archive_name* est le chemin complet de la base de données ou du journal de transactions archivés en cours de chargement. Si vous ne précisez pas le chemin complet lors de la création du fichier de sauvegarde, Adaptive Server le crée dans son répertoire de démarrage.

Exécutez la clause `stripe on` si vous compressez la base de données ou le journal de transactions avec plusieurs sauvegardes. Reportez-vous à la section "Spécification d'autres devices de sauvegarde : clause `stripe on`", page 917 pour obtenir un complément d'informations sur la clause `stripe on`.

Remarque N'utilisez pas la variable *niveau_compression* pour la commande `load`.

Exemple
`load database pubs2 from`

```
"compress::/opt/bin/Sybase/dumps/dmp090100.dmp"
```

L'identifiant de la session Backup Server est : 19. Utilisez cette valeur lors de l'exécution de la procédure système 'sp_volchanged' après la réalisation d'une requête de changement de volume émise par le Backup Server.

```
Backup Server: 4.132.1.1: Tentative d'ouverture du device de flux d'octets :
'compress::/opt/bin/Sybase/dumps/dmp090100.dmp::00'
Backup Server: 6.28.1.1: Dumpfile name 'pubs2002620A951 ' section number 1
mounted on byte stream 'compress::/opt/bin/Sybase/dumps/dmp090100.dmp::00'
Backup Server: 4.58.1.1: Database pubs2: 1382 kilobytes LOADED.
Backup Server: 4.58.1.1: Database pubs2: 3078 kilobytes LOADED.
Backup Server: 4.58.1.1: Database pubs2: 3086 kilobytes LOADED.
Backup Server: 3.42.1.1: LOAD is complete (database pubs2).
Exécutez la commande ONLINE DATABASE pour mettre cette base de données en
ligne. SQL Server ne le fera pas automatiquement.
```

La syntaxe complète de load database et de load transaction, figure dans le *Manuel de référence*.

Spécification d'un Backup Server distant

Utilisez la clause *at nom_serveur* pour envoyer, via le réseau, des requêtes de sauvegarde et de restauration à un Backup Server fonctionnant sur une autre machine.

Le tableau 27-7 donne la syntaxe des commandes de sauvegarde et des restaurations depuis un Backup Server distant.

Tableau 27-7 : Sauvegarde vers ou restauration depuis un Backup Server distant

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	dump {database tran} <i>nom_basededonnées</i> to [compress::[niveau_compression::]] <i>device_sauvegarde</i>	load {database tran} <i>nom_basededonnées</i> from [compress::] <i>device_sauvegarde</i>
Backup Server distant	[at <i>nom_serveur</i>]	[at <i>nom_serveur</i>]

Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
<pre>[density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::[<i>niveau_compression</i>::]]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>, [nodismount dismount], [nounload unload], retaindays = <i>nombre_jours</i>, [noinit init], [notify = {client operator_console}] standby_access}]</pre>	<pre>[density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>, [nodismount dismount], [nounload unload], [notify = {client operator_console}] standby_access}]</pre>

Cette méthode est parfaitement adaptée aux installations qui utilisent une seule machine dotée de plusieurs devices de sauvegarde pour l'ensemble des sauvegardes et des chargements. En effet, un opérateur peut être chargé des changements de bande sur cette machine.

Les exemples suivants correspondent à une sauvegarde vers et un chargement depuis le Backup Server distant REMOTE_BKP_SERVER :

```
dump database pubs2 to "/dev/nrmt0" at REMOTE_BKP_SERVER
load database pubs2 from "/dev/nrmt0" at REMOTE_BKP_SERVER
```

La variable *nom_serveur* doit figurer dans le fichier d'interfaces de la machine sur laquelle s'exécute Adaptive Server, mais il n'est pas nécessaire qu'elle apparaisse dans la table syssservers. Le *server_name* doit être le même dans les fichiers d'interface local et distant.

Spécification de la densité de bande, de la taille de bloc et de la capacité

Dans la plupart des cas, Backup Server utilise une taille de bloc et une densité de bande par défaut les mieux adaptées à votre système d'exploitation ; *il est conseillé de les utiliser.*

Vous pouvez préciser une densité, une taille de bloc et une capacité pour chaque device de sauvegarde. Vous pouvez également préciser les options de density, blocksize et capacity avec la clause with pour l'ensemble des devices de sauvegarde. Les caractéristiques fournies pour un device de sauvegarde particulier sont prioritaires sur celles définies dans la clause with.

Le tableau 27-8 donne la syntaxe pour la spécification de la densité de bande, de la taille de bloc et de la capacité.

Tableau 27-8 : Spécification de la densité de bande, de la taille de bloc et de la capacité

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	<pre>dump {database tran} nom_basededonnées to [compress:: [niveau_compression::]] device_sauvegarde [at nom_serveur]</pre>	<pre>load {database tran} nom_basededonnées from [compress::] device_sauvegarde [at nom_serveur]</pre>
Caractéristiques d'un unique device de sauvegarde	<pre>[density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] [stripe on [compress:: [niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] ...]</pre>	<pre>[density = densité, dumpvolume = nom_volume, file = nom_fichier] [stripe on [compress::] device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] ...]</pre>
Caractéristiques de l'ensemble des devices de sauvegarde	<pre>[with{ density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets,</pre>	<pre>[with{ density = densité,</pre>

Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
<pre>dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], retaindays = nombre_jours, [noinit init], [notify = {client operator_console}] standby_access}]</pre>	<pre>dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], [notify = {client operator_console}] standby_access}]</pre>

Les sections suivantes décrivent plus en détail les options de density, blocksize et capacity.

Remplacement de la densité par défaut

Les commandes de sauvegarde et de chargement utilisent la densité de bande par défaut de votre système d'exploitation. Dans la plupart des cas, cette valeur est la mieux adaptée aux sauvegardes sur bande.

Lorsque vous effectuez des sauvegardes sur bande sur un système OpenVMS, vous pouvez remplacer la densité par défaut à l'aide de l'option `density = densité`. Les densités correctes sont 800, 1600, 6250, 6666, 10 000 et 38 000. Tous les lecteurs de bande n'acceptant pas ces densités, indiquez une valeur adaptée à votre lecteur.

Cette option est sans incidence sur les chargements de bande sous OpenVMS, ainsi que sur les sauvegardes et chargements sur plate-forme PC ou UNIX.

Remarque Spécifiez la densité de bande uniquement lorsque vous utilisez l'option d'exploitation de bande `init`. Pour plus d'informations sur cette option, reportez-vous à la section "Réinitialisation d'un volume avant une sauvegarde", page 923.

Remplacement de la taille de bloc par défaut

Le paramètre de taille de bloc indique le nombre d'octets par opération d'E/S d'un device de sauvegarde. Par défaut, les commandes de sauvegarde et de chargement sélectionnent la taille de bloc la mieux adaptée à votre système d'exploitation. Utilisez autant que possible cette valeur par défaut.

Si vous devez utiliser une autre taille de bloc que celle par défaut pour un device de sauvegarde particulier, utilisez l'option `blocksize = nombre_octets`. La taille de bloc doit correspondre au minimum à une page de base de données (2048 octets) et doit être un multiple exact de cette unité.

Sur système OpenVMS, vous ne pouvez définir la taille de bloc que pour les sauvegardes. Utilisez une taille de bloc inférieure ou égale à 55 296.

Sur système UNIX, vous pouvez fournir une taille de bloc aussi bien pour les sauvegardes que pour les chargements. Lors du chargement d'une sauvegarde, vous devez indiquer la même taille de bloc que pour la sauvegarde.

Spécification d'une valeur de taille de bloc supérieure

Si vous sauvegardez sur bande à l'aide des commandes `dump database` ou `dump transaction` et indiquez une taille de bloc supérieure à la taille de bloc maximum du device déterminée par Backup Server, la sauvegarde ou le chargement risque d'échouer sur certains lecteurs. Un message d'erreur système s'affiche, tel que l'exemple ci-dessous correspondant à un lecteur de bande de 8 mm sur HP :

```
Backup Server: 4.141.2.22: [2] The 'write' call
failed for device 'xxx' with error number 22 (Invalid
argument) (l'appel en écriture a échoué pour le
device "xxx" - erreur 22 (argument erroné)).
Reportez-vous à la documentation de votre système
pour obtenir des informations complémentaires.
```

Vous ne devez pas indiquer de taille de bloc supérieure à celle définie dans le fichier de configuration du device de bande (`$SYBASE/backup_tape.cfg`). La taille de bloc d'un device est le cinquième champ de la ligne de ce fichier.

Cette erreur ne se produit que sur les devices exécutant la configuration automatique de la bande, c'est-à-dire les modèles qui ne possèdent pas de code spécifié dans le Backup Server.

Spécification de la capacité de la bande pour les commandes dump

Par défaut, les systèmes OpenVMS poursuivent l'écriture jusqu'à ce qu'ils atteignent le marqueur physique de fin de bande puis signalent qu'il faut changer de volume. En revanche, avec les plates-formes UNIX qui ne sont pas en mesure de détecter de manière fiable le marqueur de fin de bande, vous devez indiquer le nombre de kilo-octets pouvant être sauvegardés sur une bande.

Si vous spécifiez le chemin d'accès physique du device de sauvegarde, vous devez fournir le paramètre `capacity = nombre_kilo_octets` dans la commande de sauvegarde. Si vous spécifiez le nom d'un device logique, Backup Server utilise le paramètre `size` stocké dans la table `sysdevices`, sauf si vous indiquez le paramètre `capacity = nombre_kilo_octets`.

La capacité spécifiée doit correspondre au minimum à cinq pages de base de données (chaque page nécessite 2048 octets). Il est conseillé d'indiquer une capacité légèrement inférieure à la capacité nominale de votre device.

En général, on considère que la capacité est égale à 70 pour-cent de la capacité maximum du device indiquée par le fabricant et on réserve les 30 pour-cent restants pour les overheads (espaces inter-enregistrements, marqueurs de bande, etc.). Cette règle fonctionne dans la plupart des cas, sauf en cas de différences d'overhead entre les distributeurs et les devices.

Fonction bande non rembobinage de Backup Server

La fonction bande non rembobinage positionne automatiquement la bande à la fin des données de sauvegarde correctes, ce qui permet de gagner du temps en cas de sauvegardes multiples.

Changement d'étiquette de sauvegarde

Backup Server écrit une étiquette de fin de fichier, EOF3, à la fin de chaque sauvegarde.

Remarque Vous ne pouvez pas charger de bande avec cette étiquette dans une version d'Adaptive Server antérieure à 12.0.

Opérations relatives à la bande

Lors d'une nouvelle sauvegarde, Backup Server recherche l'étiquette EOF3.

S'il la trouve, les informations pertinentes sont sauvegardées et la bande est avancée au début du fichier suivant. Cela constitue le nouveau point de rattachement.

S'il ne la trouve pas ou s'il rencontre un autre problème, Backup Serveur rembobine la bande et cherche vers l'avant. D'éventuelles erreurs à ce stade ne mettent pas fin à la sauvegarde, mais renvoient Backup Server à son comportement par défaut (rembobiner et rechercher). Si l'erreur persiste pendant le rembobinage et la recherche, la commande dump est interrompue.

Compatibilité des versions de sauvegarde

Backup Server n'active la logique de non-rembobinage que si la version de l'étiquette de la bande est supérieure ou égale à 5. Il faut donc une commande dump avec la clause with init pour activer cette logique. Si une commande dump sans init est émise sur un volume dont la version d'étiquette est inférieure à 5, vous êtes invité à changer de volume et la sauvegarde commence sur le volume suivant. La version d'étiquette d'une sauvegarde sur plusieurs volumes ne change pas au milieu de l'ensemble de volumes.

Le tableau 27-9 indique les versions d'étiquette correspondant au nouveau comportement.

Tableau 27-9 : Compatibilité des versions d'étiquettes

Version d'étiquette	Activée
'3'	Non
'4'	Non
'5'	Oui
'6'	Oui

Spécification du nom de volume

Utilisez l'option `with dumpvolume = nom_volume` pour spécifier le nom de volume. Les commandes `dump database` et `dump transaction` écrivent le nom de volume dans l'étiquette de bande SQL. Les commandes `load database` et `load transaction` vérifient cette étiquette. Si le volume chargé est incorrect, Backup Server génère un message d'erreur.

Vous pouvez spécifier un nom de volume pour chaque device de sauvegarde. Vous pouvez également indiquer un nom de volume pour l'ensemble des devices dans la clause `with`. Les noms de volumes définis pour des devices particuliers sont prioritaires sur ceux spécifiés dans la clause `with`.

Le tableau 27-10 donne la syntaxe pour la spécification du nom de volume.

Tableau 27-10 : Spécification du nom de volume

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	<code>dump {database tran} <i>nom_basededonnées</i> to [compress::[niveau_compression::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>,</code>	<code>load {database tran} <i>nom_basededonnées</i> from [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>,</code>
Nom de volume pour un device particulier	<code>dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::[niveau_compression::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>,</code>	<code>dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with { density = <i>densité</i>,</code>

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
Nom de volume pour l'ensemble des devices	<code>dumpvolume = nom_volume,</code>	<code>dumpvolume = nom_volume,</code>
	<code>file = nom_fichier,</code> <code>[nodismount dismount],</code> <code>[nounload unload],</code> <code>retaindays = nombre_jours,</code> <code>[noinit init],</code> <code>[notify = {client operator_console}]</code> <code>standby_access}]</code>	<code>file = nom_fichier,</code> <code>[nodismount dismount],</code> <code>[nounload unload],</code> <code>[notify = {client operator_console}]</code> <code>standby_access}]</code>

Chargement depuis un volume multifichier

Lorsque vous chargez la sauvegarde d'une base de données depuis un volume contenant plusieurs fichiers de sauvegarde, spécifiez le nom du fichier de sauvegarde. Si vous omettez le nom du fichier et ne spécifiez que le nom de la base de données, Backup Server charge le premier fichier dans la base de données spécifiée. Par exemple, si vous entrez la commande suivante, le premier fichier de sauvegarde est chargé depuis la bande dans *pubs2*, que ce fichier contienne des données de *pubs2* ou non :

```
load database pubs2 from "/dev/rdisk/clt3d0s6"
```

Pour éviter ce problème, spécifiez un nom de fichier de sauvegarde unique chaque fois que vous sauvegardez ou chargez des données. Pour obtenir des informations sur les fichiers de sauvegarde contenus sur une bande, utilisez l'option `listonly = full` de la commande `load database`.

Identification d'une sauvegarde

Lorsque vous sauvegardez une base de données ou un journal de transactions, Backup Server crée un nom de fichier par défaut en concaténant :

- les 7 derniers caractères du nom de la base de données,
- les deux derniers chiffres de l'année,

- trois chiffres pour le jour de l'année (de 1 à 366),
- le nombre de secondes écoulées depuis minuit, au format hexadécimal.

Vous pouvez spécifier un autre nom de fichier avec l'option `file = nom_fichier`. Ce nom ne doit pas dépasser 17 caractères et doit respecter les conventions de dénomination des fichiers de votre système d'exploitation.

Vous pouvez spécifier un nom de fichier pour chaque device de sauvegarde. Vous pouvez également spécifier un nom de fichier pour l'ensemble des devices à l'aide de la clause `with`. Les noms de fichiers spécifiés pour des devices particuliers sont prioritaires sur ceux définis dans la clause `with`.

Le tableau 27-11 donne la syntaxe pour la spécification du nom de fichier d'une sauvegarde.

Tableau 27-11 : Spécification du nom de fichier d'une sauvegarde

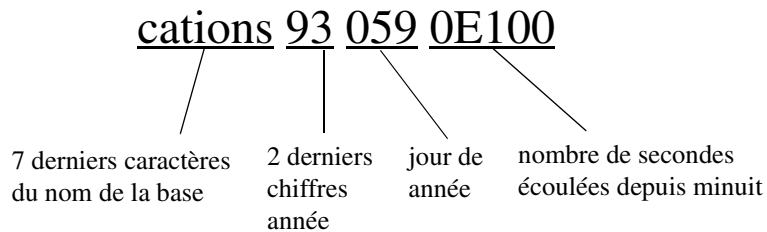
	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	<pre>dump {database tran} nom_basededonnées to [compress::[niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume,</pre>	<pre>load {database tran} nom_basededonnées from [compress::]device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume,</pre>
Nom de fichier pour un device particulier	<pre>file = nom_fichier]</pre>	<pre>file = nom_fichier]</pre>
	<pre>[stripe on [compress::[niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume,</pre>	<pre>[stripe on [compress::]device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, dumpvolume = nom_volume,</pre>

Identification d'une sauvegarde

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
Nom de fichier pour l'ensemble des devices	<code>file = nom_fichier,</code>	<code>file = nom_fichier,</code>
	<code>[nodismount dismount],</code> <code>[nounload unload],</code> <code>retaindays = nombre_jours,</code> <code>[noinit init],</code> <code>[notify = {client operator_console}]</code> <code>standby_access}]</code>	<code>[nodismount dismount],</code> <code>[nounload unload],</code> <code>[notify = {client operator_console}]</code> <code>standby_access}]</code>

Dans les exemples suivants, le journal de transactions de la base de données publications est sauvegardé sans spécification de nom de fichier. Le nom par défaut *cations930590E100* identifie la base de données, ainsi que la date et l'heure de la sauvegarde :

Figure 27-1 : Convention de dénomination pour les sauvegardes de journaux de transactions et de bases de données



Backup Server écrit le nom du fichier sur la destination par défaut des messages ou sur l'emplacement défini par `notify` de la commande de sauvegarde. N'oubliez pas de coller une étiquette indiquant les noms de volume et de fichier sur chaque bande de sauvegarde avant de l'archiver.

Lorsque vous chargez un journal de transactions ou une base de données à partir d'un volume contenant plusieurs sauvegardes, vous pouvez spécifier celle que vous souhaitez charger à l'aide de la clause `file = nom_fichier`.

Si vous chargez la sauvegarde depuis un volume contenant plusieurs fichiers, vous devez indiquer le nom du fichier correct.

```
dump tran publications
to "/dev/nrmt3"
load tran publications
from "/dev/nrmt4"
with file = "cations930590E100"
```

Dans les exemples suivants, la convention de dénomination de fichiers est définie par l'utilisateur. Le nom *mydb97jul141800* comporte 15 caractères qui identifient la base de données (mydb), la date (14 juillet 1997) et l'heure (18h00) de la sauvegarde. La commande de chargement fait défiler la bande jusqu'au fichier *mydb97jul141800* avant de lancer le chargement :

```
dump database mydb
to "/dev/nrmt3"
with file = "mydb97jul141800"
load database mydb
from "/dev/nrmt4"
with file = "mydb97jul141800"
```

Amélioration des performances de sauvegarde ou de chargement

Au démarrage de Backup Server, vous pouvez exécuter le paramètre `-m` pour améliorer les performances des commandes `dump` et `load` en configurant davantage de mémoire partagée pour le Backup Server. Vous disposez pour cela du paramètre `-m` qui permet de définir la quantité maximale par défaut de mémoire partagée disponible pour Backup Server. Vous devez également configurer le système d'exploitation pour mettre cette mémoire partagée à disposition de Backup Server. Les segments de mémoire ainsi affectés sont libérés à l'issue des opérations de sauvegarde ou de chargement.

Remarque L'amélioration des performances de sauvegarde/chargement est soumise aux limites de performances du matériel. Ainsi, augmenter la valeur de `-m` ne va pas nécessairement accélérer la sauvegarde sur un device de bande lent tel qu'un QIC, mais cela accélèrera considérablement la sauvegarde sur un device plus rapide tel qu'une bande DLT.

Compatibilité avec les versions antérieures

En fonction des versions utilisées, les formats de fichiers de sauvegarde et la version de Backup Server peuvent ne pas être compatibles. Le tableau 27-12 indique les formats de fichiers de sauvegarde admis sur les versions actuelles et antérieures de Backup Servers en local.

Tableau 27-12 : Serveur local

	Nouveau format de fichier de sauvegarde	Ancien format de fichier de sauvegarde
Serveur nouvelle version	Oui	Oui
Serveur version antérieure	Non	Oui

Les tableaux 27-13 et 27-14 indiquent les formats de fichiers de sauvegarde qui peuvent être chargés sur des versions actuelles ou antérieures de Backup Server distants. Sur ce type d'installation, le serveur maître est le Backup Server installé sur la machine de la base de données et d'Adaptive Server Enterprise et le serveur asservi est celui installé sur la machine distante du device d'archivage.

Le tableau 27-13 indique les opérations load (chargement) qui fonctionnent lorsque le serveur maître est une version actuelle de Backup Server.

Tableau 27-13 : Serveur maître nouvelle version

	Nouveau format de fichier de sauvegarde	Ancien format de fichier de sauvegarde
Serveur asservi nouvelle version	Oui	Oui
Serveur asservi version antérieure	Non	Oui

Le tableau 27-14 indique les opérations load (chargement) qui fonctionnent lorsque le serveur maître est une version antérieure de Backup Server.

Tableau 27-14 : Serveur maître version antérieure

	Nouveau format de fichier de sauvegarde	Ancien format de fichier de sauvegarde
Serveur asservi nouvelle version	Non	Oui
Serveur asservi version antérieure	Non	Oui

Étiquettes stockées au format nombre entier

Backup Server 12.0 et ultérieur stockent le numéro de device de sauvegarde au format nombre entier. Les versions antérieures de Backup Server stockaient le numéro de device de sauvegarde de 4 octets dans l'étiquette HDR1 au format ASCII. Ces versions ne peuvent pas charger un fichier de sauvegarde à un format plus récent. La version 12.0 et ultérieure de Backup Server peut néanmoins lire et écrire des versions antérieures du format de sauvegarde.

Les sauvegardes ou les chargements faisant intervenir un ou plusieurs serveurs distants sont interrompus avec un message d'erreur dans les cas suivants :

- Les versions d'un ou de plusieurs Backup Servers sont antérieures à 12.0 et la sauvegarde ou le chargement de la base de données fait intervenir plus de 32 devices de sauvegarde.
ou :
- La version du fichier de sauvegarde que lisent un ou plusieurs Backup Servers lors d'un chargement est plus ancienne et le nombre de devices de sauvegarde à partir desquels la base de données est chargée est supérieur à 32.

Configuration des ressources système

Avant de procéder à des sauvegardes ou à des chargements, vous devez configurer au démarrage les Backup Servers locaux et distants en indiquant les valeurs des ressources systèmes contrôlées par les options de commandes. Pour obtenir la liste complète de ces options, reportez-vous au guide *Utilitaires*.

La configuration erronée des ressources système peut entraîner l'échec des sauvegardes ou des chargements. Par exemple, une sauvegarde distante sur plus de 25 devices de sauvegarde faisant intervenir des Backup Servers locaux et distants démarrés avec la configuration par défaut échouera parce que le nombre maximum de connexions réseau que le Backup Server peut émettre (spécifié par l'option -N) est de 25 ; en revanche, le nombre maximum par défaut de connexions serveur au Backup Server (spécifié par l'option -C) est de 30.

Pour configurer le système de manière à ce qu'il utilise le nombre de devices de sauvegarde le plus élevé, il faut définir les paramètres de système d'exploitation suivants :

- Nombre de segments de mémoire partagée auquel un processus peut se rattacher
- Nombre d'identifiants de mémoire partagée
- Espace swap

Si la configuration de ces paramètres est incorrecte, une sauvegarde (ou un chargement) faisant intervenir un nombre élevé de devices de sauvegarde risque d'échouer par manque de ressources système. Dans ce cas, un message indique que Backup Server n'a pas pu créer un segment de mémoire partagé ou s'y rattacher, et qu'il est mis fin aux processus SYBMULTBUF.

Définition de l'usage de la mémoire partagée

La syntaxe permettant de démarrer Backup Server avec le paramètre -m est la suivante :

```
backupserv [-m nnn]
```

où *nnn* représente la quantité maximale (en mégaoctets) de mémoire partagée que Backup Server peut utiliser pour toutes ses opérations de sauvegarde et de chargement.

Le paramètre -m fixe la limite supérieure d'usage de la mémoire partagée. Backup Server peut utiliser moins de mémoire s'il estime que les performances ne seront pas meilleures avec davantage de mémoire.

Backup Server évalue la quantité de mémoire nécessaire pour chaque device de sauvegarde en divisant la valeur de -m par le nombre de threads de service défini au paramètre -P.

La valeur par défaut de -m est le nombre de threads de service multiplié par 1 Mo. La valeur par défaut de -P étant 48, la quantité maximale de mémoire partagée utilisable est de 48 Mo. Mais Backup Server n'atteint cette utilisation maximale que lorsque les 48 threads de service sont actifs simultanément. La valeur maximum de -P est le nombre maximum de threads de service, soit 12 288 (pour obtenir des informations complémentaires sur -P, reportez-vous à la section "Configuration de rôles définis par l'utilisateur", page 385.)



La quantité de mémoire partagée par device de sauvegarde à la disposition de Backup Server est inversement proportionnelle au nombre de threads de service alloués. Si vous augmentez le nombre maximum de threads de service, vous devez augmenter également la valeur de -m pour conserver la même quantité de mémoire partagée par device de sauvegarde. Si vous augmentez la valeur de -P mais pas celle de -m, la mémoire partagée allouée par device de sauvegarde risque de devenir trop faible pour autoriser la sauvegarde ou le chargement.

Pour savoir dans quelles proportions augmenter la valeur -m, utilisez la formule ci-dessous :

$$(-m \text{ valeur en Mo}) * 1024 / (\text{valeur -P})$$

Si le résultat obtenu est inférieur à 128 ko, Backup Server ne démarre pas.

La valeur minimale de -m est 6 Mo. Sa valeur maximale dépend du système d'exploitation et de ses limites en matière de mémoire partagée.

Si vous créez une sauvegarde avec un Backup Server doté de beaucoup de mémoire partagée et que vous tentez de charger la sauvegarde sur un Backup Server qui dispose de moins de mémoire partagée, les performances de chargement s'en trouveront amoindries, Backup Server ne pouvant utiliser que la mémoire disponible. Cette situation se traduit par une détérioration des performances de chargement.

Si, lors du chargement, la quantité de mémoire partagée disponible par device de sauvegarde n'est pas au moins deux fois supérieure à la taille de bloc utilisée lors de la sauvegarde, Backup Server annule le chargement et génère un message d'erreur.

Définition du nombre maximum de devices de sauvegarde

Le nombre maximum de devices de sauvegarde que Backup Server peut utiliser est limité par le nombre maximum de threads Open Server qu'il est à même de créer. Open Server impose un seuil maximum de 12 ko au nombre de threads qu'une application peut créer.

Backup Server créant un thread de service par device de sauvegarde, le nombre maximum de devices de sauvegarde locaux qu'il peut utiliser pour la sauvegarde ou le chargement est de 12 288.

De surcroît, il se sert de deux descripteurs de fichiers par device de sauvegarde, en sus de ceux associés aux fichiers de journal d'erreurs et d'interfaces et à d'autres fichiers système. Cependant le système d'exploitation impose un nombre limite de descripteurs de fichiers par thread. Avec Open Server, une application ne peut pas suivre plus de 1280 descripteurs de fichiers.

La formule permettant de déterminer le nombre maximum approximatif de devices de sauvegarde locaux sur lesquels Backup Server peut effectuer une sauvegarde est la suivante :

(limite la plus faible imposée par le système d'exploitation ou Open Server) - 2

2

La formule permettant de déterminer le nombre maximum approximatif de devices de sauvegarde distants sur lesquels Backup Server peut effectuer une sauvegarde est la suivante :

(limite la plus faible imposée par le système d'exploitation ou Open Server) - 2

3

Pour obtenir des informations complémentaires sur le nombre limites par défaut et maximum de descripteurs de fichiers, consultez la documentation de votre système d'exploitation.

Définition du nombre maximum de connexions réseau

Le nombre maximum de connexions réseau lancées par un Backup Server local est limité à 9118 par Open Server. De ce fait, le nombre maximum de devices de sauvegarde distants auxquels Backup Server peut faire appel pour une seule opération de sauvegarde ou de chargement est de 9118.

Un Backup Server distant accepte 4096 connexions serveur maximum à la fois. Le nombre maximum de devices de sauvegarde distants par Backup Server distant est donc de 4096.

Définition du nombre maximum de threads de service

Le paramètre -P de Backup Server configure le nombre de threads de service créés par Open Server. Le nombre maximum de threads de service est de 12 228. La valeur minimum est de 6. Le nombre maximum de threads est égal au nombre maximum de devices de sauvegarde disponibles. Si, après avoir démarré Backup Server sans définir une valeur -P suffisamment élevée, vous tentez de sauvegarder ou de charger une base de données sur un nombre de devices de sauvegarde supérieur au nombre de threads, la sauvegarde ou le chargement échouera.

Spécification d'autres devices de sauvegarde : clause *stripe on*

Le **striping** permet d'utiliser plusieurs devices de sauvegarde pour une seule commande de sauvegarde ou de chargement. Vous devez utiliser une clause *stripe on* distincte pour spécifier le nom (et, le cas échéant, les caractéristiques) de chaque device.

Chaque commande dump ou load peut être assortie de plusieurs clauses *stripe on*.

Le tableau 27-15 donne la syntaxe pour l'utilisation de plusieurs devices de sauvegarde.

Tableau 27-15 : Utilisation de plusieurs devices de sauvegarde

Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
<pre>dump {database tran} nom_basededonnées to [compress::[niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier]</pre>	<pre>load {database tran} nom_basededonnées from [compress::] device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier]</pre>

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
Caractéristiques d'un device de bande supplémentaire (un ensemble par device, 31 devices maximum)	<pre>[stripe on [compress::[niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] ...]</pre>	<pre>[stripe on [compress::]device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] ...]</pre>
	<pre>[with{ density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], retaindays = nombre_jours, [noinit init], [notify = {client operator_console}] standby_access}]</pre>	<pre>[with{ density = densité, dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], [notify = {client operator_console}] standby_access}]</pre>

Sauvegarde sur plusieurs devices

Backup Server divise la base de données en portions de taille sensiblement égales et envoie chacune d'elles vers un device distinct. Les sauvegardes sont effectuées simultanément sur tous les devices, ce qui diminue le temps nécessaire à la sauvegarde d'une base de données ou d'un journal de transactions particulier. En outre, comme chaque bande ne stocke qu'une partie de la base, il n'est généralement pas nécessaire de changer de bande sur l'un ou l'autre des devices.

Avertissement ! Ne sauvegardez pas la base de données master sur plusieurs devices à bande. Lors du chargement de la base de données master à partir d'une bande ou un autre support amovible, vous ne pouvez pas modifier les volumes, à moins de disposer d'un autre Adaptive Server capable de répondre à des messages de modification de volumes.

Chargement à partir de plusieurs devices

Vous pouvez utiliser plusieurs devices pour charger une base de données ou un journal de transactions. L'utilisation de plusieurs devices réduit le temps nécessaire au chargement et la probabilité d'avoir à monter plusieurs bandes sur l'un ou l'autre des devices.

Utilisation de moins de devices pour le chargement que pour la sauvegarde

Vous pouvez charger une base de données ou un journal même si l'un des devices utilisés pour la sauvegarde n'est plus disponible. Pour ce faire, spécifiez moins de clauses stripe dans la commande de chargement que dans la commande de sauvegarde.

Remarque Lorsque vous effectuez des sauvegardes et des chargements via le réseau, vous devez utiliser le même nombre de lecteurs pour les deux opérations.

Dans les exemples suivants, trois devices sont utilisés pour sauvegarder une base de données et seulement deux pour la charger :

```
dump database pubs2 to "/dev/nrmt0"
    stripe on "/dev/nrmt1"
    stripe on "/dev/nrmt2"
load database pubs2 from "/dev/nrmt0"
    stripe on "/dev/nrmt1"
```

Une fois les deux premières bandes chargées, un message demande à l'opérateur de charger la troisième.

Vous pouvez également sauvegarder une base de données dans plusieurs fichiers du système d'exploitation. L'exemple suivant s'applique à Windows NT :

```
dump database pubs2 to "d:\backups\backup1.dat"
    stripe on "d:\backups\backup2.dat"
    stripe on "d:\backups\backup3.dat"
load database pubs2 from "/dev/nrmt0"
    stripe on "d:\backups\backup2.dat"
    stripe on "d:\backups\backup3.dat"
```

Spécification des caractéristiques de devices individuels

Utilisez une clause `at nom_serveur` distincte pour chaque device de sauvegarde rattaché à un Backup Server distant. Si vous ne spécifiez pas de nom de Backup Server distant, le Backup Server local recherche le device de sauvegarde sur le système local. Si nécessaire, vous pouvez également spécifier des caractéristiques distinctes de device de sauvegarde (`density`, `blocksize`, `capacity`, `dumpvolume` et `file`) pour chaque device.

Les exemples suivants utilisent trois devices de sauvegarde, reliés chacun au Backup Server distant `REMOTE_BKP_SERVER` :

Sous UNIX :

```
dump database pubs2
  to "/dev/nrmt0" at REMOTE_BKP_SERVER
  stripe on "/dev/nrmt1" at REMOTE_BKP_SERVER
  stripe on "/dev/nrmt2" at REMOTE_BKP_SERVER
```

Sous OpenVMS :

```
dump database pubs2
  to "MTA0:" at REMOTE_BKP_SERVER
  stripe on "MTA1:" at REMOTE_BKP_SERVER
  stripe on "MTA2:" at REMOTE_BKP_SERVER
```

Options d'exploitation de bande

Les options d'exploitation de bande figurant dans la clause `with` s'appliquent à tous les devices utilisés pour la sauvegarde ou le chargement. Ces options sont les suivantes :

- `nodismount` laisse la bande disponible pour d'autres sauvegardes ou chargements ;
- `unload` rembobine et décharge la bande à l'issue de la sauvegarde ou du chargement ;
- `retaindays` protège les fichiers contre l'écrasement ;
- `init` réinitialise la bande au lieu d'ajouter les fichiers de sauvegarde à la suite du dernier marqueur de fin de bande.

Le tableau 27-16 donne la syntaxe pour les options d'exploitation de bande.

Tableau 27-16 : Options d'exploitation de bande

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	<pre>dump {database tran} nom_basededonnées to [compress::[niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] [stripe on [compress::[niveau_compression::]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier,</pre>	<pre>load {database tran} nom_basededonnées from [compress::]device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] [compress::]device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, dumpvolume = nom_volume, file = nom_fichier,</pre>
Options d'exploitation de bande	<pre>[nodismount dismount], [nounload unload], retaindays = nombre_jours, [noinit init],</pre>	<pre>nodismount dismount], [nounload unload],</pre>
	<pre>[notify = {client operator_console}] standby_access}]</pre>	<pre>[notify = {client operator_console}] standby_access}]</pre>

Spécification de démontage de la bande

Sur les plates-formes gérant les démontages logiques, telles qu'OpenVMS, les bandes sont démontées à l'issue de la sauvegarde ou du chargement. Si vous souhaitez conserver la bande en place, afin qu'elle soit disponible pour d'autres opérations, utilisez l'option `nodismount`. Cette commande est sans effet sur système UNIX ou PC.

Rembobinage de la bande

Par défaut, les commandes de sauvegarde et de chargement utilisent l'option d'exploitation de bande `nounload`.

Sur système UNIX, cette option empêche que la bande soit rebobinée à l'issue de la sauvegarde ou du chargement. Vous pouvez alors utiliser le même volume pour sauvegarder ou charger d'autres bases de données ou journaux. Pour la dernière sauvegarde sur la bande, utilisez l'option `unload` ; la bande est alors rebobinée et déchargée à la fin de l'exécution de la commande.

Sur système OpenVMS, les bandes sont toujours rebobinées à l'issue d'une sauvegarde ou d'un chargement. Utilisez l'option `unload` pour décharger la bande et l'éjecter du lecteur. (Cette opération a le même effet que le qualificatif `/UNLOAD` avec la commande OpenVMS `DISMOUNT`).

Protection des fichiers de sauvegarde contre l'écrasement

La commande `tape retention in days` indique le nombre de jours devant s'écouler entre la date de création du fichier sur la bande et la date à laquelle vous pouvez le remplacer par une autre sauvegarde. Cette variable serveur, définie dans la procédure système `sp_configure`, s'applique à toutes les demandes de sauvegarde issues d'un même Adaptive Server.

Exécutez l'option `retaindays = nombre_jours` pour ignorer le paramètre `tape retention in days` lors de la sauvegarde d'une seule base de données ou d'un seul journal de transactions. Le nombre de jours doit être un entier positif ou zéro, si le contenu de la bande peut être écrasé immédiatement.

Remarque `tape retention in days` et `retaindays` ne sont utiles que pour les disques, les cartouches 1/4 de pouce et les supports mono-fichier. En effet, pour les supports multifichier, Backup Server vérifie uniquement la date d'expiration du premier fichier.

Réinitialisation d'un volume avant une sauvegarde

Par défaut, chaque sauvegarde est ajoutée à la suite du dernier marqueur de fin de bande. Les volumes de bande ne sont pas réinitialisés. Ainsi, vous pouvez sauvegarder plusieurs bases de données sur un seul volume. (Cependant, les nouvelles sauvegardes ne peuvent être ajoutées qu'au dernier volume d'une sauvegarde sur plusieurs volumes.)

Exécutez l'option `init` pour écraser le contenu de la bande. Si vous spécifiez `init`, Backup Server réinitialise la bande et ne recherche *aucun* des éléments suivants :

- restrictions d'accès ANSI
- Fichiers dont la date d'expiration n'est pas atteinte
- Données autres que Sybase (bandes "étrangères" sous OpenVMS).

L'option par défaut `noinit` vérifie si l'un de ces éléments existe et, si tel est le cas, envoie un message de changement de volume.

Dans l'exemple suivant, deux devices sont initialisés et remplacent le contenu de la bande par les nouvelles sauvegardes du journal de transactions :

```
dump transaction pubs2
to "/dev/nrmt0"
stripe on "/dev/nrmt1"
with init
```

Lorsque vous sauvegardez une base de données dans un fichier système, vous pouvez également utiliser l'option `init` pour écraser un fichier existant. L'exemple suivant s'applique à Windows NT :

```
dump transaction pubs2
to "d:\backups\backup1.dat"
stripe on "d:\backups\backup2.dat"
with init
```

Sauvegardes multiples sur un seul volume

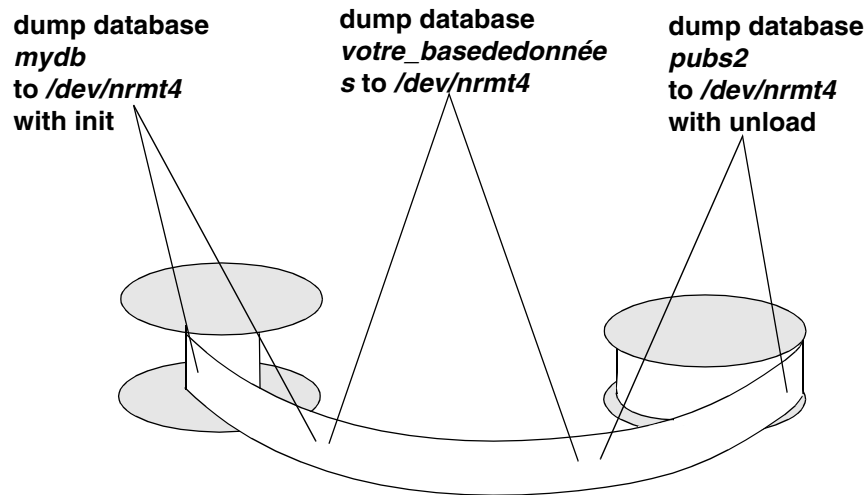
Suivez les instructions ci-dessous pour sauvegarder plusieurs bases de données sur le même volume de bande :

- 1 Utilisez l'option `init` pour la première base de données. Cette opération écrase toute sauvegarde existante et place la première sauvegarde en début de bande.

- 2 Utilisez l'option par défaut (noinit et noload) pour les bases de données suivantes. Cette opération les place l'une à la suite de l'autre sur la bande.
- 3 Utilisez l'option unload pour la dernière base de données sur la bande. Cette opération rembobine et décharge la bande une fois la dernière base sauvegardée.

La figure 27-2 représente les options à utiliser pour sauvegarder trois bases de données sur un seul volume de bande.

Figure 27-2 : Sauvegarde de plusieurs bases de données sur le même volume



Remplacement de la destination par défaut des messages

Les messages Backup Server informent l'opérateur des changements de volume de bande nécessaires, ainsi que du déroulement de la sauvegarde ou du chargement. La destination par défaut de ces messages dépend de la présence éventuelle d'une fonction de terminal d'opérateur dans le système d'exploitation.

L'option `notify` de la clause `with` permet de remplacer la cible par défaut pour une sauvegarde ou un chargement donné. Pour que cette option fonctionne, le terminal de l'opérateur ou la session de connexion depuis lequel Backup Server a été démarré doit rester actif tant que Backup Server travaille, sinon, le message `sp_volchanged` sera perdu.

Sur les systèmes d'exploitation tels qu' OpenVMS, qui offrent une fonction de terminal d'opérateur, les messages de changement de volume sont toujours envoyés vers un terminal d'opérateur de la machine accueillant Backup Server. (OpenVMS achemine les messages vers les terminaux pouvant supporter TAPES, DISKS ou CENTRAL.) Utilisez `notify = client` pour acheminer les autres messages de Backup Server vers la session de terminal où la requête `dump` ou `load` a été lancée.

Sur les systèmes tels qu'UNIX, qui ne proposent pas la fonction de terminal d'opérateur, les messages sont envoyés au client qui a lancé la requête de sauvegarde ou de chargement. Utilisez `notify = operator_console` pour acheminer les messages vers le terminal sur lequel a été démarré le Backup Server distant.

Le tableau 27-17 donne la syntaxe pour le remplacement de la destination par défaut des messages.

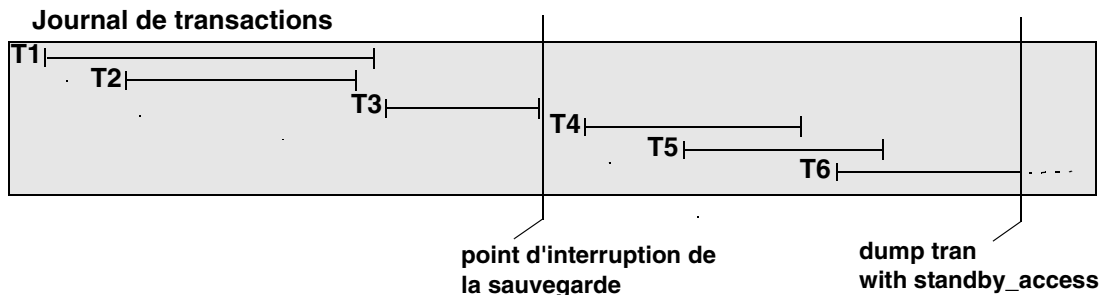
Tableau 27-17 : Remplacement de la destination par défaut des messages

	Sauvegarde d'une base de données ou d'un journal	Chargement d'une base de données ou d'un journal
	<pre>dump {database tran} nom_basededonnées to [compress::[niveau_compression:]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] [stripe on [compress::[niveau_compression:]] device_sauvegarde [at nom_serveur] [density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, blocksize = nombre_octets, capacity = nombre_kilo_octets, dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload], retaindays = nombre_jours, [noinit init],</pre>	<pre>load {database tran} nom_basededonnées from [compress::]device_sauvegarde [at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] [stripe on [compress::]device_sauvegarde at nom_serveur] [density = densité, dumpvolume = nom_volume, file = nom_fichier] ...] [with{ density = densité, dumpvolume = nom_volume, file = nom_fichier, [nodismount dismount], [nounload unload],</pre>
Destination des messages	<pre>[notify = {client operator_console}] standby_access}]</pre>	<pre>[notify = {client operator_console}] standby_access}]</pre>

Mise des bases de données en ligne avec *with standby_access*

with standby_access restreint la sauvegarde effectuée avec `dump transaction` aux transactions terminées. Le journal des transactions est sauvegardé jusqu'au point où plus aucune transaction n'est active. Si vous ne spécifiez pas *with standby_access*, le journal complet des transactions est sauvegardé, y compris les enregistrements de toutes les transactions ouvertes. Un exemple de journal de transaction sauvegardé avec *with standby_access* est illustré à la figure 27-3.

Figure 27-3 : Point d'interruption de sauvegarde avec *with standby_access*



Dans l'exemple illustré à la figure 27-3, une commande `dump transaction...with standby_access` est soumise à l'issue des transactions T1 à T5 alors que la transaction T6 est encore ouverte. La transaction T5 ne pourra être incluse dans la sauvegarde puisque T6 est encore ouverte, de même que T4 ne pourra être incluse puisque T5 est, elle aussi, ouverte. La sauvegarde s'arrêtera donc à la fin de la transaction T3 et comprendra les transactions achevées T1 à T3.

Syntaxe

La syntaxe de *with standby_access* est la suivante :

```
dump tran[saction] nom_basededonnées to...
[with standby_access]
```

Pour obtenir des informations complémentaires sur l'option avec `dump transaction...with standby_access`, Reportez-vous au *Manuel de référence d'Adaptive Server Enterprise*.

Dans quels cas utiliser *with standby_access* ?

Utilisez `dump tran[saction]...with standby_access` lorsque vous devez charger deux ou plusieurs journaux de transaction à la suite, tout en gardant la base de données en ligne entre deux chargements. C'est par exemple le cas lorsque votre base de données est en lecture seule et qu'elle récupère ses données à partir des sauvegardes de transaction d'une base de données primaire. Si la base de données en lecture seule sert à générer des rapports journaliers sur les transactions de la base de données primaire et que le journal de cette base est sauvegardé à la fin de chaque journée, le cycle journalier des opérations est le suivant :

- 1 Sur la base de données primaire : `dump tran[saction]...with standby_access`
- 2 Sur la base de données en lecture seule : `load tran[saction]...`
- 3 Sur la base de données en lecture seule : `online database for standby_access`

Avertissement ! Si un journal de transaction inclut des transactions ouvertes et que vous le sauvegardez sans spécifier l'option `with standby_access`, Adaptive Server ne vous autorisera pas à charger le journal ni à mettre la base en ligne et vous ne pourrez pas non plus charger une sauvegarde de transaction ultérieure. Si vous envisagez de charger une série de sauvegardes de transactions, vous ne pourrez mettre la base de données en ligne qu'après avoir chargé une sauvegarde effectuée initialement avec `with standby_access` ou la série complète des sauvegardes.

Mise des bases de données en ligne avec *for standby_access*

La commande `online database` inclut également une option `for standby_access`. Exécutez `for standby_access` pour mettre en ligne une base de données après avoir chargé une sauvegarde effectuée avec l'option `with standby_access`.

Avertissement ! Si vous tentez d'utiliser la commande `online database for standby_access` avec un journal de transaction qui n'a pas été sauvegardé avec l'option `with standby_access`, la commande échoue.

Syntaxe

La syntaxe de la commande online database est la suivante :

```
online database nom_basededonnées [for standby_access]
```

Pour obtenir des informations complémentaires sur l'option avec online database...for standby_access, reportez-vous au *Manuel de référence d'Adaptive Server Enterprise*.

Obtention d'informations sur les fichiers de sauvegarde

En cas de doute sur le contenu d'une bande, utilisez l'option with headeronly ou with listonly des commandes de chargement pour obtenir ces informations.

Le tableau 27-18 donne la syntaxe permettant de connaître le contenu d'une bande.

Tableau 27-18 : Liste des noms de fichiers ou des en-têtes de sauvegarde

	Liste d'informations sur une sauvegarde
	<pre>load {database tran} <i>nom_basededonnées</i> from [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>, [nodismount dismount], [nounload unload],</pre>
Liste des informations relatives aux en-têtes	<pre>[headeronly [, file = <i>nom_fichier</i>]], [listonly [= full]],</pre>
Liste des fichiers sur la bande	<pre>[notify = {client operator_console}] standby_access}}</pre>

Remarque Les options `with headeronly` et `with listonly` ne chargent pas les fichiers de sauvegarde après l'affichage du rapport.

Requête d'informations relatives aux en-têtes de sauvegarde

L'option `with headeronly` renvoie le contenu de l'en-tête d'un seul fichier donné. Si vous n'indiquez pas de nom de fichier, les informations renvoyées `with headeronly` sont celles du premier fichier de la bande.

L'en-tête indique s'il s'agit d'une sauvegarde de base de données ou de journal de transactions et fournit l'ID de la base, le nom du fichier, ainsi que la date de la sauvegarde. Dans le cas d'une sauvegarde de base, il mentionne également le jeu de caractères, l'ordre de tri, le nombre de pages et l'ID d'objet suivant. Dans le cas d'une sauvegarde de journal de transactions, l'en-tête indique l'emplacement du point de reprise dans le journal, l'emplacement de l'enregistrement `begin transaction` le plus ancien, ainsi que l'ancienne et la nouvelle date de séquence.

Dans l'exemple suivant les informations d'en-tête du premier fichier de la bande s'affichent, puis celles du fichier `mydb9229510945` :

```
load database mydb
  from "/dev/nrmt4"
  with headeronly
load database mydb
  from "/dev/nrmt4"
  with headeronly, file = "mydb9229510945"
```

Voici un exemple de résultat obtenu avec `headeronly` :

```
L'identifiant de la session Backup Server est : 44. Utilisez cette valeur lors
de l'exécution de la procédure système 'sp_volchanged' après la réalisation
d'une requête de changement de volume émise par le Backup Server.
Backup Server: 4.28.1.1: Dumpfile name 'mydb9232610BC8 ' section number 0001
mounted on device 'backup/SQL_SERVER/mydb.db.dump'
This is a database dump of database ID 5 from Nov 21 1992 7:02PM.
Database contains 1536 pages; checkpoint RID=(Rid pageid = 0x404; row num =
0xa); next object ID=3031; sort order ID=50, status=0; charset ID=1.
```

Détermination de la base de données, du device, du nom de fichier et de la date

L'option `with listonly` renvoie une brève description de chaque fichier de sauvegarde d'un volume. Elle indique le nom de la base de données, le device utilisé pour effectuer la sauvegarde, le nom de fichier, la date et l'heure de la sauvegarde, ainsi que la date et l'heure à partir desquelles l'écrasement est possible. L'option `with listonly = full` fournit davantage de détails. Ces deux rapports sont triés selon l'étiquette de bande SQL.

Voici un exemple de résultat obtenu avec la commande `load database` et l'option `with listonly` :

```
Backup Server: 4.36.1.1: Device '/dev/nrst0':
File name: 'model9320715138 '
Create date & time: Monday, Jul 26, 1993, 23:58:48
Expiration date & time: Monday, Jul 26, 1993, 00:00:00
Database name: 'model'
```

et un exemple de résultat avec l'option `with listonly = full` :

```
Backup Server: 4.37.1.1: Device '/dev/nrst0':
Label id: 'HDR1'
File name: 'model9320715138 '
Stripe count:0001
Device typecount:01
Archive volume number:0001
Stripe position:0000
Generation number:0001
Generation version:00
Create date & time:Monday, Jul 26, 1993, 23:58:48
Expiration date & time:Monday, Jul 26, 1993, 00:00:00
Access code:' '
File block count:000000
Sybase id string:
'Sybase 'Reserved:'
Backup Server: 4.38.1.1: Device '/dev/nrst0':
Label id:'HDR2'
Record format:'F'
Max. bytes/block:55296
Record length:02048
Backup format version:01
Reserved:' '
Database name:'model'
Buffer offset length:00
Reserved:' '
```

Après avoir répertorié tous les fichiers d'un volume, Backup Server envoie une requête de changement de volume :

```
Backup Server: 6.30.1.2: Device /dev/nrst0: Volume cataloguing
complete.
Backup Server: 6.51.1.1: OPERATOR: Mount the next volume to search.
```

```
Backup Server: 6.78.1.1: EXECUTE sp_volchanged
    @session_id = 5,
    @devname = '/dev/nrst0',
    @action = { 'PROCEED' | 'RETRY' | 'ABORT' },
    @fname = '
```

L'opérateur peut monter un autre volume et signaler ce changement à l'aide de `sp_volchanged` ou mettre fin à l'opération de recherche pour tous les devices de sauvegarde.

Copie du journal après une panne de device

En principe, `dump transaction` tronque la portion inactive du journal après la copie. Exécutez `with no_truncate` pour copier le journal sans le tronquer.

`no_truncate` permet de copier le journal de transactions après un incident sur le device contenant vos données. Cette commande utilise des pointeurs dans les tables système `sysdatabases` et `sysindexes` pour déterminer l'emplacement physique du journal de transactions. Elle ne peut s'utiliser que si le journal de transactions réside sur un segment distinct et que votre base de données master est accessible.

Avertissement ! Exécutez `no_truncate` uniquement lorsqu'une panne de disque interdit l'accès à votre segment de données. N'utilisez jamais `no_truncate` sur une base de données en cours d'utilisation.

La copie du journal à l'aide de `with no_truncate` est la première étape décrite dans la section "Restauration d'une base de données : instructions détaillées", page 944.

Le tableau 27-19 donne la syntaxe permettant de copier un journal après une panne de device.

Tableau 27-19 : Copie du fichier de journal après une panne de device

Copie à l'aide de l'option <i>no_truncate</i>	
	<pre> dump transaction <i>nom_basededonnées</i> to [compress::[<i>niveau_compression</i>::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>] [stripe on [compress::[<i>niveau_compression</i>::]] <i>device_sauvegarde</i> [at <i>nom_serveur</i>] [density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dump volume = <i>nom_volume</i>, file = <i>nom_fichier</i>] ...] [with{ density = <i>densité</i>, blocksize = <i>nombre_octets</i>, capacity = <i>nombre_kilo_octets</i>, dumpvolume = <i>nom_volume</i>, file = <i>nom_fichier</i>, [nodismount dismount], [nounload unload], retaindays = <i>nombre_jours</i>, [noinit init], </pre>
Ne pas tronquer le journal	<pre> no_truncate, </pre>
	<pre> [notify = {client operator_console}] standby_access}] </pre>

Vous pouvez utiliser l'option *no_truncate* avec des sauvegardes multibandes, une initialisation de bande et des Backup Server distants. Par exemple :

```

dump transaction mydb
to "/dev/nrmt0" at REMOTE_BKP_SERVER
with init, no_truncate,
notify = "operator_console"

```

Troncature d'un journal ne se trouvant pas sur un segment distinct

Lorsqu'une base de données ne dispose pas d'un segment de journal sur un device distinct des segments de données, vous ne pouvez pas utiliser la commande `dump transaction` pour copier le journal puis le tronquer. Dans ce cas, vous devez procéder comme suit :

- 1 Utilisez l'option spéciale `with truncate_only` de la commande `dump transaction` pour tronquer le journal afin qu'il dispose de suffisamment d'espace.
- 2 Exécutez `dump database` pour copier l'intégralité de la base de données, journal compris

Comme elle ne sauvegarde pas de données, l'option `with truncate_only` nécessite uniquement le nom de la base de données :

```
dump transaction nom_basededonnées with
truncate_only
```

L'exemple suivant sauvegarde la base de données `mydb` qui ne dispose pas d'un segment de journalisation dans un device distinct de segments de données, puis tronque ce dernier :

```
dump database mydb to mydevice
dump transaction mydb with truncate_only
```

Troncature du journal en environnement de début de développement

Dans les environnements de début de développement, le journal de transactions est rapidement saturé par les processus de création, de suppression et de recréation des procédures stockées et des triggers, ainsi que de vérification des contraintes d'intégrité. Il peut alors être moins important de restaurer les données que de garantir que l'espace sur les devices de bases de données reste suffisant.

L'option `with truncate_only` permet de tronquer le journal de transactions sans effectuer de copie de sauvegarde :

```
dump transaction nom_basededonnées with
truncate_only
```

Après l'exécution de `dump transaction with truncate_only`, vous devez sauvegarder la base de données avant de pouvoir recommencer à sauvegarder le journal de transactions.

Troncature d'un journal sans espace libre

Lorsque le journal de transactions est très rempli, il est possible que vous ne puissiez pas le sauvegarder à l'aide de la méthode habituelle. Si la commande `dump transaction` ou `dump transaction with truncate_only` a échoué par manque d'un espace suffisant dans le journal, exécutez l'option spéciale `with no_log` de `dump transaction` :

```
dump transaction nom_basededonnées with no_log
```

Cette option tronque le journal sans y consigner l'événement de troncature. Comme elle ne sauvegarde pas de données, seul le nom de la base de données est requis.

Avertissement ! N'utilisez `dump transaction with no_log` qu'en dernier recours et uniquement en cas d'échec de la commande `dump transaction with truncate_only`. Si vous continuez à charger des données après avoir entré la commande `dump transaction with no_log`, vous risquez de saturer le journal, ce qui engendrera l'échec de toute commande `dump transaction` ultérieure. Utilisez la commande `alter database` pour allouer de l'espace supplémentaire à la base de données.

Toutes les occurrences de `dump tran with no_log` sont répertoriées dans le journal d'erreurs d'Adaptive Server. Le message inclut l'ID de l'utilisateur exécutant la commande. Les messages indiquant le succès ou l'échec sont également envoyés vers le journal d'erreurs. `no_log` est la seule option de sauvegarde qui génère des messages dans le journal d'erreurs.

Risques liés à l'utilisation de *with truncate_only* et de *with no_log*

Les options `with truncate_only` et `with no_log` servent à tronquer un journal dont l'espace libre est nettement insuffisant. Ni l'une, ni l'autre de ces options ne permet de restaurer des transactions validées depuis la dernière sauvegarde systématique.

Avertissement ! Exécutez `dump database` dès que vous en avez la possibilité pour vous assurer que vos données peuvent être récupérées.

L'exemple suivant tronque le journal de transactions de `mydb`, puis sauvegarde la base de données :

```
dump transaction mydb
with no_log
dump database mydb to ...
```

Aménagement d'un espace suffisant pour le journal

Chaque utilisation de `dump transaction...with no_log` est considérée comme une erreur et donc enregistrée dans le journal d'erreurs du serveur. Si vous avez créé vos bases de données avec des segments de journal sur un device distinct des segments de données, écrivez une procédure de seuil ultime qui sauvegarde régulièrement votre journal de transactions et alloué suffisamment d'espace au journal et à la base de données, vous n'aurez en principe jamais à utiliser cette option.

Cependant, certaines situations peuvent encore engendrer la saturation du journal de transactions, même s'il est fréquemment sauvegardé. La commande `dump transaction` tronque le journal en supprimant toutes les pages depuis le début de celui-ci jusqu'à la page précédant celle qui contient un enregistrement de transaction non validée (transaction active la plus ancienne). Tant que cette transaction active n'est pas validée, la quantité d'espace disponible dans le journal de transactions est réduite, puisque `dump transaction` ne peut pas tronquer de pages supplémentaires.

Cela peut se produire lorsque des applications comportant des transactions très longues modifient les tables d'une base de données avec un journal de transactions de petite taille : augmentez alors la taille de ce journal. C'est également le cas lorsque des transactions ne sont pas validées pendant de longues périodes, lorsqu'une commande `begin transaction` implicite est générée par le mode de transaction chaîné ou lorsqu'un utilisateur oublie de terminer une transaction. Vous pouvez déterminer la transaction active la plus ancienne de chaque base de données en interrogeant la table système `syslogshold`.

Table `syslogshold`

La table `syslogshold` se trouve dans la base de données master. Chacune de ses lignes représente l'un des deux éléments suivants :

- la transaction active la plus ancienne d'une base de données ou
- le point de troncature de Replication Server pour le journal de la base de données.

La table `syslogshold` peut ne comporter aucune ligne, être constituée d'une ligne représentant l'un des éléments ci-dessus ou de deux lignes représentant les deux éléments d'une base de données cités précédemment. Pour plus d'informations sur la manière dont un point de troncature de Replication Server influe sur la troncature du journal de transactions de la base de données, reportez-vous à la documentation de Replication Server.

La table `syslogshold` donne un aperçu de la situation courante dans chaque base de données. La plupart des transactions étant de courte durée, les résultats de la requête peuvent être incohérents. Par exemple, la transaction active la plus ancienne décrite dans la première ligne de `syslogshold` peut s'achever avant qu'Adaptive Server n'ait achevé la requête sur la table. Cependant, lorsque plusieurs interrogations de `syslogshold` sur une période aboutissent à l'obtention de la même ligne pour une base de données, cette transaction peut empêcher une commande `dump transaction` de tronquer de l'espace de journalisation.

Lorsque le journal de transactions atteint le seuil ultime et que `dump transaction` ne parvient pas à libérer de l'espace dans le journal, vous pouvez interroger les tables `syslogshold` et `sysindexes` pour identifier la transaction qui empêche la troncature. Exemple :

```
select H.spid, H.name
from master..syslogshold H, threshdb..sysindexes I
where H.dbid = db_id("threshdb")
et I.id = 8
```

```
and H.page = I.first
spid      name
-----
      8  $user_transaction
```

(1 row affected)

Cette requête utilise l'ID de l'objet associé à syslogs (8) dans la base de données threshdb afin de faire coïncider la première page de son journal de transactions avec celle de la transaction active la plus ancienne dans syslogshold.

Vous pouvez également interroger syslogshold et sysprocesses dans la base de données master pour identifier le nom de l'ordinateur et de l'application auxquels appartiennent les transactions actives les plus anciennes. Exemple :

```
select P.hostname, P.hostprocess, P.program_name,
       H.name, H.starttime
from sysprocesses P, syslogshold H
where P.spid = H.spid
and H.spid != 0
```

hostname	hostprocess	program_name	name	starttime
eagle	15826	isql	\$user_transaction	Sep 6 1997 4:29PM
hawk	15859	isql	\$user_transaction	Sep 6 1997 5:00PM
condor	15866	isql	\$user_transaction	Sep 6 1997 5:08PM

(3 rows affected)

A l'aide des informations ci-dessus, vous pouvez avertir ou supprimer (kill) le processus utilisateur propriétaire de la transaction active la plus ancienne et poursuivre la transaction dump transaction. Vous pouvez également inclure les types de requête ci-dessus dans les procédures associées aux seuils pour la base de données en tant que mécanisme d'avertissement automatique. Par exemple, vous pouvez décider que le journal de transactions n'atteigne jamais son seuil ultime. S'il l'atteint, la procédure associée au seuil ultime (sp_thresholdaction) peut générer une alerte à l'aide d'informations sur la transaction active la plus ancienne empêchant la sauvegarde de la transaction.

Remarque Les premiers enregistrements du journal pour une transaction donnée peuvent résider dans le cache du journal et ne pas apparaître dans syslogshold tant que les enregistrements ne sont pas vidés vers le journal (après un point de reprise, par exemple).

Pour obtenir des informations complémentaires sur la table système `syslogshold`, reportez-vous au *Manuel de référence d'Adaptive Server Enterprise*. Pour plus d'informations sur le seuil ultime et les procédures associées aux seuils, reportez-vous au chapitre 29, "Gestion de l'espace libre avec des seuils".

Réponse aux requêtes de changement de volume

Sur systèmes UNIX ou PC, utilisez `sp_volchanged` pour signaler à Backup Server que les volumes corrects ont été montés. Sur système OpenVMS, utilisez pour ce faire la commande `REPLY`.

Pour utiliser `sp_volchanged`, connectez-vous à un Adaptive Server capable de communiquer à la fois avec le Backup Server ayant émis la requête de changement de volume et l'Adaptive Server à l'origine de la sauvegarde ou du chargement.

Syntaxe de `sp_volchanged`

La syntaxe de `sp_volchanged` est la suivante :

```
sp_volchanged id_session, nom_device, action  
[ ,fname [ , nom_volume ] ]
```

- Utilisez les paramètres *id_session* et *nom_device* spécifiés dans la requête de changement de volume.
- *action* précise s'il faut appliquer les commandes `abort`, `proceed with` ou `retry` à la sauvegarde ou au chargement.
- *nom_fichier* désigne le fichier à charger. Si vous ne spécifiez pas de nom de fichier, Backup Server charge le paramètre `file = nom_fichier` de la commande `load`. Si ni `sp_volchanged` ni la commande de chargement ne désignent un fichier à charger, Backup Server charge le premier fichier de la bande.

- En cas d'écrasement d'une sauvegarde existante, de sauvegarde sur une bande neuve ou sur une bande dont le contenu n'est pas reconnu, le *nom_volume* est écrit par Backup Server dans l'étiquette de bande ANSI. Lors des chargements, Backup Server se sert du nom de volume *nom_volume* pour vérifier que la bande appropriée a été montée. Si vous ne spécifiez pas de *nom_volume*, Backup Server prend en compte le nom de volume spécifié dans la commande de sauvegarde ou de chargement. Si ni *sp_volchanged*, ni la commande ne désignent un nom de volume, Backup Server ne contrôle pas le champ correspondant dans l'étiquette de bande ANSI.

Invites de changement de volume lors des sauvegardes

Cette section présente les invites de changement de volume apparaissant au cours de la sauvegarde d'une base de données ou d'un journal de transactions. Elle répertorie, pour chaque invite, les actions que l'opérateur peut effectuer et la réponse *sp_volchanged* appropriée.

- Mount the next volume to search.

Lorsque vous ajoutez une sauvegarde à un volume existant, Backup Server envoie ce message s'il ne peut trouver le marqueur de fin de fichier.

L'opérateur peut	En répondant
Annuler la sauvegarde	<i>sp_volchanged id_session, nom_device, abort</i>
Monter un nouveau volume et procéder à la sauvegarde	<i>sp_volchanged id_session, nom_device, proceed [, nom_fichier [, nom_volume]]</i>

- Mount the next volume to write.

Backup Server renvoie ce message lorsqu'il parvient en fin de bande. Cela se produit lorsqu'il détecte le marqueur de fin de bande, lorsque la sauvegarde atteint le nombre de kilo-octets spécifié par le paramètre *capacity* de la commande *dump* ou qu'elle atteint la valeur *haute* spécifiée pour le device dans la table système *sysdevices*.

L'opérateur peut	En répondant
Annuler la sauvegarde	<i>sp_volchanged id_session, nom_device, abort</i>
Monter le volume suivant et procéder à la sauvegarde	<i>sp_volchanged id_session, nom_device, proceed [, nom_fichier [, nom_volume]]</i>

- Volume on device devname has restricted access (code access_code). (l'accès au volume du device nom_device fait l'objet de restrictions (code code_accès))

Les sauvegardes avec l'option `init` écrasent intégralement le contenu de la bande. Backup Server renvoie ce message si vous tentez de sauvegarder sur une bande soumise à des restrictions d'accès ANSI sans spécifier l'option `init`.

L'opérateur peut	En répondant
Annuler la sauvegarde	<code>sp_volchanged id_session, nom_device, abort</code>
Monter un autre volume et retenter la sauvegarde	<code>sp_volchanged id_session, nom_device, retry [, nom_fichier [, nom_volume]]</code>
Procéder à la sauvegarde en écrasant le contenu de la bande	<code>sp_volchanged id_session, nom_device, proceed [, nom_fichier [, nom_volume]]</code>

- Volume on device devname is expired and will be overwritten. (le volume du device nom_device a expiré et sera écrasé)

Les sauvegardes avec l'option `init` écrasent intégralement le contenu de la bande. Lors d'une sauvegarde sur un support mono-fichier, Backup Server renvoie ce message si vous n'avez pas spécifié l'option `init` et que la bande contient une sauvegarde dont la date d'expiration est dépassée.

L'opérateur peut	En répondant
Annuler la sauvegarde	<code>sp_volchanged id_session, nom_device, abort</code>
Monter un autre volume et retenter la sauvegarde	<code>sp_volchanged id_session, id_session, retry [, id_session [, id_session]]</code>
Procéder à la sauvegarde en écrasant le contenu de la bande	<code>sp_volchanged id_session, id_session, proceed [, id_session [, id_session]]</code>

- Volume to be overwritten on 'devname' has not expired: (le volume à écraser sur 'nom_device' n'est pas arrivé à expiration) creation date on this volume is creation_date, expiration date is expiration_date. (la date de création de ce volume est date_création, sa date d'expiration est date_expiration)

Sur les supports mono-fichier, lorsque vous ne spécifiez pas l'option `init`, Backup Server vérifie la date d'expiration de toute sauvegarde existante. Il renvoie ce message si la date d'expiration de la sauvegarde n'est pas encore atteinte.

L'opérateur peut	En répondant
Annuler la sauvegarde	<code>sp_volchanged id_session, id_session, abort</code>
Monter un autre volume et retenter la sauvegarde	<code>sp_volchanged id_session, id_session, retry [, id_session [, id_session]]</code>
Procéder à la sauvegarde en écrasant le contenu de la bande	<code>sp_volchanged id_session, id_session, proceed [, id_session [, id_session]]</code>

- `Volume to be overwritten on 'devname' has unrecognized label data.` (le volume à écraser sur '`nom_device`' comporte des données d'étiquettes non reconnues)

Les sauvegardes avec l'option `init` écrasent intégralement le contenu de la bande. Backup Server renvoie ce message si vous tentez d'effectuer la sauvegarde sur une nouvelle bande ou une bande contenant des données autres que Sybase, sans spécifier l'option `init`.

L'opérateur peut	En répondant
Annuler la sauvegarde	<code>sp_volchanged id_session, id_session, abort</code>
Monter un autre volume et retenter la sauvegarde	<code>sp_volchanged id_session, id_session, retry [, id_session [, id_session]]</code>
Procéder à la sauvegarde en écrasant le contenu de la bande	<code>sp_volchanged id_session, id_session, proceed [, id_session [, id_session]]</code>

Invites de changement de volume lors des chargements

Cette section présente les invites de changement de volume apparaissant lors des chargements et les actions que l'opérateur peut entreprendre.

- `Dumpfile 'fname' section vname found instead of 'fname' section vname.` (trouvé section `nom_volume` du fichier de sauvegarde `nom_fichier` au lieu de section `nom_volume` de `nom_fichier`)

Backup Server renvoie ce message s'il ne parvient pas à trouver le fichier spécifié sur un support mono-fichier.

L'opérateur peut	En répondant
Annuler le chargement	<code>sp_volchanged id_session, id_session, abort</code>
Monter un autre volume et tenter un nouveau chargement	<code>sp_volchanged id_session, id_session, retry [, id_session [, id_session]]</code>
Charger le fichier dans le volume en cours, même si ce n'est pas le fichier spécifié (non recommandé)	<code>sp_volchanged id_session, id_session, proceed [, id_session [, id_session]]</code>

- `Mount the next volume to read.` (montez le volume suivant à lire)

Backup Server renvoie ce message lorsqu'il est prêt à lire la section suivante d'un fichier de sauvegarde réparti sur plusieurs volumes.

L'opérateur peut	En répondant
Annuler le chargement	<code>sp_volchanged id_session, id_session, abort</code>
Monter le volume suivant et poursuivre le chargement	<code>sp_volchanged id_session, id_session, proceed [, id_session [, id_session]]</code>

- `Mount the next volume to search.`

Backup Server renvoie ce message s'il ne parvient pas à trouver le fichier spécifié sur un support multifichier.

L'opérateur peut	En répondant
Annuler le chargement	<code>sp_volchanged id_session, id_session, abort</code>
Monter un autre volume et poursuivre le chargement	<code>sp_volchanged id_session, id_session, proceed [, id_session [, id_session]]</code>

Restauration d'une base de données : instructions détaillées

Les pannes de disque se manifestent par des symptômes aussi divers que les causes. Si un seul bloc du disque est défectueux, la base de données peut, dans un premier temps, continuer à fonctionner parfaitement en apparence, sauf si vous exécutez fréquemment des commandes dbcc. Lorsque l'incident touche un contrôleur de disque ou un disque entier, vous n'êtes en revanche plus en mesure d'utiliser la base de données. Adaptive Server la marque alors comme suspecte et affiche un message d'avertissement. Si le disque contenant la base de données master tombe en panne, les utilisateurs ne peuvent plus se connecter au serveur et ceux qui y sont déjà connectés ne peuvent plus exécuter la moindre opération d'accès aux tables système de master.

En cas de panne sur votre device de base de données, Sybase conseille de procéder comme suit :

- 1 Procédez à une sauvegarde du journal de *chaque base de données se trouvant sur le device*.
- 2 Vérifiez l'utilisation de l'espace de *chaque base de données se trouvant sur le device*.
- 3 Une fois que vous avez collecté ces informations pour toutes les bases de données présentes, supprimez chacune d'elles.
- 4 Supprimez le device défectueux.
- 5 Initialisez de nouveaux devices.
- 6 Recréez une à une les bases de données.
- 7 Chargez la version de sauvegarde la plus récente dans chaque base de données.
- 8 Appliquez toutes les sauvegardes du journal de transactions dans leur ordre de création.

Toutes ces actions sont décrites en détail dans les sections ci-après.

Réalisation d'une sauvegarde courante du journal de transactions

Exécutez la commande `dump transaction with no_truncate` pour obtenir une sauvegarde courante du journal de transactions *pour chaque base de données se trouvant sur le device défectueux*. Par exemple, voici la commande pour obtenir une sauvegarde du journal de transactions courant de `mydb` :

```
dump transaction mydb
to "/dev/nrmt0" at REMOTE_BKP_SERVER
with init, no_truncate,
notify = "operator_console"
```

Contrôle de l'utilisation de l'espace

Les opérations ci-après permettent de déterminer les devices utilisés par votre base de données, l'espace alloué à chacun d'eux et si l'espace est alloué aux données, au journal ou aux deux. Vous pouvez utiliser ces informations lors de la recréation de vos bases de données, pour vous assurer que le journal, les données et les index résident sur des devices distincts et pour préserver la portée des segments utilisateur que vous avez créés.

Remarque Vous pouvez également effectuer ces opérations pour conserver les configurations de segment lors du transfert d'une sauvegarde de base de données d'un serveur vers un autre (sur la même plate-forme matérielle et logicielle).

Si vous n'utilisez pas ces informations afin de recréer les allocations de device pour les bases de données endommagées, *Adaptive Server modifie* la table `sysusages` après `load database` pour prendre en compte les anomalies. Cela signifie que les segments définis par le système et par l'utilisateur de la base de données ne coïncident plus avec les allocations de device appropriées. Des informations incorrectes dans `sysusages` peuvent engendrer le stockage du journal sur les mêmes devices que les données, même si le journal et les données étaient distincts avant la restauration. Cela peut également modifier de manière imprévisible les segments définis par l'utilisateur et résulter en une base de données qui ne pourrait pas être créée à l'aide de la commande standard `create database`.

Suivez les instructions ci-après pour examiner et enregistrer les allocations de device pour toutes les bases de données endommagées :

- 1 Dans master, utilisez la requête suivante pour examiner les allocations de devices et les devices utilisés par la base de données endommagée :

```
select segmap, size from sysusages
      où dbid = db_id("nom_basededonnées")
```

- 2 Examinez le résultat de la requête. Chaque ligne avec une valeur segmap de 3 signale une allocation de données ; une valeur segmap de 4 signale une allocation de journal. Les valeurs plus élevées caractérisent des segments définis par l'utilisateur ; considérez-les comme des allocations de données afin de conserver l'étendue de ces segments. La colonne size indique le nombre de blocs de données. Notez l'ordre, l'utilisation et la taille de chaque élément de disque.

Par exemple, ce résultat obtenu d'un serveur utilisant des pages logiques de 2 ko :

segmap	size
3	10240
3	5120
4	5120
8	1024
4	2048

correspond aux utilisations et aux tailles décrites dans le tableau 27-20.

Tableau 27-20 : Exemple d'allocation de device

Allocation de device	Méga-octets
Données	20
Données	10
Journal	10
Données (segment utilisateur)	2
Journal	4

Remarque Si la colonne segmap contient des 7, cela signifie que les données et le journal se trouvent sur le même device et que la reprise ne peut remonter que jusqu'à la dernière sauvegarde de la base de données. N'utilisez *pas* l'option log on pour créer une base de données à l'aide de create database. En revanche, allouez un espace égal (ou supérieur) au total indiqué par sysusages.

- 3 Exécutez `sp_helpdb nom_basedonnees` sur la base de données souhaitée. Cette requête répertorie les devices sur lesquels se trouvent les données et les journaux :

```

name          db_size owner  dbid   created
-----
mydb          46.0 MB sa      15     Apr  9 1991

status        device_fragments  size  usage
-----
no options set  datadev1          20 MB  data only
                datadev2          10 MB  data only
                datadev3           2 MB  data only
                logdev1           10 MB  log only
                logdev1            4 MB  log only

```

Suppression des bases de données

Une fois que vous avez exécuté ces actions *pour toutes les bases de données du device défectueux*, utilisez la commande `drop database` pour supprimer chacune d'elles.

Remarque Si certaines tables d'autres bases de données font référence à une table de la base de données que vous tentez de supprimer, vous devez d'abord supprimer les contraintes d'intégrité référentielle au moyen de la commande `alter table`.

Si le système génère des messages d'erreur rapportant que la base de données a été endommagée lorsque vous lancez la `drop database`, utilisez l'option `dropdb` de la commande `dbcc dbrepair` :

```
dbcc dbrepair (mydb, dropdb)
```

Reportez-vous au *Guide de dépannage* pour plus d'informations sur l'utilisation de `dbcc dbrepair`.

Suppression des devices défectueux

Après avoir supprimé chacune des bases de données, utilisez `sp_dropdevice` pour supprimer le device défectueux. Pour obtenir des informations complémentaires, reportez-vous au *Manuel de référence d'Adaptive Server Enterprise*.

Initialisation de nouveaux devices

Exécutez `disk init` pour initialiser les nouveaux devices de bases de données. Pour plus d'informations, reportez-vous au chapitre 16, "Initialisation des devices de base de données".

Recréation des bases de données

Exécutez les actions suivantes pour recréer chacune des bases de données à l'aide des informations sur les segments précédemment collectées.

Remarque Si vous avez choisi de ne pas collecter les informations sur l'utilisation des segments, exécutez la commande `create database for load` pour créer une nouvelle base de données au moins aussi grande que la base d'origine.

- 1 Exécutez la commande `create database` avec l'option `for load`. Dupliquez toutes les configurations des fragments de devices et les tailles de chaque ligne de votre base de données à partir de la table `sysusages`, *jusqu'au premier device de journal inclus*. Utilisez l'ordre des lignes de `sysusages`. (Les résultats de `sp_helpdb` sont classés en ordre alphabétique par nom de device et non dans l'ordre d'allocation.) Par exemple, pour recréer les allocations de la base de données `mydb` présentées dans le tableau 27-20, page 946, entrez la commande suivante :

```
create database mydb
  on datadev1 = 20,
  datadev2 = 10
log on logdev1 = 10
for load
```

Remarque La commande `create database... for load` verrouille temporairement la nouvelle base en empêchant les utilisateurs d'y accéder et `load database` désactive la base de données pour une utilisation ordinaire. Cela permet d'empêcher les utilisateurs d'effectuer des transactions journalisées pendant la restauration.

- 2 Exécutez la commande `alter database` avec l'option `for load` pour recréer, dans l'ordre, les entrées restantes. Considérez les allocations de devices pour les segments utilisateur comme des allocations de données.

La commande ci-dessous permet d'allouer de l'espace de données supplémentaire sur datadev3 et de l'espace de journalisation supplémentaire sur logdev1 :

```
alter database mydb
  on datadev3 = "2M"
log on logdev1= "4M"
for load
```

Chargement de la base de données

Rechargez la base de données à l'aide de load database. Si la base de données d'origine stockait des objets sur des segments définis par l'utilisateur (sysusages affiche un segmap supérieur à 7) et que vos nouvelles allocations de devices coïncident avec celles de la base de données sauvegardée, Adaptive Server conserve les configurations de segment utilisateur.

Si vous n'avez pas créé les allocations de devices afin qu'elles coïncident avec celles de la base de données sauvegardée, Adaptive Server reconfigure les segments en fonction des allocations de devices disponibles. Cette reconfiguration peut également placer le journal et les données sur le même device physique.

Remarque En cas de nouvel incident pendant le chargement d'une base de données, Adaptive Server ne récupère pas la base de données partiellement chargée et en informe l'utilisateur. Vous devez alors relancer la restauration de la base en répétant la commande load.

Chargement des journaux de transactions

Exécutez la commande load transaction pour appliquer les sauvegardes de journal de transactions *dans l'ordre dans lequel elles ont été effectuées*.

Adaptive Server contrôle l'estampille de chaque journal de transactions et base de données sauvegardés. Si les sauvegardes sont chargées dans un ordre incorrect ou si le journal de transactions a été modifié par des transactions entre deux chargements, la restauration échoue.

Si vous sauvegardez un journal de transaction à l'aide de `with standby_access`, vous devez également charger la base de données avec `with standby_access`.

Une fois la base de données mise à jour, utilisez les commandes `dbcc` pour contrôler sa cohérence.

Chargement d'un journal de transactions à un instant T

Vous pouvez restaurer une base de données jusqu'à un instant T spécifié dans son journal de transactions. Pour ce faire, exécutez l'option `until_time` de `load transaction`. Cette fonctionnalité est utile si, par exemple, un utilisateur supprime accidentellement une table importante. Vous pouvez utiliser l'option `until_time` pour récupérer toutes les modifications apportées à la base de données contenant la table avant la suppression de cette dernière.

Pour une utilisation efficace de l'option `until_time` après la suppression des données, vous devez connaître l'heure exacte à laquelle l'erreur s'est produite. Vous pouvez l'obtenir en exécutant une instruction `select getdate` au moment de l'erreur. Par exemple, supposez qu'un utilisateur supprime accidentellement une table importante, quelques minutes plus tard, vous obtenez l'heure courante en millisecondes :

```
select convert(char(26), getdate(), 109)
-----
Mar 26 1997 12:45:59:650PM
```

Après avoir sauvegardé le journal de transactions contenant l'erreur et chargé la dernière sauvegarde de la base de données, chargez les journaux de transactions créés après la dernière sauvegarde de la base de données. Ensuite, chargez le journal de transactions contenant l'erreur en utilisant `until_time` ; par exemple :

```
load transaction employees_db
from "/dev/nrmt5"
with until_time = "Mar 26 1997 12:35:59: 650PM"
```

Une fois que vous avez chargé un journal de transactions à l'aide de l'option `until_time`, Adaptive Server relance la séquence de journalisation sur la base de données. Cela signifie que jusqu'à la prochaine sauvegarde de la base de données, vous ne pourrez pas charger d'autres journaux de transactions après l'exécution de `load transaction` avec `until_time`. Vous devrez sauvegarder la base de données avant de pouvoir sauvegarder un autre journal de transactions.

Activation des bases de données

Dans cet exemple, le journal de transactions est chargé jusqu'à une heure précise précédant la suppression de la table. Après avoir appliqué toutes les sauvegardes du journal de transactions à une base de données, utilisez la commande `online database` afin qu'elle puisse être utilisée. La commande permettant d'activer la base de données `mydb` est la suivante :

```
online database mydb
```

Bases de données répliquées

Vous devez activer les bases de données répliquées avant de les mettre au niveau de la version courante d'Adaptive Server. Cependant, vous ne pouvez pas activer ces bases tant que les journaux ne sont pas vidés. Si vous tentez de le faire, Adaptive Server renvoie le message suivant :

```
Database is replicated, but the log is not yet drained. (la base de données est répliquée, mais le journal n'est pas encore vidé). This database will come online automatically after the log is drained. (cette base sera automatiquement activée une fois le journal vidé)
```

Lorsque Replication Server, via le gestionnaire de transfert de journal (LTM), vide le journal, la commande `online database` est exécutée automatiquement.

Mise à niveau à la version actuelle d'Adaptive Server

Reportez-vous au Guide d'installation pour votre plate-forme pour obtenir des instructions de mise à niveau pour les utilisateurs d'Adaptive Server disposant de bases de données répliquées.

Séquence de chargement

La séquence de chargement des bases de données répliquées est la suivante : `load database`, `replicate`, `load transaction`, `replicate`, etc. A la fin de la séquence de chargement, exécutez `online database` pour activer les bases de données. Les bases de données désactivées, car en cours de chargement, ne sont pas automatiquement activées par Replication Server.

Avertissement ! Il est primordial de ne pas exécuter la commande `online database` tant que tous les journaux de transactions ne sont pas chargés.

Chargement des sauvegardes de bases de données effectuées avec des versions antérieures

Lors de la mise à niveau d'un Adaptive Server avec une nouvelle version, toutes les bases de données associées à ce serveur sont mises à niveau automatiquement.

Par conséquent, il faut mettre à jour les sauvegardes de bases de données et de journaux de transactions créées avec une version précédente d'Adaptive Server avant de pouvoir les utiliser avec la version courante d'Adaptive Server.

Adaptive Server comporte une fonctionnalité de mise à niveau automatique, base par base, des sauvegardes de bases de données ou des journaux de transactions réalisés avec Backup Server vers la version courante d'Adaptive Server, qui rend les sauvegardes compatibles et utilisables. Cette caractéristique d'Adaptive Server ne requiert aucun programme externe. Elle offre la souplesse de la mise à niveau individuelle des sauvegardes en fonction des besoins.

Les tâches suivantes ne sont pas supportées par cette fonctionnalité de mise à niveau automatique :

- Chargement d'une ancienne version de la base de données master. Une fois la version d'Adaptive Server mise à jour, il est impossible de charger une sauvegarde de l'ancienne base de données master.
- Installation de procédures stockées nouvelles ou modifiées. Continuez à exécuter installmaster.
- Chargement et mise à niveau de sauvegardes générées avant SQL Server version 10.0.

Mise à niveau d'une sauvegarde vers Adaptive Server

Pour mettre à niveau une sauvegarde de base de données utilisateur ou un journal de transactions avec la version courante d'Adaptive Server, procédez comme suit :

- 1 Exécutez load database et load transaction pour charger la sauvegarde à mettre à niveau.

Adaptive Server détermine la version qu'il charge à partir de l'en-tête de la sauvegarde. Lorsque l'en-tête est lu et avant que Backup Server ne commence le chargement, la base de données est désactivée par les commandes `load database` ou `load transaction`. La base de données cesse donc d'être disponible pour une utilisation ordinaire (les requêtes et la commande `use basededonnées` ne sont pas autorisées) et fournit à l'utilisateur un plus grand contrôle sur les séquences de chargement et les risques que d'autres utilisateurs interrompent accidentellement une séquence de chargement disparaissent.

- 2 Exécutez la commande `online database` lorsque la sauvegarde a été restaurée pour activer le processus de mise à niveau.

Remarque N'exécutez *pas* la commande `online database` tant que toutes les sauvegardes de journaux de transactions ne sont pas chargées.

Dans les version antérieures à SQL Server version 11.0, une base de données était automatiquement disponible au terme d'une séquence de chargement réussie. Avec la version courante d'Adaptive Server, l'utilisateur doit activer la base de données après toute séquence de chargement à l'aide de la commande `online database`.

Pour les sauvegardes restaurées à partir de SQL Server version 10.0, la commande `online database` active le processus de mise à niveau des sauvegardes venant d'être chargées. Lorsque la mise à niveau a abouti, Adaptive Server active la base de données, qui devient donc accessible.

Pour les sauvegardes chargées à partir de la version courante d'Adaptive Server, aucun processus de mise à niveau n'est activé. Vous devez quand même exécuter la commande `online database` pour activer la base de données et la rendre accessible à tous les utilisateurs. La commande `load database` désactive la base de données.

Au cours de la mise à niveau, chaque étape du processus génère un message indiquant l'opération qui va être effectuée.

Si la mise à niveau échoue, la base de données reste inactive et un message est généré, indiquant que l'utilisateur doit y remédier.

Pour obtenir des informations complémentaires à propos de `online database`, reportez-vous au *Manuel de référence d'Adaptive Server Enterprise*.

- 3 Lorsque l'exécution de la commande `online database` a abouti, utilisez la commande `dump database`. La base de données doit être sauvegardée avant qu'une commande `dump transaction` ne soit autorisée. Cette dernière n'est pas autorisée sur une base de données nouvellement créée ou mise à niveau tant qu'une commande `dump database` n'a pas abouti.

Bit d'état *base de données désactivée*

Le bit d'état "base de données désactivée" (database offline) indique que la base de données n'est pas disponible pour une utilisation ordinaire. Vous pouvez déterminer si une base est désactivée à l'aide de la commande `sp_helpdb`. Elle indique que la base de données est désactivée si le bit d'état correspondant est défini.

Lorsqu'une base de données est désactivée par `load database`, un bit d'état est défini dans la table `sysdatabases` et il le reste jusqu'à l'aboutissement de la commande `online database`.

Le bit d'état "base de données désactivée" fonctionne conjointement avec les autres bits d'état existants. Il augmente le bit d'état ci-dessous afin de procurer un plus grand contrôle :

- En cours de restauration (in recovery)

Le bit d'état "base de données désactivée" remplace les bits d'état suivants :

- utilisation par le `dbo` seulement (DBO use only),
- lecture seule (read only).

Les bits d'état ci-dessous remplacent le bit d'état "base de données désactivée" :

- mise à niveau commencée (began upgrade),
- pas de restauration (bypass recovery),
- en cours de chargement (in load),
- non restaurée (not recovered),
- suspect (suspect),
- utilisation non restaurée (use not recovered).

Bien que la base de données ne soit pas accessible pour une utilisation ordinaire, les commandes ci-dessous sont autorisées lorsque la base est désactivée :

- dump database et dump transaction
- load database et load transaction
- alter database on device
- drop database
- online database
- diagnostics dbcc (soumis aux restrictions de dbcc).

Identificateurs de versions

La caractéristique de mise à niveau automatique fournit des identificateurs de version pour Adaptive Server, les bases de données et les formats des enregistrements du journal.

- L'identificateur de version de mise à niveau de configuration (configuration upgrade Version 10) indique la version courante d'Adaptive Server ; il est stocké dans la table système sysconfigures. sp_configure affiche la version courante d'Adaptive Server en tant que "upgrade version" (version de mise à niveau).
- L'identificateur de version de mise à jour indique la version courante d'une base de données ; il est stocké dans les en-têtes de la base de données et de la sauvegarde. Le mécanisme de reprise d'Adaptive Server utilise cette valeur pour déterminer si la base de données doit être mise à niveau avant d'être disponible pour une utilisation ordinaire.
- Spécificateur de version de compatibilité de journal. Cet identificateur différencie les journaux de la version 10.x des journaux de la version 11.x en affichant le format des enregistrements de journal d'une base de données, d'une sauvegarde de base de données ou de journal de transactions. Cette constante est stockée dans l'en-tête de la base de données et de la sauvegarde ; elle est utilisée par Adaptive Server pour détecter le format des enregistrements du journal pendant la reprise.

Liaisons de cache et chargement de bases de données

Vous devez connaître les liaisons de cache d'une base de données et de ses objets lorsque vous la sauvegardez et la chargez sur un serveur disposant de plusieurs liaisons. Vous pouvez décider de charger la base de données sur un autre serveur pour effectuer des travaux d'optimisation et de développement ou vous pouvez avoir besoin de charger une base de données qui a été supprimée d'un serveur dont les liaisons de cache ont changé depuis la sauvegarde.

Lorsqu'une base de données est activée après restauration ou à l'aide de la commande `online database` après un chargement, Adaptive Server contrôle toutes les liaisons de cache pour cette base de données et ses objets. S'il n'existe pas de cache, Adaptive Server consigne dans le journal d'erreurs un avertissement et la liaison dans `sysattributes` est marquée comme incorrecte. Ce message peut être le suivant :

```
Cache binding for database '5', object '208003772',  
index '3' is being marked invalid in Sysattributes.  
(la liaison de cache de la base de données 5, objet  
208003772, index 3 est marquée incorrecte dans  
Sysattributes)
```

Les liaisons de cache incorrectes ne sont pas supprimées. Si vous créez un cache de même nom et redémarrez Adaptive Server, la liaison est marquée comme correcte et le cache est utilisé. Si vous ne le faites pas, vous pouvez lier l'objet à un autre cache ou l'autoriser à utiliser le cache par défaut.

Dans les sections suivantes, qui traitent des liaisons de cache, *serveur de destination* (ou cible) fait référence au serveur vers lequel est chargée la base de données et *serveur d'origine*, au serveur sur lequel a été effectuée la sauvegarde.

Recréez si possible les caches ayant les mêmes noms sur le serveur de destination que les liaisons sur le serveur d'origine. Vous pouvez configurer les zones de la même manière si vous utilisez la base de données cible dans le même but ou dans le cadre de tests de performances ou de développement que vous voulez porter sur le serveur d'origine. Si vous utilisez la base de données cible comme base décisionnelle ou pour exécuter des commandes `dbcc`, vous pouvez configurer des pools, afin d'augmenter la quantité d'espace dans les pools de mémoire de 16 ko.

Bases de données et liaisons de cache

Les informations de liaison des bases de données sont stockées dans `master.sysattributes`. Aucune information de ce type n'est stockée dans la base elle-même. Si vous utilisez `load database` pour restaurer une sauvegarde sur une base de données existante liée à un cache et que vous ne supprimez pas la base de données avant d'exécuter la commande de chargement, la liaison n'est pas modifiée.

Si la base de données que vous chargez était liée à un cache sur le serveur d'origine, deux options s'offrent à vous :

- liez la base de données du serveur cible à un cache configuré pour répondre aux besoins de ce serveur ou
- configurez les zones du cache de données par défaut sur le serveur cible en fonction des besoins de l'application et ne liez pas la base de données à un cache de données nommé.

Objets de base de données et liaisons de cache

Les informations de liaison relatives aux objets sont stockées dans la table `sysattributes` de la base de données. Si vous chargez fréquemment la base sur le serveur cible, la solution la plus simple consiste à configurer des caches de même nom sur le serveur cible.

Si le serveur cible n'est pas configuré avec des caches de même nom que ceux du serveur d'origine, liez les objets aux caches appropriés sur le serveur cible après avoir activé la base de données ou assurez-vous que le cache par défaut est configuré de manière à répondre à vos besoins sur ce serveur.

Contrôle des liaisons de cache

La commande `Use sp_helpcache` affiche les liaisons de cache pour les objets d'une base de données même si elles sont incorrectes.

Les requêtes SQL suivantes reproduisent les commandes de liaison de cache à partir des informations de la table `sysattributes` d'une base de données utilisateur :

```
/* create a bindcache statement for tables */

select "sp_bindcache "+ char_value + ", "
      + db_name() + ", " + object_name(object)
from sysattributes
where class = 3
      and object_type = "T"

/* create a bindcache statement for indexes */

select "sp_bindcache "+ char_value + ", "
      + db_name() + ", " + i.name
from sysattributes, sysindexes i
where class = 3
      and object_type = "I"
      and i.indid = convert(tinyint, object_info1)
      and i.id = object
```

Contraintes entre bases de données et chargement de bases de données

Si vous utilisez la contrainte `references` de `create table` ou `alter database` pour référencer des tables dans des bases de données, vous pouvez rencontrer des problèmes si vous tentez de charger une sauvegarde d'une de ces bases de données.

- Si les tables d'une base de données référencent une base sauvegardée, des erreurs d'intégrité référentielle se produiront si vous restaurez la base de données sous un nom différent ou sur un serveur différent de celui sur lequel elle a été sauvegardée. Si vous voulez modifier le nom ou l'emplacement d'une base de données lorsque vous la restaurez, utilisez `alter table` dans la base de données référençante afin de supprimer toutes les contraintes d'intégrité référentielle externes avant de sauvegarder la base.
- Le chargement de la sauvegarde d'une base de données référencée antérieure à la base référençante pourrait poser des problèmes de cohérence ou endommager les données. Par précaution, chaque fois que vous ajoutez ou supprimez une contrainte portant sur plusieurs bases de données ou que vous supprimez une table contenant une contrainte de ce type, sauvegardez les deux bases concernées.

- Sauvegardez simultanément toutes les bases de données qui se référencent les unes les autres. Afin d'éviter des problèmes de synchronisation, mettez les deux bases de données en mode mono-utilisateur pour les sauvegardes. Lorsque vous chargez les bases de données, activez-les toutes les deux en même temps.

Les contraintes entre plusieurs bases de données peuvent créer des incohérences si vous :

- ne chargez pas les sauvegardes de bases de données dans l'ordre chronologique (par exemple, vous chargez une sauvegarde créée le 12 août 1997 après une sauvegarde créée le 13) ou
- chargez une sauvegarde dans une base de données avec un nouveau nom.

Si vous n'effectuez aucun chargement, les contraintes entre plusieurs bases de données peuvent devenir incohérentes.

Afin de remédier à ce problème, suivez les instructions ci-après :

- 1 Mettez les deux bases de données en mode mono-utilisateur.
- 2 Supprimez la contrainte référentielle incohérente.
- 3 Vérifiez la cohérence des données avec une requête telle que la requête ci-après :

```
select foreign_key_col from table1
where foreign_key not in
(select primary_key_col from otherdb..othertable)
```

- 4 Corrigez les problèmes d'incohérence de données.
- 5 Recréez la contrainte.

Restauration des bases de données système

Ce chapitre explique comment restaurer les bases de données master, model et sybssystemprocs.

Les sujets abordés dans ce chapitre sont les suivants :

Sujet	Page
En quoi consiste la restauration d'une base de données système ?	961
Symptômes indiquant que la base de données master est endommagée	962
Restauration de la base de données master	962
Restauration de la base model	977
Restauration de la base sybssystemprocs	979
Restauration des tables système à l'aide de disk reinit et de disk refit	981

En quoi consiste la restauration d'une base de données système ?

La procédure de restauration d'une base de données système dépend de la base elle-même et de la nature de l'incident. En général, elle comprend les étapes suivantes :

- exécution de load database pour charger les sauvegardes des bases de données concernées ;
- exécution de dataserver, installmaster et installmodel afin de restaurer l'état initial des bases de données concernées ou bien ;
- une combinaison de ces tâches.

Pour assurer la restauration des bases de données système la plus efficace possible :

- ne stockez, ni bases utilisateur, ni bases autres que master, tempdb et model sur le device master ;

- conservez toujours des impressions papier à jour des tables système importantes ;
- sauvegardez toujours la base master après l'initialisation des devices de base de données, la création ou la modification de bases ou l'ajout de login serveur.

Symptômes indiquant que la base de données *master* est endommagée

Une base de données master peut être endommagée suite à une erreur de support de stockage survenant dans la zone où la base master est stockée ou par corruption interne. Votre base de données master est endommagée si :

- Adaptive Server ne parvient pas à démarrer,
- vous observez des erreurs de stacks traces répétitifs ou des erreurs fréquentes,
- dbcc détecte toutes les erreurs à l'occasion du contrôle régulier de vos bases de données.

Restauration de la base de données *master*

La présente section décrit la procédure de restauration de la base de données master et de reconstruction du device master. Elle suppose ce qui suit :

- La base de données master est corrompue ou bien le device master est endommagé.
- Vous disposez d'un listing à jour des tables systèmes figurant dans la section "Sauvegarde de master et conservation de copies des tables système", page 26.
- Le device master contient *uniquement* les bases de données master, tempdb et model.

- Vous disposez d'une sauvegarde à jour de la base de données master et vous n'avez pas initialisé de devices, ni créé ou modifié de bases de données depuis la dernière sauvegarde de master.
- Votre serveur utilise l'ordre de tri par défaut.

Vous pouvez également utiliser ces procédures pour déplacer votre base de données master vers un device master de plus grande taille.

Le *Guide de dépannage* décrit plus en détail les scénarios de restauration.

A propos de la restauration

Du fait de leur nature indispensable et de leur rôle de contrôle, la base de données master et le device master nécessitent des procédures spéciales. Les tables de master configurent et contrôlent l'ensemble des fonctions, des bases de données et des devices de données d'Adaptive Server. La restauration :

- reconstruit le device master à l'état par défaut qui était le sien lors de l'installation initiale du serveur ;
- restaure la base de données master à son état par défaut ;
- restaure la base de données master à l'état qui était le sien lors de votre dernière sauvegarde.

Il n'est pas possible d'utiliser les procédures stockées dans le système pendant les phases initiales de la restauration de la base de données master.

Résumé de la procédure de restauration

Procédez comme suit pour restaurer un device master endommagé. *Les étapes sont décrites de manière plus approfondie dans les pages suivantes.*

Etape	Voir
Procurez-vous les copies papier des tables système requises pour restaurer les disques, les bases de données et les logins.	"Etape 1 : Se procurer des copies des tables système", page 965
Arrêtez Adaptive Server et exécutez dataserver pour construire une nouvelle base de données master et un device master.	"Etape 2 : Construction d'un nouveau device master", page 965

Etape	Voir
Redémarrez Adaptive Server en mode restauration de master.	"Etape 3 : Démarrage d'Adaptive Server en mode restauration de master.", page 966.
Recréez les allocations de la base de données master dans sysusages à l'identique.	"Etape 4 : Recréation des allocations de device pour master", page 967.
Mettez à jour le nom réseau de Backup Server dans la table syssservers.	"Etape 5 : Vérification des informations syssservers de Backup Server", page 972.
Vérifiez que Backup Server est actif.	"Etape 6 : Vérification de l'exécution de Backup Server", page 973.
Exécutez load database pour charger la sauvegarde la plus récente de la base de données master. Adaptive Server s'arrête automatiquement à la fin du chargement de master.	"Etape 7 : Chargement d'une sauvegarde de master", page 973.
Mettez à jour le paramètre de configuration number of devices dans le fichier de configuration.	"Etape 8 : Mise à jour du paramètre de configuration number of devices", page 974.
Redémarrez Adaptive Server en mode mono-utilisateur.	"Etape 9 : Redémarrage d'Adaptive Server en mode restauration de master", page 974.
Vérifiez que la sauvegarde de master dispose des informations les plus récentes des tables système.	"Etape 10 : Comparaison des tables système pour vérifier la sauvegarde courante de master", page 975.
Redémarrez Adaptive Server.	"Etape 11 : Redémarrage d'Adaptive Server", page 975.
Vérifiez syslogins si vous avez ajouté des logins depuis la dernière sauvegarde de master.	"Etape 12 : Restauration des ID utilisateur du serveur", page 975.
Restaurez la base de données model.	"Etape 13 : Restauration de la base model", page 976.
Comparez les exemplaires papier de sysusages et de sysdatabases à leur nouvelle version en ligne, exécutez dbcc checkalloc sur chaque base et examinez les tables importantes.	"Etape 14 : Vérification d'Adaptive Server", page 977.
Sauvegardez la base de données master.	"Etape 15 : Sauvegarde de master", page 977.

Etape 1 : Se procurer des copies des tables système

Procurez-vous des copies des tables systèmes enregistrées dans un fichier : sysdatabases, sysdevices, sysusages, sysloginroles et syslogins. Utilisez-les pour être certain que votre système aura été restauré dans son intégralité à la fin de ce processus.

Pour des informations sur la préparation de la restauration après incident par copie des tables système dans un fichier, reportez-vous à la section "Sauvegarde de master et conservation de copies des tables système", page 26.

Etape 2 : Construction d'un nouveau device master

Avant d'exécuter `dataserver`, vérifiez votre copie la plus récente de `sysusages`. Si `dbid 1` ne comporte qu'une seule ligne, le master n'est constituée que d'un fragment de base de données et vous pouvez passer à l'"Etape 5 : Vérification des informations `syssservers` de Backup Server", page 972.

Arrêtez Adaptive Server, s'il est en service, et reconstruisez le device master. Cette opération nécessite d'indiquer la taille du device.

Avant de commencer, n'oubliez pas ce qui suit :

- Utilisez un nouveau device et conservez l'ancien en cas de problèmes car celui-ci risque de contenir des informations cruciales.
- Arrêtez Adaptive Server avant d'exécuter une commande `dataserver` quelle qu'elle soit. Si vous exécutez `dataserver` sur un device master en cours d'utilisation par Adaptive Server, la procédure de restauration échouera lorsque vous tenterez de charger la sauvegarde la plus récente de master.

Exécutez `dataserver` (UNIX) ou `sqlsrvr` (Windows NT) pour construire un nouveau device master et installer une copie d'une base de données master "générique". Indiquez le nom et la taille complets de votre device master.

Remarque Vous devez attribuer à `dataserver` une taille égale ou supérieure à celle définie lors de la configuration d'origine d'Adaptive Server. Si sa taille est trop petite, vous recevrez des messages d'erreur lors du chargement de vos bases.

L'exemple suivant reconstruit un device master de 17 Mo.

Sur plates-formes UNIX :

```
dataserver -d /dev/rsd1f -b17M
```

Sous Windows NT :

```
sqlsrvr -d d:\devices\master.dat -b17M
```

Après l'exécution de `dataserver`, le mot de passe du compte "sa" par défaut devient NULL.

Pour plus d'informations sur l'utilitaire `dataserver`, reportez-vous au guide *Utilitaires*.

Etape 3 : Démarrage d'Adaptive Server en mode restauration de master.

Démarrez Adaptive Server en mode restauration de master en exécutant l'option `-m` (UNIX et Windows NT).

Sur les plates-formes UNIX, copiez le fichier `runserver` et nommez-le `m_RUN_nom_server`. Dans la ligne de commande `dataserver` du nouveau fichier, ajoutez le paramètre `-m`. Démarrez ensuite le serveur en mode restauration de master :

```
startserver -f ./RUN_nom_serveur
```

Sous Windows NT, démarrez Adaptive Server à partir de la ligne de commande en exécutant `sqlsrver`. Spécifiez le paramètre `-m` en plus des autres paramètres requis. Exemple :

```
sqlsrver.exe -dD:\Sybase\DATA\MASTER.dat -sPIANO -eD:\Sybase\install\errorlog  
-iD:\Sybase\ini -MD:\Sybase -m
```

Reportez-vous au guide *Utilitaires* pour la syntaxe complète de ces commandes.

Lors du démarrage d'Adaptive Server en mode restauration de master, seul le login de l'administrateur système est autorisé. Immédiatement après l'exécution d'une commande `dataserver` sur la base master, seul le compte "sa" existe et le mot de passe est NULL.

Avertissement ! Certains sites disposent de tâches automatiques qui se connectent au serveur au démarrage à l'aide du login "sa". Veillez à ce qu'elles soient désactivées.

Il faut passer par le mode restauration de master car la base master générique créée à l'aide de `dataserver` ne correspond pas à l'état d'Adaptive Server. Par exemple, la base ne connaît pas vos devices de base de données. Toute opération effectuée sur la base master risque de rendre la restauration impossible ou, dans le meilleurs des cas, beaucoup plus longue et compliquée.

Un Adaptive Server démarré en mode restauration de master est configuré automatiquement pour autoriser la mise à jour directe des tables système. Certaines autres opérations (le processus "checkpoint", par exemple) sont désactivées.

Avertissement ! Les modifications ad hoc apportées aux tables système sont dangereuses ; certaines peuvent empêcher l'exécution d'Adaptive Server. Apportez uniquement les modifications décrites dans le présent chapitre et toujours dans une transaction définie par l'utilisateur.

Etape 4 : Recréation des allocations de device pour *master*

Si votre copie de sauvegarde de la table `sysusages` contient plusieurs lignes pour `dbid 1`, vous devez augmenter la taille de master pour pouvoir en charger la sauvegarde. Il vous faut dupliquer la valeur de `vstart` de chaque allocation de master dans `sysusages`. C'est la méthode de copie la plus simple si vous disposez d'une copie de `sysusages` triée par la valeur `vstart`.

Dans les cas les plus simples, les allocations supplémentaires de master requièrent uniquement l'exécution de `alter database`. Dans les cas plus complexes, vous devez allouer de l'espace à d'autres bases de données pour reconstruire les valeurs de `vstart` exactes nécessaires à la reconstruction de master.

Outre la base de données master, `tempdb` (`dbid = 2`) et `model` (`dbid = 3`) se trouvent en totalité ou en partie sur le device master.

Détermination des allocations présentes sur le device master

Remarque Les exemples de la présente section supposent un serveur utilisant des pages logiques de 2 ko.

Pour déterminer les valeurs de vstart qui représentent des allocations sur le device master, examinez la table sysdevices. Elle indique les valeurs basse et haute pour chaque device. Les devices de base de données ont un cntrltype de 0 ; l'exemple suivant ne comprend pas les lignes correspondant aux devices à bandes.

Figure 28-1 : Détermination des allocations sur le device master

low	high	status	cntrltype	name	phyname	mirrorname
0	8703	3	0	master	d_master	NULL
16777216	16782335	2	0	sprocdev	/sybase/ sp_dev	NULL

**plage de pages
du device master**

Dans cet exemple, les numéros de page sur le device master vont de 0 à 8703. Par conséquent, toute allocation de base de données dont les valeurs sysusages.vstart sont comprises dans cette plage représentent des allocations sur le device master.

Vérifiez toutes les lignes concernant la base de données master à partir de la copie de la table système sysusages. Voici un exemple d'informations sysusages demandées par vstart :

Figure 28-2 : Exemple de résultat de sysusages

dbid	segmap	lstart	size	vstart	pad	unreservedpgs
1	7	0	1536	4	NULL	480
3	7	0	1024	1540	NULL	680
2	7	0	1024	2564	NULL	680
1	7	1536	1024	3588	NULL	1024
4	7	0	5120	4432	NULL	1560

dbid = 1, allocation supplémentaire pour la base master

taille de cette allocation

vstart compris entre 0 et 8703

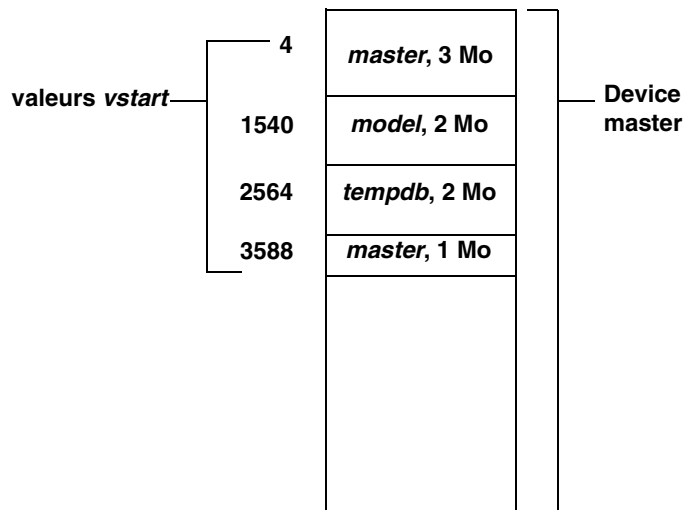
Dans cet exemple, les quatre premières lignes comportent des valeurs de vstart comprises entre 4 et 3588. Seul dbid 4 se trouve sur un autre device.

dataserver recrée les trois premières lignes afin que, dans votre master nouvellement reconstruite, sysusages corresponde à la copie papier en votre possession.

La quatrième ligne comporte une allocation supplémentaire pour master avec *vstart* = 3588 et *size* = 1024.

La figure 28-3 représente les allocations de stockage pour les données sysusages ci-dessus.

Figure 28-3 : Allocations sur un device master



Dans la figure 28-3, il suffit d'exécuter la commande `alter database` pour augmenter la taille de la base master. Pour déterminer la taille à fournir pour cette commande, consultez la colonne *size* de la seconde allocation de master et divisez par 512. Dans cet exemple, la ligne supplémentaire de master indiquant une allocation de 1024 pages de données, le paramètre correct est 2, résultat de la division de 1024 par 512.

Utilisez ce résultat pour la commande `alter database`. Connectez-vous au serveur en tant que "sa". N'oubliez pas que dataserver a défini pour ce compte le mot de passe NULL. Exécutez la commande `alter database`. Pour l'exemple ci-dessus, utilisez :

```
alter database master on master = "2M"
```

Vérifiez les valeurs de *size* et *vstart* pour la nouvelle ligne de sysusages.

Création d'allocations supplémentaires

Le résultat de sysusages peut avoir d'autres allocations sur le device master si :

- vous avez mis à niveau la version d'Adaptive Server ;
- Un administrateur système a augmenté la taille de master, model ou tempdb sur le device master.

Vous devez restaurer ces allocations jusqu'à la dernière ligne de sysusages pour la base de données master, dbid 1. Voici un exemple de sysusages indiquant d'autres allocations sur le device master dans l'ordre de vstart :

Figure 28-4 : Exemple de résultat de sysusages avec d'autres allocations

dbid	segmap	lstart	size	vstart	pad	unreservedpgs
1	7	0	1536	4	NULL	80
3	7	0	1024	1540	NULL	632
2	7	0	1024	2564	NULL	624
1	7	1536	1024	3588	NULL	1016
2	7	2560	512	4612	NULL	512
1	7	1024	1024	5124	NULL	1024
4	7	0	14336	33554432	NULL	13944
5	3	0	1024	50331648	NULL	632
5	4	1024	1024	67108864	NULL	1024
6	7	0	1024	83886080	NULL	632

dbid = 1, allocation supplémentaire pour la base master

vstart entre 0 et 8703

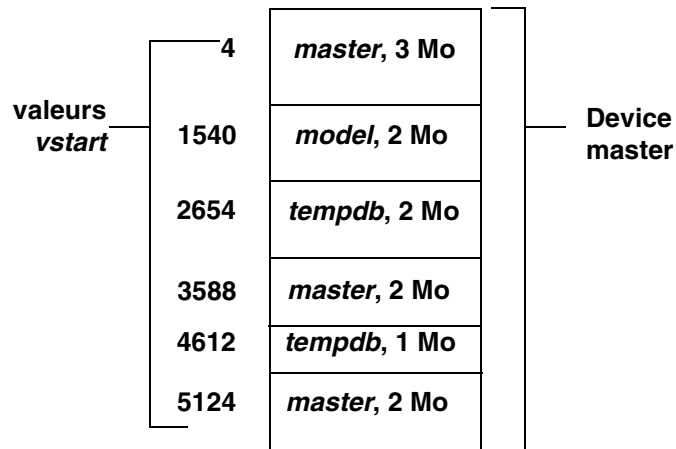
Cette copie de sysusages représente les allocations sur le device master suivantes (à l'exception des trois créées par dataserver) :

- une pour la base master, dbid = 1, size = 1024, vstart = 3588 ;
- une pour la base tempdb, dbid = 2, size = 512, vstart = 4612 ;
- une autre pour la base master, dbid = 1, size = 1024, vstart = 5124.

Les autres allocations finales de ce résultat ne résident pas sur le device master.

La figure 28-5 montre les allocations sur le device master.

Figure 28-5 : Allocations complexes sur un device master



Vous devez exécuter plusieurs commandes `alter database` et `create database` pour recréer toutes les allocations avec les tailles et les valeurs `vstart` correctes. Si votre copie de `sysusages` contient des allocations supplémentaires sur le device master après la dernière entrée concernant la base de données master, il n'est pas nécessaire de les recréer.

Pour déterminer la taille requise par les commandes `create database` et `alter database` divisez la valeur figurant dans la colonne `size` de `sysusages` par 512.

Par exemple, exécutez ces commandes dans l'ordre suivant :

- Pour restaurer la première allocation de master, `dbid 1`, `size = 1024` :

```
alter database master on default = "2M"
```
- Pour allouer davantage d'espace à tempdb, `dbid 2`, `size = 512` :

```
alter database tempdb on default = "1M"
```
- Pour ajouter l'allocation finale à master, `dbid 1`, `size = 1024` :

```
alter database master on default = "1M"
```

Vous ne devez restaurer que les allocations allant jusqu'à la dernière ligne (comprise) de la base de données master. Lorsque vous chargerez la sauvegarde de master, `sysusages` sera entièrement restaurée à partir de la sauvegarde.

A ce stade, comparez soigneusement les valeurs existantes de `sysusages` à celles de votre copie de sauvegarde :

- Si l'ensemble des valeurs de `vstart` et de `size` de master sont identiques, passez à l'"Etape 5 : Vérification des informations `syssservers` de Backup Server", page 972.
- Si ces valeurs diffèrent, il est probable que le chargement de la base master échouera. Arrêtez le serveur et recommencez en exécutant `dataserver`. Reportez-vous à la section "Etape 2 : Construction d'un nouveau device master", page 965.
- Si les valeurs de `sysusages` vous paraissent correctes, passez à l'"Etape 5 : Vérification des informations `syssservers` de Backup Server", page 972.

Etape 5 : Vérification des informations `syssservers` de Backup Server

Connectez-vous au serveur en tant que "sa" avec le mot de passe null.

Si le nom réseau de votre Backup Server n'est pas `SYB_BACKUP`, vous devez mettre à jour `syssservers` afin qu'Adaptive Server puisse communiquer avec son Backup Server. Vérifiez le nom du Backup Server dans votre fichier interfaces et exécutez la requête suivante :

```
select *
from syssservers
where srvname = "SYB_BACKUP"
```

Vérifiez la colonne `srvnetname` dans le résultat de cette requête. S'il correspond à l'entrée du fichier d'interfaces pour le Backup Server de votre serveur, passez à l'"Etape 6 : Vérification de l'exécution de Backup Server", page 973.

Si la colonne `srvnetname` n'est *pas* identique, vous devez mettre à jour `syssservers`. L'exemple ci-dessous passe le nom réseau du Backup Server à `PRODUCTION_BSRV` :

```
begin transaction
update syssservers
set srvnetname = "PRODUCTION_BSRV"
where srvname = "SYB_BACKUP"
```

Exécutez cette commande et assurez-vous qu'elle n'a modifié qu'une seule ligne. Exécutez à nouveau la commande `select`. Vérifiez que la bonne ligne a été modifiée et qu'elle contient la valeur correcte. Si l'`update` a modifié plusieurs lignes ou la mauvaise ligne, exécutez la commande `rollback transaction` et procédez à une nouvelle tentative de mise à jour.

Si la modification est correcte, exécutez la commande `commit transaction`.

Etape 6 : Vérification de l'exécution de Backup Server

Sur les plates-formes UNIX, exécutez la commande `showserver` pour vérifier que votre Backup Server est en cours d'exécution et redémarrez-le si nécessaire. Voir `showserver` et `startserver` dans le guide *Utilitaires*.

Sous Windows NT, Sybase Central installé localement ou le Gestionnaire de services permettent de vérifier si le Backup Server est démarré.

Pour plus d'informations sur les commandes de démarrage de Backup Server, reportez-vous au guide *Utilitaires*.

Etape 7 : Chargement d'une sauvegarde de *master*

Chargez la version de sauvegarde la plus récente de la base *master*. Voici des exemples de commandes de chargement :

Sur plates-formes UNIX :

```
load database master from "/dev/nrmt4"
```

Sous Windows NT :

```
load database master from "\\.\TAPE0"
```

Pour plus d'informations sur la syntaxe de ces commandes, reportez-vous au chapitre 27, "Sauvegarde et restauration de bases de données utilisateur".

Lorsque l'exécution de `load database` s'est accomplie avec succès, Adaptive Server s'arrête. Soyez attentifs aux éventuels messages d'erreur pendant la procédure de chargement et d'arrêt.

Etape 8 : Mise à jour du paramètre de configuration *number of devices*

Effectuez cette étape uniquement si vous utilisez un nombre de devices de base de données supérieur au nombre par défaut. Sinon, passez à l'"Etape 9 : Redémarrage d'Adaptive Server en mode restauration de master", page 974.

Les valeurs de configuration n'étant accessibles à Adaptive Server qu'après le recovery de la base master, vous devez fournir à Adaptive Server la valeur du paramètre *number of devices* à partir du fichier de configuration lu au démarrage.

Si votre fichier de configuration le plus récent n'est pas disponible, insérez la valeur correcte du paramètre *number of devices* dans le fichier de configuration.

Modifiez le serveur `runserver`. Ajoutez le paramètre `-c` à la fin de la commande `dataserver` ou `sqlsrvr` pour spécifier le nom et l'emplacement du fichier de configuration. Au démarrage, Adaptive Server lit les valeurs de paramètres dans le fichier spécifié.

Etape 9 : Redémarrage d'Adaptive Server en mode restauration de master

Exécutez `startserver` pour redémarrer Adaptive Server en mode restauration de master (voir "Etape 3 : Démarrage d'Adaptive Server en mode restauration de master.", page 966). Soyez attentif à l'affichage de messages d'erreur pendant la restauration.

Le chargement de la sauvegarde de master repasse le compte "sa" à son état antérieur et en restaure le mot de passe, s'il existe. Si vous avez exécuté `sp_locklogin` pour verrouiller ce compte avant la sauvegarde, il sera verrouillé. Effectuez les étapes restantes avec un compte bénéficiant du rôle d'administrateur système.

Etape 10 : Comparaison des tables système pour vérifier la sauvegarde courante de *master*

Si vous avez sauvegardé la base *master* depuis la dernière exécution de `disk init`, `create database` ou `alter database`, le contenu de `sysusages`, `sysdatabases` et `sysdevices` sera identique à celui de vos copies de sauvegarde.

Comparez les tables `sysusages`, `sysdatabases` et `sysdevices` du serveur restauré avec votre copie de sauvegarde. Soyez plus particulièrement attentif aux problèmes suivants :

- Si un ou plusieurs devices de votre copie de sauvegarde n'apparaissent pas dans la base `sysdevices` restaurée, cela signifie que vous avez ajouté des devices depuis votre dernière sauvegarde et que vous devez exécuter `disk reinit` et `disk refit`. Pour plus d'informations sur l'utilisation de ces commandes, reportez-vous à la section "Restauration des tables système à l'aide de `disk reinit` et de `disk refit`", page 981.
- Si une base de données figurant dans votre copie de sauvegarde n'apparaît pas dans la table `sysdatabases` restaurée, cela signifie que vous avez ajouté une base de données depuis la dernière sauvegarde de *master*. Vous devez exécuter `disk refit` (voir "Restauration des tables système à l'aide de `disk reinit` et de `disk refit`", page 981).

Etape 11 : Redémarrage d'Adaptive Server

Redémarrez Adaptive Server en mode normal (multi-utilisateur).

Etape 12 : Restauration des ID utilisateur du serveur

Comparez la copie de sauvegarde de la table `syslogins` à la table `syslogins` restaurée. Soyez particulièrement attentif aux situations suivantes et, si besoin, exécutez à nouveau les commandes suivantes :

- Si vous avez ajouté des logins serveur depuis la dernière sauvegarde de *master*, exécutez à nouveau les commandes `sp_addlogin` .
- Si vous avez supprimé des logins serveur, exécutez à nouveau les commandes `sp_droplogin`.

- Si vous avez verrouillé des comptes serveur, exécutez à nouveau les commandes `sp_locklogin`.
- Vérifiez la présence d'autres différences dues à l'utilisation de `sp_modifylogin` par des utilisateurs ou des administrateurs système.

Assurez-vous que les suids affectés aux utilisateurs sont corrects. Des valeurs `suid` divergentes dans les bases de données peuvent entraîner des problèmes d'autorisation et les utilisateurs risquent de ne pas pouvoir accéder aux tables ou exécuter des commandes.

Une technique efficace pour vérifier les valeurs existantes des suids est d'exécuter une union sur chaque table `sysusers` des bases de données utilisateur. Vous pouvez inclure `master` dans cette procédure si ces utilisateurs ont l'autorisation de l'utiliser.

Par exemple :

```
select suid, name from master..sysusers
union
select suid, name from sales..sysusers
union
select suid, name from parts..sysusers
union
select suid, name from accounting..sysusers
```

Si la liste de résultats indique que des valeurs `suid` ont été ignorées dans la plage où vous souhaitez refaire les logins, vous devez ajouter des marques de réservation pour ces valeurs puis les supprimer à l'aide de `sp_droplogin` ou les verrouiller à l'aide de `sp_locklogin`.

Etape 13 : Restauration de la base *model*

Restaurez la base de données `model` :

- Chargez votre sauvegarde de `model`, si vous en avez effectué une.
- En l'absence d'une sauvegarde :
 - exécutez le script `installmodel` :

Sur la plupart des plates-formes :

```
cd $SYBASE/scripts
isql -Usa -Pmot de passe -Snom_serveur < installmodel
```


Sous Windows NT :

```
cd $SYBASE/scripts
isql -Usa -Pmot de passe -Snom_serveur < instmodl
```

- Répétez les modifications éventuellement apportées à model.

Etape 14 : Vérification d'Adaptive Server

Vérifiez Adaptive Server soigneusement :

- 1 comparez votre copie de sauvegarde de sysusages à la nouvelle version en ligne ;
- 2 comparez votre copie de sauvegarde de sysdatabases à la nouvelle version en ligne ;
- 3 exécutez dbcc checkalloc sur chaque base de données ;
- 4 examinez les tables importantes de chaque base de données.

Avertissement ! Si vous découvrez des anomalies avec la table sysusages, appelez le support technique Sybase.

Etape 15 : Sauvegarde de *master*

Lorsque la restauration de la base master est terminée et après l'exécution de contrôles d'intégrité dbcc complets, sauvegardez la base à l'aide des commandes de sauvegarde que vous utilisez habituellement.

Restauration de la base *model*

La présente section décrit la restauration de la base model quand elle seule a besoin d'être restaurée. Elle fournit des consignes pour les scénarios suivants :

- Vous n'avez pas modifié model : il vous suffit de restaurer la base model générique.
- Vous avez modifié model et vous disposez d'une sauvegarde.
- Vous avez modifié model et vous ne disposez pas d'une sauvegarde.

Restauration de la base *model* générique

dataserver peut restaurer la base de données *model* sans incidence sur master.

Avertissement ! Arrêtez Adaptive Server avant d'exécuter une commande `dataserver` quelle qu'elle soit.

Sur plates-formes UNIX :

```
dataserver -d /devname -x
```

Sous Windows NT :

```
sqlsrvr -d nom_physique -x
```

Restauration de *model* à partir d'une sauvegarde

Si vous avez modifié le contenu de la base *model*, vous pouvez la restaurer à partir d'une sauvegarde à l'aide de `load database`.

Si vous n'êtes pas en mesure d'utiliser la base de données :

- 1 Suivez les consignes fournies dans "Restauration de la base *model* générique", page 978.
- 2 Si vous avez modifié la taille de *model*, exécutez à nouveau `alter database`.
- 3 Chargez la sauvegarde à l'aide de `load database`.

Restauration de *model* sans sauvegarde

Si vous avez modifié *model* sans disposer d'une sauvegarde.

- Suivez les étapes décrites pour "Restauration de la base *model* générique", page 978.
- Exécutez à nouveau toutes les commandes émises pour modifier *model*.

Restauration de la base *sybssystemprocs*

La base sybssystemprocs stocke les procédures système servant à modifier les tables systèmes et à établir les rapports les concernant. Si vos contrôles dbcc systématiques signalent des dommages alors que vous ne possédez pas de sauvegarde de la base, vous pouvez la restaurer à l'aide de installmaster. Si vous possédez des sauvegardes de sybssystemprocs, vous pouvez la restaurer avec load database.

Restauration de *sybssystemprocs* à l'aide d'*installmaster*

- 1 Recherchez le nom du device logique qui contient la base de données. Si vous pouvez toujours exécuter sp_helpdb, exécutez ce qui suit :

```

                sp_helpdb sybssystemprocs
name           db_size      owner          dbid
      created
      status
-----
sybssystemprocs      28.0 MB sa              4
      Aug 07, 1993
      trunc log on chkpt

device_fragments  size          usage          free kbytes
-----
sprocdev          28.0 MB      data and log    3120

```

La colonne "device_fragments" indique que la base de données est stockée sur sprocdev.

Si vous n'êtes pas en mesure d'exécuter sp_helpdb, la requête suivante retourne le nom des devices utilisés par la base de données sybssystemprocs ainsi que la et la quantité d'espace utilisée sur chacun d'entre eux :

```

select sysdevices.name, sysusages.size / 512
from sysdevices, sysdatabases, sysusages
where sysdatabases.name = "sybssystemprocs"
  and sysdatabases.dbid = sysusages.dbid
  and sysdevices.low <= sysusages.size + vstart
  and sysdevices.high >= sysusages.size + vstart -1
name
-----
sprocdev          28

```

- 2 Supprimez la base de données :

```
drop database sybsystemprocs
```

Si le disque est endommagé, exécutez `dbcc dbrepair` pour supprimer la base de données puis `sp_dropdevice` pour supprimer le device. Si nécessaire, exécutez `disk init` pour initialiser les nouveaux devices de base de données. Pour plus d'informations sur `disk init`, reportez-vous au chapitre 16, "Initialisation des devices de base de données".

- 3 Recréez la base sybsystemprocs sur le device en utilisant la taille renvoyée par la requête à l'étape 1 :

```
create database sybsystemprocs
on sprocdev = 28
```

Remarque La taille requise pour sybsystemprocs par votre système d'exploitation n'est peut-être pas la même. Reportez-vous au Guide d'installation pour votre plate-forme pour connaître la bonne taille.

- 4 Exécutez le script `installmaster`.

Avertissement ! L'exécution répétée d'*installmaster* risque de modifier la répartition des valeurs d'index de telle manière que la table `sysprocedures` nécessitera davantage d'espace disque pour stocker la même quantité de données. Pour éviter ce problème, supprimez et recréez sybsystemprocs avant d'exécuter *installmaster*.

Sur plates-formes UNIX :

```
cd $SYBASE/scripts
isql -Usa -Pmot de passe -Snom_serveur <
installmaster
```

Sous Windows NT :

```
cd $SYBASE/scripts
isql -Usa -Pmot de passe -Snom_serveur < instmstr
```

- 5 Si vous avez modifié les autorisations dans sybsystemprocs ou ajouté vos propres procédures à la base de données, vous devez recommencer ces modifications.

Restauration de *sybssystemprocs* à l'aide de *load database*

Si vous stockez dans *sybssystemprocs* des procédures système écrites par vous, vous avez le choix entre deux méthodes pour les restaurer si la base de données est endommagée :

- restaurez la base de données à partir d'*installmaster* comme décrit à l'étape 4 "Restauration de *sybssystemprocs* à l'aide d'*installmaster*", page 979. Recréez ensuite les procédures en exécutant à nouveau les commandes *create procedure*.
- Conservez des sauvegardes de la base de données et chargez-les à l'aide de *load database*.

Si vous décidez de conserver une sauvegarde de la base de données, assurez-vous qu'elle tient en entier sur un volume de bande ou que plusieurs Adaptive Server sont à même de communiquer avec votre Backup Server. Lorsqu'une sauvegarde occupe plusieurs volumes, il est nécessaire d'exécuter la commande de changement de volume *sp_volchanged* stockée dans *sybssystemprocs*. Vous ne pourrez pas exécuter cette commande si la restauration de la base de données.

Vous trouverez ci-après des exemples de commandes de chargement :

Sous UNIX :

```
load database sybssystemprocs from "/dev/nrmt4"
```

Sous Windows NT :

```
load database sybssystemprocs from "\\.\Tape0"
```

Restauration des tables système à l'aide de *disk reinit* et de *disk refit*

Lors d'une restauration de la base de données master à partir d'une sauvegarde ne contenant pas les informations stockées par l'exécution des commandes *disk init* ou *create database* ou *alter database* les plus récentes, suivez les procédures décrites ici pour restaurer les informations appropriées dans les tables *sysusages*, *sysdatabases* et *sysdevices*.

Restauration de *sysdevices* à l'aide de *disk reinit*

Si vous avez ajouté des devices de base de données depuis la dernière sauvegarde – autrement dit, si vous avez exécuté la commande *disk init* – vous devez de nouveau ajouter les devices manquants *sysdevices* à l'aide de *disk reinit*. Si vous avez sauvegardé les scripts de vos commandes *disk init* d'origine, servez-vous en pour définir les paramètres de *disk reinit* (dont la valeur initiale de *vstart*). Si la taille que vous indiquez est trop petite ou si vous utilisez une valeur *vstart* différente, vous risquez d'endommager votre base de données.

Si vous n'avez pas sauvegardé vos scripts *disk init*, consultez votre copie de sauvegarde la plus récente de *sysdevices* pour définir certains paramètres corrects de *disk reinit*. Il vous faudra encore déterminer la valeur de *vstart* si vous avez utilisé un *vstart* personnalisé dans la commande *disk init* d'origine.

Le tableau 28-1 décrit les paramètres de *disk reinit*, ainsi que les données de *sysdevices* correspondantes :

Tableau 28-1 : Utilisation de *sysdevices* pour définir les paramètres de *disk reinit*

paramètre <i>disk reinit</i>	données <i>sysdevices</i>	Notes
name	name	Utilisez le même nom, surtout si vos scripts créent ou modifient des bases de données ou ajoutent des segments.
physname	<i>nom_physique</i>	Doit être le chemin complet du device.
vdevno	<i>low</i> /16777216	Il n'est pas nécessaire d'utiliser la même valeur pour <i>vdevno</i> , mais veillez à ne pas vous servir d'une valeur déjà utilisée.
size	$(high-low) + 1$	Il est extrêmement important de fournir la taille correcte.

Vous pouvez également obtenir des informations sur les devices en recherchant leurs *nom*, *nom physique* et *vdevno* dans le journal d'erreurs et en vous servant des commandes du système d'exploitation pour en déterminer la taille.

Si vous stockez votre base de données *sybsystemprocs* sur un device physique distinct, n'oubliez pas d'inclure une commande *disk reinit* pour *sybsystemprocs* si elle n'apparaît pas dans *sysdevices*.

Après l'exécution de `disk reinit`, comparez votre table `sysdevices` à la copie que vous en avez effectuée avant d'exécuter `dataserver`.

`disk reinit` ne peut s'exécuter qu'à partir de la base de données master et seulement par un administrateur système. Ces autorisations ne peuvent pas être transférées. Sa syntaxe est la suivante :

```
disk reinit
  name = "nom_device",
  physname = "nom_physique",
  [vdevno = numéro_device_virtuel,]
  size = nombre_blocs
  [, vstart = adresse_virtuelle ,
  cntrltype = numéro_contrôleur]
```

Pour plus d'informations sur `disk reinit`, reportez-vous à la section qui lui est consacrée dans le chapitre 16, "Initialisation des devics de base de données", ou dans le *Manuel de référence d'Adaptive Server*.

Restauration de `sysusages` et de `sysdatabase` à l'aide de `disk refit`

Si vous avez ajouté des devics de base de données ou bien créé ou modifié des bases depuis la dernière sauvegarde, exécutez `disk refit` pour reconstruire les tables `sysusages` et `sysdatabases`.

`disk refit` ne peut s'exécuter qu'à partir de la base de données master et seulement par un administrateur système. Ces autorisations ne peuvent pas être transférées. Sa syntaxe est la suivante :

```
disk refit
```

Adaptive Server s'arrête automatiquement après que `disk refit` ait reconstruit les tables système. Examinez le résultat pendant l'exécution de `disk refit` et le processus d'arrêt afin de déterminer si des erreurs se sont produites.

Avertissement ! Des informations inexacts dans la commande `disk reinit` risquent d'endommager les données de manière permanente lors de leur mise à jour. N'oubliez pas de contrôler Adaptive Server à l'aide de `dbcc` après l'exécution de `disk refit`.

Gestion de l'espace libre avec des seuils

Lors de la création ou de la modification d'une base de données (par alter database), vous allouez une quantité d'espace déterminée pour ses segments de données et de journal. La création d'objets et l'insertion de données fait diminuer la quantité d'espace libre dans la base de données.

Ce chapitre explique comment utiliser des seuils pour contrôler la quantité d'espace libre dans un segment de base de données.

Les sujets abordés sont les suivants :

Sujet	Page
Contrôle d'espace libre avec le seuil ultime	986
Enregistrement d'annulations et seuil ultime	988
Seuil ultime et caches de journal utilisateur pour des segments de journal et de données partagés	993
Utilisation de alter database quand la base de données master atteint le seuil ultime	996
Annulation automatique ou suspension des processus	996
Reprise des processus suspendus	997
Ajout, modification et suppression de seuils	998
Création d'un seuil d'espace libre pour le segment de journal	1001
Création de seuils supplémentaires sur d'autres segments	1005
Création de procédures de seuil	1006
Désactivation de la comptabilisation de l'espace libre pour les segments de données	1012

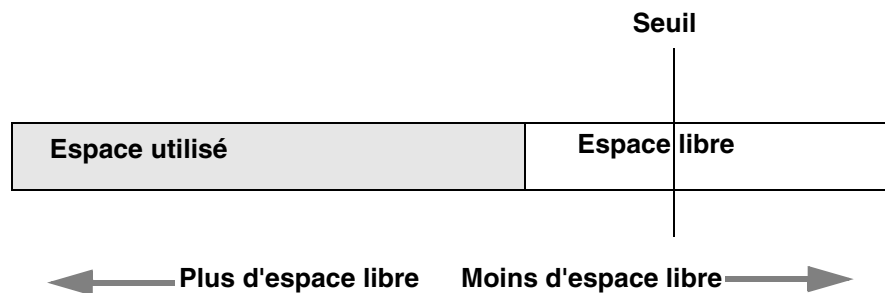
Contrôle d'espace libre avec le seuil ultime

Toutes les bases de données disposent d'un **seuil ultime**, y compris master. Ce seuil est une estimation du nombre de pages disponibles nécessaires pour sauvegarder le journal de transactions. Quand vous allouez plus d'espace au segment de journal, Adaptive Server règle automatiquement le seuil ultime.

Quand la quantité d'espace libre dans le segment de journal tombe sous le seuil ultime, Adaptive Server exécute automatiquement la procédure stockée spéciale `sp_thresholdaction`. (Vous pouvez spécifier une procédure de seuil ultime différente par `sp_modifythreshold`.)

La figure 29-1 illustre un segment de journal avec un seuil ultime. La zone ombrée représente l'espace de journalisation déjà utilisé ; la zone non ombrée représente l'espace de journalisation libre. Le seuil ultime n'a pas encore été dépassé.

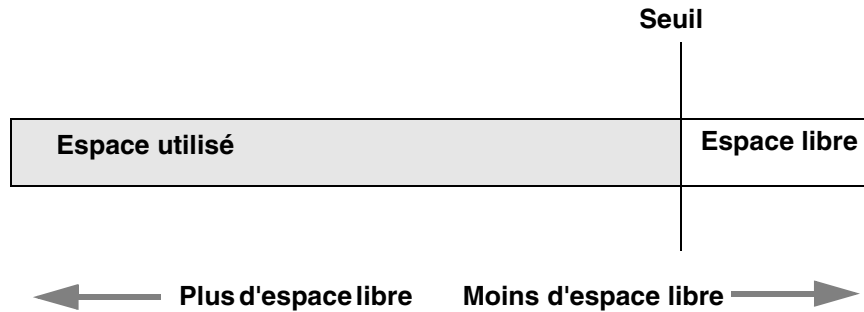
Figure 29-1 : Segment de journal avec un seuil ultime



Passage du seuil

Quand les utilisateurs exécutent des transactions, la quantité d'espace de journalisation libre diminue. Quand la quantité d'espace libre franchit le seuil ultime, Adaptive Server exécute `sp_thresholdaction` :

Figure 29-2 : Exécution de `sp_thresholdaction` au passage du seuil ultime



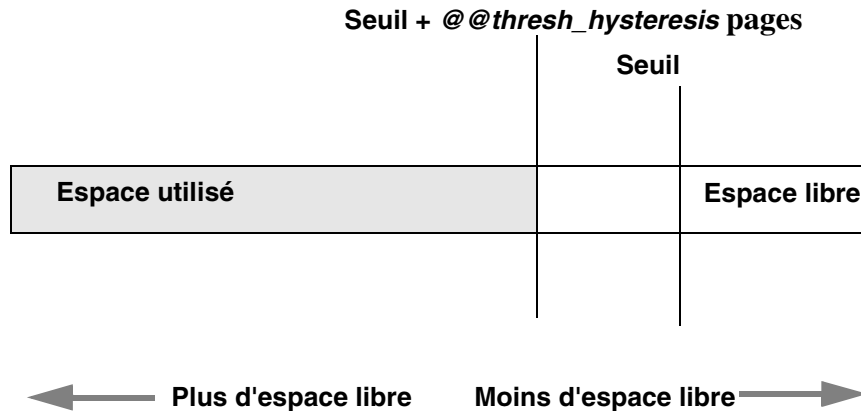
Contrôle de la fréquence d'exécution de `sp_thresholdaction`

Adaptive Server utilise une *valeur d'hystérésis*, la variable globale `@@thresh_hysteresis`, qui permet de contrôler le degré de sensibilité des seuils face aux variations de l'espace disponible.

Un seuil est désactivé après exécution de sa procédure, et reste inactif jusqu'à ce que la quantité d'espace libre dans le segment atteigne `@@thresh_hysteresis` pages au-dessus du seuil. Cette protection empêche les seuils d'exécuter leur procédure à chaque fluctuation mineure de l'espace disponible. Vous ne pouvez pas modifier la valeur `@@thresh_hysteresis`.

Par exemple, quand le passage du seuil de la figure 29-2 fait exécuter `sp_thresholdaction`, ce seuil est désactivé. A la figure 29-3, le seuil est réactivé quand la quantité d'espace libre augmente de la valeur de `@@thresh_hysteresis` :

Figure 29-3 : L'espace libre doit augmenter de @@thresh_hysteresis pour réactiver le seuil



Enregistrement d'annulations et seuil ultime

Adaptive Server version 11.9 et ultérieures utilise des **enregistrements d'annulation** dans les journaux de transactions. Les enregistrements d'annulation sont journalisés à chaque annulation d'une transaction. Les serveurs conservent un espace suffisant pour un enregistrement d'annulation correspondant à chaque mise à jour effectuée dans une transaction ouverte. Si une transaction s'achève normalement, aucun enregistrement d'annulation n'est journalisé et l'espace réservé pour eux est libéré.

Dans des transactions à exécution longue, les enregistrements d'annulation peuvent réserver de grandes quantités d'espace. Mais comme Adaptive Server n'alloue pas l'espace réservé pour les enregistrements d'annulation au journal de transactions, cet espace n'est pas mentionné par `sp_spaceused` ni `dbcc checktable`.

Si le seuil ultime du journal de transactions se déclenche alors qu'il semble y avoir suffisamment d'espace, c'est peut-être l'espace réservé pour les annulations qui crée le problème. Pour plus d'informations, reportez-vous à la section "Détermination de l'espace utilisé par les enregistrements d'annulation", page 990.

Calcul de l'espace pour les enregistrements d'annulation

Pour calculer le supplément d'espace à ajouter à un journal de transactions pour recevoir les enregistrements d'annulation, estimez :

- le nombre d'enregistrements de mise à jour dans le journal de transactions qui sont susceptibles d'appartenir à des transactions déjà annulées ;
- le nombre maximum d'enregistrements de mise à jour dans le journal de transactions qui sont susceptibles d'appartenir à des transactions ouvertes à chaque instant.

Les enregistrements de mise à jour sont ceux qui modifient la valeur timestamp. Ceci inclut les modifications de pages de données, de pages d'index, de pages d'allocation, etc.

Chaque enregistrement d'annulation nécessite environ 80 octets, ou 3 centièmes de page. Le calcul pour l'inclusion des enregistrements d'annulation (RR = rollback record) dans le journal de transactions est le suivant :

Espace à ajouter, en pages = (RR journalisés + nb mises à jour ouvertes) X 3/100

Vous pouvez aussi ajouter de l'espace de journalisation pour compenser l'effet des enregistrements d'annulation sur le seuil ultime et sur les seuils définis par l'utilisateur, comme indiqué dans les sections ci-dessous.

Utilisation de `lct_admin` pour déterminer l'espace de journalisation libre

Utilisez `logsegment_freepages` pour connaître la quantité d'espace libre de votre segment de journal dédié. Respectez la syntaxe suivante :

```
lct_admin("logsegment_freepages", dbid)
```

par exemple, pour connaître la quantité de pages libres pour les segments de journal de la base de données `pubs2` :

```
select lct_admin("logsegment_freepages", 4
```

Détermination de l'espace utilisé par les enregistrements d'annulation

Pour déterminer le nombre de pages réservées pour les annulations par une base de données, utilisez `lct_admin` avec le paramètre `reserved_for_rollback`. La syntaxe partielle de `lct_admin` est la suivante :

```
select lct_admin ("reserved_for_rollback", dbid, 0)
```

Le nombre renvoyé est le nombre de pages réservées, mais pas encore allouées aux enregistrements d'annulation.

Par exemple, pour connaître le nombre de pages réservées aux enregistrements d'annulation dans la base de données `pubs2` (dont le `pubid` est 5), utilisez la commande suivante :

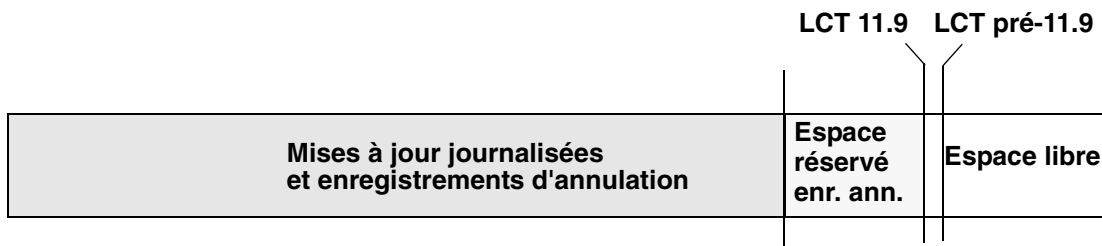
```
select lct_admin("reserved_for_rollback", 5, 0)
```

Pour plus d'informations sur `lct_admin`, reportez-vous au *Manuel de référence d'Adaptive Server*.

Effet des enregistrements d'annulation sur le seuil ultime

Adaptive Server doit réserver de la place supplémentaire pour le seuil ultime quand il utilise des enregistrements d'annulation. Le seuil ultime ou LCT (last-chance threshold) peut aussi être atteint plus tôt à cause de l'espace utilisé par des enregistrements d'annulation déjà journalisés et de l'espace réservé pour les enregistrements d'annulation éventuels des transactions ouvertes. La figure 29-4 illustre l'utilisation de l'espace dans un journal de transactions avec des enregistrements d'annulation :

Figure 29-4 : Espace utilisé dans un journal avec enregistrements d'annulation



Dans la figure 29-4, dans Adaptive Server 11.9, l'espace de journalisation est occupé par les enregistrements d'annulation journalisés pour les transactions fermées qui ne se sont pas terminées correctement. De plus, de l'espace est réservé pour les enregistrements d'annulation qui peuvent avoir à être journalisés, si une des transactions ouvertes ne se termine pas normalement. L'espace total pour les enregistrements d'annulation journalisés et les enregistrements d'annulation potentiels peut devenir nettement plus grand que l'espace supplémentaire pour le seuil ultime. En conséquence, les journaux de transactions qui utilisent des enregistrements d'annulation atteignent le seuil ultime nettement plus tôt que les journaux de transactions qui n'en utilisent pas, même si la taille du journal de transactions n'est pas augmentée.

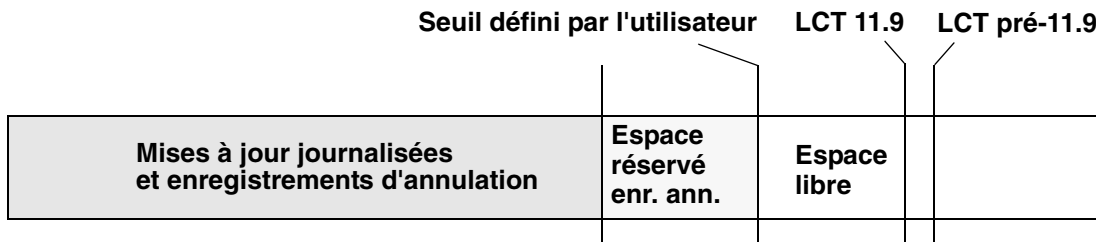
En général, les versions d'Adaptive Server 11.9 et ultérieures réservent 18 pour cent d'espace de journalisation en plus pour le seuil ultime que les versions précédentes. Par exemple, pour un journal de transactions de 5000 pages, la version 11.5 réserve 264 pages et la version 11.9 réserve 312 pages. C'est une augmentation de 18,18 pour cent entre les versions 11.5 et 11.9.

Seuils définis par l'utilisateur

Les enregistrements d'annulation occupent de l'espace supplémentaire dans le journal de transactions, il y a donc moins d'espace libre après le seuil défini par l'utilisateur pour effectuer une sauvegarde que dans les versions d'Adaptive Server qui n'utilisent pas d'enregistrements d'annulation (voir figure 29-4). Mais la perte d'espace pour sauvegarde, conséquence de l'augmentation du seuil ultime est sans doute plus que compensée par l'espace réservé pour les enregistrements d'annulation des transactions ouvertes.

La mise à niveau vers une version qui utilise des enregistrements d'annulation modifie les seuils définis par l'utilisateur de façon comparable à son influence sur les seuils ultimes. La figure 29-5 illustre l'effet d'une mise à niveau vers un Adaptive Server utilisant des enregistrements d'annulation sur un seuil défini par l'utilisateur :

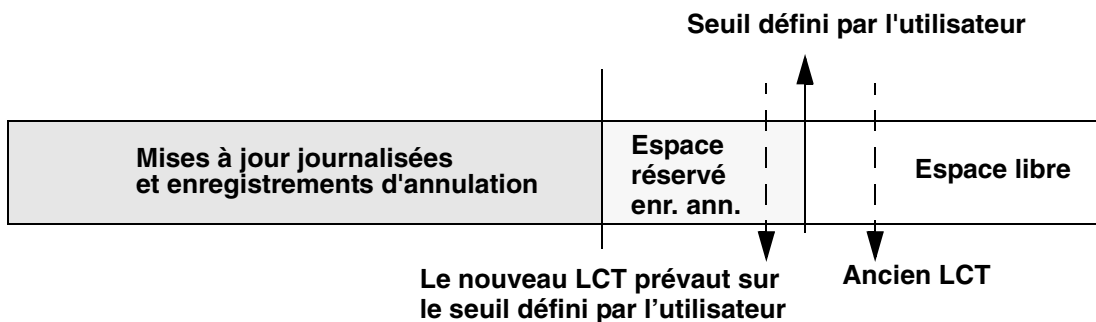
Figure 29-5 : Effet d'une mise à niveau sur des seuils définis par l'utilisateur



Un seuil défini par l'utilisateur tel que celui de la figure 29-5 est souvent utilisé pour exécuter la commande dump transaction. Le seuil est défini de façon à conserver suffisamment de place pour achever la sauvegarde avant d'atteindre le seuil ultime et avant de suspendre toutes les transactions ouvertes dans le journal.

Dans des bases de données qui mélangent l'espace de journalisation et l'espace de données, le seuil ultime se déplace dynamiquement, sa valeur peut être configurée automatiquement pour être inférieure au seuil défini par l'utilisateur. Dans ce cas, le seuil défini par l'utilisateur est désactivé, et le seuil ultime se déclenche avant que le seuil défini par l'utilisateur soit atteint, comme indiqué à la figure 29-6 :

Figure 29-6 : Déclenchement du seuil ultime avant le seuil défini par l'utilisateur



Le seuil défini par l'utilisateur est réactivé si la valeur du seuil ultime est configurée à une valeur supérieure à celle du seuil défini par l'utilisateur (par exemple si le seuil ultime est reconfiguré avec la valeur "ancien LCT" à la figure 29-6).

Dans les bases de données à segment de journal séparé, le journal dispose d'un espace dédié et le seuil ultime est statique. Le seuil défini par l'utilisateur n'est pas modifié par le seuil ultime.

Seuil ultime et caches de journal utilisateur pour des segments de journal et de données partagés

Chaque base de données d'un serveur Adaptive Server dispose d'un seuil ultime et toute transaction est mise en buffer dans un cache de journal utilisateur. Lors de la création d'une base de données avec segments partagés de données et de journal, le seuil ultime dépend de la taille de la base de données model. Dès le début de l'ajout de données et le démarrage de l'activité de journalisation, le seuil ultime est recalculé de façon dynamique en fonction de l'espace disponible et des transactions ouvertes. Le seuil ultime d'une base de données avec segments de journal et de données séparés dépend de la taille du segment de journal et ne varie pas dynamiquement.

Pour connaître le seuil ultime d'une base de données, vous pouvez utiliser `lct_admin` avec le paramètre `reserve` et une spécification de 0 page de journal :

```
select lct_admin("reserve",0)
```

Le seuil ultime pour une base de données est enregistré dans la table `systhresholds`, aussi accessible par `sp_helpthreshold`. Mais n'oubliez pas que :

- `sp_helpthreshold` renvoie les seuils définis par l'utilisateur et d'autres données, en plus de la valeur à jour du seuil ultime. L'utilisation de `lct_admin` est plus simple si vous n'avez besoin que de la valeur du seuil ultime. L'une ou l'autre de ces valeurs donne la valeur en cours du seuil ultime.
- Pour une base de données avec segments partagés de données et de journal, la valeur du seuil ultime dans `systhresholds` peut *ne pas* être égale à la valeur en cours du seuil ultime.

Le franchissement du seuil ultime suspend les transactions

Le comportement par défaut d'Adaptive Server est de suspendre les transactions ouvertes jusqu'à la création d'un espace de journalisation supplémentaire. Les transactions suspendues à cause du seuil ultime peuvent être interrompues par le paramètre abort de la fonction système `lct_admin`, décrite dans la section "Utilisation de `lct_admin abort` pour annuler les transactions suspendues" ci-dessous. Pour plus d'informations sur la configuration d'Adaptive Server pour annuler automatiquement les processus suspendus, reportez-vous à la section "Annulation automatique ou suspension des processus", page 996.

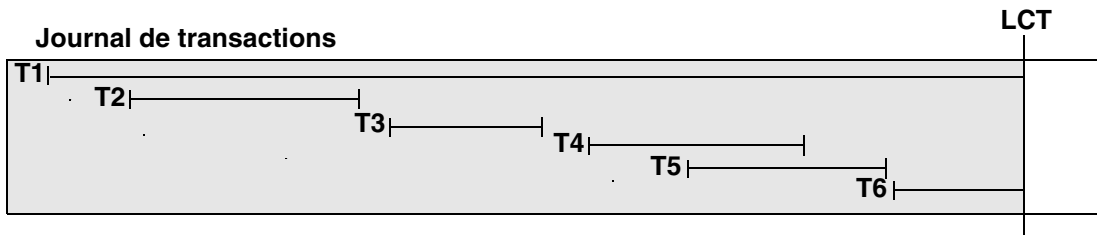
Utilisation de `lct_admin abort` pour annuler les transactions suspendues

Le plus souvent, la sauvegarde du journal de transactions crée de l'espace, parce qu'elle supprime les transactions validées au début du journal. Mais si une ou plusieurs transactions du début du journal sont toujours ouvertes, ceci empêche la sauvegarde et donc la troncature du journal de transactions.

Utilisez `lct_admin abort` pour interrompre les transactions suspendues qui empêchent la sauvegarde du journal de transactions. L'interruption d'une transaction

la ferme, donc ceci autorise la poursuite de la sauvegarde. La figure 29-7 illustre un scénario d'utilisation possible pour `lct_admin abort` :

Figure 29-7 : Exemple d'utilisation de `lct_admin abort`



A la figure 29-7, un journal de transactions a atteint son seuil ultime, et les transactions ouvertes T1 et T6 sont suspendues. T1 est au début du journal, donc elle empêche une sauvegarde de supprimer les transactions fermées T3 à T5 pour créer de l'espace qui permettrait de poursuivre la journalisation. L'interruption de T1 par `lct_admin abort` permet de fermer T1 pour qu'une sauvegarde permette d'effacer du journal les transactions T1 à T5.

`lct_admin abort` remplace `lct_admin unsuspend`.

Syntaxe de `lct_admin abort`

La syntaxe de `lct_admin abort` est :

```
lct_admin("abort", {spid [, dbid]})
```

Pour annuler une transaction, il faut déterminer son ID. Reportez-vous à la section "Obtention du spid de la transaction ouverte la plus ancienne" ci-dessous pour plus d'informations sur la détermination du spid de la transaction.

Pour interrompre la transaction la plus ancienne, tapez l'ID (spid) du processus qui a commencé la transaction. Ceci interrompt aussi toute autre transaction suspendue dans le journal qui appartient au processus spécifié.

Par exemple, si le processus 83 détient la transaction ouverte la plus ancienne dans un journal suspendu, pour interrompre la transaction, tapez :

```
select lct_admin("abort", 83)
```

Ceci interrompt aussi toute autre transaction ouverte appartenant au processus 83 dans le même journal de transactions.

Pour interrompre toutes les transactions ouvertes dans le journal, tapez :

```
select lct_admin("abort", 0, 12)
```

Obtention du spid de la transaction ouverte la plus ancienne

Utilisez la requête ci-dessous pour connaître le spid de la transaction ouverte la plus ancienne dans un journal de transactions qui a atteint son seuil ultime :

```
use master
go
select dbid, spid from syslogshold
where dbid = db_id("nom_base_de_données")
```

Par exemple, pour connaître la transaction en cours la plus ancienne sur la base de données pubs2 :

```
select dbid, spid from syslogshold
where dbid = db_id("pubs2")
dbid    spid
-----
       7      1
```

Utilisation de *alter database* quand la base de données master atteint le seuil ultime

Quand le seuil ultime de la base de données master est atteint, vous pouvez utiliser *alter database* pour ajouter de l'espace au journal de transactions de la base de données master. Ceci autorise plus d'activités sur le serveur en réactivant les transactions suspendues dans le journal. Mais tant que le journal de transactions de master est à son seuil ultime, vous ne pouvez pas utiliser *alter database* pour faire des modifications dans d'autres bases. Donc, si le seuil ultime est atteint en même temps par la base de données master et une autre base, vous devez d'abord utiliser *alter database* pour ajouter de l'espace de journalisation sur la base master, avant de l'utiliser à nouveau pour ajouter de l'espace de journalisation sur la deuxième base de données.

Annulation automatique ou suspension des processus

Par définition, le seuil ultime conserve suffisamment d'espace de journalisation libre pour enregistrer une commande *dump transaction*. Il est possible qu'il n'y ait pas assez d'espace pour enregistrer d'autres transactions utilisateur sur la base de données.

Quand le seuil ultime est franchi, Adaptive Server suspend les processus utilisateur et affiche le message :

```
Space available in the log segment has fallen
critically low in database 'mydb'. All future
modifications to this database will be suspended
until the log is successfully dumped and space
becomes available.
```

Seules les commandes non enregistrées dans le journal de transactions (select ou readtext) ainsi que celles nécessaires à la libération d'espace supplémentaire de journalisation (dump transaction, dump database et alter database) sont exécutables.

Utilisation de *abort tran on log full* pour annuler les transactions

Pour configurer le seuil ultime de façon à annuler automatiquement les transactions ouvertes, plutôt que de les suspendre :

```
sp_dboption nom_base "abort tran on log full", true
```

En cas de mise à niveau depuis une version précédente d'Adaptive Server, le serveur mis à niveau conserve le paramètre abort tran on log full.

Reprise des processus suspendus

Dès que la commande dump transaction libère suffisamment d'espace de journalisation, les processus suspendus sont automatiquement réactivés. Si writetext ou select into a effectué des modifications non journalisées sur la base de données depuis la dernière sauvegarde, la procédure du seuil ultime ne peut pas exécuter la commande dump transaction. Dans ce cas, effectuez une copie de la base de données avec dump database, puis tronquez le journal avec dump transaction.

Si cette opération ne libère pas suffisamment d'espace pour reprendre les processus suspendus, augmentez la taille du journal de transactions. Pour ce faire, utilisez l'option log on de la commande alter database.

En dernier recours, les administrateurs système peuvent utiliser la commande sp_who, afin de déterminer les processus dont l'état d'exécution est suspendus (log suspend) avant d'utiliser la commande kill pour les supprimer.

Ajout, modification et suppression de seuils

Le propriétaire de la base de données ou l'administrateur système peuvent créer des seuils supplémentaires pour contrôler l'espace libre sur n'importe quel segment de la base de données. Les seuils supplémentaires s'appellent **seuils d'espace libre**. Chaque base de données peut comporter jusqu'à 256 seuils, seuil ultime inclus.

`sp_addthreshold`, `sp_modifythreshold` et `sp_dropthreshold` permettent de créer, de modifier et de supprimer des seuils. Pour empêcher les utilisateurs de modifier par inadvertance les seuils d'une autre base de données, ces procédures imposent la spécification du nom de la base de données en cours.

Affichage d'informations sur les seuils existants

Utilisez `sp_helpthreshold` pour obtenir des informations sur tous les seuils d'une base de données. Utilisez `sp_helpthreshold nom_segment` pour obtenir des informations sur les seuils d'un segment particulier.

L'exemple ci-dessous affiche des informations sur les seuils du segment par défaut de la base de données. Vous devez inclure "default" entre guillemets parce que c'est un mot réservé. Le résultat de `sp_helpthreshold` montre qu'il y a un seuil sur ce segment, défini à 200 pages. Le 0 dans la colonne "last chance" indique qu'il s'agit d'un seuil d'espace libre et non pas d'un seuil ultime :

```
sp_helpthreshold "default"
segment name    free pages    last chance?  threshold procedure
-----
default         200          0             space_dataseg
```

```
(1 row affected, return status = 0)
```

Seuils et tables système

La table système `systhresholds` contient des informations sur les seuils. `sp_helpthreshold` utilise cette table pour fournir ces informations. En plus des informations concernant le nom du segment, les pages libres, l'état ultime ou non ainsi que le nom de la procédure de seuil, cette table contient également l'ID utilisateur sur le serveur de l'utilisateur qui a créé le seuil et les rôles qu'il avait au moment de la création du seuil.

Adaptive Server permet d'obtenir des informations sur l'espace libre restant dans un segment par la fonction système intégrée `curunreservedpgs()`.

Ajout d'un seuil d'espace libre

Utilisez `sp_addthreshold` pour ajouter des seuils d'espace libre. Sa syntaxe est la suivante :

```
sp_addthreshold nom_base, nom_seg, espace_libre, nom_proc
```

Le `nom_base` doit spécifier le nom de la base de données en cours. Les autres paramètres spécifient le segment dont l'espace doit être contrôlé, la valeur du seuil en pages de base de données et le nom d'une procédure stockée.

Quand la quantité d'espace libre sur le segment descend en dessous du seuil, un processus interne à Adaptive Server exécute la procédure associée. Ce processus dispose des autorisations de l'utilisateur qui a créé le seuil au moment où il a exécuté `sp_addthreshold`, moins les autorisations qui ont été révoquées depuis.

Les seuils peuvent exécuter une procédure dans la même base de données, dans une autre base de données utilisateur, dans `sybssystemprocs` ou dans `master`. Ils peuvent aussi appeler une procédure distante sur un Open Server. `sp_addthreshold` ne vérifie pas que la procédure de seuil existe quand vous créez le seuil.

Modification d'un seuil d'espace libre

Utilisez `sp_modifythreshold` pour associer un seuil d'espace libre à une nouvelle procédure de seuil, un nouveau segment ou modifier la valeur d'espace libre. `sp_modifythreshold` supprime le seuil existant et en crée un autre à la place. Sa syntaxe est la suivante :

```
sp_modifythreshold nom_base , nom_segment, , espace_disponible  
[, nom_nouvelle_procedure [, nouvel_espace_libre  
[, nouveau_nom_segment]]]
```

où `nom_base` est le nom de la base de données en cours, `nom_segment` et `espace_disponible` identifient le seuil à modifier.

Par exemple, pour exécuter une procédure de seuil quand l'espace libre sur le segment tombe en dessous de 175 pages plutôt qu'en dessous de 200 pages, tapez :

```
sp_modifythreshold mydb, "default", 200, NULL, 175
```

Dans cet exemple, NULL est une marque de réservation qui permet de placer *nouvel_espace_libre* à la place appropriée dans la liste de paramètres. Le nom de la procédure de seuil n'est pas modifié.

La personne qui modifie le seuil devient le nouveau propriétaire de ce seuil. Quand la quantité d'espace libre sur le segment tombe en dessous du seuil, Adaptive Server exécute la procédure de seuil avec les autorisations du propriétaire au moment où il a exécuté `sp_modifythreshold`, moins les autorisations révoquées depuis.

Spécification d'une nouvelle procédure de seuil ultime

Vous pouvez utiliser `sp_modifythreshold` pour changer le nom de la procédure associée au seuil ultime. Vous *ne pouvez pas* l'utiliser pour modifier le volume d'espace disponible ou le nom du segment du seuil ultime.

`sp_modifythreshold` impose de spécifier le nombre de pages libres associées au seuil ultime. Utilisez `sp_helpthreshold` pour connaître cette valeur.

L'exemple ci-dessous affiche des informations sur le seuil ultime, puis spécifie une nouvelle procédure, `sp_new_thresh_proc`, à exécuter au dépassement du seuil :

```
sp_helpthreshold logsegment
```

```
segment name    free pages    last chance?  threshold procedure
-----
logsegment      40            1             sp_thresholdaction
```

```
(1 row affected, return status = 0)
```

```
sp_modifythreshold mydb, logsegment, 40,
sp_new_thresh_proc
```


Suppression d'un seuil

Utilisez `sp_droptreshold` pour supprimer un seuil d'espace libre dans un segment. Sa syntaxe est la suivante :

```
sp_droptreshold nom_base, nom_segment, espace_disponible
```

Le *nom_base* doit spécifier le nom de la base de données en cours. Vous devez spécifier à la fois le nom du segment et le nombre de pages libres, parce qu'il peut y avoir plusieurs seuils sur un segment. Exemple :

```
sp_droptreshold mydb, "default", 200
```

Création d'un seuil d'espace libre pour le segment de journal

Quand le seuil ultime est atteint, toutes les transactions sont annulées ou suspendues jusqu'à la libération de suffisamment d'espace de journalisation. Dans un environnement de production, cet événement peut avoir des conséquences importantes sur les utilisateurs. L'ajout d'un seuil d'espace libre défini correctement sur votre segment de journalisation peut réduire la probabilité d'atteindre le seuil ultime.

Le seuil supplémentaire doit permettre de sauvegarder le journal de transactions suffisamment souvent afin d'éviter d'atteindre le seuil ultime. Mais il ne doit pas le sauvegarder trop souvent non plus, sinon la restauration de la base de données nécessiterait le chargement de beaucoup de bandes.

Cette section vous aide à déterminer la meilleure valeur pour un deuxième seuil de journalisation. Elle commence par ajouter un seuil avec une valeur de *espace_libre* fixée à 45 pour cent de la taille du journal et règle ce seuil en fonction de l'utilisation de l'espace sur votre site.

Ajout d'un seuil de journalisation à 45 pour cent de la taille du journal

Utilisez la procédure ci-dessous pour ajouter un seuil de journal avec une valeur *espace_libre* égale à 45 pour cent de la taille du journal.

- 1 Déterminez la taille du journal en pages :

```
select sum(size)
from master..sysusages
where dbid = db_id("nom_base")
and (segmap & 4) = 4
```

- 2 Utilisez `sp_addthreshold` pour ajouter un seuil avec une valeur *espace_libre* fixée à 45 pour cent. Par exemple, si la capacité du journal est de 2048 pages, ajoutez un seuil avec une valeur *espace_libre* de 922 pages :

```
sp_addthreshold mydb, logsegment, 922,
thresh_proc
```

- 3 Créez une procédure de seuil simple qui sauvegarde le journal de transactions vers les devices appropriés. Pour plus d'informations sur la création de procédures relatives aux seuils, reportez-vous à la section "Création de procédures de seuil", page 1006.

Test et réglage du nouveau seuil

Utilisez `dump transaction` pour vérifier que votre journal de transactions est saturé à moins de 55 pour cent. Utilisez ensuite la procédure ci-dessous pour tester le nouveau seuil :

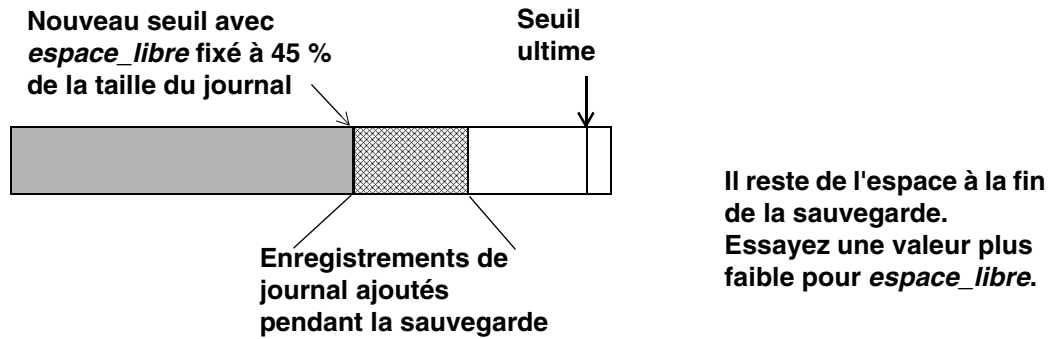
- 1 Remplissez le journal de transactions en simulant des actions quotidiennes d'utilisateur. Utilisez des scripts automatisés effectuant des transactions courantes à la fréquence prévue.

Quand le seuil de 45 pour cent d'espace libre est atteint, votre procédure de seuil doit sauvegarder le journal de transactions. Ce n'est pas un seuil ultime, donc les transactions ne seront ni suspendues ni annulées ; le journal continuera à grandir pendant la sauvegarde.

- 2 Pendant la sauvegarde, utilisez `sp_helpsegment` pour contrôler l'utilisation de l'espace sur le segment de journal. Enregistrez la taille maximale du journal de transactions juste avant l'achèvement de la sauvegarde.

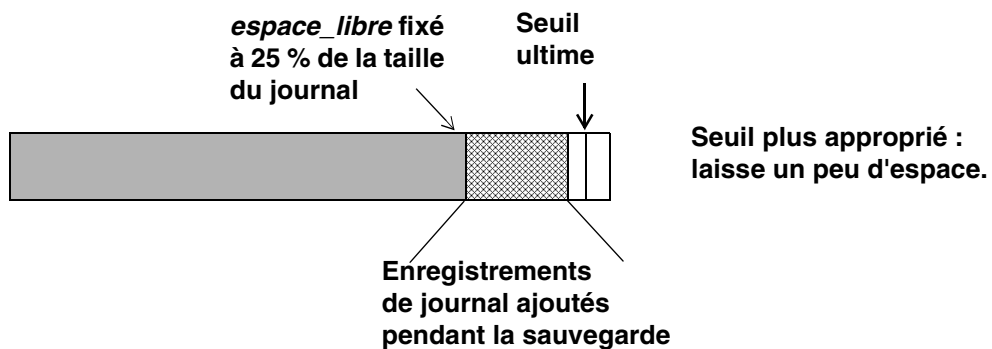
- 3 S'il restait beaucoup d'espaces dans le journal au moment de l'achèvement de la sauvegarde, il peut ne pas être nécessaire de sauvegarder le journal de transactions si tôt, comme indiqué à la figure 29-8 :

Figure 29-8 : Journal de transactions avec un seuil supplémentaire à 45 pour cent



Essayez d'attendre jusqu'à ce qu'il reste 25 pour cent de l'espace du journal, comme indiqué à la figure 29-9 :

Figure 29-9 : Le déplacement du seuil laisse moins d'espace libre après la sauvegarde

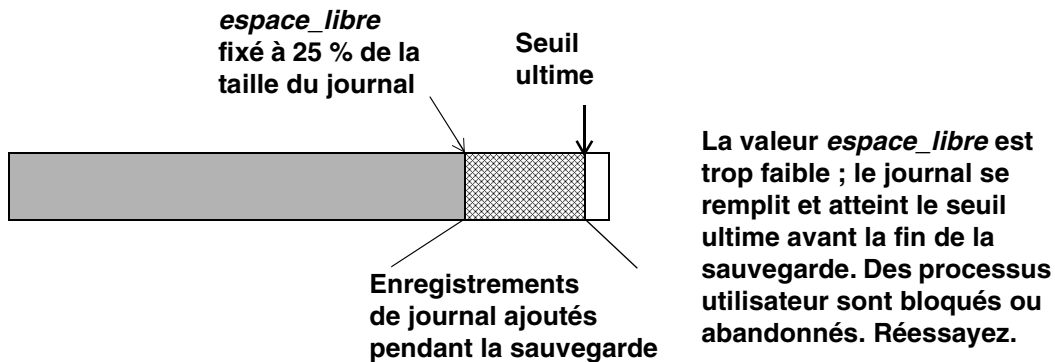


Utilisez `sp_modifythreshold` pour régler la valeur `espace_libre` à 25 pour cent de la taille du journal. Exemple :

```
sp_modifythreshold mydb, logsegment, 512,
    thresh_proc
```

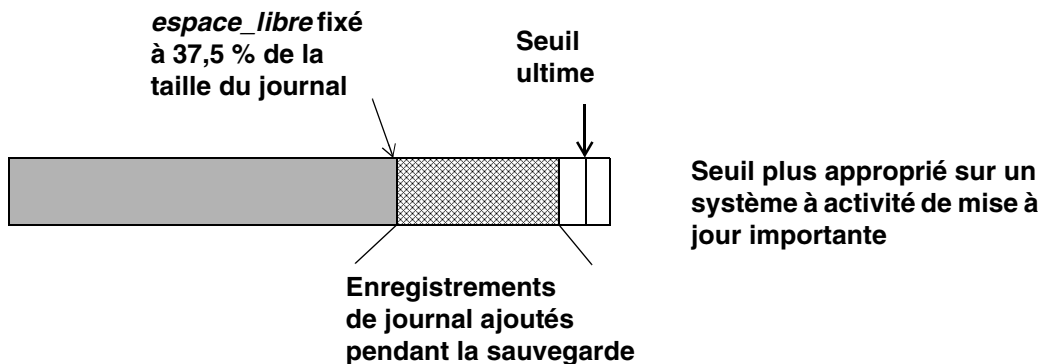
- 4 Sauvegardez le journal de transactions et testez la nouvelle valeur `espace_libre`. Si le seuil ultime est atteint pendant la sauvegarde, vous ne commencez pas la sauvegarde par dump transaction assez tôt, comme l'indique la figure 29-10 :

Figure 29-10 : Le seuil de journal supplémentaire ne commence pas la sauvegarde assez tôt



25 pour cent d'espace libre n'est pas suffisant. Essayez de commencer la transaction de sauvegarde quand le journal a encore 37,5 pour cent d'espace libre, comme indiqué à la figure 29-11 :

Figure 29-11 : Le déplacement du seuil laisse suffisamment d'espace libre pour terminer la sauvegarde



Utilisez `sp_modifythreshold` pour modifier la valeur `espace_libre` à 37,5 pour cent de la capacité du journal. Exemple :

```
sp_modifythreshold mydb, logsegment, 768,  
thresh_proc
```

Création de seuils supplémentaires sur d'autres segments

Vous pouvez créer des seuils d'espace libre sur des segments de données comme sur des segments de journal. Par exemple, vous pouvez créer un seuil d'espace libre sur le segment "default" utilisé pour les tables et les index. Vous devez alors créer aussi une procédure stockée associée capable d'imprimer des messages dans votre journal d'erreurs quand l'espace sur le segment default tombe sous ce seuil. Si vous contrôlez ces messages dans le journal d'erreurs, il est possible d'ajouter de l'espace au device de base de données avant que vos utilisateurs rencontrent des problèmes.

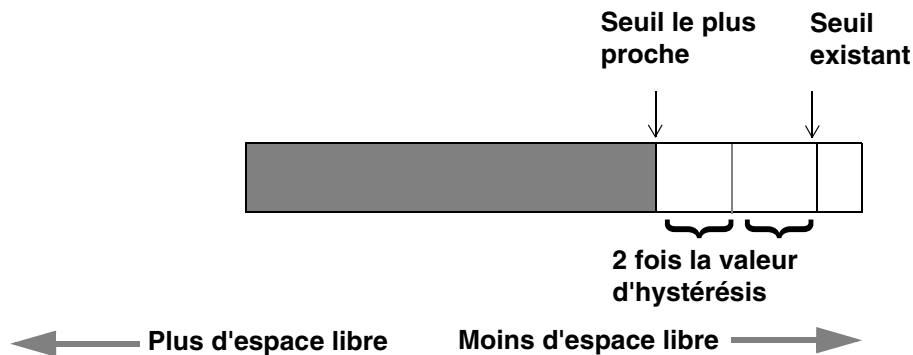
L'exemple ci-dessous crée un seuil d'espace libre sur le segment default de mydb. Quand l'espace libre sur ce segment tombe en dessous de 200 pages, Adaptive Server exécute la procédure `space_dataseg` :

```
sp_addthreshold mydb, "default", 200, space_dataseg
```

Détermination de la valeur du seuil

Chaque nouveau seuil doit être distant du seuil le plus proche d'au moins deux fois la valeur de `@@thresh_hysteresis`, comme l'indique la figure 29-12 :

Figure 29-12 : Détermination de l'emplacement d'un seuil



Pour connaître la valeur d'hystérésis d'une base de données, utilisez :

```
select @@thresh_hysteresis
```

Dans cet exemple, un segment a un seuil défini à 100 pages, et la valeur d'hystérésis pour la base de données est de 64 pages. Le seuil suivant doit être au moins égal à $100 + (2 * 64)$, soit 228 pages.

```
select @@thresh_hysteresis
-----
          64
sp_addthreshold mydb, user_log_dev, 228,
sp_thresholdaction
```

Création de procédures de seuil

Sybase ne fournit pas de procédure de seuil. Vous devez les créer vous-même et vérifier qu'elles sont adaptées aux besoins de votre site.

Les actions suggérées pour une procédure de seuil sont notamment l'écriture dans le journal d'erreurs du serveur et la sauvegarde du journal de transactions pour augmenter l'espace de journalisation. Vous pouvez aussi exécuter des appels de procédures distantes vers un Open Server ou un XP Server. Par exemple, si vous incluez les commandes suivantes dans `sp_thresholdaction`, celle-ci exécute la procédure `mail_me` sur un Open Server :

```
exec openserv...mail_me @dbname, @segment
```

Reportez-vous au chapitre 15, "Utilisation des procédures stockées étendues", dans le *Guide de l'utilisateur Transact-SQL* pour plus d'informations sur l'utilisation de procédures stockées étendues et de XP Server.

Cette section fournit quelques règles d'écriture de procédures de seuil ainsi que deux exemples de procédures.

Déclaration des paramètres de la procédure

Adaptive Server transmet les quatre paramètres suivants à une procédure de seuil :

- `@nom_base`, varchar(30), qui contient le nom de la base de données ;
- `@segmentname`, varchar(30), qui contient le nom du segment ;
- `@space_left`, int qui contient la valeur de l'espace restant pour le seuil ;

- `@status`, int, possède la valeur 1 pour les seuils ultimes et 0 pour les autres seuils.

Ces paramètres sont passés par position plutôt que par nom. Votre procédure peut utiliser d'autres noms pour ces paramètres, mais doit les déclarer dans l'ordre présenté et avec les types de données indiqués.

Génération de messages dans le journal d'erreurs

Vous devez inclure une instruction `print` près du début de votre procédure pour enregistrer le nom de la base de données, le nom du segment et la valeur du seuil dans le journal d'erreurs. Si votre procédure ne contient pas d'instruction `print` ou `raiserror`, le journal d'erreurs ne contiendra aucun enregistrement de l'événement seuil.

Le processus qui exécute les procédures de seuil est un processus interne Adaptive Server. Il n'a pas de connexion réseau ou terminal utilisateur associée. Si vous testez vos procédures de seuil en les exécutant directement (c'est-à-dire par `execute nom_procedure`) dans une session terminal, la sortie des messages `print` et `raiserror` apparaît sur votre écran. Lors de l'exécution des mêmes procédures par franchissement d'un seuil, les messages vont dans le journal d'erreurs. Les messages dans le journal d'erreurs incluent la date et l'heure.

Par exemple, si `sp_thresholdaction` inclut cette instruction :

```
print "LOG DUMP: log for '%1!' dumped", @dbname
```

Adaptive Server écrit ce message dans le journal d'erreurs :

```
00: 92/09/04 15:44:23.04 server: background task message: LOG DUMP: log for  
'pubs2' dumped
```

Sauvegarde du journal de transactions

Si votre procédure `sp_thresholdaction` inclut une commande `dump transaction`, Adaptive Server sauvegarde le journal vers les devices indiqués dans la procédure. `dump transaction` tronque le journal de transactions en supprimant toutes les pages depuis le début du journal, jusqu'à la page juste avant celle qui contient un enregistrement de transactions encore ouverte.

Quand il y a suffisamment d'espace de journalisation, les transactions suspendues sont réveillées. Si vous annulez les transactions plutôt que de les suspendre, les utilisateurs doivent les soumettre à nouveau.

Généralement, la sauvegarde vers un disque *n'est pas* recommandée, en particulier vers un disque se trouvant sur la même machine ou le même contrôleur que le disque de base de données. Mais comme les sauvegardes peuvent être déclenchées par les seuils à tout moment, vous pouvez envisager d'effectuer la sauvegarde sur disque avant de copier les fichiers résultants vers des supports hors ligne. (Vous devrez alors recopier les fichiers sur le disque pour les recharger.)

Votre choix peut dépendre de plusieurs facteurs :

- Présence d'un device de sauvegarde dédié en ligne, chargé et prêt à recevoir les données,
- Présence d'opérateurs pour monter les volumes de bande aux heures de disponibilité de votre base de données,
- Taille du journal de transactions,
- Débit de transactions,
- Programme régulier de sauvegarde des bases de données et des journaux de transactions,
- Espace disque disponible,
- Autres ressources et contraintes de sauvegarde spécifiques du site.

Une procédure de seuil simple

La procédure ci-dessous est très simple, elle sauvegarde le journal de transactions et imprime un message dans le journal d'erreurs. Cette procédure utilise la variable (*@dbname*) pour le nom de la base de données, elle est donc utilisable pour toutes les bases de données d'Adaptive Server :

```
create procedure sp_thresholdaction
    @dbname varchar(30),
    @segmentname varchar(30),
    @free_space int,
    @status int
as
dump transaction @dbname
to tapedump1
print "LOG DUMP: '%1!' for '%2!' dumped",
    @segmentname, @dbname
```


Une procédure plus complexe

La procédure de seuil ci-dessous effectue différentes actions selon la valeur des paramètres qui lui sont passés. Sa logique conditionnelle la rend utilisable avec les segments de journal et de données.

La procédure :

- Imprime un message "LOG FULL" si la procédure a été appelée suite au franchissement du seuil ultime du journal. Le bit d'état est à 1 pour le seuil ultime et à 0 pour tous les autres seuils. Le test if (@status&1) = 1 renvoie la valeur "true" seulement pour le seuil ultime.
- Vérifie que le nom de segment fourni est celui du segment de journal. L'ID du segment de journal est toujours 2, même si son nom a été modifié.
- Imprime les informations de taille "avant" et "après" dans le journal de transactions. Si la taille du journal n'a pas diminué de façon significative, une transaction à exécution longue peut causer une saturation du journal.
- Imprime l'heure de début et de fin de sauvegarde du journal de transactions, pour faciliter la collecte des données de durée de sauvegarde.
- Imprime un message dans le journal d'erreurs si le seuil n'est pas sur le segment de journal. Le message donne le nom de la base de données, le nom du segment et la valeur du seuil, pour vous faire savoir que le segment de données d'une base est en cours de saturation.

```
create procedure sp_thresholdaction
    @dbname      varchar(30),
    @segmentname varchar(30),
    @space_left  int,
    @status      int
as
declare @devname varchar(100),
        @before_size int,
        @after_size int,
        @before_time datetime,
        @after_time datetime,
        @error int
```

```
/*
** if this is a last-chance threshold, print a LOG FULL msg
** @status is 1 for last-chance thresholds, 0 for all others
*/
if (@status&1) = 1
begin
    print "LOG FULL: database '%1!'", @dbname
end

/*
** if the segment is the logsegment, dump the log
** log segment is always "2" in syssegments
*/
if @segmentname = (select name from syssegments
                    where segment = 2)
begin
    /* get the time and log size
    ** just before the dump starts
    */
    select @before_time = getdate(),
           @before_size = reserved_pgs(id, doampg)
    from sysindexes
    where sysindexes.name = "syslogs"

    print "LOG DUMP: database '%1!', threshold '%2!'",
          @dbname, @space_left

    select @devname = "/backup/" + @dbname + "_" +
              convert(char(8), getdate(), 4) + "_" +
              convert(char(8), getdate(), 8)

    dump transaction @dbname to @devname
    /* error checking */
    select @error = @@error
    if @error != 0
    begin
        print "LOG DUMP ERROR: %1!", @error
    end

    /* get size of log and time after dump */
    select @after_time = getdate(),
           @after_size = reserved_pgs(id, doampg)
    from sysindexes
    where sysindexes.name = "syslogs"
```

```
/* print messages to error log */
print "LOG DUMPED TO: device '%1!', @devname
print "LOG DUMP PAGES: Before: '%1!', After '%2!'",
    @before_size, @after_size
print "LOG DUMP TIME: %1!, %2!", @before_time, @after_time
end /* end of 'if segment = 2' section */
else /* this is a data segment, print a message */
begin
print "THRESHOLD WARNING: database '%1!', segment '%2!' at '%3!'
pages", @dbname, @segmentname, @space_left
end
```

Où mettre en place une procédure de seuil

Vous pouvez créer une procédure séparée pour sauvegarder le journal de transactions pour chaque seuil, mais il est plus facile de créer une procédure de seuil unique exécutée par tous les seuils de segment de journal. Quand la quantité d'espace libre sur un segment tombe sous un seuil, Adaptive Server lit la table `systhresholds` de la base de données concernée pour y trouver le nom de la procédure stockée associée, qui peut être au choix :

- Un appel de procédure distante vers un Open Server
- Un nom de procédure qualifié par un nom de base de données (par exemple `sybsystemprocs.dbo.sp_thresholdaction`)
- Un nom de procédure non qualifié

Si le nom de procédure n'inclut pas des qualificatifs de base de données, Adaptive Server recherche dans la base où le manque d'espace est apparu. S'il ne trouve pas la procédure, et si le nom commence par les caractères "sp_", Adaptive Server recherche la procédure dans la base de données `sybsystemprocs` puis dans la base de données `master`.

Si Adaptive Server ne trouve pas la procédure de seuil, ou s'il ne peut pas l'exécuter, il imprime un message dans le journal d'erreurs.

Désactivation de la comptabilisation de l'espace libre pour les segments de données

Utilisez l'option `no free space acctg` de `sp_dboption`, suivie par la commande `checkpoint`, pour désactiver la comptabilisation de l'espace libre sur les segments autres que de journal. Vous *ne pouvez pas* désactiver la fonction de comptabilisation de l'espace disponible sur le segment de journal.

Quand vous désactivez la comptabilisation d'espace libre, seuls les seuils de votre segment de journal contrôlent l'utilisation de l'espace ; les procédures de seuil sur vos segments de données ne s'exécutent pas au franchissement de ces seuils. La désactivation de la comptabilisation de l'espace libre accélère le temps de reprise en évitant les temps de recalcul de l'espace libre de tous les segments sauf le segment de journal.

L'exemple ci-dessous désactive la comptabilisation de l'espace libre pour la base de données `production` :

```
sp_dboption production,  
"no free space acctg", true
```

Avertissement ! Si vous désactivez la comptabilisation de l'espace libre, les procédures système ne peuvent pas fournir d'informations fiables sur l'allocation d'espace.

Index

Symboles

- ! guillemets (" ")
 - conversion en signe livre (£) dans les noms de login 550
- { } (accolades)
 - conversion en signe dollar (\$) dans les noms de logins 549
 - dans les instructions SQL xxxv
- ' (apostrophe) conversion en trait de soulignement dans les noms de login 549
- * (astérisque)
 - conversion en signe livre (£) dans les noms de login 550
 - select** et 467
- / (barre oblique)
 - conversion en signe livre (£) dans les noms de login 550
- \ (barre oblique inverse)
 - conversion en trait de soulignement dans les noms de login 549
- | (barre verticale)
 - conversion en signe livre (£) dans les noms de login 550
- ^ (caret)
 - conversion en signe dollar (\$) dans les noms de login 549
- [] (crochets)
 - conversion en signe livre (£) dans les noms de login 550
 - dans les instructions SQL xxxvi
- : (deux-points)
 - conversion en trait de soulignement dans les noms de login 549
- ... (ellipse) dans les instructions SQL xxxvi
- ' (guillemet droit) conversion en trait de soulignement dans les noms de login 549
- ' (guillemet gauche) conversion en trait de soulignement dans les noms de login 549
- " " (guillemets)
 - conversion en signe livre (£) dans les noms de login 550
- encadrement de valeurs 371
- inclusion de ponctuation 373
- syntaxe des valeurs de paramètres 12
- valeurs 755
- () (parenthèses)
 - conversion en signe dollar (\$) dans les noms de login 549
- & (perluète)
 - conversion en trait de soulignement dans les noms de login 549
- + (plus)
 - conversion en signe livre (£) dans les noms de login 550
- . (point)
 - conversion en signe dollar (\$) dans les noms de login 549
- ! (point d'exclamation)
 - conversion en signe dollar (\$) dans les noms de login 549
- ? (point d'interrogation) conversion en signe dollar (\$) dans les noms de login 549
- ?? (points d'interrogation)
 - pour les caractères suspects 326
- ; (point-virgule) conversion en signe livre (£) dans les noms de login 550
- % (pourcentage)
 - conversion en trait de soulignement dans les noms de login 549
 - marque de réservation de message d'erreur 54
- = (signe égal)
 - conversion en trait de soulignement dans les noms de login 549
- < (signe inférieur à)
 - conversion en signe dollar (\$) dans les noms de login 549
- (signe moins)
 - conversion en signe livre (£) dans les noms de login 550

Index

- > (signe supérieur à)
 - conversion en trait de soulignement dans les noms de login 549
- ~ (tilde)
 - conversion en trait de soulignement dans les noms de login 549
- , (virgule)
 - conversion en trait de soulignement dans les noms de login 549
 - dans les instructions SQL xxxiv

A

- accès
 - à distance 873
 - restriction des utilisateurs guest 378
 - restrictions ANSI sur bandes 923
- accès refusé à un utilisateur 394, 396
- accolades ({})
 - conversion en signe dollar (\$) dans les noms de login 549
 - dans les instructions SQL xxxv
 - voir* crochets []
- accréditation, mécanisme de sécurité et 535
- activation
 - audit 334, 479
- activation de rôles 391
- activation de SSL 347
- administrateur système 1–6
 - (SA), mode mono-utilisateur du serveur 966
 - (SA), mot de passe et **dataserver** 966
 - autorisations 419–422
 - résolution des problèmes système 57, 60
 - responsabilité en matière d'erreurs 57, 60–63
 - tâches pour les débutants 35–49
- administration de base de données 1–6
- administration de la sécurité
 - exemple 335
 - mise en route 332–336
 - recommandations 334
- adresse logique 714
- adresse virtuelle 586
- adresse, serveur 16
- affichage
 - fichiers sauvegardés sur une bande 852
 - option de base de données 722
- ajout
 - chaîne de date 311
 - commentaires dans la trace d'audit 480
 - connexions au serveur 372–374
 - connexions distantes 379
 - d'espace à une base de données 709
 - device de sauvegarde 877
 - devices de base de données 247, 580–587
 - groupe dans une base de données 374–375
 - intervalles de temps nommés 259
 - limites d'utilisation des ressources 273–275
 - logins au serveur 374
 - logins distants 525–527
 - mois de l'année 311
 - serveurs distants 519–532
 - seuils 998–1005
 - utilisateurs à un groupe 376
 - utilisateurs à une base de données 247, 370
 - utilisateurs guest 377
- alias
 - noms de device 876
 - serveur 520
 - utilisateur, aide sur 404
 - utilisateur, création 402
 - utilisateur, suppression 404, 432
 - utilisateur, transfert de propriété de base de données et 426, 709
 - utilisateur, *voir aussi* connexions; utilisateurs
- allemand
 - jeux de caractères supportés 289
- allocation d'espace
 - voir aussi* devices de base de données; segments; gestion du stockage
 - après affectation, attribution 700
 - après affectation, contigu 712, 714
 - après affectation, extents et rapport **sp_spaceused** 717
 - après affectation, impact de **drop database** 711
 - après affectation, mise en miroir des disques 594
 - après affectation, pages 717
 - après affectation, rôle du serveur 712
 - après affectation, unités 712
 - attribution 948
 - commande de contrôle **dbcc** 788–789
 - correction d'une erreur avec **dbcc** 788

- extents 775
- faire coïncider de nouvelles bases de données à des bases existantes 946
- méthodes de sauvegarde et 946
- OAM (Object Allocation Map) 777
- page 745, 775
- pour la base de données *dbccdb* 814
- récapitulatif des commandes 570
- recréation 848, 949
- réparation d'un extent non référencé 792
- répartition et fractionnement de tables 737
- restauration/performances et 572, 737
- rôle du serveur 775
- segments et 949
- sur un device existant 949
- table *sysusages* 576
- unités 775, 946
- alphabet latin 290
- anglais
 - jeux de caractères supportés 289
- annulation d'un lien 444
- annulation des processus
 - capacité de la pile du serveur et 251
 - intervalle de restauration et 106
 - transactions non validées 831
- apostrophe conversion en trait de soulignement dans les noms de login 549
- appartenance
 - chaînes 469
- appels de procédure
 - voir* appels de procédure à distance
- appels de procédure à distance 517–532
 - définition des options de sécurité 557
 - exemple de configuration de sécurité 561
 - mémoire 636
 - modèles de sécurité 558
 - paramètres de configuration pour 531–532
 - procédure globale du modèle de sécurité B 559
 - sauvegardes 869
 - sécurité réseau 555
 - temporisations 522
 - unification des logins et 556
- appels de procédure distante
 - seuils et 1011
- applications
 - identification des grands consommateurs de ressources 263
 - informations sur les limites d'utilisation des ressources 276
 - limites d'utilisation des ressources 262
 - mémoire 608
 - modification des limites d'utilisation des ressources 278
 - noms 262
 - procuration et 460
 - suppression des limites d'utilisation des ressources 280
- arabe
 - jeux de caractères supportés 289
- architecture
 - serveur SMP 682
- arrêt
 - Backup Server 77, 873
 - serveurs 76, 77
- astérisque (*)
 - conversion en signe livre (£) dans les noms de login 550
 - select** et 467
- attribution
 - de rôles à d'autres rôles 386
 - noms de login 335
- audit 356, 475, 475–508
 - voir aussi* options d'audit
 - activation 334, 479
 - activation et désactivation 493
 - affichage des options 479
 - ajout de commentaires dans la trace d'audit 480
 - base de données *sybsecurity* 30, 477
 - désactivation 479
 - devices 481
 - gestion de la trace d'audit 484
 - gestion du journal de transactions 491
 - installation 481
 - paramètres de configuration 479
 - présentation 475
 - procédure relative aux seuils 485
 - procédures système 479
 - tables *sysaudits_01...sysaudits_08* 509
 - taille de la file d'attente 235, 478

Index

authentification 533, 535
 du serveur, certificats de serveur 345
 réciproque 536

autorisations
 voir aussi DAC (contrôle d'accès discrétionnaire)

accès aux objets 428, 428–434

administrateur système 419–422

affectation sélective 436

alias et 402

attribution 435

base de données *master* 26

base de données *model* 27, 28

base de données *tempdb* 29

chaînes d'appartenance et 469

create database 422, 696–697

création d'objets 428, 434

de création de triggers 431

disk init 587

groupe public 427, 430, 438

groupes et 374

hiérarchie d'utilisateurs 440

identification concrète 432

informations 460–464

insuffisantes 59

insuffisantes (degré 14) 59

objet 5, 427

octroi 428–438
 par un propriétaire de base de données 435

opérateur 423

option **ansi_permissions** et 432

options par défaut 27

procédures relatives au seuil 999, 1000

procédures stockées 427, 430, 529

procédures système 425

procuration 457

propriétaires de bases de données 5, 419, 424

propriétaires des objets de la base de données 5

refus 59

révocation 428–438

suids divergents 976

synthèse 419

tables 427, 430

tables comparées à des vues 466

tables système 425

transferts et 426, 709

triggers et 473

utilisateurs distants 529

utilisateurs guest 377

valeur par défaut 421

vues 466–468
 sur les vues au lieu des colonnes 467

autorisations d'accès
 voir autorisations d'accès aux objets

autorisations d'accès aux objets
 grant all 429, 435
 voir autorisations

autre identité
 voir alias, utilisateur

B

Backup Server 868–871
 accès à distance 873
 arrêt 77
 chargement à partir de plusieurs devices 919
 compatibilité des sauvegardes selon les versions 912
 conditions de sauvegarde 871
 configuration des ressources système 913–917
 définition de l'usage de la mémoire partagée 914
 démarrage 873
 descripteurs de fichiers 916
 distant 901
 emplacement de 871
 fichier d'interface et 871
 format d'étiquette 913
 message d'exploitation de volume 939–943
 messages 924
 messages d'erreur 64
 nom réseau 972
 nombre de connexions réseau 916
 nombre de threads de service 914, 917
 nombre limite de devices de sauvegarde 913
 noms de device et 875
 paramètre de configuration **tape retention in days** 108
 striping de sauvegarde et 918
 support multifichier et expiration de bande 922
 table *syssservers* 871
 utilisation de moins de devices pour le chargement que pour la sauvegarde 919

- vérification avec **showserver** 973
- balayages décroissants
 - interblocages et 194
- balte
 - jeux de caractères supportés 289
- barre oblique (/)
 - conversion en signe livre (£) dans les noms de login 550
- barre oblique inverse (\)
 - conversion en trait de soulignement dans les noms de login 549
- base 4
- base de données
 - courante 58
 - dbccdb*
 - contrôle de cohérence de 819
 - création d'espace de travail 809
 - informations sur la configuration du rapport 821
 - informations sur les erreurs de rapport 822
 - installation 815
 - suppression de l'historique de **dbcc checkstorage** 819
 - suppression des informations sur la configuration 819
- maintenance 774
- master* 9, 25–27, 43
 - appartenance 426
 - clés pour les tables système dans la 10
 - commandes modifiant la 880
 - conditions pour une sauvegarde sur un seul volume 875
 - création 574
 - en tant que base de données par défaut 372
 - en tant que base de données utilisateur par défaut 372
 - extension à l'aide de **alter database** 710
 - modification de la définition des options 722
 - propriété de 709
 - sauvegarde 43, 875, 880–881
 - sauvegarde du journal de transactions 881
 - suppression d'utilisateurs guest de 377
 - symptômes de dommages 944
 - table *sysdevices* 587
 - utilisateur guest dans 377
- model* 27
 - clés pour les tables système dans la 10
 - création 574
 - modification des options 722
 - modification des options de base de données 728
 - reprise automatique et 831
 - restauration 977
 - sauvegarde 882
 - sauvegarde du journal de transactions 882
 - taille 200, 583, 701
- par défaut
 - utilisateur, changement de 399
- primaire 726
- pubs2*
 - administration 31
 - informations *image* dans 32
- pubs3*
 - administration 31
- secondaire 726
- sybsecurity* 30, 477
 - reprise automatique et 831
- sybssystemdb* 31
 - reprise automatique et 831
- sybssystemprocs* 12, 14, 28
 - autorisations et 426
 - reprise automatique et 831
 - restauration 979–981
 - sauvegarde 882
 - seuils et 1011
 - voir aussi* bases de données
- tempdb* 29
 - cache de données 679
 - création 574
 - option de base de données **auto identity** 724
 - option de base de données **unique auto_identity index** 729
 - reprise automatique et 831
 - taille 29
 - voir aussi* bases de données
- master*
 - voir aussi* mise en miroir de disques; tables système
- bases de données
 - voir aussi* objets de base de données; bases de données utilisateur
 - ajout d'utilisateurs 375–379, 699

Index

- appartenance 422
 - attribution à des devices de base de données 700
 - audit 482
 - augmentation de la taille 709
 - changement de base utilisateur par défaut 372
 - chargement 949
 - chargement après changement de l'ordre de tri 305
 - chargement après changement du jeu de caractères 305
 - corrompues
 - base système et base utilisateur 835
 - estimation du nombre de pages suspectes 845
 - isolement des pages suspectes 836
 - mode d'isolement des pannes en cours de reprise 835
 - création avec un segment de journal distinct 702
 - création d'un utilisateur 697–709
 - droit de création 422
 - erreurs affectant les 62
 - espace insuffisant dans 935
 - gestion de l'espace utilisé 717
 - informations sur l'espace de stockage utilisé 716
 - informations sur le stockage 712
 - interactions sauvegarde/journal et 831
 - liaison à des caches de données 654
 - maintenance de 798
 - mise à niveau des sauvegardes de bases de données 952
 - nom 697
 - nombre de bases de données ouvertes 156
 - nouvelles 27
 - numéro de séquence pour la reprise 726
 - option **checkalloc (dbcc)** 788
 - option **checkdb (dbcc)** 787, 794
 - option **checkstorage (dbcc)** 781
 - option **checktable (dbcc)** 784
 - option **indexalloc (dbcc)** 790
 - option **tablealloc (dbcc)** 790
 - options 722–730
 - options par défaut 26
 - par défaut 372, 373, 399
 - problèmes de cohérence 63, 774–798
 - restauration 944
 - restrictions relatives aux copies externes 857
 - sauvegarde 28, 43, 797, 846
 - stockage par défaut 24, 589
 - suppression 711, 805
 - suppression d'utilisateurs 392, 699
 - suppression et réparation de bases corrompues 947
 - système 23, 23–31
 - ordre de reprise 832
 - taille 27
 - taille par défaut 701
 - transfert sur une autre machine 707, 849, 945
 - utilisateur 696
 - contrôle de la base de données *master* 25
 - messages définis par l'utilisateur 57
 - processus de création 698
 - reprise automatique et 831
 - tables système 27
 - voir aussi* bases de données; autorisations
 - batches de requête
 - intervalles de temps actifs et 262
 - portée des limites d'utilisation des ressources et 267
 - restriction du temps écoulé 271
 - bcp** (utilitaire bulk copy)
 - changements de l'ordre de tri et 305
 - commande **dump database** et 879
 - conversion de jeux de caractères et 328, 329
 - mode rapide 727
 - règles d'accès et 447
 - select into/bulkcopy/pilsort** 727
 - services de sécurité et 563
 - Big 5
 - similitudes avec CP 950 289
 - bits d'état dans *sysdevices* 588
 - bloc de device, mise en miroir 599
 - "proc buffers" 631
- ## C
- cache de données
 - chargement de base de données et 956–957
 - configuration 639, 674–679
 - configuration de partitions 110, 670
 - dbcc** et 794
 - erreurs d'intégrité de bases de données et 63
 - état 644
 - fichier de configuration et 674
 - informations 641–643, 656

- modification de la taille 664, 678
- modification des liaisons 656
- nombre global de partition de caches 670
- overhead 657
- par défaut 638, 646, 678
- partitionnement 670
- partitions de cache 671
- partitions de cache local 670
- résumé des commandes 639
- suppression 666
- suppression de liaisons 659
- taille d'E/S et 678
- cache de métadonnées
 - description 632
 - paramètres de configuration 155–163
 - recherche de statistiques d'utilisation 626
- cache de procédures 113
 - paramètre de configuration **procedure cache percent** 612
- cache nommé **dbcc checkstorage** 812
- caractères
 - ASCII, conversion de jeux de caractères et impossibles à convertir 320
 - non admis dans les noms de login 549
- catalogues système
 - voir* tables système
- certificats
 - administration 351
 - autorisation 349
 - CA auto-signée 349
 - certificats de CA 342
 - certificats de serveur 342
 - cryptage par clé publique 342
 - définition 342
 - demande 349
 - obtention 348
- certificats d'origine sécurisés
 - certificat de CA 342
 - emplacement 345
- certificats de CA 342
 - certificats d'origine sécurisés 342
 - emplacement 345
- certificats de serveur 342
 - authentification du serveur 345
 - emplacement 345
- chaîne de pages
 - données *text* ou *image* 747
- chaînes, appartenance 469
- chargement, base de données 949
 - cache de données et 956–957
 - chargement de bases de données utilisant des contraintes référentielles entre bases de données 958
 - invites **sp_volchanged** 942
 - modification de nom 891
 - paramètre de configuration **number of large i/o buffers** 105, 121, 883
 - reconfiguration automatique 949
 - spécification de device 891
- chargement, journal de transactions
 - invites **sp_volchanged** 942
 - ordre des sauvegardes 949
 - spécification de device 891
- chemin d'accès
 - device miroir 599
- chevauchement des intervalles de temps 258
- chinois simplifié
 - jeux de caractères supportés 289
- chinois traditionnel
 - jeux de caractères supportés 289
- clés, table
 - sur les tables système 10
- client
 - attribution de nom de client, nom d'hôte et nom d'application 400
 - conversion de jeux de caractères 329
- codage des caractères 317
- cohérence
 - vérification des bases de données 45, 846
- colonne *dbid*, table *sysusages* 713
- colonne IDENTITY
 - automatique 724, 729
 - index non unique 726
- colonne *lstart*, table *sysusages* 714
- colonne *segmap*, table *sysusages* 714
 - procédures qui changent 577
 - valeurs de segment 946
- colonne *segment*, table *syssegments* 714
- colonne *srvname*, table *sys.servers* 521
- colonne *srvnetname*, table *sys.servers* 521
- colonne *vstart* 714

Index

- colonnes
 - autorisations 430, 464
- commande
 - alter database**, 709
 - alter database**, omission du device de base de données et 588, 589
 - alter database**, option **for load** 711
 - alter database**, option **with override** 711
 - alter database**, sauvegarde de *master* après 880
 - alter database**, tables système et 576, 752
 - alter database**, taille de la base de données et 700
 - alter database**, *voir aussi* commande **create database**
 - alter role**, 386
 - alter role**, déverrouillage des rôles 360
 - alter table** 447
 - checkpoint** 830
 - checkpoint**, définition des options de base de données 730
 - create database**, 697–709
 - create database**, allocation d'espace de stockage 700
 - create database**, autorisation 696
 - create database**, base de données *model* et 27
 - create database**, droit d'utilisation 422
 - create database**, mot-clé **on** 700
 - create database**, omission de la clause **on** 702
 - create database**, omission du device de base de données et 587, 589
 - create database**, option **for load** et 706, 711
 - create database**, option **log on** 700, 702
 - create database**, option omission de l'option **log on** 705
 - create database**, option **with override** 707, 711
 - create database**, paramètre de configuration **default database size** et 199, 701
 - create database**, paramètre *size* 701
 - create database**, sauvegarde de *master* après 880
 - create database**, tables système et 9, 752
 - create index** 572, 577, 743
 - create index**, déplacement de tables avec 757
 - create index**, sauvegarde de base de données et 879
 - create procedure** 14
 - create role** 385
 - create table** 572, 743
 - create table**, index clusterisés et 757
 - create trigger** 431
 - cs_connection, number of user connections** et 247
 - dataserver**, redémarrage du serveur avec **-q** 859
 - dbprocess, number of user connections** et 247
 - delete**, journal de transactions et 827
 - disk init** 571, 576, 580–587
 - disk init**, allocation et 774
 - disk init**, device miroir 599
 - disk init**, sauvegarde de la base de données *master*, après 880
 - disk mirror** 571, 593, 598–603
 - disk refit** 983
 - disk reinit** 982
 - disk reinit**, *voir aussi*
 - disk reinit**, *voir aussi* commande **disk init**
 - disk remirror** 602
 - disk remirror**, *voir aussi* mise en miroir des disques
 - disk unmirror** 600
 - disk unmirror**, *voir aussi* mise en miroir des disques
 - drop database** 711
 - drop database**, bases de données endommagées et 805
 - drop role** 393
 - dump database** 885–952
 - dump database**, base de données *master* et 43
 - dump database**, base de données *model* et 28
 - dump database, disk init** et 580
 - dump database**, droits d'exécution 867
 - dump database**, interdite dans les bases de données hors ligne 841
 - dump database**, planification de **dbcc** et 797
 - dump database**, utilisation 797, 877–883
 - dump database**, *voir aussi* sauvegarde, base de données
 - dump transaction** 702, 704, 706, 886–952
 - dump transaction**, dans la base de données *master* 881
 - dump transaction**, dans la base de données *model* 882
 - dump transaction**, droits d'exécution 867
 - dump transaction**, interdite dans les bases de données hors ligne 841
 - dump transaction**, option **compress** 894
 - dump transaction**, option **standby_access** 928
 - dump transaction**, option **with no_log** 879, 935
 - dump transaction**, option **with no_truncate** 932
 - dump transaction**, option **with truncate_only** 934
 - dump transaction**, procédures relatives au seuil et 1007

- dump transaction, trunc log on chkpt** et 728–729, 829
- dump transaction**, voir aussi sauvegarde, journal de transactions
- grant** 420, 428–438
- grant**, création de base de données 697
- grant**, groupe public et 430, 457
- grant**, mot-clé **all** 435
- grant**, rôles et 439, 457
- insert**, journal de transactions et 827
- kill** 66–69
- load database** 886–952
- load database**, droits d'exécution 867
- load database**, pour base de données *model* 978
- load database**, pour base de données *sysystemprocs* 981
- load database**, pour la base de données *master* 973
- load database**, voir aussi chargement de base de données
- load transaction** 886–952
- load transaction**, droits d'exécution 867
- load transaction**, voir aussi chargement du journal de transactions
- online database** 849, 951
- online database** option **standby_accessonline database** 929
- online database**, activation de bases de données 951
- online database**, bases de données répliquées 951
- online database**, bits d'état 954
- online database**, mise à niveau d'une base de données 851, 953
- online database**, reprise d'une base de données 853, 855
- quiesce database** 855–866
- quiesce database**, méthode de l'actualisation itérative 862
- quiesce database**, méthode de la reprise semi-automatique 864
- quiesce database**, mise à jour des enregistrements de journal 859
- quiesce database**, mise à jour du numéro de séquence de sauvegarde 860
- quiesce database**, recommandations 857
- quiesce database**, sauvegarde des devices secondaires 862
- quiesce database**, syntaxe 855
- reconfigure** 101
- reorg** 759–768
- reorg**, option **compact** 764
- reorg**, option **forwarded_rows** 762
- reorg**, option **rebuild** 765
- reorg**, option **reclaim_space** 763
- REPLY**, (OpenVMS) 939
- revoke** 420, 428–438
- revoke**, groupe public et 430
- select ***, message d'erreur 467
- select into**, sauvegarde de base de données et 879
- set**, rôles et 391
- setuser ,show_role** et 410
- shutdown** 76–78
- shutdown**, Backup Server 873
- shutdown**, point de reprise automatique et 830
- shutdown**, reprise automatique après 831
- update statistics** 762
- update**, journal de transactions et 704, 827
- utilitaire **defncopy**, conversion de jeux de caractères et 328, 329
- utilitaire **defncopy**, voir aussi le manuel Utilitaires
- utilitaire **isql**, administration du système et 6
- utilitaire **isql**, conversion de jeux de caractères et 328, 329
- utilitaire **isql**, messages d'état et d'information 58
- utilitaire **isql**, mots de passe et 530
- utilitaire **isql**, **number of user connections** et 246
- utilitaire **isql**, services de sécurité et 563
- utilitaire **startserver** Backup Server et 873
- waitfor mirrorexit** mise en miroir des disques 602
- writetext**, option de base de données **select into/bulkcopy/plisort** 727
- writetext**, sauvegarde de base de données et 879
- commande **dump database**, option **compress** 894
- commandes de sauvegarde
 - voir **dump database**; dump transaction
- commandes du système d'exploitation
 - exécution 14
- commandes non consignées 879
- commandes utilitaires
 - voir aussi le manuel Utilitaires
- buildmaster** 966

Index

- jeux de caractères et 328
- showserver** 973
- startserver** 966
- commentaires
 - ajout à la trace d'audit 480, 506
- commit à deux phases
 - transactions 31
- comparaison de valeurs
 - problèmes liés aux types de données 58
- compartiments de hachage de verrou 151
- compression aux fins de sauvegarde
 - définie 894
 - exemple 896
 - niveaux de compression 897
 - syntaxe 895
- comptabilisation, refacturation 415
- comptabilité individuelle 334
- comptes de visiteur 378
- comptes, serveur
 - voir* connexions; utilisateurs
- conception d'applications 246
- condensé de messages
 - définition 341
 - hachage 341
- configuration (serveur)
 - voir aussi* paramètre de configuration
 - cache de données nommés 674
 - environnement SMP 684–693
 - fichier de configuration et caches 674
 - jeux de caractères 303
 - langue des messages 303–306
 - limites d'utilisation des ressources 257
 - ordres de tri 303–309
 - pour base de données *dbccdb* 810
 - sécurité réseau 538
- conflit d'autorisations 437
 - voir aussi* autorisations
- connexion à Adaptive Server 16
- connexions
 - voir aussi* connexions distantes; utilisateurs
 - affectation de noms 335
 - affichage des informations relatives au mot de passe 361
 - ajout aux serveurs 372–374
 - alias 432
 - déverrouillage 360
 - fichiers d'interface 16
 - identification et authentification 337
 - intervalles de temps actifs et 262
 - nom utilisateur "dbo" 2, 4
 - nombre maximal de tentatives de connexion, définition 357
 - nombre maximal de tentatives de connexion, modification 359
 - nombre maximum d'utilisateurs 246
 - noms incorrects 549
 - propriétaire des objets de la base de données "sa" 334
 - services de répertoire 17
 - verrouillage 357, 360
- connexions distantes
 - paramètres de configuration pour 164
- connexions SSL
 - Open Client 346, 347
 - pour les RPC 346
 - pour serveurs compagnons 347
- connexions utilisateur
 - mémoire allouée par 245–247
- conservation des enregistrements 47–49
 - configuration 48
 - contacts 47
 - maintenance 48
 - système 49
- constantes xxxvii
- contextes applicatifs 442, 453
- contraintes d'intégrité référentielle
 - chargement de bases de données 958
- contraintes d'intégrité référentielle entre bases de données
 - chargement de base de données 959
- contrôle
 - CPU occupation 689
 - d'accès 441
 - table *spt_monitor* 13
 - texte SQL 187
- contrôleurs de disque 587, 737
- conventions
 - syntaxe Transact-SQL xxxiv–xxxvii
 - utilisées dans les manuels xxxiv
- conversion de jeux de caractères 318, 327–328
- copie
 - fichiers et disques de sauvegarde 875

- copie de données sélectionnées
 - voir* commande **insert**; **select**
 - copies externes de bases de données 855
 - coréen
 - jeux de caractères supportés 289
 - correspondance
 - utilisateurs distants 529
 - corruption de la base de données
 - causée par la copie de devices de base de données 846
 - coût
 - E/S 269
 - coût estimé
 - limites d'utilisation des ressources pour les E/S 266, 268
 - CP 1252
 - similitudes avec ISO 8859-1 289
 - CP 950
 - similitudes avec Big 5 289
 - CPU occupation
 - contrôle 689
 - nombre de moteurs et 684
 - traitement symétrique et 682
 - création 444, 453
 - alias d'utilisateur 402
 - base de données *master* 574
 - base de données *model* 574
 - base de données *sybsecurity* 482
 - base de données *tempdb* 574
 - bases de données 422, 697, 709
 - groupes 374–375
 - intervalles de temps nommés 259
 - limites d'utilisation des ressources 273–275
 - messages d'erreur définis par l'utilisateur 57
 - noms logiques 875
 - objets de base de données 572
 - objets de base de données sur des segments 742
 - procédures stockées 13
 - procédures système 13
 - segments 574, 740
 - seuils 998–1005
 - tables système 9
 - triggers 431
 - utilisateurs guest 377
 - création d'un contexte applicatif 453
 - créneau de vulnérabilité 196
 - crochets []
 - conversion en signe livre (£) dans les noms de login 550
 - dans les instructions SQL xxxvi
 - cryptage
 - clé publique/privée 340
 - clé symétrique 340
 - cryptage par clé publique 340
 - données 534
 - échange de clés 340
 - par clé publique/privée 341
 - par clés symétriques 340
 - cryptage par clé publique
 - certificats 340
 - cryptage 340
 - définition 340
 - signature électronique 340
 - curseurs
 - nombre de lignes, valeur 117
 - restriction du coût des E/S 270
 - restriction du nombre de lignes renvoyées 273
 - cyrillique
 - jeux de caractères supportés 289
- ## D
- DAC (contrôle d'accès discrétionnaire) 419–473
 - voir aussi* autorisations
 - administrateurs système et 419
 - alias utilisateur et 455
 - octroi et révocation d'autorisations 428
 - présentation 337
 - procédures stockées et 468
 - vues 466
 - dates
 - ajout d'éléments de date 311
 - autre langue 311
 - format des messages d'erreur 56
 - formats d'affichage 315
 - dbcc** (Database Consistency Checker) 45, 773–824
 - altération des bases de données et 57, 62, 774
 - commandes comparées 794
 - maintenance d'une base de données et 774–798
 - maintenance d'une base de données et 944
 - planification 795–798

Index

- rapports 798
- réalisation de contrôles par 779
- résultat 798
- sauvegardes et 846
- utilisation 62
- déchargement de fichiers compressés 899
- déchargement de journaux de transactions compressés 899
 - syntaxe 899
- déchargement de sauvegardes compressées
 - exemple 899
 - syntaxe 899
- degrés de sévérité, erreur 51, 57
 - Backup Server 65
 - degrés 10 à 18 (erreurs utilisateur) 58
 - degrés 19 à 24 (erreurs fatales) 61
- démarrage de serveurs
 - Backup Server 873
 - mémoire requise 618
 - mode restauration de master 966
 - services de sécurité et 551
- dénomination
 - serveurs 520
- désactivation de l'audit 479
- désactivation de rôles 391
- descripteurs d'index
 - nombre maximum ouverts 158
- descripteurs de balayage 217–220
- descripteurs de fichier 246
 - nombre maximum configuré par processus pour votre système d'exploitation 176
- descripteurs de fichiers, nombre pour Backup Server 916
- détection de l'altération, signature électronique 341
- détection des ré-exécutions 536
- deux-points (:)
 - conversion en trait de soulignement dans les noms de login 549
- déverrouillage d'un compte de connexion 394
- déverrouillage d'un compte de connexion 361
- déverrouillage des rôles 361
- device de base de données
 - annulation de la mise en miroir 600
 - placement des journaux de transactions sur un device distinct 594
- device de sauvegarde
 - ajout 877
- autorisations 871
- bandes 874
- disques 875
- fichiers 875
- informations 587
- liste 876
- maximum autorisé 917
- multiple 910, 911–920
- noms logiques pour 875–877
- redéfinition 877
- spécification 889, 891
- suppression 589, 877
- table *sysdevices* et 575, 876
- tape retention in days** et **retaindays** utiles pour 922
 - voir* devices de sauvegarde
- device disque
 - voir aussi* devices de base de données; devices de sauvegarde; allocation d'espace
 - ajout 877
 - annulation de la mise en miroir 600
 - mise en miroir 593–603
 - sauvegarde 875
- device master 24, 582, 588
 - voir aussi* devices de base de données
 - mise en miroir des 595
 - mise en miroir des disques 603
 - sp_diskdefault** et 589
 - suppression de la zone d'espace par défaut 589
- device miroir 594, 599
- device, désactivation de la mise en miroir
 - voir* mise en miroir des disques
- devices 579
 - voir aussi* devices de base de données; devices de sauvegarde; device master
 - ajout 580–587
 - contraintes du système d'exploitation 582
 - création d'alias de 876
 - fractionnement de tables sur plusieurs devices 745–747
 - initialisation 579–587
 - liste 876
 - listes d'informations 587, 947
 - noms des devices physiques 875–877
 - noms physiques 581
 - number of user connections** et 246, 247

- suppression 589, 947
- système d'audit 481
- utilisation d'un device distinct 573, 702, 736
- devices de base de données 579
 - voir aussi* mise en miroir des disques; devices de sauvegarde; device master
- ajout 580–587
- amélioration des performances 737–757
- attribution d'une base de données 948
- attribution d'une base de données 700, 711
- fragments 576
- informations 587, 715
- initialisation 579–587
- nombre utilisable par le serveur 122, 633
- noms 574, 581
- numérotation 699
- par défaut 589–591
- placement d'objets 573, 742
- reprise et 982
- suppression 589
- valeurs par défaut 702
- devices de base de données par défaut 702
- désignation 589
- devices de sauvegarde (bande)
 - démontage 922
 - marqueur de fin de bande 905
 - nom de volume 907
 - réinitialisation des volumes 923
 - rembobinage 922
- devices de sauvegarde sur bande
 - ajout 877
 - prévention des réécritures 874
 - pour les sauvegardes 874
- devices de sauvegarde, nombre maximum par Backup Server 915
- devices miroir 602
- devices secondaires
 - sauvegarde à l'aide de **quiesce database** 862
- devise
 - formats locaux 315
- dictionnaire de données
 - voir* tables système
- didacticiel de création de segments 753–757
- disques
 - partitionnés 595
 - voir* devices de base de données; devices; devices de sauvegarde
- distinction majuscules/minuscules
 - dans SQL xxxv
- données
 - voir aussi* autorisations
 - confidentialité 534
 - cryptage 534
 - intégrité 534, 550
 - non consignées dans le journal et perdues 728
 - paquets 532
 - données confidentielles 534
 - données de caractères ASCII 7 bits, conversion de jeux de caractères de 320
 - données de type entier
 - dans SQL xxxvii
 - données en virgule flottante xxxvii
 - données non consignées dans le journal et perdues 728
 - droits
 - voir* autorisations
 - dscp** utilitaire pour la spécification du mécanisme de sécurité 544
 - DTP (Distributed Transaction Processing) 31
 - dump database
 - striping de sauvegarde 917
 - durée
 - pour acquérir des verrous 153
- E**
- E/S
 - configuration de la taille 650–654
 - erreurs 962
 - évaluation des coûts 270
 - évaluation du coût 268–271
 - informations statistiques 270
 - mise en miroir des disques 599
 - restriction 265
 - restriction au moment de la pré-exécution 266
 - statistiques d'utilisation 416
- E/S asynchrone
 - activation 120
 - limitation des requêtes du serveur pour 173
 - mise en miroir de device 599
- E/S disque
 - chargements de bases de données et 105

Index

- device en miroir 597
- mémoire 633
- paramètres de configuration pour 120
- échange de clés
 - clé publique/privée 341
 - clé symétrique 341
 - cryptage 341
- éléments de date
 - autre langue 311
- ellipse dans les instructions SQL (...) xxxvi
- emprunt d'identité
 - voir* commande **setuser**
- enregistrements, audit 478
- entrées de répertoire, création 350
- environnement multi-utilisateur, fractionnement de tables 745
- erreur allocation et correction avec **dbcc** 791
- erreur interne, non fatale 61
- erreurs
 - voir aussi* journaux d'erreurs; messages d'erreur
 - allocation 788, 791
 - conversion de caractères 325
 - correction avec **dbcc** 791
 - entrée/sortie 962
 - fatales 61–63
 - journalisation 55
 - multiples 52
 - numéros d'état 51
 - rapport 64
 - réponses du serveur 51–63
 - segmentation 962
 - types d'informations journalisées 15
 - utilisateur 58, 58–61
- erreurs de débordement
 - pile du serveur 250
- erreurs de l'utilisateur
 - voir* erreurs; degrés de sévérité, erreur
- erreurs de ressources insuffisantes (degré 17) 60
- erreurs de segmentation 962
- erreurs de trace
 - voir* journaux d'erreurs
- erreurs fatales
 - degrés de sévérité 19 et au-delà 61–63
 - messages d'erreur 61–63
 - trace du noyau 55, 61
- erreurs utilisateur 58, 58–61
 - voir* erreurs; degrés de sévérité, erreur
- erreurs utilisateur diverses 59
- espace
 - voir aussi* taille; allocation d'espace
 - ajout à une base de données 709
 - effet sur **sp_dropsegment** 749
 - entre seuils 1005–1006
 - évaluation de la taille des tables et des index 702
 - extension d'une base de données 709
 - informations sur l'utilisation 716, 946
 - insuffisant 60, 729, 935
 - non réservé 717
 - partage entre les segments 736
 - réservé 717
 - taille du journal de la base de données 703
- espace du device de base de données
 - voir* segments; allocation d'espace
- espace insuffisant
 - voir* espace
- espace libre, segment de journal et 986–1012
- espaces de travail
 - suppression 819
- espagnol
 - jeux de caractères supportés 289
- et (&)
 - conversion en trait de soulignement dans les noms de login 549
- étapes
 - administration de la sécurité 332
- état
 - messages d'information (degré 10) 58
- état de retour
 - procédures système 13
- état de statistiques d'utilisation 416
- étendue 444
- étiquette bande ANSI 940
- étiquettes
 - volumes de sauvegarde 931
- étiquettes de bande
 - informations sur les fichiers sauvegardés 852
- étiquettes HDR1 913
- étiquettes, device
 - voir* segments
- Europe de l'Est
 - jeux de caractères supportés 289

- Europe de l’Ouest
 - jeux de caractères supportés 289
 - exclusivité mutuelle des rôles 339
 - exécution
 - ESP et priorité XP Server 137
 - limites d’utilisation des ressources et 266
 - expiration des mots de passe 241, 365
 - exploitation des volumes 907
 - expression de type caractère xxxvii
 - expressions
 - types de xxxvi–xxxvii
 - expressions binaires xxxvii
 - expressions logiques xxxvi
 - expressions numériques xxxvii
 - extension de segments 741
 - extents
 - allocation d’espace et 775
 - rapport **sp_spaceused** et 717
 - taille d’E/S et 638
- F**
- fermeture transitive par jointure
 - activation au niveau du serveur 202
 - fichier *charset.loc* 313
 - fichier *common.loc* 315
 - fichier d’interface 16, 544
 - Backup Server 871, 901
 - fichier de configuration
 - nom et emplacement par défaut 86
 - spécification au démarrage 91
 - stockage de la valeur configurée 87
 - fichier journal
 - voir* journaux d’erreurs
 - fichier keytab
 - spécification 552
 - spécification pour les programmes utilitaires 564
 - fichier *libtcl.cfg* 18
 - exemple 542
 - outils de modification 541
 - préparation pour la sécurité réseau 539
 - fichier *locales.dat* 315
 - fichier *objectid.dat* 543
 - fichier *server.loc* 315
 - fichiers
 - device miroir 599
 - fichier *libtcl.cfg* 18
 - interfaces 16
 - interfaces et Backup Serveur 901
 - internationalisation 313
 - journal d’erreurs 16, 55
 - localisation 315
 - rapports relatifs aux problèmes résolus (CPR) 79
 - rapports relatifs aux problèmes système (SPR) 79
 - sauvegarde 875
 - suppression 589
 - traduction des jeux de caractères (*.xlt*) 313
 - fichiers CPR 79
 - fichiers SPR 79
 - fichiers *.srt* 313
 - fichiers *.xlt* 313
 - file d’attente d’audit 478, 490
 - fillfactor
 - paramètre de configuration **default fill factor percent** 200
 - filtre SSL
 - définition 344
 - fonction
 - lct_admin**, option **reserve** 993–995
 - fonction de sécurité **is_sec_service_on** 568
 - fonction de sécurité **show_sec_services** 568
 - fonction système
 - mut_excl_roles**, exclusivité mutuelle des rôles 411
 - proc_role**, procédures stockées et 411, 468
 - role_contain** 410
 - show_role** 410
 - suser_id** 407–408
 - suser_name** 407–408
 - user_id** 408
 - user_name** 408
 - fonctions de contexte applicatif 452
 - fonctions Java définies par l’utilisateur 448
 - formats
 - date, heure et devise 315
 - paramètres locaux, non supportés 311
 - paramètres régionaux, non supportés 312
 - formats de fichiers de sauvegarde, compatibilité entre versions 912
 - formules
 - exigences utilisateur et 246

Index

fragments de device 576
fragments, espace sur un device 576, 712, 753
français
 jeux de caractères supportés 289
fréquence
 point de reprise automatique 827

G

gestion d'utilisateurs
 voir utilisateurs
gestion des transactions distribuées 31
gestion des transactions distribuées (DTM) 125
gestion du stockage 569
 voir aussi espace; allocation d'espace
 création de bases de données utilisateur 697–709
 devices de base de données par défaut 589–591
 informations 717
 initialisation du device de base de données 579–588
 mise en miroir des disques 593–603
 modification de la propriété d'une base de données
 708
 problèmes 39–42
 récapitulatif des commandes 570
 sujets 572, 737
 suppression de bases de données 711
 tables système et 574–577
 utilisation de segments 736–757
 valeurs par défaut à l'installation 574
gestion, espace
 voir allocation d'espace; gestion du stockage
gestionnaires de répertoire 539
 exemple d'entrée dans le fichier *libtcl.cfg* 542
gestionnaires de réseau 539
 exemple d'entrée dans le fichier *libtcl.cfg* 543
 syntaxe du fichier *libtcl.cfg* 539
gestionnaires de sécurité
 exemple d'entrée dans le fichier *libtcl.cfg* 543
 syntaxe des entrées dans le fichier *libtcl.cfg* 540
gestionnaires de site 532
get_appcontext 453
grec
 jeux de caractères supportés 289
groupe d'utilisateurs
 voir groupes; groupe public

groupe public 374
 voir aussi groupes
 autorisations 427, 438
 autorisations de l'utilisateur "guest" 377
 commande **revoke** et 430
 grant et 430, 435, 457
 procédure **sp_adduser** 376
 sp_changegroup et 399
groupes
 voir aussi groupe public
 commande **revoke** et 434
 conflit d'autorisations et 437
 création 374–375
 grant et 433
 modification 399
 nom 374
 suppression 393
groupes de langues 288, 289

H

hachage
 condensé de messages 341
 définition 341
haute disponibilité
 paramètre **enable HA** 207
 script *installhasvss* 207
 script *insthasv* 207
hébreux
 jeux de caractères supportés 289
hiérarchie de rôles 339
 affichage 411
 affichage avec **sp_displayroles** 410
 création 439
 voir hiérarchie des rôles
hiérarchie des autorisations
 voir autorisations
historique, suppression de la base de données *dbccdb*
 819

- I**
- ID de login, nombre de 379
 - ID utilisateur 421
 - affichage 406
 - comparaison après sauvegarde et reprise 881
 - comparaison après sauvegarde et restauration 975
 - numéro 1, propriétaire de la base de données 14
 - recherche 407
 - ID, intervalle de temps 259
 - ID, utilisateur 407, 421
 - procédures système et 14
 - identificateurs de versions, mise à jour automatique et 955
 - identification concrète 432
 - identification et authentification
 - voir aussi* connexions
 - contrôles 337
 - identité d'utilisateur
 - voir* alias; connexions; utilisateurs
 - identités
 - autorisations de session et 456
 - autre 402
 - procurations et 456
 - index
 - allocation à des segments spécifiques 737
 - changements de l'ordre de tri 308
 - changements des jeux de caractères 309
 - colonne IDENTITY dans index non unique 726
 - liaison à des caches de données 655
 - modification de l'ordre de tri 784
 - orientés caractères 307
 - positionnement sur un device unique 737
 - pourcentage **default fill factor percent** pour 200
 - reconstruction 308, 879
 - sauvegarde de la base après création 879
 - suspects 62, 308
 - table d'allocation d'objets (Object Allocation Map) de 777
 - tables d'allocation d'objets (Object Allocation Maps) 112
 - index clusterisés
 - segments et 748, 757
 - transfert de tables vers 748, 757
 - index non clusterisés
 - transfert sur un autre device 745
 - index suspects
 - mise en ligne forcée 840
 - suppression 840
 - indicateur de trace **T1204** (appelé maintenant paramètre de configuration **print deadlock information**) 224
 - indicateur de trace **T1603** (appelé maintenant paramètre de configuration **allow sql server async i/o**) 120
 - indicateur de trace **T1610** (appelé maintenant paramètre de configuration **tcp no delay**) 172
 - indicateur de trace **T1611** (appelé maintenant paramètre de configuration **lock shared memory**) 186
 - informations (serveur)
 - alias d'utilisateur 404
 - autorisations 460–464
 - cache de données 641
 - device de sauvegarde 587, 876
 - devices 587
 - devices de base de données 587, 715
 - limites d'utilisation des ressources 276–278
 - logins 407
 - logins du serveur distant 530
 - logins verrouillés 395
 - messages d'erreur 53–63
 - modification utilisateur 397–400
 - noms de device 876
 - options de base de données 722–723
 - paramètres de configuration 90
 - problèmes 55
 - résultat **dbcc** 798
 - segments 715–719, 749, 793
 - serveurs distants 523
 - seuils 998
 - stockage d'une base de données 712
 - taille de base de données 701, 717
 - utilisateurs, base de données 405–417
 - utilisation de l'espace 715–719
 - informations confidentielles, vues 466
 - informations de configuration, suppression dans *dbccdb* 819
 - initialisation
 - devices de base de données 579–587
 - mis en miroir des disques 599
 - insertions
 - récupération d'espace pendant 762

Index

- installation
 - bases de données exemples 31
- installation, serveur
 - état après 574
 - fichier d'interface 17
 - mise en place de la sécurité 336
 - mise en place de la sécurité après 334
 - système d'audit 481
- intégrité référentielle
 - mémoire 636
- interblocages 59
 - balayages décroissants et 194
- internationalisation
 - avantages 284
 - définition 283
 - exemple de système 285
 - fichiers 313
 - structure des répertoires de jeux de caractères 314
- intervalle d'expiration des mots de passe 365
- intervalles de temps 258
 - "at all times" 258
 - ajout 259
 - chevauchement 258
 - création 259
 - modification 260–261
 - modification des intervalles actifs 262
 - priorité 282
 - restriction 265
 - sauvegardes de bases de données 878
 - suppression 261
 - suppression des limites d'utilisation des ressources 280
 - utilisation 258–262
- intervalles de temps nommés 258
 - "at all times" 258
 - ajout 259
 - chevauchement 258
 - création 259
 - modification 260–261
 - modification des intervalles actifs 262
 - priorité 282
 - suppression 261
 - suppression des limites d'utilisation des ressources 280
 - utilisation 258–262
- ISO 8859-1
 - similitudes avec CP 1252 289
- isolement des pannes en cours de reprise 835–845
- J**
- japonais
 - jeux de caractères supportés 289
- JDBC 448
- jeu de caractères
 - définition 288
 - jeu de caractères cp437 144
 - jeu de caractères cp850 144
 - jeu de caractères deckanji 144
 - jeu de caractères eucjis 144
 - jeu de caractères IBM 144
 - jeu de caractères iso_1 144
 - jeu de caractères Macintosh 144, 325
 - jeu de caractères Microsoft 144
 - jeu de caractères multilingue 144
 - jeu de caractères roman8 144
 - jeu de caractères sjis (Shift-JIS)
 - voir* jeux de caractères japonais
 - jeu de caractères sun 144
 - jeu de caractères UNIX étendu 144
 - jeux de caractères 144
 - voir aussi* jeux de caractères japonais
 - arabe 289
 - balte 289
 - changement 303
 - chemins de conversion supportés 318–325
 - chinois simplifié 289
 - chinois traditionnel 289
 - codage dans différents 317
 - codés sur plusieurs octets 309
 - codés sur plusieurs octets, changement 311
 - conversion entre client et serveur 318–320
 - conversion entre client et système de fichiers 329
 - conversion entre client et terminal 329
 - coréen 289
 - erreurs de conversion 325
 - Europe de l'Est 289
 - Europe de l'Ouest 289
 - fichiers de conversion, propres au terminal 313, 329

- fichiers de définition 313
 - grec 289
 - hébreux 289
 - japonais 289
 - mise à niveau des valeurs de type text après
 - changement 309
 - numéro d'ID 144
 - pour les groupes de langues 289
 - réindexation après configuration 307–311
 - russe 289
 - sauvegardes de base de données et 930
 - script en cyrillique 289
 - symbole de la devise européenne et 290
 - thaï 289
 - turc 289
 - Unicode 290
 - valeurs par défaut 290
 - vietnamien 290
 - jeux de caractères codés sur plusieurs octets 309
 - changement 311
 - incompatibles 325
 - paramètre de configuration **default character set id** 144
 - jeux de caractères japonais 144
 - voir aussi* langues, autres
 - sjis (Shift-JIS) 144
 - jointures
 - vues et 466
 - jointures par fusion
 - activation au niveau du serveur 202
 - désactivation au niveau du serveur 202
 - journal courant
 - voir* journaux de transactions
 - journal de préécriture
 - voir* journaux de transactions
 - journal de transactions *syslogs* de *sybsecurity* 491
 - journalisation
 - connexions réussies 136
 - échecs de connexion 136
 - journalisation des événements de Windows NT 134, 135
 - journaux
 - voir* journaux de transactions
 - journaux d'erreurs 47, 61, 630
 - contrôle de la taille des caches 630
 - création et propriété 55
 - emplacement 15
 - format 56
 - troncature 55
 - journaux de transactions
 - voir aussi* sauvegarde, journal de transactions; commande **dump transaction**; table *syslogs*
 - alter database** et 576
 - base de données *master* 881
 - base de données *model* 882
 - bases de données primaire et secondaire 726
 - caches de données et 648
 - caches et 648
 - commandes non consignées 879
 - copie 827
 - create database** et 576, 702
 - déplacement pour libérer de l'espace 706
 - device distinct 594, 847
 - espace disponible 827
 - espace insuffisant 852
 - fonction 826
 - initialisation après points de reprise 829
 - modifications entre chargements 949
 - option de base de données **select into/bulkcopy/plisort** 727
 - option **trunc log on chkpt** 107, 728–729
 - placement d'un device 573, 576, 702, 706, 707
 - purge 935
 - sauvegarde 847
 - sauvegarde consécutive à une panne de disque 932
 - sur le même device 851, 934
 - synchronisation avec la base de données 827–830
 - taille 703, 729, 827
 - troncature 310, 934–936
 - vérification de l'espace utilisé 703
 - jours
 - autre langue 311
- ## K
- kanji
 - voir* jeux de caractères japonais

Index

L

- langages de manipulation des données 441
 - langue par défaut 373
 - changement de la langue de l'utilisateur 307
 - langue us_english (anglais américain) 144
 - langues
 - supportées par un jeu de caractères 288
 - sur le serveur 288
 - langues par défaut 144
 - us_english (anglais américain) 144
 - langues, autres 313
 - voir aussi* jeux de caractères; *charset.loc* fichier; jeux de caractères japonais
 - fichiers de localisation 299–316
 - formats de date dans les langues non supportées 311
 - langues supportées 284
 - LDAP
 - définition 18
 - et fichier d'interface 20
 - restrictions d'accès 19
 - services de répertoire multiples 19
 - lecture
 - physique 573, 593
 - liaison utilisateurs
 - voir* alias, utilisateur
 - lien, page 779, 784
 - voir aussi* page, données
 - lier 444
 - lignes de données
 - contrôle avec les commandes **dbcc** 784
 - lignes renvoyées
 - commande **reorg** 762–765
 - élimination par **reorg forwarded_rows** 762–765
 - réduction avec le paramètre de configuration **default exp_row_size** 201
 - lignes, table
 - restriction du renvoi 265, 272
 - sysindexes* 577
 - limites d'utilisation des ressources 255
 - activation 257
 - configuration 195
 - création 273–275
 - exemples d'informations 277
 - exemples de création 275
 - exemples de modification 279
 - exemples de suppression 281
 - identification des utilisateurs et des limites 262–268
 - informations 276–278
 - modification 278
 - moment d'application 266
 - planification de !@#\$ 256
 - portée 266
 - pré-exécution et exécution 266
 - présentation des types de limite 268–273
 - priorité 282
 - restriction du coût des E/S 268–271
 - suppression 280–281
- liste
 - fichiers sauvegardés sur bande 929–932
 - messages **sp_volchanged** 940–942
 - load database
 - sauvegardes compressées 899
 - load transaction
 - fichiers compressés 899
 - localisation 284
 - voir aussi* langues, autres
 - fichiers pour 315
 - lock scheme
 - par défaut 152
 - login par défaut sécurisé 548
 - login "sa" 334, 969, 974
 - configuration avec les rôles d'administrateur système et de responsable de la sécurité du système 334
 - modification du mot de passe 334
 - mot de passe 966
 - recommandations de sécurité pour l'utilisation 334
 - logins
 - alias 403
 - déverrouillage 394
 - informations sur 407
 - mot de passe "sa" 966
 - recherche 407
 - "sa" 969, 974
 - suppression 396
 - verrouillage 394
 - logins distants
 - ajout 525–527
 - mode sécurisé et non sécurisé 526
 - options pour 529
 - paramètres de configuration pour 531–532

- suppression 524, 525
 - temporisation 522
- M**
- mappage
 - du nom du device au nom physique 580
 - utilisateurs distants 524
 - marques de réservation
 - pourcentage (%) dans les messages d'erreur 54
 - marqueur de fin de bande 905
 - matériel
 - annulation de la mise en miroir 600
 - erreurs 63
 - mécanisme de sécurité CyberSAFE Kerberos 534, 545
 - mécanisme de sécurité DCE 534, 545
 - mécanisme de sécurité Kerberos 534, 545
 - mécanisme de sécurité LAN Manager 534, 545
 - mécanisme de sécurité NT LAN Manager 534, 545
 - mécanisme de sécurité Windows NT LAN Manager 534, 545
 - mécanismes de protection
 - voir* fonctions de sécurité; procédures stockées; vues
 - mécanismes de sécurité 534, 566
 - détermination du mécanisme supporté par le serveur 552
 - supportés 534
 - mémoire
 - voir aussi* allocation d'espace
 - appels de procédure à distance 636
 - code exécutable 628
 - configuration 607–636
 - connexions utilisateur 631
 - enregistrements d'audit 235, 490
 - intégrité référentielle 636
 - libération de XP Server 138
 - maximisation 607
 - messages du journal d'erreurs 630
 - number of open databases** et 156
 - partagée 682
 - principales utilisations 627–633
 - procédures système utilisées 622–626
 - processus de production 634
 - sécurité réseau et 551
 - segment de mémoire 611
 - serveurs distants 635
 - traitement parallèle 634
 - utilisation par Component Integration Services 628
 - utilisation par le serveur 608
 - mémoire partagée
 - définition pour Backup Server 914
 - quantité à la disposition de chaque device de sauvegarde 915
 - mémoire segmentée 611
 - messages
 - Backup Server 924
 - confidentialité 536, 550
 - définis par l'utilisateur 57
 - définition de la langue 284
 - démarrage 15
 - erreur 15, 53–64
 - erreur fatale 15
 - intégrité 536, 550
 - liste de **sp_volchanged** 940
 - services de protection 535
 - système 53–63
 - vérifications de l'origine 536
 - messages d'erreur 53–64
 - allocation via **tablealloc** 798
 - conversion de caractères 326
 - création de messages définis par l'utilisateur 57
 - définis par l'utilisateur 57
 - degrés de sévérité 57–63
 - erreur d'allocation 791
 - modification des messages du serveur 57, 315
 - numérotation 53
 - pour les erreurs fatales 61–64
 - seuils et 1007
 - utilisation de la mémoire 630
 - messages d'information (serveur)
 - voir* messages d'erreur; degrés de sévérité
 - messages système
 - voir* messages d'erreur; essais
 - mise à jour
 - voir aussi* modification
 - page courante du journal de transactions 706
 - paramètre de configuration **allow updates to system tables** et 14

Index

- procédures système et 468
- tables système 967
- texte après changement du jeu de caractères 309
- mise à niveau, restauration après 970
- mise en miroir des disques 593–606
 - activation 121
 - annulation de la mise en miroir 600
 - désactivation 121
 - didacticiel 603
 - E/S asynchrone 599, 602
 - état dans la table *sysdevices* 588
 - impact sur *sysdevices* 601, 603–606
 - initialisation 599
 - redémarrage 602
 - restauration et 573
- mise en miroir, désactivation
 - voir* commande **disk unmirror**
- mises à jour directes
 - des tables système 196
- mode mono-utilisateur 196, 307, 964
- mode non sécurisé, logins distants et 529
- mode restauration de master 966
- mode sécurisé
 - logins distants et 529
- modèles de sécurité 555
 - configuration du modèle B pour RPC 558
 - exemple du modèle B 561
 - modèle B 559
 - pour les RPC 556
- modification
 - voir aussi* mise à jour
 - allocation d'espace
 - après affectation, allocation d'espace, modification 702
 - après affectation, modification 709
 - base de données par défaut 399
 - connexions au serveur 399
 - groupe d'un utilisateur 399
 - identité de l'utilisateur 455
 - informations utilisateur 397–400
 - intervalles de temps nommés 260–261
 - limites d'utilisation des ressources 278
 - matériel 600
 - mots de passe comptes de connexion 398
 - options de base de données 722–731
 - ordre de tri 787
 - paramètres de configuration 101, 531
 - propriétaire de base de données 426
 - propriétaire de bases de données 708
 - seuils 999
 - tables système, risques 11, 14, 967
 - taille de base de données 709
 - voir* modification; mise à jour
- monopolisation de buffer, taille de la zone de vidage et 661
- mot-clé
 - all, grant** 429, 435
 - all, revoke** 435
 - membership, alter role** 386
 - null**, dans **sp_addlogin** 373
 - on, alter database** 710
 - on, create database** 700, 702
 - on, create index** 743
 - on, create table** 743
 - on, grant** 430
 - on, revoke** 430
 - public, grant** 435, 457
- moteur de serveur
 - voir* moteurs
- moteurs 682
 - désactivation d'un moteur à l'aide de la commande **dbcc engine** 687
 - fonctions et planification 682
 - gestion 684–689
 - nombre de 190, 684
 - numéros d'identification 56
- mots de passe 397
 - affichage des informations 361
 - choix 371
 - choix sécurisé 371
 - cryptage sur le réseau 522
 - date de dernière modification 406
 - expiration 365
 - intervalle d'expiration 365
 - longueur minimale 363
 - modification 398
 - null 398, 966
 - oubli 423
 - pour les rôles 365
 - protection 371
 - protection contre les mots de passe devinés 357
 - recherche d'au moins un caractère 362
 - règles pour 371

- rôles et 391
 - sp_password** 397
 - utilisateurs distants 522, 529
 - mots de passe null 398, 966
 - MSDTC 127
- N**
- names
 - partiel, pour la spécification de l'option 731
 - négociation SSL
 - SSL 343
 - niveau d'isolement
 - lecture au niveau 0 726
 - nom
 - fichiers de sauvegarde 908–911
 - groupes 374
 - rôles définis par l'utilisateur 384
 - nom de colonne
 - non qualifié 59
 - nom device
 - device de sauvegarde 947
 - nom du device 581
 - device de sauvegarde 875
 - liste *sysdevices* 576
 - nom logique pour device physique 877
 - nom et ID d'utilisateur du serveur 407
 - nom utilisateur "dbo" 2, 4
 - nombre (quantité de)
 - bases de données ouvertes sur le serveur 156
 - connexions utilisateur (@@*max_connections*) 246
 - d'ID de login 379
 - device de sauvegarde 917
 - devices de base de données 122, 633
 - extents 775
 - lignes renvoyées 265, 272
 - moteurs 189, 685
 - moteurs de système SMP 685
 - objets ouverts 160
 - secondes pour acquérir des verrous 153
 - segments 734
 - sites distants 532
 - threads de service de Backup Server 914
 - utilisateurs 379
 - verrous 147
 - nombre de connexions réseau pour Backup Server 916
 - nombre limite de devices de sauvegarde pour Backup Server 913
 - nombre maximal de tentatives de connexion 357
 - nombres
 - valeur de segment 946
 - noms
 - voir aussi* informations (serveur); connexions
 - alias 403, 432, 455
 - applications 262
 - colonnes, dans les commandes 59
 - correspondance de l'utilisateur distant 525
 - groupe 430
 - identité initiale 455
 - pour les connexions 335
 - procédures stockées étendues système 14
 - procédures système 12
 - recherche d'utilisateur 407
 - segment 740
 - serveur 521
 - serveur distant 519
 - utilisateur distant 525
 - utilisateurs 375, 407, 427, 430
 - noms d'utilisateurs 427
 - modification 399
 - préférences 375
 - recherche 407
 - noms de fichiers
 - sauvegarde de base de données 908–911
 - sauvegardes du journal de transactions 908–911
 - noms de login
 - voir* connexions
 - noms de suffixe, table temporaire 29
 - noms logiques 875
 - non-répudiation, signature électronique 341
 - noyau
 - messages d'erreur 55, 61
 - utilisation de la mémoire 628
 - numéro
 - device virtuel 714
 - page virtuelle 584
 - numéros
 - bit d'état (*sysdevices*) 588
 - device 714
 - device virtuel 714

Index

message d'erreur 53
moteur 56
ordre de tri 145, 243
valeur d'un segment 714

O

objectid.dat
 emplacement 351
objets
 voir objets de base de données
objets compilés
 buffers de procédures 631
objets d'utilisateur
 voir objets de base de données
objets de base de données
 voir aussi noms d'objets individuels
 allocation à des devices 572, 736, 742
 amélioration des performances et 737
 autorisations 427
 autorisations d'accès 5, 429
 contrôle de la création par les utilisateurs 26, 880
 création 26, 427, 572
 dépendants 470
 erreurs affectant les 62
 espace utilisé 717
 nombre maximum ouverts 160
 placement sur des segments 736, 742, 745, 949
 propriété 5, 393, 427
 recherche 58
 suppression 427, 712
 suppression d'utilisateurs propriétaires 393
 suppression de segments et 748
 triggers 473
octets
 buffers de procédures 631
 capacité de bande 905
 caractère 325
 taille de bloc 904
octroi
 autorisation d'accès 5
 autorisation de procuration 457
 autorisations de création d'un objet 5
 droit de création de triggers 431
 rôle avec **grant role** 439
opération d'écriture
 mise en miroir des disques 593
opération non journalisée 728
opérations automatiques
 conversions des caractères dans les logins 549
 points de reprise 828
 reprise 831
 sauvegarde de la base de données primaire dans la
 base de données secondaire 726
opérations d'écriture
 physique 573
optimisation
 contrôle des performances 101
optimiseur 638
option **ansi_permissions, set**
 autorisations et 432
option **at** 900
 striping de sauvegarde et 920
option base de données **trunc log on chkpt
recovery interval in minutes** et 107
option **blocksize** 904
option **capacity** 905
option **cascade, revoke** 431
option **checkalloc, dbcc** 776, 788, 794
option **checkcatalog, dbcc** 749, 793, 794
option **checkdb, dbcc** 787, 794
option **checkstorage, dbcc** 781, 794
 vérification des erreurs 802
option **checktable, dbcc** 308, 784, 794
 option fix_spacebits 785
 taille du journal de transactions et 704
option **checkverify, dbcc** 802–804
option **cntrltype**
 disk init 587
option **dbrepair, dbcc** 805, 947
 drop database et 805
option de base de données 722–732
 affichage 722
 affichage des valeurs 723
 modification 730
 valeur 723–729
option de base de données **abort tran on log full** 723,
 997
option de base de données **allow nulls by default** 723
option de base de données **auto identity** 724
option de base de données **dbo use only** 724

- option de base de données **ddl in tran** 724
- option de base de données **identity in nonunique index** 726
- option de base de données **no chkpt on recovery** 726
- option de base de données **no free space acctg** 727, 1012
- option de base de données **read only** 308, 727, 730
- option de base de données **select into/bulkcopy/pllsort**
 - base de données *model* et 28
 - journal de transactions 727
- option de base de données **single user** 728
- option de base de données **trunc log on chkpt** 728–729
- option de base de données **unique auto_identity index** 729
- option **defaulton | defaultoff, sp_diskdefault** 589
- option **density** 903
- option **drop logins, sp_dropserver** 524
- option **dropdb, dbcc dbrepair** 805, 947
- option **dsync**
 - disk init** 584–586, 588, 701
- option **dumpvolume** 907
- option **engine, dbcc** 687
- option **fast**
 - dbcc indexalloc** 790, 791, 792, 794
 - dbcc tablealloc** 792, 794
- option **file** 909
- option **fix**
 - dbcc** 791
 - dbcc checkalloc** 788
 - dbcc indexalloc** 792
 - dbcc tablealloc** 798
 - utilisation en mode mono-utilisateur 791
- option **fix_spacebits**
 - dbcc checktable** 785
- option **fix_text, dbcc** 309–310
- option **for load**
 - alter database** 711
 - create database** 706
- option **full**
 - dbcc indexalloc** 790, 791, 792, 794
 - dbcc tablealloc** 792, 794
- option **grant**
 - sp_helprotect** 462
- option **grant option for, revoke** 431
- option **headeronly** 852, 929–932
- option **indexalloc, dbcc** 790, 794
- option **init** 920–924
- option **listonly** 852, 929–932
- option **local, sp_addserver** 520
- option **log on**
 - alter database** 710
 - create database** 576, 702, 705
- option **mode, disk unmirror** 600
- option **net password encryption** 522
- option **no_log, dump transaction** 935
- option **no_truncate, dump transaction** 932–933
- option **nodismount** 920
- option **nofix, dbcc** 791
- option **noserial, disk mirror** 599
- option **notify** 925, 926
- option **nounload** 920–922
- option **primary, disk unmirror** 600
- option **reclaim_space**, commande **reorg** 763
- option **remove, disk unmirror** 600, 601
- option **retain, disk unmirror** 600
- option **retaindays** 920–922
 - dump database** 109, 874
 - dump transaction** 109, 874
- option **secondary, disk unmirror** 600
- option **serial, disk mirror** 599
- option serveur **mutual authentication** 557
- option serveur **security mechanism** 557
- option serveur **use message confidentiality** 557
- option serveur **use message integrity** 557
- option **session authorization, set** 458
- option **showplan, set**
 - limites d'utilisation des ressources et 264, 265
- option **side, disk unmirror** 600
- option **statistics io, set**
 - limites d'utilisation des ressources et 264, 265
- option **statistics time, set**
 - détermination du temps de traitement 271
 - limites d'utilisation des ressources et 264, 265
- option **stripe on** 917–920
- option **tablealloc, dbcc** 790, 794
- option **timeouts, sp_serveroption** 522
- option **truncate_only, dump transaction** 934
- option **unload** 920–922
- option **vstart**
 - disk init** 586
- option **with grant option, grant** 431

Index

- option **with no_log, dump transaction** 935
 - option **with no_truncate, dump transaction** 932–933
 - option **with nowait, shutdown** 77, 78
 - option **with override**
 - create database** 707
 - drop role** 394
 - option **with truncate_only, dump transaction** 934
 - option **writes, disk mirror** 599
 - options
 - base de données 722, 722–732
 - chaîne unique 731
 - logins distants 529
 - serveur 521
 - serveurs distants 521
 - options d'audit
 - affichage 479
 - définition 499
 - exemples 500
 - options d'information du serveur
 - voir* information (serveur)
 - options par défaut
 - bases de données système à l'installation 574
 - changement de l'ordre de tri 304–309
 - changement du jeu de caractères 303–311
 - taille de base de données 701
 - ordre de reprise
 - bases de données 832–834
 - bases de données système et 832
 - ordre de tri
 - changement 304–307
 - cohérence entre les serveurs 304
 - dbcc checktable** et 787
 - default sortorder id** 145, 243
 - fichiers de définition 313
 - installation 313
 - modification 787
 - numéros 145
 - reconstruction des index après changement 308
 - sauvegardes de base de données et 930
 - ordre de tri binaire des jeux de caractères
 - changements des jeux de caractères et sauvegardes des bases de données 305
 - dbcc checktable** et 787
 - ordre des commandes
 - contrôle par **dbcc** au niveau objet 797
 - création d'index clusterisé et 745
 - instructions **grant** et **revoke** 428–440
 - sauvegarde du journal et de la base de données 728
 - overhead
 - code exécutable 628
 - Component Integration Services et 628
 - mémoire 628
- ## P
- pages corrompues
 - affichage 838
 - estimation 845
 - isolement lors de la reprise 835–845
 - pages d'allocation
 - dbcc tablealloc** et 792
 - pages d'allocation 580, 712, 774
 - pages hors ligne 836
 - affichage 838
 - incidences 840
 - pages modifiées 107, 827
 - pages OAM (Object Allocation Map) 777
 - contrôle avec les commandes **dbcc** 787, 790, 792
 - pages suspectes
 - affichage 838
 - estimation 845
 - isolement lors de la reprise 835–845
 - pages, données 580
 - allocation 712, 774
 - début (*lstart*) 714
 - gestion avec les extents 775
 - lien dans une table ou un index 779, 784
 - modifiées 107, 827
 - numérotation dans la base de données 714
 - remplissage du journal de transactions 705, 935
 - taille de bloc et 904
 - pages, OAM (Object Allocation Map) 777
 - panne de device 846
 - sauvegarde du journal de transactions après 886, 932
 - panne disque 63
 - base de données utilisateur 595
 - copie du device du journal après 847, 932–933
 - device master 595
 - diagnostic 944
 - reprise et 846

- paquets, réseau
 - lecture anticipée 532
 - taille, configuration 167–169
- paramètre de configuration
 - abstract plan cache** 192
 - abstract plan dump** 192
 - abstract plan load** 193
 - abstract plan replace** 193
 - additional network memory** 184
 - allow backward scans** 193
 - allow nested triggers** 195
 - allow procedure grouping** 233
 - allow remote access** 164, 531
 - allow remote access**, Backup Server et 873
 - allow resource limits** 195
 - allow sendmsg** 165
 - allow sql server async i/o** 120
 - allow updates** (maintenant appelé **allow updates to system tables**) 14, 195
 - allow updates to system tables** 14, 196
 - audit queue size** 235, 478, 490
 - auditing** 234, 493
 - calignment** (maintenant appelé **memory alignment boundary**) 110
 - cckrate** (maintenant appelé **sql server clock tick length**) 227
 - cfgcprot** (maintenant appelé **permission cache entries**) 248
 - cguardsz** (maintenant appelé **stack guard size**) 249
 - cindextrips** (maintenant appelé **number of index trips**) 111
 - cis bulk insert batch size** 115
 - cis connect timeout** 116
 - cis cursor rows** 116
 - cis packet size** 117
 - cis rpc handling** 118
 - cmaxnetworks** (maintenant appelé **max number network listeners**) 169
 - cmaxschedules** (appelé maintenant **i/o polling process count**) 212
 - cnalarm** (maintenant appelé **number of alarms**) 216
 - cnblkio** (appelé maintenant **disk i/o structures**) 121
 - cnmaxaio_engine** (maintenant appelé **max async i/os per engine**) 173
 - cnmaxaio_server** (maintenant appelé **max async i/os per server**) 174
 - cnmbox** (maintenant appelé **number of mailboxes**) 220
 - cnmsg** (maintenant appelé **number of messages**) 220
 - coamtrips** (maintenant appelé **number of oam trips**) 112
 - configuration file** 140, 141, 142, 143
 - cprealloctxt** (maintenant appelé **number of pre-allocated extents**) 221
 - cpu accounting flush interval** 197, 416
 - cpu flush** (maintenant appelé **cpu accounting flush interval**) 197
 - cpu grace time** 198
 - cschedspins** (appelé maintenant **runnable process search count**) 224
 - csortbufsize** (maintenant appelé **number of sort buffers**) 222
 - ctimemax** (maintenant appelé **cpu grace time**) 198
 - current audit table** 236, 485
 - database size** (maintenant appelé **default database size**) 199
 - deadlock checking period** 148
 - deadlock retries** 149
 - default character set id** 144
 - default database size** 199, 701
 - default exp_row_size percent** 201
 - default fill factor percent** 200
 - default language** (maintenant appelé **default language id**) 144
 - default language id** 144
 - default network packet size** 165
 - default sortorder id** 145, 243
 - devices** (maintenant appelé **number of devices**) 122
 - disable character set conversions** 145
 - disable disk mirroring** 121
 - disk i/o structures** 121
 - dtm detach timeout period** 125
 - dtm lock timeout period** 125
 - dump on conditions** 202
 - enable cis** 118, 119, 203, 237
 - enable DTM** 127

Index

- enable housekeeper GC 208
- enable java 141, 142
- enable rep agent threads 191
- enable row level access control 203
- enable sort-merge joins and JTC 202
- enable unicode conversions 323
- enable xact coordination 127
- esp execution priority 137
- esp execution stacksize 137
- esp unload dll 138
- event buffers per engine 204
- event log computer name 134
- event logging 135
- executable code size + overhead 155
- expand_down, sp_activeroles 411
- fillfactor (maintenant appelé **default fill factor percent**) 200
- global async prefetch limit 109
- global cache partition number 670
- global cache partition number 110
- heap memory per user 186
- housekeeper free write percent 205
- i/o accounting flush interval 211, 417
- i/o flush (maintenant appelé **i/o accounting flush interval**) 211
- i/o polling process count 212
- identity burning set factor 209
- identity grab size 210
- license information 232, 413
- lock address spinlock ratio 146
- lock hashtable size 152
- lock scheme 152
- lock shared memory 185, 186
- lock spinlock ratio 150, 151
- lock table spinlock ratio 154
- lock wait period 153
- locks (maintenant appelé **number of locks**) 147
- log audit logon failure 136
- log audit logon success 136
- max async i/os per engine 173
- max async i/os per server 174
- max cis remote connections 119
- max network packet size 167
- max number network listeners 169
- max online engines 189, 685
- max parallel degree 178
- max roles enabled per user 237, 385
- max scan parallel degree 179
- max SQL text monitored 187
- maximum dump conditions 216
- maximum network packet size (maintenant appelé **max network packet size**) 166
- memory (maintenant appelé **total memory**) 187, 188
- memory alignment boundary 110
- memory per worker process 180
- mrstart (maintenant appelé **shared memory starting address**) 176
- msg confidentiality reqd 237
- msg integrity reqd 238
- nested trigger (maintenant appelé **allow nested triggers**) 194
- number of alarms 216
- number of aux scan descriptors 217
- number of devices 122, 633
- number of dtx participants 128
- number of index trips 111
- number of large i/o buffers 105, 883
- number of locks 147
- number of mailboxes 220
- number of messages 221
- number of oam trips 112
- number of open databases 155
- number of open indexes 158
- number of open objects 160
- number of pre-allocated extents 221
- number of remote connections 170, 532
- number of remote logins 170, 531
- number of remote sites 170, 532
- number of sort buffers 222
- number of user connections 101, 245–247
- number of worker processes 178
- o/s file descriptors 176
- open databases (maintenant appelé **number of open databases**) 155
- open index hash spinlock ratio 161
- open index spinlock ratio 162, 691
- open object spinlock ratio 163
- open objects (maintenant appelé **number of open objects**) 159
- page lock promotion HWM 213
- page lock promotion LWM 214, 231

- page lock promotion PCT** 215
- partition groups** 222
- partition spinlock ratio** 223
- password expiration interval** (maintenant appelé **systemwide password expiration**) 241
- permission cache entries** 248
- pre-read packets** (maintenant appelé **remote server pre-read packets**) 171
- print deadlock information** 224
- print recovery information** 105, 831
- read committed with lock** 153
- recovery flags** (maintenant appelé **print recovery information**) 105
- recovery interval** (maintenant appelé **recovery interval in minutes**) 106
- recovery interval in minutes** 106–108
- recovery interval in minutes**, transactions longues et 106, 828
- remote access** (maintenant appelé **allow remote access**) 164
- remote connections** (maintenant appelé **number of remote connections**) 169
- remote logins** (maintenant appelé **number of remote logins**) 170
- remote server pre-read packets** 171, 532
- remote sites** (maintenant appelé **number of remote sites**) 170
- row lock promotion HWM** 230
- row lock promotion LWM** 231
- row lock promotion PCT** 232
- runnable process search count** 225
- secure default login** 238
- select on syscomments.text column** 239
- shared memory starting address** 176
- size of auto identity column** 226, 724
- size of auto identity column**, option de base de données **unique auto_identity index** 729
- size of global fixed heap** 142
- size of process object fixed heap** 143
- size of shared class heap** 143
- size of unilib cache** 245
- sp_cacheconfig** 671
- sql server clock tick length** 227
- sql server code size** (maintenant appelé **executable code size**) 155
- stack guard size** 249
- stack size** 251
- start mail session** 138
- strict dtm enforcement** 130
- suspend audit when device full** 240, 490
- syb_sendmsg port number** 171, 172
- systemwide password expiration** 241
- tape retention** (maintenant appelé **tape retention in days**) 108
- tape retention in days** 108, 874
- tcp no delay** 172
- text prefetch size** 228
- time slice** 229
- total data cache size** 114
- total memory** 608–618
- txn to pss ratio** 131
- upgrade version** 229
- use security services** 242, 546
- user connections** (maintenant appelé **number of user connections**) 245
- user log cache size** 253
- user log cache spinlock ratio** 254
- xact coordination interval** 132
- xp_cmdshell context** 139
- paramètres de configuration
 - 104–254
 - aide sur 89, 624
 - comptabilisation de refacturation 416
 - connexions distantes et 164
 - dynamiques 87
 - énumération des valeurs des 90
 - Java 141–143
 - limites d'utilisation des ressources 257
 - logins distants et 531–532
 - relatif à l'audit 479
 - statiques 87
 - valeurs par défaut des 86
- paramètres par défaut
 - bases de données 372, 373
 - langue 373
- paramètres, limite d'utilisation des ressources 255
- paramètres, procédure 373
- parenthèses ()
 - conversion en signe dollar (\$) dans les noms de login 549
- partitions
 - disque 581

Index

- partitions de cache 671
 - configuration 110, 670
- performances
 - allocation d'espace et 573, 737
 - commandes **dbcc** 794
 - comptabilisation d'espace libre et 1012
 - configuration de caches et 673
 - effet de **default fill factor percent** sur 200
 - environnement SMP 684–690
 - ESP et priorité XP Server 137
 - mémoire et 607, 608
 - mise en miroir des disques et 572, 596
 - positionnement d'un objet de base de données et 737
 - taille de la file d'attente 235
 - utilisation de segments et 737
 - utilisation du système à multifenêtrage et 608
 - vitesse et 573
- pile de débordement (paramètre de configuration **stack guard size**) 249
- plan de verrouillage
 - par défaut au niveau du serveur 152
- planification, serveur
 - commande **dbcc** 795–798
 - sauvegardes de bases de données 878
- plates-formes UNIX, partition de disque 581
- plus (+)
 - conversion en signe livre (£) dans les noms de login 550
- point (.)
 - conversion en signe dollar (\$) dans les noms de login 549
- point d'exclamation (!)
 - conversion en signe dollar (\$) dans les noms de login 549
- point de reprise
 - option de base de données **no chkpt on recovery** 726
 - option de base de données **trunc log on chkpt** 728–729
- point moyen entre les seuils 1005
- pointeurs, device
 - voir* segments
- points d'interrogation (??)
 - pour les caractères suspects 326
- points de sauvegarde
 - erreur (degré 13) 59
- portée des limites d'utilisation des ressources 266
 - coût des E/S 271
 - décompte de lignes 273
 - temps écoulé 272
- pourcentage (%)
 - conversion en trait de soulignement dans les noms de login 549
 - marque de réservation de message d'erreur 54
- préférences, nom d'utilisateur 375
- prélecture asynchrone
 - configuration 109
 - configuration des limites 663
- priorité
 - caractéristiques de **dump** et **load** 902
 - intervalles de temps 282
 - limites d'utilisation des ressources 282
 - XP Server 137
- problème Halloween
 - éviter avec l'option de base de données **unique auto_identity index** 730
- problèmes connus 79
- problèmes système
 - voir aussi* erreurs
 - degrés de sévérité 10 à 18 58–61
 - degrés de sévérité 19 à 24 61–63
 - rapports relatifs aux problèmes système (SPR) 79
 - réponses du serveur 51–63
- proc headers
 - informations d'en-tête, "proc headers" 631
- procédure cache 62
- procédure stockée catalogue **sp_column_privileges** 464
- procédure stockée catalogue **sp_table_privileges** 464
- procédure stockée étendue système **xp_cmdshell** 14
- procédure système
 - sp_activeroles** 411
 - sp_add_resource_limit** 273
 - sp_add_time_range** 259
 - sp_addalias** 403
 - sp_addauditrecord** 506
 - sp_addgroup** 374
 - sp_addlanguage** 311
 - sp_addlogin** 365, 367, 372–374
 - sp_addlogin**, réexécution après restauration 975
 - sp_addremotelogin** 525–527
 - sp_addsegment** 740, 752

- sp_addsegment**, *sysusages* et 577
- sp_addserver** 519–521
- sp_addthreshold** 999–1005
- sp_addumpdevice** 877
- sp_adduser** 27, 375–379, 699
- sp_audit**, définition des options d’audit 499
- sp_cacheconfig** 641–648
- sp_changedbowner** 426, 696, 708
- sp_changegroup** 374, 399
- sp_configure** 90
- sp_configure**, configuration du serveur pour les services de sécurité 546
- sp_configure**, logins distants et 531
- sp_configure**, reprise automatique et 827
- sp_configure**, *voir aussi les noms des paramètres de configuration individuels*
- sp_countmetadata** 156, 158, 160
- sp_dboption** 722–731
- sp_dboption**, annulation de la mise en miroir des disques 602
- sp_dboption**, annulation de processus 997
- sp_dboption**, désactivation de la comptabilisation de l’espace disponible 1012
- sp_dboption**, modification des options par défaut 699
- sp_dboption**, points de reprise et 830
- sp_dboption**, seuils et 997
- sp_dbrecovery_order** 832–834
- sp_deviceattr** 571, 586
- sp_diskdefault** 571, 589–591
- sp_displaylogin** 406
- sp_displayroles** 410
- sp_drop_resource_limit** 280
- sp_drop_time_range** 261
- sp_dropalias** 404, 708
- sp_dropalias** 432
- sp_dropdevice** 589, 712, 877
- sp_dropdevice**, pour device défectueux 947
- sp_dropgroup** 392, 393
- sp_droplogin** 395, 396
- sp_droplogin**, réexécution après restauration 975
- sp_dropremotelogin** 525
- sp_dropsegment** 748
- sp_dropsegment**, *sysusages* et 577
- sp_dropserver** 524
- sp_droptreshold** 1001
- sp_dropuser** 392, 708
- sp_estspace** 702
- sp_extendsegment** 741
- sp_extendsegment**, inversion des effets 742
- sp_extendsegment**, *sysusages* et 577
- sp_forceonline_db** 839
- sp_forceonline_object** 840
- sp_forceonline_page** 839
- sp_help_resource_limit** 276, 277
- sp_helpcache** 656
- sp_helpconfig** 156, 158, 160, 624
- sp_helpdb** 13
- sp_helpdb**, informations sur le stockage 715
- sp_helpdb**, informations sur les options de base de données 723
- sp_helpdb**, informations sur un segment 751
- sp_helpdevice** 13, 587, 876
- sp_helpindex** 13
- sp_helpjoins** 10
- sp_helpkey** 10
- sp_helplog** 706
- sp_helpremotelogin** 530
- sp_helpprotect** 462–463
- sp_helpsegment** 750, 752
- sp_helpsegment**, contrôle de l’espace disponible 827
- sp_helpserver** 523
- sp_helptext** 13
- sp_helpthreshold** 998
- sp_helpuser** 404
- sp_indsuspect** 308, 309
- sp_listsuspect_db** 838
- sp_listsuspect_object** 840
- sp_listsuspect_page** 838
- sp_locklogin** 395
- sp_locklogin**, réexécution après restauration 976
- sp_logdevice** 705, 737
- sp_modify_resource_limit** 278
- sp_modify_time_range** 260
- sp_modifylogin** 307, 365, 368, 399
- sp_modifylogin**, changement de la base utilisateur par défaut en 372
- sp_modifylogin**, changement du nom complet d’utilisateur 373
- sp_modifythreshold** 999

Index

- sp_monitorconfig** 626
- sp_monitorconfig**, configuration de **number of open databases** et 157
- sp_monitorconfig**, configuration de **number of open indexes** et 159
- sp_monitorconfig**, configuration de **number of open objects** et 161
- sp_password** 397
- sp_placeobject** 747
- sp_remoteoption** 529–530
- sp_reportstats** 416
- sp_reportstats**, limites d'utilisation des ressources et 263
- sp_serveroption** 521, 557
- sp_setsuspect_granularity** 836–838
- sp_setsuspect_threshold** 837
- sp_showplan** 75
- sp_spaceused** 716
- sp_spaceused**, contrôle du journal de transactions 827
- sp_sysmon**, taille de la zone de vidage et 662
- sp_thresholdaction** 986
- sp_thresholdaction**, création 1006–1011
- sp_thresholdaction**, exemple de procédure 1008–1011
- sp_thresholdaction**, messages d'erreur et 1007
- sp_thresholdaction**, paramètres transmis 1006
- sp_thresholdaction**, sauvegarde d'un journal de transactions 1007
- sp_volchanged** 939
- sp_who** 405, 461
- sp_who**, état LOG SUSPEND 997
- sp_who**, processus de point de reprise 828
- procédure système **sp_sysmon**
 - taille de la zone de vidage et 662
- procédures
 - voir* procédures stockées; procédures système
- procédures relatives au seuil
 - autorisations 999, 1000
 - création 1006–1011
 - création, noms logiques et 876
 - emplacement 999, 1011
 - messages d'erreur et 1007
 - paramètres transmis 1006
 - sauvegarde d'un journal de transactions et 1007
 - trace d'audit 485
- procédures stockées
 - voir aussi* objets de base de données; procédures système
 - accès utilisateur distant à 529
 - autorisations 427, 430
 - autorisations sur 529
 - cache de procédure et 113
 - chaînes d'appartenance 469
 - comme mécanismes de sécurité 468
 - contrôle des rôles dans 411
 - création 14
 - liaison de caches et 673
 - modification des tables système et 14
 - octroi d'autorisation d'exécution à des rôles 411
 - octroi d'autorisations 430
 - portée des limites d'utilisation des ressources et 267
 - rôles et 468
- procédures stockées étendues
 - paramètres de configuration 137–140
- procédures stockées étendues système 14
- procédures système 12–14
 - voir aussi* informations (serveur); procédures stockées; *noms de procédures individuelles*
 - autorisations 425
 - création 14
 - de suppression d'alias 432
 - gestion des serveurs distants 523
 - pour ajouter des utilisateurs 370
 - pour la gestion des serveurs distants 519
 - pour modifier les informations utilisateur 397–400
 - sur les tables temporaires 30
 - utilisation 12
- processus
 - nombre maximal de tentatives 357
- processus (tâches serveur) 66, 69, 682
 - voir aussi* serveurs
 - administration d'Adaptive Server 332
 - annulation quand le journal est saturé 997
 - en cours sur le serveur 405
 - informations sur 405
 - suppression 66–69
 - suspension quand le journal est saturé 996
- processus de login
 - authentification 535
- processus de point de reprise 107, 827–830

initialisation des journaux de transactions 829
 journal de transactions et 830
 option base de données **trunc log on chkpt** 107, 829
 paramètre **recovery interval** et 108
 processus point de reprise en veille
 voir processus de point de reprise
 processus, SMP
 voir systèmes SMP
 procurement 455–465
 octroi d'autorisation 457
 présentation 456
 utilisation 458
 utilisation par les applications 460
 utilisation par les utilisateurs 458
 produit cartésien 256
 programmes DB-Bibliothèque
 number of user connections et 247
 propriétaires
 voir propriétaires d'objets de base de données;
 propriétaires de bases de données
 voir propriétaires d'objets de base de
 données; propriétaires de bases de données
 propriétaires d'objets
 voir propriétaires d'objets de base de données
 propriétaires d'objets de base de données 5
 voir aussi propriétaires de bases de données
 autorisations 5, 420–421, 456
 statut non transférable 393
 propriétaires de bases de données 4
 voir aussi propriétaires d'objets de base de
 données; autorisations
 autorisations 5, 419, 424
 commande **setuser** et 455
 modification 426, 708
 nom dans la base de données 393, 403
 nom de login 2, 4
 objets non transférés entre 393
 octroi d'autorisations 435
 oubli du mot de passe 423
 plusieurs utilisateurs identiques 402
 responsabilité en matière d'erreurs 58, 60
 tâches 4
 propriétaires de tables
 voir propriétaires d'objets de base de données
 propriétaires des objets de la base de données

tâches 5
 protection contextuelle 467
 protection de l'accès
 voir autorisations; fonctions de sécurité

R

rapport d'erreurs 58, 60, 64
 rapport **optimized**
 dbcc indexalloc 790, 791, 794
 dbcc tablealloc 794
 rapports
 voir aussi informations (serveur)
 dbcc 781, 782, 789, 820
 pour **dbcc checkalloc** 792
 pour **dbcc indexalloc** 792
 utilisation du serveur 415
 rapports relatifs aux problèmes résolus 79
 recherche
 ID utilisateur 407
 noms d'utilisateur 407
 objets de base de données 58
 utilisateurs dans une base de données 407
 recherche d'au moins un caractère dans les mots de passe
 362
 recommandations, sécurité 334
 reconstruction
 base de données *master* 965
 récupération de l'espace
 paramètre de configuration **enable housekeeper GC**
 208
 reorg reclaim_space pour 763, 765
 redéfinition de la configuration.
 voir paramètre de configuration; commande
 reconfigure
 redémarrage, serveur 551
 après reconfiguration 307
 bases de données au repos 859
 à partir du même répertoire 55
 point de reprise 726
 réindexation après 307
 reprise automatique après 831
 tables système et 307
 tables temporaires et 30
 voir redémarrage, serveur

Index

- redondance totale
 - voir* mise en miroir des disques
- règle d'accès
 - AND 444
 - OR 444
- règles
 - voir aussi* objets de base de données
 - hiérarchie de la protection 473
- règles d'accès 442, 447
 - bcp et 447
- répartition
 - tables sur deux disques 573
 - tables sur plusieurs segments 745–747
- répertoire
 - charsets* 314
 - locales* 300
- réplication
 - reprise et 951
- Replication Server 951
- reprise
 - à partir de sauvegardes 846–855
 - accès utilisateur refusé pendant 831
 - après des pannes 831, 846
 - base de données *sysystemprocs* 979–981
 - bases de données utilisant la reprise semi-automatique 865
 - cache de données par défaut et 678
 - délai requis 831
 - interactions sauvegarde base de données/journal 831
 - isolement des pannes 834–845
 - modification des ID utilisateur 881
 - rapide 596
- reprise de la base *master*
 - automatique 831
 - changements de volume pendant la sauvegarde 881
 - ID utilisateur et 881
 - planification des sauvegardes 880
- reprise semi-automatique
 - reprise d'une base de données marquée in quiesce 865
- requêtes
 - erreurs de conversion, exécution impossible 326
 - évaluation de l'utilisation des ressources 264
 - portée des limites d'utilisation des ressources et 266
 - restriction à l'aide de **sp_add_resource_limit** 255
 - restriction de l'utilisation des ressources avant ou lors de l'exécution 266
- réseaux
 - chargements sur 892
 - connexions 16
 - fichiers d'interface 16
 - logiciel 38
 - restauration sur des 972
 - sauvegardes sur 892, 900
 - services de répertoire 17
 - striping de sauvegarde et 919
- responsable de la sécurité du système 3
- ressources physiques, gestion
 - voir* gestion du stockage
- restauration
 - voir aussi* mise en miroir des disques
 - à partir du journal courant 932
 - à une heure spécifiée dans le journal de transactions 950
 - allocation d'espace et 572, 949
 - après mise à niveau 970
 - après reconfiguration 305
 - base de données *master* 43, 580
 - base de données *model* 977
 - changements de l'ordre de tri et 305
 - chargement des bases de données 305
 - instantanée 573, 596
 - instructions détaillées 944–951
 - méthode de copie actualisée d'une base de données 726
 - moteurs SMP et 685
 - option **for load** et 707
 - pannes pendant 949
 - paramètres de configuration pour 106–108
 - planification d'une sauvegarde 797
 - planification d'une sauvegarde 28
 - rapide 596
 - reconfiguration automatique 949
 - recréation de base de données 948
 - temps et comptabilisation d'espace libre 1012
- restauration de l'identité initiale 455
- restauration de la base *master* 962–977
 - reconstruction 965
 - sauvegarde après 977
 - utilisateurs supprimés et 975
- restauration instantanée 573, 596
- résultats
 - restriction du nombre de lignes renvoyées 272

révocation
 droit de création de triggers 431
 privilèges de rôle avec **with override** 394
 rôle avec **revoke role** 440

rôle d'opérateur 3
 autorisations 423
 tâches 867

rôles
 activation 391
 autorisations de procédure stockée et 411
 autorisations et 440
 configuration pour le login "sa" 334
 des serveurs 858
 désactivation 391
 déverrouillage 360
 dans les instructions **grant** et **revoke** 430, 435
 mots de passe 365
 nombre maximal de tentatives de connexion,
 définition 358
 nombre maximal de tentatives de connexion,
 modification 359
 octroi 457
 procédures stockées et 439, 468
 verrouillage 357, 360

rôles définis par l'utilisateur
 activation 391
 configuration 385
 désactivation 391
 nombre de 385
 octroi avec **grant role** 439
 suppression 393

rôles définis par l'utilisateur
 planification 383

rôles système
 activation 391
 administrateur système 2
 désactivation 391
 octroi avec **grant role** 439
 opérateur 3
 paramètre de configuration **max_roles_enabled**
 385
 responsable de la sécurité du système 3

RPC
 voir appels de procédure à distance

russe
 jeux de caractères supportés 289

S

sauvegarde d'un device secondaire 862

sauvegarde, base de données 43, 797, 885–925
 démontage de bandes 922
 destination des messages 924
 device de sauvegarde 891
 étiquettes de volumes 907
 initialisation/ajout 923
 invite **sp_volchanged** 940–942
 mise à niveau des sauvegardes de bases utilisateur
 952
 multiples sur un seul volume 923
 nom de base de données 889, 891
 nom de fichier 908–911
 nom de volume 907
 rembobinage de bandes après 922
 systématique 846
 taille de bloc 904

sauvegarde, journal de transactions 847, 885–925
 capacité de bande 905
 démontage de bandes 922
 destination des messages 924
 device de sauvegarde 889, 891
 espace de journal insuffisant 888
 invite **sp_volchanged** 940–942
 maximisation de l'espace 934
 nom de base de données 891
 nom de fichier 908–911
 nom de volume 907
 rembobinage de bandes après 922
 sauvegarde consécutive à une panne de disque 932
 striping de sauvegarde 917

sauvegardes 42–46, 825–884
 à distance 868, 873
 indications 42–46
 modification des ID utilisateur 881
 plusieurs bases de données sur un seul volume 924
 prévention des réécritures sur bande 874

scénarios de règle d'accès 448

script 288
 installhasvss 207
 insthasv 207

script **installmaster**
 reprise avec *sybssystemprocs* 882

scripts
 installdbccdb 816

Index

- installmaster** 979
- installmodel** 976
 - noms logiques de device dans 876
 - pour les sauvegardes 880
- sécurité
 - audit 356
 - contrôle d'accès discrétionnaire 337
 - contrôles d'identification et d'authentification 337
 - fonctions de login 356
 - mise en place après l'installation 334–336
 - rôles 338
- sécurité réseau 533–568
 - ajout de logins pour l'unification des logins 553
 - appels de procédure à distance 555
 - configuration du serveur 546
 - connexion au serveur 563
 - définition des fichiers de configuration 538
 - espace mémoire nécessaire 551
 - identification des utilisateurs et des serveurs 545
 - mécanisme de sécurité 545
 - mécanismes de sécurité supportés 534
 - présentation 533
 - procédure d'administration 537
 - recherche d'informations 563, 567
 - redémarrage du serveur pour l'activation 551
 - utilisation 563
- segment de journal
 - seuils 1001–1004
- segment *default* 574, 734
 - réduction du champ d'application 742
- segments 577, 715–719, 735–757
 - voir aussi* devices de base de données; allocation d'espace
 - affichage des seuils 998
 - amélioration des performances et 737
 - colonnes *text/image* et 747
 - comptabilisation d'espace libre et 1012
 - création 574, 740
 - création d'objets de base de données 742
 - default* 574
 - définis par l'utilisateur 946
 - didacticiel de création 753–757
 - entrées d'une table système 714, 752
 - extension 741
 - gestion d'espace libre 985–1012
 - index clusterisés 748, 757
 - index non clusterisés 745
 - informations 715–719, 749, 793
 - logsegment* 574, 734, 986–1012
 - options par défaut* 734
 - partage de l'espace 736
 - placement d'objets 736, 745
 - placement d'objets de bases de données sur 736, 742, 745, 949
 - placement d'objets sur 949
 - rapport **sp_helpthreshold** 998
 - segment *system* 574, 734
 - seuils et 1005
 - suppression 748
 - suppression de devices 748
 - system* 574, 734
 - table *syssegments* 577
 - tableau de valeurs 714
- segments d'allocation de disque 714
- segments de base de données
 - voir* segments
- segments utilisateur, création 753–757
 - voir aussi* segments
- séparation des rôles 338
- séparation physique
 - du device de journal de transactions 594
- server_name.cfg*, nom par défaut du fichier de configuration 86
- serveur de développement 35
- serveur de production 35
- serveur local et distant
 - voir* serveurs distants
- serveurs
 - voir aussi* processus (tâches serveur); serveurs distants
 - ajout d'utilisateurs aux 372–374
 - ajout de logins aux 372–374
 - architecture de système SMP 682
 - arrêt 76
 - cohérence de l'ordre de tri 304
 - connexion 16
 - connexions utilisateur aux 247
 - contrôle des performances 101
 - degrés de sévérité des messages d'erreur 57–63
 - déverrouillage des logins ou des rôles au démarrage 361
 - distants 519–526

- erreurs de syntaxe 59
- erreurs fatales et 61–63
- erreurs internes non fatales 61
- étapes de création d'une base de données 698
- étapes de l'allocation d'espace 712
- fichiers d'interface 16
- informations utilisateur 405–417
- installation 37, 574
- local 520
- messages d'erreur 55
- mode mono-utilisateur 196, 728, 964, 966
- mode restauration de master 966
- mots de passe 529
- mots de passe sur 522
- multiprocesseur 681–693
- noms de 520, 521
- ordonnanceur 229
- placement d'objets sur des segments 745
- placement d'objets sur un segment 949
- problèmes de démarrage et mémoire 618
- spécifications mémoire 607
- suppression de logins de 396
- test 35–37
- valeurs des paramètres de configuration 86
- serveurs distants 519–523
 - ajout 519–532
 - informations sur 523
 - mémoire 635
 - noms de 519
 - options pour 521
 - suppression 524
- serveurs multiprocesseur
 - voir* systèmes SMP
- services de répertoire dans le fichier *libtcl.cfg* 17, 540
- services de répertoire multiples
 - LDAP 19
- services de sécurité
 - exemple 534–535
 - présentation 533
 - support par Adaptive Server 536
- session de messagerie, démarrage 138
- sessions
 - voir* connexions
- set_appcontext 453
- seuil de suspicion 837
- seuil ultime 986–1012
 - exemple de procédure 1008–1011
 - fonction **lct_admin** 993
 - nombre de pages libres 1000
 - procédure, création 1006–1011
 - procédure, nouvelle 1000
 - sauvegarde d'un journal de transactions 1007
- seuils 985–1012
 - ajout 999–1005
 - ajout pour le segment de journal 1001–1004
 - création 998–1005
 - désactivation 1012
 - informations 998
 - intervalle 1005–1006
 - modification 999
 - nombre maximal 998
 - point moyen entre deux 1005
 - recherche de procédure associée 1011
 - segments et 1005
 - seuil ultime 986
 - suppression 1001
 - table *systhresholds* 1011
 - ultimes 1012
 - valeur d'hystérésis 987
- seuils de conversion des verrous
 - paramétrage avec **sp_configure** 213–232
- seuils de conversion des verrous de ligne
 - paramétrage avec **sp_configure** 230–232
- signature électronique
 - cryptage par clé publique 341
 - définition 341
 - détection d'altération 341
 - non-répudiation 341
- signe moins (–)
 - conversion en signe livre (£) dans les noms de login 550
- sites, distants 532
- sp_bindrule 443
- sp_dbcc_runcheck**
 - dbcc checkverify** et 804
- spécialisation
 - processus 682
 - processus vers moteur 684
- spécification *secmech* 543
- SSL
 - activation de SSL 347
 - définition 343

Index

- filtre SSL 344
- négoce SSL 343
- statistiques
 - coût des E/S 270
 - résultat **dbcc** 799
 - sauvegarde et reprise 884
 - utilisation des E/S 415, 416
 - vidage avec la tâche housekeeper 205
 - vidage housekeeper et 206
- statistiques d'utilisation 416
- stockage de disque pour la base de données *dbccdb* 814
- stockage du journal dans *logsegment* 574, 734
- striping de sauvegarde 910, 911–920
 - sauvegarde de bases de données sur plusieurs devices 868
- structure
 - configuration en environnement SMP 684–693
 - répertoire des fichiers d'internationalisation 314
 - répertoire des fichiers de localisation 316
- structure des répertoires
 - fichiers d'internationalisation 314
 - fichiers de localisation 316
 - fichiers *.loc 316
 - jeux de caractères 314
- structures internes
 - utilisation de la mémoire 628
- suid* (ID utilisateur serveur) 374
- suites Cipher
 - définition 354
 - supportées 354
- superutilisateur
 - voir* administrateur système
- support de l'archivage de fichiers compressés 894
- support des langues
 - voir* jeux de caractères; langues
- suppression 445
 - alias d'utilisateur 404
 - alias utilisateur 432
 - base de données endommagée 805
 - bases de données 711, 805
 - device de sauvegarde 589, 877
 - device master de la zone d'espace par défaut 589
 - devices de base de données 589, 712, 947
 - espaces de travail 819
 - fichiers 589
 - groupes 393
 - historique de **dbcc checkstorage** à partir de la base de données *dbccdb* 819
 - informations sur la configuration dans *dbccdb* 819
 - intervalles de temps nommés 261
 - limites d'utilisation des ressources 280–281
 - logins de serveur 396
 - logins distants 524, 525
 - rôles définis par l'utilisateur 393
 - segment de base de données 748
 - serveurs 524
 - seuils 1001
 - utilisateur base de données 392
 - utilisateur propriétaire d'objets de base de données 393
 - utilisateurs de serveur 396
 - utilisateurs guest de *master* 377
- supprime
 - récupération d'espace avec **reorg** 763
- surveillance
 - analyseur de performances Windows NT 226
- Sybase Central, utilisation pour les tâches d'administration du système 8
- symbole de la devise européenne
 - jeux de caractères 290
- symboles
 - voir aussi* section Symboles de cet index
 - dans les instructions SQL xxxiv
- symptômes de dommages, base de données *master*, *voir* *base de données master*
- syntaxe 442, 444, 445
 - Conventions Transact-SQL xxxiv–xxxvii
 - erreurs 59
- syntaxe de règle d'accès 444
- syntaxe de règle d'accès étendue 444
- sys_sessions 452
- système d'exploitation
 - commandes de copie corrompant les bases 846
 - contraintes 582
 - mise en miroir des fichiers 599
 - pannes et reprise automatique 831
 - planification des tâches Sybase et 682
- système de protection
 - contextuel 467
 - hiérarchie (chaînes d'appartenance) 469
 - rapports 460–464
 - synthèse 419

système OpenVMS
 commande **REPLY** 939
 device externe 581
 prévention des démontages de bande 922
 systèmes à multifenêtrage 608
 systèmes multiprocesseur symétriques
 voir systèmes SMP
 systèmes SMP (symmetric multiprocessing)
 architecture 682
 configuration de l'environnement 684–693
 gestion des serveurs 681–693

T

table

spt_committab 13
spt_limit_types 265
spt_monitor 13
spt_values 13
syblicenseslog 414
sysalternates 403
syscolumns 793
sysconfigures 103
syscurconfigs 103
sysindexes 308, 577, 747
sysmessages 52, 54
sysobjects 308
sysremotelogins 527
sysresourcelimits 276
syssegments 577, 714, 752
systhresholds 1011
 table d'audit système 509
 table de hachage de verrou
 configuration de la taille 152
 table offset de ligne et contrôle des entrées 784
 table *sysalternates*
 voir aussi table *sysusers*
 table *sysdatabases*
create database et 699
disk refit et 983
 table *sysdevices* 575, 587
 bits d'état 588, 714
 commandes de mise en miroir des disques 599
create database et 702
 devices de sauvegarde et 876

disk init et 576
sp_dropdevice et 589
sp_helpdevice et 587
 table *syslogins*
 incidence de **sp_addlogin** sur 374
 limites d'utilisation des ressources et 264
 sauvegarde et reprise 881
 table *syslogs* 826
 voir aussi journaux de transactions
create database et 698, 702
 gestion de l'espace utilisé 718
 modification 11
 placement sur un device distinct 594
 table *sysprocesses*
 limites d'utilisation des ressources et 263
 table *syssservers* 517, 518, 519, 524
 Backup Server et 871
 colonne *srvname* 521
 colonne *srvnetname* 521
sp_helpserver et 523, 563
 table *sys timeranges*
 ID d'intervalle 259
 suppression des intervalles de temps 261
 table *sysusages* 576, 752
 allocation d'espace de base de données et 712, 946
 altération 63
create database et 699, 948
 différences dans 977
disk refit et 982–983
 restauration et 965
 table *sysusers*
 autorisations et 426
 table *sysalternates* et 403
 table temporaire
#spdevtab 13
#spindtab 13
 tables
 voir aussi objets de base de données; tables système
 altération de l'intégrité 62
 autorisations 427, 430
 autorisations, comparées à des vues 466
 chaînes d'appartenance pour les 469
 contrôle de cohérence avec **dbcc** 784
dbcc checkdb et 787, 794
dbcc checktable et 308, 704, 706, 784, 794
 données indispensables 797

Index

- fractionnement sur plusieurs segments 745–747
- informations sur les autorisations 464
- lecture seule 308
- liaison à des caches de données 655
- ordre de tri de 787
- procédure système 13
- protection contextuelle 467
- répartition sur deux disques 573
- sans index 309
- sous-jacentes 466
- table d'allocation d'objets (Object Allocation Map) de 112, 777
- temporaires 28
- transfert sur un autre device 745, 748, 757
- transfert vers un index clusterisé 748, 757
- tables de procédures système 13
- tables de vues sous-jacentes (tables standard) 466
- tables standard
 - voir* tables
- tables système 9–10
 - voir aussi* tables; *noms de tables individuelles, sous-table*
 - altération 63
 - autorisations 425
 - clés 10
 - create database** et 9, 576, 752
 - création 9
 - dangers des mises à jour directes 967
 - dbcc checkcatalog** et 793
 - dbcc reindex** et 309
 - informations sur un segment et 752
 - mise à jour 10, 14, 967
 - modification dangereuse 14
 - modifications autorisées 425
 - option **dbcc nofix** 791
 - pour les bases de données utilisateur 27
 - procédures stockées et 9, 14
 - redémarrage du serveur et 307
 - réindexation et 309
 - relations de la gestion du stockage 574–577
 - requête 9, 14
- tables temporaires 28
 - option de base de données **select into/bulkcopy/plisort** 728
- tâche du gestionnaire
 - récupération d'espace et 762
- tâche housekeeper
 - configuration 205
 - contrôle d'utilisation de licences 413
 - récupération d'espace et 208
 - vidage des statistiques 205
- tâches d'administration du système
 - exécution avec Sybase Central 8
- taille
 - voir aussi* espace
 - base de données 700
 - base de données *model* 200, 583, 701
 - base de données *tempdb* 29
 - bases de données, estimation 702
 - device de sauvegarde sur bande 877
 - E/S 654
 - extension de segments 741
 - index 702
 - journal d'erreurs 16
 - journaux de transactions 703, 729
 - modification d'une base de données 709
 - nouvelle base de données 27, 701
 - stockage de texte*</type de données> 718
 - tables 702
 - transaction **dbcc fix_text** 309
 - unités d'allocation 712
- taille de bloc
 - device de sauvegarde 868
 - sauvegardes et chargements de bases de données 904
- temporisations de verrous
 - configuration au niveau du serveur 153
- temps de réponse 229
- terminaux
 - conversion de jeux de caractères de 329
 - installation de nouvelles définitions 313
- thai
 - jeux de caractères supportés 289
- thread de service
 - définition pour Backup Server 917
- trace d'audit 30, 475, 509
 - ajout de commentaires 480, 506
 - changement de table d'audit courante 485
 - gestion 484
 - illustration avec plusieurs tables d'audit 477
 - procédure relative aux seuils 485
 - requête 508

- trace des messages d'erreur 55
 - traduction
 - voir* jeux de caractères
 - traitement batch
 - intervalles de temps actifs et 262
 - portée des limites d'utilisation des ressources et 267
 - restriction du temps écoulé 271
 - traitement parallèle des requêtes
 - mémoire 634
 - transactions
 - voir aussi* verrouillages; journaux de transactions
 - commit à deux phases 31
 - définition 826
 - détachées 125
 - erreurs 59
 - intervalles de temps actifs et 262
 - longues 106, 828
 - portée des limites d'utilisation des ressources et 268
 - reprise et 828
 - restauration et 106
 - restriction à l'aide de **sp_add_resource_limit** 255
 - restriction du temps écoulé 271
 - transfert
 - index non clusterisés 745
 - journaux de transactions 705
 - tables 745, 748, 757
 - de tables vers des index clusterisés 748, 757
 - transfert de propriété
 - voir* objets de base de données, propriété
 - triggers
 - voir aussi* objets de base de données; procédures stockées
 - autorisations et 473
 - création 431
 - imbriqués 195
 - triggers procédure stockée
 - voir* triggers
 - turc
 - jeux de caractères supportés 289
 - type de données *image*
 - effets sur les performances 739
 - stockage sur un device distinct 747
 - table *sysindexes* et 747, 752
 - type de données *text*
 - chaîne de pages de texte 747
 - changement des jeux de caractères et 309
 - effets sur les performances 739
 - jeux de caractères codés sur plusieurs octets et 309
 - stockage sur un device distinct 747
 - table *sysindexes* et 747, 752
 - volume de stockage 718
 - types de limite 265, 268, 273
 - coût des E/S 268
 - nombre de lignes renvoyées 272
 - temps écoulé 271
 - types de machine, transfert de bases de données entre 707
- ## U
- unbindrule 444
 - unicode 289
 - jeux de caractères 290
 - unification des logins 536, 547
 - correspondance des noms de login 549
 - login par défaut sécurisé 548
 - modèles de sécurité des procédures distantes 556
 - nécessité 547
 - unified login required** 242
 - unités d'allocation 580, 712, 774
 - voir aussi* taille; allocation d'espace
 - sur le device master 967
 - reprise et 946
 - utilisateur courant
 - set proxy** et 459
 - utilisateurs
 - voir aussi* alias; groupes; connexions; connexions distantes
 - actuellement sur la base de données 405
 - actuellement sur le serveur 405
 - ajout 370–375, 375–379
 - ajout à des bases de données 699
 - ajoutés et restauration de *master* 975
 - alias 402
 - autorisations générales ou spécifiques 436, 467
 - contrôle d'utilisation de licences 412
 - distants 524, 529
 - erreurs 58, 58–61

Index

guest 377
ID 407, 421
identification des grands consommateurs de ressources 263
informations sur 405–417
informations sur les limites d'utilisation des ressources 276
invité 426
mode mono-utilisateur 196, 728
modification des limites d'utilisation des ressources 278
multiples et performances 745
nom d'application, définition 400
nom d'hôte client, définition 400
nom de client, définition 400
number of user connections et 246
suppression d'une base de données 392–393, 699
suppression de serveur 396
suppression des groupes 400
suppression des limites d'utilisation des ressources 280
supprimés et restauration de *master* 975
visiteur 378
vues pour des utilisateurs spécifiques 466
utilisateurs distants
 voir connexions distantes
utilisateurs distants du serveur
 voir connexions distantes
utilisateurs guest 426
 ajout 377
 autorisations 377
 bases de données exemples et 32, 378
 création 377
utilisateurs, nombre ou 379
utilisateurs, objet
 voir propriétaires d'objets de base de données
utilisation de Java 442
utilisation de licences
 contrôle 412
 messages du journal d'erreurs 414
utilisation des ressources, évaluation 264
utilisation du processeur
 par utilisateur 415
utilitaire
 dataserver 965

dataserver, utilisation pour déverrouiller les logins et les rôles 361
services de sécurité **dsedit** 545
showserver 973
showserver, *voir aussi* le manuel Programmes Utilitaires
startserver, mode restauration de master 966
utilitaires autonomes et jeux de caractères 328

V

valeur d'hystérésis, variable globale
 @@*thresh_hysteresis* 1005
valeurs d'heure
 format d'affichage 315
valeurs de mois
 autre langue 311
valeurs par défaut
 voir aussi objets de base de données
 autorisations 27, 421
 bases de données 26
 langue 144
 numéro d'ID du jeu de caractères 144
 ordre de tri 145, 243
 paramètres de configuration 86
valeurs *text*, **dbcc fix_text** mise à niveau de 309
variable globale
 @@*char_convert* 314
 @@*client_csid* 314
 @@*client_csname* 314
 @@*langid* 316
 @@*language* 316
 @@*max_connections* 246
 @@*maxcharlen* 314
 @@*ncharsize* 314
 @@*thresh_hysteresis* 987
variable globale @@*rowcount*
 limites d'utilisation des ressources et 265
 limites de décompte de lignes et 273
variable globale @@*thresh_hysteresis*
 emplacement de seuil et 1005
variables
 dans les messages d'erreur 54
vérification, accès utilisateur 522, 526
vérifications de l'ordre 536

vérifications de l'ordre des données transmises 536
verrouillage
 connexions 357
 commandes **dbcc** 310
 par commande **dbcc** 794
 liaison de caches et 673
 logins 394
verrous
 quantité de 147
verrous d'attente
 paramètres de configuration les affectant 691
 table de hachage de verrou 151
vietnamien
 jeux de caractères supportés 290
virgule (,)
 conversion en trait de soulignement dans les noms
 de login 549
 dans les instructions SQL xxxiv
Virtual Server Architecture 681
vitesse (serveur)
 de commande **dbcc** 794
 de croissance du journal de transactions 704
 performances du système 573, 596
 utilisation de segments 733
vues
 voir aussi objets de base de données
 autorisations 430, 466–468
 chaînes d'appartenance 469
 dépendantes 470
 sécurité et 465

X

X/Open XA 127
XP Server
 libération de la mémoire de 138
 priorité 137

Z

zone de vidage
 configuration 659–663
 configuration par défaut 660
zones mémoire
 configuration 650–654
 configuration de limites de prélecture asynchrone
 663
 configuration du pourcentage de la zone de vidage
 659–663
 modification de la taille 666
 suppression 672

Index