
Tips zur Optimierung der Leistung von Microsoft Access, Version 1.1
April 1993

(C) Copyright Microsoft Corporation, 1993

Dieses Dokument bietet Tips zur Leistungssteigerung von Microsoft Access durch Hinzufügen oder Ändern von Einträgen in der Initialisierungsdatei MSACCESS.INI und durch das Verwenden von Transaktionen.

So verwenden Sie dieses Dokument

Maximieren Sie zum Lesen dieses Dokuments mit Microsoft Windows Write das Fenster von Write.

Zum Drucken dieses Dokuments öffnen Sie es in Windows Write, Microsoft Word, oder einem anderen Textverarbeitungsprogramm. Markieren Sie dann das gesamte Dokument und formatieren Sie den Text in der Schriftart Courier, Schriftgröße 10, bevor Sie es drucken.

Inhalt

Einstellen von [ISAM]-Einträgen in MSACCESS.INI
Verwenden von Transaktionen zur Maximierung des Datendurchsatzes

Einstellen von [ISAM]-Einträgen in MSACCESS.INI

Microsoft Access bietet automatisch interne Standardeinstellungen, die normalerweise die beste Leistung für die meisten Datenbankoperationen liefern. Fortgeschrittene Benutzer können diese Einstellungen allerdings ändern, um höchste Leistung für eine bestimmte Systemkonfiguration oder Anwendung zu erhalten.

Sie können diese Leistungssteigerung für Microsoft Access erzielen, indem Sie Einträge im Abschnitt [ISAM] der Initialisierungsdatei MSACCESS.INI hinzufügen oder ändern. (Setup installiert diese Datei automatisch im Windows-Verzeichnis). Diese Einträge bestimmen die Größe der Datenseite und des Read-Ahead-Cache-Speichers im Arbeitsspeicher, wie lange Daten in einem Seiten-Cache-Speicher aufbewahrt werden und wie oft Microsoft Access einen Sperrungsvorgang wiederholt.

WARNUNG: Das Bestimmen der besten Einstellungen für das System kann ein zeitaufwendiger und schwieriger Prozeß sein und auch empirische Methoden erfordern. Unerfahrene Benutzer sollten diese Einträge nicht ändern oder ihnen etwas hinzuzufügen.

Microsoft Access enthält automatisch einen PageTimeout-Eintrag im Abschnitt [ISAM] der Datei MSACCESS.INI. Dieser Eintrag legt fest, wie lange Microsoft Access eine Datenseite im Speicher aufbewahrt. Zur zusätzlichen Optimierung können Sie auch die folgenden Einträge in diesen Abschnitt einfügen: MaxBufferSize, ReadAheadPages, LockRetry und CommitLockRetry. Diese Einträge werden alle in diesem Dokument beschrieben.

Diese Einstellungen werden beim Starten von Microsoft Access gelesen. Sie können sie verändern, während Sie Microsoft Access ausführen, aber die Veränderungen werden erst gültig, nachdem Sie Microsoft Access neu gestartet haben.

PageTimeout (nur bei gemeinsamem Datenzugriff)

Der Eintrag PageTimeout setzt in Zehntelsekunden fest, wie lange Microsoft Access eine Datenseite in einem Seiten-Cache-Speicher im Arbeitsspeicher aufbewahrt, falls die Datenbank für gemeinsamen Datenzugriff geöffnet wurde. Microsoft Access liest Daten jeweils in Datensatzblöcken von 2 KB oder "Seiten".

Wenn Microsoft Access beispielsweise eine Datenseite liest, kopiert es die Daten in den Seiten-Cache-Speicher. Wenn Microsoft Access eine weitere Leseanforderung für dieselbe Datenseite während der durch PageTimeout festgelegten Zeit erhält, liest es die Daten direkt vom Seiten-Cache-Speicher und nicht vom Datenträger.

ANMERKUNG: Access Basic beachtet die Einstellung PageTimeout nicht, außer wenn Ihr Code Hintergrundverarbeitung zuläßt, indem dieser die Anweisung bzw. Funktion DoEvents in periodischen Abständen aufruft.

Maximaleinstellung: 2147483647 (maximal möglicher Long Integer-Wert)

Minimaleinstellung: 0

Standardeinstellung: Setup legt den Eintrag PageTimeout in der Datei MSACCESS.INI automatisch auf 300 fest. Wenn Sie den Eintrag PageTimeout entfernen, verwendet Microsoft Access für PageTimeout die Standardeinstellung von 5.

BEISPIEL: Dieses Beispiel legt PageTimeout auf 2 Sekunden fest.

PageTimeout=20

MaxBufferSize

Der Eintrag MaxBufferSize legt die Größe des Arbeitsspeichers in KB fest, der für die Verwendung als Seiten-Cache-Speicher reserviert wird. Microsoft Access liest Daten in Seitengrößen von 2 KB und

kopiert die Daten in den Seiten-Cache-Speicher. Nachdem die Daten im Cache-Speicher sind, kann Microsoft Access sie je nach Bedarf für Tabellen, Abfragen, Formulare oder Berichte verwenden.

Wenn Microsoft Access eine Leseanforderung erhält, überprüft es zuerst die Datenseite im Seiten-Cache-Speicher. Falls die Seite nicht im Cache-Speicher gefunden werden kann, liest Microsoft Access die Datenseite vom Datenträger und kopiert sie dann in den Seiten-Cache-Speicher. Microsoft Access verwendet physischen Speicher und, falls notwendig, virtuellen Speicher, um den Cache-Speicher zu erstellen. Alle Seiten bleiben im Cache-Speicher, bis er voll ist und Seiten aus dem Puffer geleert werden müssen, um Platz für neu gelesene Daten zu schaffen

Maximaleinstellung: 4096

Minimaleinstellung: 18

Standardeinstellung: Falls kein Eintrag MaxBufferSize in Ihrer Datei MSACCESS.INI vorhanden ist, verwendet Microsoft Access eine Standardeinstellung von 512.

BEMERKUNG: Da Microsoft Access Daten jeweils in Seitengrößen von 2 KB liest, verwendet es immer eine gerade Einstellung für MaxBufferSize. Wenn Sie eine ungerade Zahl eingeben, verwendet Microsoft Access als Einstellung für MaxBufferSize die jeweilige Zahl minus 1.

BEISPIEL: Dieses Beispiel legt MaxBufferSize auf 4 MB fest.

MaxBufferSize=4096

ReadAheadPages

Der Eintrag ReadAheadPages legt die Größe in Seiten des Read-Ahead-Cache-Speichers fest, der von Microsoft Access für sequentielles Lesen von Seiten verwendet wird. Sequentielles Lesen von Seiten wird dann ausgeführt, wenn Microsoft Access feststellt, daß sich Daten einer laufenden Leseanforderung auf dem physischen Datenträger in einer Datenseite neben der Datenseite aus der vorhergehenden Anforderung befinden. Microsoft Access verwendet den Read-Ahead-Cache-Speicher nur dann, wenn es feststellt, daß ein sequentieller Lesevorgang ausgeführt wird.

Wenn Microsoft Access sequentielles Lesen von Seiten feststellt, liest es die angeforderte Seite und die nächsten N-1 Seiten in dieser Richtung (N ist dabei die Einstellung für ReadAheadPages) und kopiert die Datenseiten in den Read-Ahead-Cache-Speicher. Sooft Microsoft Access dann sequentielles Lesen feststellt, kann es die nächsten N Leseanforderungen direkt aus dem Read-Ahead-Cache-Speicher vornehmen.

Der Read-Ahead-Cache-Speicher erhöht die Geschwindigkeit von sequentiellen Lesevorgängen, besonders für das Lesen von im Netzwerk gespeicherten Daten. Er erhöht die Anzahl der Aktualisierungen von

Datensätzen pro Sekunde (Durchsatz) auf einem Netzwerk, indem er wenige große Pakete anstelle von vielen kleinen Paketen über das Netzwerk schickt.

Microsoft Access kopiert den Read-Ahead-Cache-Speicher möglichst in die ersten 640 KB Speicher, um die Fähigkeit von Windows auszunutzen, direkt im konventionellen Arbeitsspeicher zu lesen und zu schreiben.

Wenn der Read-Ahead-Cache-Speicher nicht in den konventionellen Arbeitsspeicher kopiert werden kann, kopiert ihn Microsoft Access in den hohen Speicherbereich. Das Kopieren des Cache-Speichers in den hohen Speicherbereich ist allerdings arbeitsaufwendiger, da Windows alle gelesenen und geschriebenen Daten in den eigenen Puffer speichern muß, bevor der Speichervorgang ausgeführt werden kann.

Maximaleinstellung: 31

Minimaleinstellung: 0

Standardeinstellung: Wenn sich kein Eintrag ReadAheadPages in der Datei MSACCESS.INI befindet, verwendet Microsoft Access eine Standardeinstellung von 8.

BEISPIEL: ReadAheadPages=16

ANMERKUNG: Microsoft Access erstellt einen separaten Read-Ahead-Cache-Speicher für jede offene Datenbank auf dem Computer. Jede Bibliotheksdatenbank besitzt ihren eigenen Read-Ahead-Cache-Speicher.

LockRetry

Der Eintrag LockRetry bestimmt, wie oft Microsoft Access eine Seitensperrungsoperation wiederholt, bevor ein Fehler gemeldet wird. Wenn ein Benutzer beispielsweise versucht, eine Datenseite zu sperren, die bereits von einem anderen Benutzer gesperrt wurde, schlägt dieser Versuch fehl. Microsoft Access wird noch N Male versuchen, die Seite zu sperren, wobei N gleich der Einstellung LockRetry ist.

Falls der Versuch, die Seite zu sperren, nach der N-ten Wiederholung immer noch fehlschlägt, meldet Microsoft Access einen Fehler.

Maximaleinstellung: 2147483647 (maximal möglicher Long Integer-Wert)

Minimaleinstellung: 0

Standardeinstellung: Wenn sich in der Datei MSACCESS.INI kein Eintrag LockRetry befindet, verwendet Microsoft Access eine Standardeinstellung von 20.

BEISPIEL: LockRetry=6

CommitLockRetry

Der Eintrag CommitLockRetry wird in Zusammenhang mit dem Eintrag

LockRetry verwendet, um festzulegen, wie viele Versuche unternommen werden sollen, einen Datensatz auf einer Datenseite zu sperren, die bereits durch eine Transaktion gesperrt wurde. (Weitere Informationen zu Transaktionen finden Sie unter "Verwenden von Transaktionen zur Maximierung des Datendurchsatzes" weiter unten in dieser Textdatei).

Wenn ein Benutzer versucht, eine Datenseite zu sperren, die bereits durch eine Transaktion gesperrt wurde, versucht Microsoft Access noch N Male, die Seite zu sperren, wobei N das Produkt der Einstellungen LockRetry und CommitLockRetry ist. Wenn beispielsweise für LockRetry die Einstellung 5 und für CommitLockRetry 6 lautet, versucht Microsoft Access noch 30 Male, die Seite zu sperren.

Maximaleinstellung: 2147483647 (maximal möglicher Long Integer-Wert)
Minimaleinstellung: 0
Standardeinstellung: Wenn sich in der Datei MSACCESS.INI kein Eintrag CommitLockRetry befindet, verwendet Microsoft Access eine Standardeinstellung von 20.

BEISPIEL: Angenommen, die Einstellung LockRetry ist 6, dann wird dieses Beispiel dazu führen, daß Microsoft Access noch 36 Male versuchen wird, eine durch eine Transaktion gesperrte Seite zu sperren.

CommitLockRetry=6

Verwenden von Transaktionen zur Maximierung des Datendurchsatzes

In einer Mehrbenutzerumgebung können Sie die Leistung von Microsoft Access noch weiter optimieren, indem Sie Access Basic-Transaktionen für Operationen zur Aktualisierung von Daten verwenden.

Eine Transaktion ist eine Reihe von Operationen, die als Ganzes oder überhaupt nicht ausgeführt werden müssen. Hierbei wird der Anfang einer Transaktion mit der Anweisung BeginTrans markiert. Die Anweisungen Rollback oder CommitTrans beenden dann die Transaktion. (Weitere Informationen zu Transaktionen finden Sie in der Online-Hilfe unter "Transaktionen" oder "Verwenden von Transaktionen" in Kapitel 8, "Manipulieren von Daten", im Handbuch "Einführung in die Programmierung.")

Sie können die Anzahl der Aktualisierungen von Datensätzen pro Sekunde (Durchsatz) einer Anwendung beschleunigen, indem Sie Operationen, die Daten aktualisieren, in eine Access Basic-Transaktion einfügen. Das folgende Beispiel aus Kapitel 8, "Einführung in die Programmierung", (S. 131-132) zeigt, wie man Transaktionen in einer Funktion verwenden kann, die Beträge von einem Bankkonto auf ein anderes überweist.

Function GeldTransfer(VonKonto, ZuKonto, Kunde, Betrag)

```

' Überweist Betrag vom VonKonto des Kunden zum ZuKonto.
' Gibt im Erfolgsfall 0 zurück, andernfalls einen Wert ungleich 0.
Dim db As Database, Herkunft As Dynaset, Ziel As Dynaset
On Error Resume Next
Set db = CurrentDB()
Set Herkunft = db.CreateDynaset(VonKonto)
Set Ziel = db.CreateDynaset(ZuKonto)
If Err Then ' Fehler bei einem Feld oder einer Tabelle.
    On Error GoTo 0
    GeldTransfer = Err
    Exit Function
End If
Herkunft.FindFirst "[Kunden-Nr] = " & Kunde
Ziel.FindFirst "[Kunden-Nr] = " & Kunde
If Not (Herkunft.NoMatch Or Ziel.NoMatch) Then
    If Herkunft!Kontostand >= Betrag Then
        BeginTrans
        Herkunft.Edit
        Herkunft![Kontostand] = Herkunft![Kontostand] - Betrag
        Herkunft.Update
        Ziel.Edit
        Ziel![Kontostand] = Ziel![Kontostand] + Betrag
        Ziel.Update
        If Err Then
            GeldTransfer = -1 ' Eine oder beide Aktualisierungen schlugen fehl.
            RollBack ' Änderungen zurücknehmen.
        Else
            GeldTransfer = 0 ' Kein Fehler bei Transaktion.
            CommitTrans ' Änderungen übernehmen.
        End If
    Else
        GeldTransfer = -2 ' Guthaben nicht ausreichend.
    End If
Else
    GeldTransfer = -3 ' Konto existiert nicht.
End If
Herkunft.Close
Ziel.Close
End Function

```

TIP: Da Microsoft Access die in einer Transaktion verwendeten Datenseiten sperrt, bis die Transaktion beendet ist, verhindert die Verwendung von Transaktionen den Zugriff auf diese Datenseiten, während die Transaktion noch ausgeführt wird. Versuchen Sie, wenn Sie Transaktionen verwenden möchten, ein Gleichgewicht zwischen Datendurchsatz und Datenzugriff zu erreichen.