



HP Operations Manager i

Softwareversion: 10.00

Datenbankhandbuch

Dokument-Releasedatum: Januar 2015
Software-Releasedatum: Januar 2015

Rechtliche Hinweise

Garantie

Die Garantiebedingungen für Produkte und Services von HP sind in der Garantieerklärung festgelegt, die diesen Produkten und Services beiliegt. Keine der folgenden Aussagen kann als zusätzliche Garantie interpretiert werden. HP haftet nicht für technische oder redaktionelle Fehler oder Auslassungen.

Die hierin enthaltenen Informationen können ohne vorherige Ankündigung geändert werden.

Eingeschränkte Rechte

Vertrauliche Computersoftware. Gültige Lizenz von HP für den Besitz, Gebrauch oder die Anfertigung von Kopien erforderlich. Entspricht FAR 12.211 und 12.212; kommerzielle Computersoftware, Computersoftwareokumentation und technische Daten für kommerzielle Komponenten werden an die US-Regierung per Standardlizenz lizenziert.

Copyright-Hinweis

© Copyright 2015 Hewlett-Packard Development Company, L.P.

Marken

Adobe® und Acrobat® sind Marken der Adobe Systems Incorporated.

AMD und das AMD Arrow-Logo sind eingetragene Marken von Advanced Micro Devices, Inc.

Citrix® und XenDesktop® sind eingetragene Marken der Citrix Systems, Inc. und/oder der zugehörigen Tochtergesellschaften. Sie können beim US-Patentamt und Markenämtern in anderen Ländern registriert sein.

Google™ und Google Maps™ sind Marken von Google Inc.

Intel®, Itanium®, Pentium®, Intel® Xeon® und Lync® sind Marken der Intel Corporation in den USA und anderen Ländern.

Linux® ist eine eingetragene Marke von Linus Torvalds in den USA und anderen Ländern.

Java ist eine eingetragene Marke von Oracle und/oder der zugehörigen Tochtergesellschaften.

Microsoft®, Windows®, Windows NT®, Windows® XP und Windows Vista® sind in den USA eingetragene Marken der Microsoft Corporation.

Oracle ist eine eingetragene Marke der Oracle Corporation und/oder der zugehörigen Tochtergesellschaften.

Red Hat® ist in den USA und anderen Ländern eine eingetragene Marke Red Hat, Inc.

UNIX® ist eine eingetragene Marke von The Open Group.

Hinweise

Dokumentationsaktualisierungen

Die Titelseite dieses Dokuments enthält die folgenden bezeichnenden Informationen:

- Software-Versionsnummer zur Angabe der Software-Version.
- Dokument-Releasedatum, das sich mit jeder Aktualisierung des Dokuments ändert.
- Software-Releasedatum zur Angabe des Releasedatums der Software-Version.

Um nach Aktualisierungen des Dokuments zu suchen oder um zu überprüfen, ob Sie die aktuellste Version des Dokuments verwenden, wechseln Sie zu:

<https://softwaresupport.hp.com/group/softwaresupport/search-result?keyword=>

Auf dieser Website ist ein HP Passport-Konto erforderlich. Wenn Sie nicht über ein Konto verfügen, klicken Sie auf der Seite **HP Passport Sign in** auf die Schaltfläche **Create an account**.

Support

Besuchen Sie die HP-Website zur Software-Unterstützung unter: <https://softwaresupport.hp.com>

Auf dieser Website finden Sie Kontaktinformationen und Details zu Produkten, Services und Supportleistungen von HP Software.

Der HP Software Support bietet den Kunden Möglichkeiten, Probleme selbst zu lösen. Sie bietet schnelle und effiziente Wege, auf interaktive Werkzeuge für den technischen Support zuzugreifen, die für die Führung des Unternehmens erforderlich sind. Als Kunde mit Supportvertrag stehen Ihnen beim Support folgende Optionen zur Verfügung:

- Suchen nach interessanten Wissensdokumenten
- Absenden und Verfolgen von Support-Fällen und Erweiterungsanforderungen
- Herunterladen von Software-Patches
- Verwalten von Supportverträgen
- Nachschlagen von HP-Supportkontakten
- Einsehen von Informationen über verfügbare Services
- Führen von Diskussionen mit anderen Softwarekunden
- Suchen und Registrieren für Softwareschulungen

Für die meisten Supportfelder müssen Sie als HP Passport-Benutzer registriert und angemeldet sein. Häufig ist auch ein Supportvertrag erforderlich. Um sich für eine HP Passport-ID zu registrieren, wechseln Sie zu <https://softwaresupport.hp.com> und klicken Sie auf **Register**.

Weitere Informationen zu den Zugriffsebenen finden Sie unter: <https://softwaresupport.hp.com/web/softwaresupport/access-levels>

HP Software-Lösungen und -Integrationen sowie bewährte Vorgehensweisen

Besuchen Sie die Website **HP Software Solutions Now** unter <https://h20230.www2.hp.com/sc/solutions/index.jsp> und erfahren Sie, wie die Produkte im HP Software-Katalog zusammenarbeiten, Informationen austauschen und geschäftliche Anforderungen erfüllen.

Besuchen Sie die Website **Cross Portfolio Best Practices Library** unter <https://hpln.hp.com/group/best-practices-hpsw> und greifen Sie auf umfassende Dokumente und Materialien zu bewährten Vorgehensweisen zu.

Inhalt

Teil I: Einführung in die Datenbankumgebung	9
Kapitel 1: Einführung in das Vorbereiten der Datenbankumgebung	10
Datenbank – Übersicht	10
Anforderungen	11
Konfiguration des Datenbankzeichensatzes	11
Teil II: Bereitstellen und Verwalten der Microsoft SQL Server-Datenbanken	13
Kapitel 2: Microsoft SQL Server-Bereitstellung - Übersicht	14
Informationen über die Microsoft SQL Server-Bereitstellung	14
Systemanforderungen	14
Hardwareanforderungen	14
Softwareanforderungen	15
Zeichensatzunterstützung in Microsoft SQL Server	15
Kapitel 3: Manuelles Erstellen der Microsoft SQL Server-Datenbank	17
Microsoft SQL Server-Datenbanken - Übersicht	17
Manuelles Erstellen der OMi-Datenbanken	17
Erstellen von Datenbanken	17
Erstellen der Objekte	18
Erstellen der Management-Objekte	18
Erstellen der RTSM-Objekte	19
Erstellen der Ereignisobjekte	19
Kapitel 4: Installieren und Konfigurieren von Microsoft SQL Server	21
Workflow für die Microsoft SQL Server-Bereitstellung	21
Voraussetzungen für die Installation	22
Hinweise und Einschränkungen	22
Prüfliste für Support und Zertifizierung	23
Installieren von Microsoft SQL Server	24
Konfigurieren von Microsoft SQL Server	25
Optionen für die Servicekonfiguration	25
Serverkonfigurationsoptionen	26
Überprüfen und Ändern von Server- und Datenbankeinstellungen	26
Portzuweisungsoptionen für benannte Instanzen	28
Kapitel 5: Erstellen und Konfigurieren von Microsoft SQL Server-Datenbanken	30
Erstellen von Datenbanken	30
Datenbankberechtigungen	30
Datenbankdateilayout	31
Speicherort für Daten und Protokoll	31
Datei- und Datenbankeigenschaften	31
Einstellungen für die tempdb-Datenbank	32
Dateigruppen	33
Systemdatenbanken	33

- Konfigurieren von Datenbanken 34
 - Datenbankdateikonfiguration 34
 - Hinzufügen von Dateien 34
 - Löschen von Dateien 35
 - Ändern der Dateieigenschaften 35
 - Datenbankkonfigurationsoptionen 35
- Kapitel 6: Warten von Microsoft SQL Server-Datenbanken 39
 - Sichern von Datenbanken 39
 - Vollständige Sicherung 40
 - Differenzielle Sicherung 40
 - Protokollsicherung 40
 - Datei und Dateigruppensicherung 41
 - Wartungsplan 41
 - Probleme mit dem Transaktionsprotokoll 41
 - Datenbankintegrität und -fragmentierung 42
 - Datenbankintegrität 42
 - Grundlegende Informationen zur Dateisystemfragmentierung 42
 - Grundlegende Informationen zur internen Fragmentierung 43
 - Grundlegende Informationen zur externen Fragmentierung 43
 - Ermitteln und Verarbeiten der Indexfragmentierung 44
 - Verwenden von "sys.dm_db_index_physical_stats" zum Erkennen der
Fragmentierung 44
 - Fragmentierungsbehandlung 44
 - Bereitgestellte Dienstprogramme zur Überwachung und Wiederherstellung von Indizes 45
 - Dienstprogramm für das Wiederherstellen aller Indizes in der Datenbank 45
 - Dienstprogramm für das Wiederherstellen von Indizes basierend auf dem
Fragmentierungslevel für jeden Index 46
 - Verteilungsstatistik 46
 - Sammeln von Statistiken für RTSM 47
 - Dienstprogramm für das Aktualisieren von Statistiken 47
- Kapitel 7: Verwenden der Windows-Authentifizierung für den Zugriff auf Microsoft SQL Server-
Datenbanken 49
 - Aktivieren von OMi für die Arbeit mit der Windows-Authentifizierung 49
 - Konfigurieren von Microsoft SQL Server zur Verwendung der Windows-Authentifizierung 49
 - Starten des OMi-Dienstes mit einem Windows-Benutzer 50
 - Erstellen einer Verbindung zu einer Microsoft SQL-Datenbank mit Windows
Authentifizierung oder Herstellen der Verbindung 50
- Teil III: Bereitstellen und Verwalten der Oracle Server-Datenbank 51**
 - Kapitel 8: Übersicht über die Oracle Server-Bereitstellung 52
 - Informationen über die Oracle Server-Bereitstellung 52
 - Systemanforderungen 53
 - Hardwareanforderungen 53
 - Softwareanforderungen 53
 - Oracle-Instanzen 53
 - Zeichensatzunterstützung in Oracle 53

Kapitel 9: Manuelles Erstellen der Oracle-Schemas	55
Oracle Server-Datenbanken - Übersicht	55
Manuelles Erstellen eines OMi-Schemas	55
Manuelles Erstellen eines Management-Benutzerschemas	56
Manuelles Erstellen eines RTSM-Benutzerschemas	57
Manuelles Erstellen eines Ereignisschemas	57
OMi – Berechtigungen für Oracle-Schema	58
Kapitel 10: Richtlinien für die Oracle Server-Konfiguration und -Dimensionierung	59
Oracle-Parametereinstellungen	59
Oracle-Tablespaces	62
Lokal verwaltete Tablespaces	63
Erstellen eines Oracle-Tablespace	63
Einstellungen für Oracle-Tablespace	64
Einstellungen für den Daten-Tablespace	64
Standardspeichereinstellungen für den Daten-Tablespace	65
System-Tablespace-Einstellungen	65
Einstellungen für temporäre Tablespaces	65
Einstellungen für das Wiederholen-Protokoll	66
Einstellungen für Rückgängigmachen-Segmente	66
Verwenden der RAID-Konfiguration	67
ASM-Speicher	68
Kapitel 11: Warten einer Oracle Server-Datenbank	69
Datenbankwartung und -abstimmung	69
System Global Area (SGA)	69
Datenbankauslastung - Verhalten	70
CPU und E/A	70
Oracle-Warnungsdatei	70
Archivprotokoll - Dateisystem	70
Tablespace-Speicherplatz	71
Speicherplatzverwaltung	71
Sammeln von Statistiken für Datenbanken	71
Oracle - Automatisierte Statistiksammlung	71
Sammeln von Statistiken für RTSM	72
Indexfragmentierung	72
Dienstprogramm zur Indexwartung	73
Oracle-Datenbank - Sicherung und Wiederherstellung	74
Verfügbare Sicherungsmethoden	75
Cold Backup	75
Hot Backup	75
Data Pump - Export/Import	76
Oracle Recovery Manager - RMAN	76
Kapitel 12: Konfigurieren des Oracle-Clients für OMi	77
Kapitel 13: Oracle-Prüfliste	79
Teil IV: Bereitstellen und Verwalten der PostgreSQL Server-Datenbanken	83
Kapitel 14: Übersicht über die PostgreSQL-Bereitstellung	84

Informationen zur PostgreSQL-Bereitstellung	84
Systemanforderungen	85
Hardwareanforderungen	85
Softwareanforderungen	85
Zeichensatzunterstützung in PostgreSQL	86
Kapitel 15: Manuelles Erstellen der OMi-Datenbanken	87
Übersicht über die PostgreSQL-Datenbank	87
Manuelles Erstellen von OMi-Datenbanken	87
Manuelles Erstellen einer Management-Datenbank	87
Manuelles Erstellen einer RTSM-Datenbank	88
Manuelles Erstellen einer Ereignisdatenbank	88
Kapitel 16: Warten einer PostgreSQL-Datenbank	90
Datenbankwartung	90
Sichern und Wiederherstellen der PostgreSQL-Datenbank	91
Sichern und Wiederherstellen einer eingebetteten PostgreSQL-Datenbank	91
Sicherung der eingebetteten PostgreSQL-Datenbank	91
Wiederherstellung der eingebetteten PostgreSQL-Datenbank	92
Sichern und Wiederherstellen einer externen PostgreSQL-Datenbank	92
SQL-Dump	92
Dateisystemebene	93
Kontinuierliche Archivierung	93
Zurücksetzen oder Ändern des Benutzerkennworts für die eingebettete PostgreSQL-Datenbank	93
Kapitel 17: Migrieren einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank	95
Gründe für die Migration Ihrer eingebetteten PostgreSQL-Datenbank	95
Verfahren zum Migrieren einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank	95
Teil V: Anhänge	97
Anhang A: Empfehlungen zum Microsoft SQL Server-Datenspeicher	98
Erstellen von Microsoft SQL Server-Dateigruppen	98
Vorteile von mehreren Dateigruppen	98
Anhang B: Microsoft SQL Server-Konfigurationsoptionen	101
Anhang C: Ändern des Datenbank-Hosts	106
Anhang D: Unterstützung für Oracle Real Applikation Cluster	107
Informationen über Oracle Real Applikation Cluster (RAC)	107
Single Client Access Name	108
Clientseitige Konfiguration für Oracle RAC	108
Serverseitige Konfiguration	111
Empfehlungen für OMi-Datenbanken	111
Erstellen von oder Verbinden mit einer der OMi-Datenbanken	112
Senden von Feedback zur Dokumentation	113

Teil I: Einführung in die Datenbankumgebung

Kapitel 1: Einführung in das Vorbereiten der Datenbankumgebung

Dieses Kapitel enthält Informationen zu den Datenbanktypen, die mit HP Operations Manager i (OMi) verwendet werden.

Datenbank – Übersicht

Für das Arbeiten mit OMi müssen Sie die folgenden Datenbanken einrichten:

- **Management.** Zur Speicherung von systemweiten und verwaltungsbezogenen Metadaten für die OMi-Umgebung. Für OMi ist eine Management-Datenbank erforderlich. Sie können diese Datenbank manuell oder mithilfe des Konfigurations-Assistenten erstellen.
- **Ereignis.** Zur Speicherung von Ereignissen und der zugehörigen Daten, wie z. B. Anmerkungen, sowie zur Speicherung von Konfigurationsdaten, wie z. B. Ereigniskorrelationsregeln.
- **Run-time Service Model (RTSM).** Zur Speicherung von Konfigurationsinformationen, die von verschiedenen OMi-Applikationen und -Werkzeugen sowie Applikationen und Werkzeugen von Drittanbietern gesammelt werden. Diese Informationen werden beim Erstellen von OMi-Ansichten verwendet.

Sie können OMi-Datenbanken auf einem Microsoft SQL-Server, einem Oracle-Server oder einem PostgreSQL-Server einrichten, je nach Typ des Datenbankservers, der in Ihrer Organisation verwendet wird:

- Wenn Sie mit einer Microsoft SQL Server-Datenbank arbeiten, finden Sie weitere Informationen unter "[Bereitstellen und Verwalten der Microsoft SQL Server-Datenbanken](#)" auf Seite 13.
- Wenn Sie mit einer Oracle Server-Datenbank arbeiten, finden Sie weitere Informationen unter "[Bereitstellen und Verwalten der Oracle Server-Datenbank](#)" auf Seite 51.
- Wenn Sie mit einer PostgreSQL Server-Datenbank arbeiten, finden Sie weitere Informationen unter "[Bereitstellen und Verwalten der PostgreSQL Server-Datenbanken](#)" auf Seite 83.

In den Anhängen sind zusätzliche Informationen zu Microsoft SQL Server-, Oracle Server- und PostgreSQL Server-Datenbanken enthalten.

Hinweis: Datenbankserver müssen so eingestellt sein, dass die Zeitzone, die Sommerzeiteinstellungen und die Uhrzeit den entsprechenden Einstellungen der OMi-Server entsprechen.

Anforderungen

In der folgenden Tabelle werden die empfohlenen Hardwareanforderungen (CPU und Speicher) des OMi-Datenbankservers beschrieben:

Größe der Bereitstellung	Anzahl von Prozessoren	Physischer Speicher
Klein (bis zu 2.000 Knoten)	2 CPU-Kerne	Minimum: 2 GB RAM Empfohlen: 4 GB RAM
Mittel (bis zu 5.000 Knoten)	2 CPU-Kerne	Minimum: 2 GB RAM Empfohlen: 6 GB RAM
Groß (mehr als 5.000 Knoten)	Mindestens 4 CPU-Kerne	Minimum: 4 GB RAM Empfohlen: 8 GB RAM und mehr

Informationen zu den Softwareanforderungen finden Sie in der Tabelle zur OMi-Unterstützung unter folgender Adresse:

<http://support.openview.hp.com/selfsolve/document/KM323488>

Konfiguration des Datenbankzeichensatzes

Beim Erstellen einer Datenbank und Angeben eines Zeichensatzes, mit dem festgelegt wird, welche Sprachen gespeichert und in der Datenbank und OMi dargestellt werden können, müssen Sie den Zeichensatz auswählen, der die in Ihrer OMi-Umgebung benötigten Zeichen unterstützt. Sie müssen besondere Vorsicht bei der Verwendung von Sprachen mit nicht lateinischen Zeichen oder bei der Arbeit in einer mehrsprachigen Umgebung walten lassen, in der das Speichern und Verarbeiten von Daten in mehreren Sprachen gleichzeitig durchgeführt wird.

Für eine erfolgreiche Kommunikation müssen sowohl OMi als auch die Datenbank einen kompatiblen Zeichensatz verwenden. Andernfalls werden die Daten nicht ordnungsgemäß gespeichert und es treten Fehler auf. Darüber hinaus kann das Verwenden desselben Zeichensatzes die Leistung erhöhen, da die Notwendigkeit der Zeichenkonvertierung reduziert oder eliminiert ist. Weitere Informationen zu Überlegungen und aktuellen Einschränkungen bei der Verwendung von nicht lateinischen Zeichen finden Sie im OMi-Verwaltungshandbuch.

Hinweis: Sie können die Oracle- und PostgreSQL Server-Datenbanken für das Speichern von mehrsprachigen Daten in OMi verwenden. Mit der aktuellen Implementierung ist es jedoch für OMi nicht möglich, mehrsprachige Daten in der Microsoft SQL Server-Datenbank zu speichern; derzeit können nicht mehrere Zeichensätze gleichzeitig verwendet werden.

Abhängig vom Typ des Datenbankservers finden Sie weitere Informationen in einem der folgenden Abschnitte:

- "Zeichensatzunterstützung in Microsoft SQL Server" auf Seite 15
- "Zeichensatzunterstützung in Oracle" auf Seite 53

- ["Zeichensatzunterstützung in PostgreSQL" auf Seite 86](#)

Teil II: Bereitstellen und Verwalten der Microsoft SQL Server-Datenbanken

Kapitel 2: Microsoft SQL Server-Bereitstellung - Übersicht

Sie können Microsoft SQL Server für die Bereitstellung von OMi-Datenbanken verwenden. In diesem Kapitel werden die folgenden Themen zur Bereitstellung von Microsoft SQL Server für die Verwendung mit OMi beschrieben:

- ["Informationen über die Microsoft SQL Server-Bereitstellung" unten](#)
- ["Systemanforderungen" unten](#)
- ["Zeichensatzunterstützung in Microsoft SQL Server" auf der nächsten Seite](#)

Informationen über die Microsoft SQL Server-Bereitstellung

Zum Bereitstellen von Microsoft SQL Server für die Verwendung mit OMi führen Sie die folgenden Verfahren durch:

- **Installieren und konfigurieren Sie Microsoft SQL Server.**
Weitere Informationen zum Installieren und Konfigurieren von Microsoft SQL Server finden Sie unter ["Installieren und Konfigurieren von Microsoft SQL Server" auf Seite 21](#).
- **Erstellen Sie Datenbanken unter Microsoft SQL Server.**
Sie können OMi-Datenbanken manuell erstellen oder den Konfigurationsassistenten zum Erstellen der Datenbanken verwenden.
Weitere Informationen zum manuellen Erstellen von Datenbanken für OMi finden Sie unter ["Manuelles Erstellen der OMi-Datenbanken" auf Seite 17](#).
Weitere Informationen zum Erstellen der Microsoft SQL-Datenbank finden Sie unter ["Erstellen und Konfigurieren von Microsoft SQL Server-Datenbanken" auf Seite 30](#).

Systemanforderungen

In diesem Abschnitt werden die Systemanforderungen für das Arbeiten mit Microsoft SQL Server in Kombination mit OMi beschrieben.

Hardwareanforderungen

Richtlinien für die OMi-Hardware-Dimensionierung finden Sie unter ["Anforderungen" auf Seite 11](#). Die Hardwareanforderungen für Microsoft SQL Server finden Sie im Installationshandbuch für die Microsoft SQL Server-Version für Ihr Betriebssystem.

Softwareanforderungen

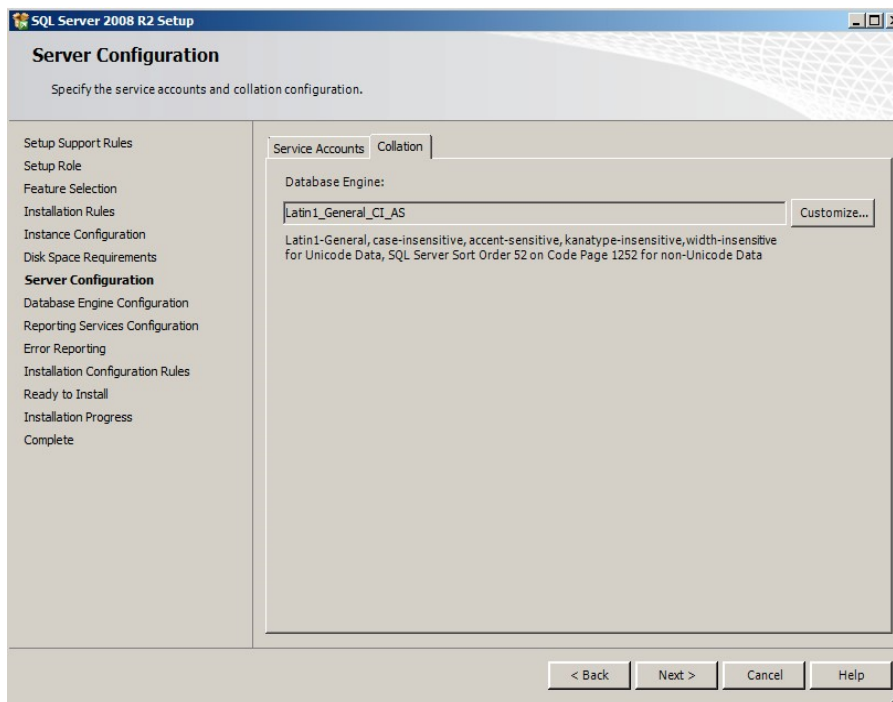
Eine Liste der unterstützten Microsoft SQL Server-Datenbankversionen finden Sie in der Tabelle zur OMi-Unterstützung unter:

<http://support.openview.hp.com/selfsolve/document/KM323488>

Zeichensatzunterstützung in Microsoft SQL Server

Da Microsoft SQL Server keine UTF-8-Codierung unterstützt (obwohl die Daten intern in den Feldern NVARCHAR als UCS oder UTF-16 gespeichert werden können), kann OMi keine mehrsprachigen Daten speichern, wenn eine Microsoft SQL Server-Datenbank verwendet wird. Es ist jedoch noch möglich, eine oder mehrere Sprachen mit nicht lateinischen Zeichen zu verwenden, die allerdings denselben Zeichensatz verwenden müssen. In diesem Fall müssen Sie eine Sortierung angeben, die der OMi-Sprache und dem Gebietsschema des Computers entspricht.

Während der Datenbankinstallation ermittelt das Setupprogramm das aktuelle Gebietsschema und schlägt die am besten geeignete Sortierung vor.



Die folgende Tabelle enthält Sprachen und ihre standardmäßigen Sortierungen:

Sprache	Standardsortierung
Englisch	Latin1_General_CI_AS
Französisch	French_CI_AS

Sprache	Standardsortierung
Spanisch	Modern_Spanish_CI_AS
Deutsch	Latin1_General_CI_AS
Russisch	Cyrillic_General_CI_AS
Japanisch	Japanese_CI_AS
Koreanisch	Korean_Wansung_CI_AS
Vereinfachtes Chinesisch	Chinese_PRC_CI_AS

Kapitel 3: Manuelles Erstellen der Microsoft SQL Server-Datenbank

In diesem Kapitel wird die manuell Erstellung von OMi-Datenbanken unter Microsoft SQL Server beschrieben.

Dieses Kapitel umfasst die folgenden Themen:

- ["Microsoft SQL Server-Datenbanken - Übersicht" unten](#)
- ["Manuelles Erstellen der OMi-Datenbanken" unten](#)

Microsoft SQL Server-Datenbanken - Übersicht

OMi verwendet die Management-, RTSM- und Ereignisdatenbanken.

Während des OMi-Setups können neue Datenbanken automatisch über das Setupverfahren eingerichtet werden oder vorhandene Datenbanken genutzt werden. Die vorhandenen Datenbanken können entweder manuell im Voraus (z. B. aufgrund von Sicherheitsbeschränkungen im Unternehmen) oder durch eine frühere Installation der gleichen OMi-Version erstellt werden.

Weitere Informationen zum Installieren des OMi-Servers finden Sie im OMi-Installationshandbuch.

Manuelles Erstellen der OMi-Datenbanken

Nachdem Sie Microsoft SQL Server installiert und konfiguriert haben, erstellen Sie OMi-Datenbanken für die Speicherung von Daten, die mit OMi gesammelt wurden.

Sie können es OMi auch ermöglichen, OMi-Datenbanken mithilfe des Konfigurations-Assistenten zu erstellen.

Wenn Sie die Datenbanken im Voraus erstellt haben (siehe unten), können Sie über den Konfigurations-Assistenten eine Verbindung von OMi zu den vorhandenen Datenbanken herstellen.

Informationen zum Konfigurations-Assistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.

Dieser Abschnitt umfasst die folgenden Themen:

- ["Erstellen von Datenbanken" unten](#)
- ["Erstellen der Objekte" auf der nächsten Seite](#)

Erstellen von Datenbanken

Der Datenbankadministrator sollte die folgenden Microsoft SQL Server-Datenbanken erstellen: eine für Management-Objekte, eine für RTSM-Objekte und eine für Ereignis-Objekte.

Zum Erstellen einer Datenbank müssen Sie über die Berechtigung **DATENBANK ERSTELLEN** verfügen. Um eine Verbindung zu einer bestehenden Datenbank herzustellen, muss das Anmeldekonto, mit dem Sie verbunden sind, zu **dbo** in der Datenbank zugewiesen werden.

Hinweis: Mitglieder der Serverrolle **sysadmin** verfügen automatisch über die Berechtigungen **DATENBANK ERSTELLEN** und werden auch in allen Datenbanken zu **dbo** zugewiesen. Ein Datenbankbesitzer wird automatisch der Rolle **dbo** in der Datenbank zugeordnet.

Weitere Informationen zum manuellen Erstellen finden Sie unter "[Erstellen und Konfigurieren von Microsoft SQL Server-Datenbanken](#)" auf Seite 30.

Hinweis: Wenn das OMi-Setupverfahren die Datenbanken automatisch erstellt, verwendet es zwei Dateigruppen für jede Datenbank: eine für die Aufnahme der Systemtabellen und eine für die Applikationsdaten.

Erstellen der Objekte

Nach dem Erstellen der einzelnen OMi-Datenbanken erstellen Sie Objekte, indem Sie SQL-Skripts ausführen. Die Skripts zum Erstellen von Objekten befinden sich im folgenden Verzeichnis:

```
<OMi_GW_Stammverzeichnis>/AppServer/webapps/site.war/DataBases/SQL_Svr_DB_Utils
```

Für die Management-Datenbank wird das SQL-Skript dynamisch durch das Dienstprogramm `generate_create_script` erstellt:

```
<OMi_GW_Stammverzeichnis>/dbverify/bin/generate_create_script
```

Hinweis: Die SQL-Skriptdateien, die vom Dienstprogramm `generate_create_script` erstellt werden, erhalten Verweise auf die Dateigruppe [PRIMARY]. Um eine andere Dateigruppe auszuwählen, öffnen Sie manuell die .SQL-Datei in einem Texteditor und ersetzen alle Instanzen von [PRIMARY] durch die gewünschte Dateigruppe.

Erstellen der Management-Objekte

Zum Erstellen der OMi-Management-Objekte führen Sie das unten beschriebene Verfahren zum Generieren und Ausführen des SQL-Skripts für die Management-Datenbank aus:

So erstellen Sie das SQL-Skript für die Management-Datenbank:

Führen Sie eine der folgenden Skriptdateien im Verzeichnis `<OMi_GW_Stammverzeichnis>/dbverify/bin` aus:

Für Windows:

```
generate_create_script.bat [Bereitstellungstyp] [Name des generierten Skripts] [DB-Typ] [Sitzungstyp]
```

Für Linux:

```
./generate_create_script.sh [Bereitstellungstyp] [Name des generierten Skripts] [DB-Typ] [Sitzungstyp]
```

Die folgenden Parameter müssen für das Management-Schema verwendet werden:

- **deployment_type.** EE (Enterprise Edition)
- **Name des generierten Skripts.** Der Name der zu generierenden SQL-Datei, z. B. `Management.sql`.

- **DB-Typ.** SqlServer
- **Sitzungstyp.** Management

Beispiel:

```
/opt/HP/BSM/dbverify/bin/generate_create_script.sh EE Management.sql SqlServer  
Management
```

So führen Sie das SQL-Skript für die Management-Datenbank aus:

1. Erstellen Sie die Management-Datenbank. Weitere Informationen finden Sie unter "[Erstellen von Datenbanken](#)" auf Seite 17.
2. Stellen Sie die Verbindung zu der Management-Datenbank her und führen Sie das im vorherigen Abschnitt generierte Skript zum Bereitstellen der Management-Objekte aus.
3. Führen Sie das folgende SQL-Skript aus dem Verzeichnis der Objektskripts aus:

```
management_sql_inserts.sql
```

Erstellen der RTSM-Objekte

Erstellen Sie eine Datenbank zum Speichern der RTSM-Objekte. Es sind keine Skripts erforderlich, um die Objekte zum Füllen dieser Datenbank zu erstellen.

Erstellen der Ereignisobjekte

Erstellen Sie die Ereignisdatenbank zum Speichern von Ereignisobjekten. Führen Sie das untenstehende Verfahren durch, um Ereignisobjekte zu erstellen:

So erstellen Sie das SQL-Skript für die Ereignisdatenbank:

Führen Sie eine der folgenden Skriptdateien aus:

Für Windows:

```
<OMi_STAMMVERZEICHNIS>\opr\bin\opr-schema-script-generator.bat
```

Für Linux:

```
<OMi_STAMMVERZEICHNIS>/opr/bin/opr-schema-script-generator.sh
```

Für den Parameter **dbType** muss **SQL** als Ereignisschema festgelegt werden.

Geben Sie den folgenden Befehl ein, um beispielsweise Datenbanktabellen für eine SQL-Datenbank für die Datei **oprGenerateDbTablesForSqlServer.sql** zu erstellen:

```
<OMi_HOME>\opr\bin\opr-schema-script-generator.bat -d  
SQL -o C:\temp\oprGenerateDbTablesForSqlServer.sql
```

So führen Sie das SQL-Skript für die Ereignisdatenbank aus:

1. Erstellen Sie die Ereignisdatenbank. Weitere Informationen finden Sie unter "[Erstellen von Datenbanken](#)" auf Seite 17.
2. Stellen Sie die Verbindung zu der Ereignisdatenbank her und führen Sie das im vorherigen Abschnitt generierte Skript zum Bereitstellen der Operations-Objekte aus.

Hinweis: Führen Sie die oben genannten Verfahren nur durch, wenn Sie ein erfahrener Datenbankadministrator für Microsoft SQL Server sind.

Kapitel 4: Installieren und Konfigurieren von Microsoft SQL Server

Dieses Kapitel enthält Informationen über das Microsoft SQL Server-Installationsverfahren und die Konfigurationseinstellungen.

Dieses Kapitel umfasst die folgenden Themen:

- ["Workflow für die Microsoft SQL Server-Bereitstellung"](#) unten
- ["Voraussetzungen für die Installation"](#) auf der nächsten Seite
- ["Hinweise und Einschränkungen"](#) auf der nächsten Seite
- ["Prüfliste für Support und Zertifizierung"](#) auf Seite 23
- ["Installieren von Microsoft SQL Server"](#) auf Seite 24
- ["Konfigurieren von Microsoft SQL Server"](#) auf Seite 25
- ["Überprüfen und Ändern von Server- und Datenbankeinstellungen"](#) auf Seite 26

Workflow für die Microsoft SQL Server-Bereitstellung

Führen Sie die folgenden Schritte aus, um Microsoft SQL Server für die Verwendung mit OMi bereitzustellen:

- 1. Überprüfen der Richtlinien für die Größe von Microsoft SQL Server**
Weitere Informationen finden Sie unter ["Datenbankdateilayout"](#) auf Seite 31.
- 2. Prüfen der Voraussetzungen für die Installation**
Weitere Informationen finden Sie unter ["Voraussetzungen für die Installation"](#) auf der nächsten Seite.
- 3. Überprüfen von Server- und Datenbankprüflisten**
In diesen Prüflisten werden die Server- und Datenbankkonfigurationsoptionen zusammengefasst, die für OMi unterstützt und empfohlen werden.
Weitere Informationen finden Sie unter ["Prüfliste für Support und Zertifizierung"](#) auf Seite 23.
- 4. Installieren der Microsoft SQL Server Enterprise Edition**
Weitere Informationen finden Sie unter ["Installieren von Microsoft SQL Server"](#) auf Seite 24.
- 5. Konfigurieren von Microsoft SQL Server**
Weitere Informationen finden Sie unter ["Konfigurieren von Microsoft SQL Server"](#) auf Seite 25.
- 6. Erstellen von OMi-Datenbanken in Microsoft SQL Server**
Weitere Informationen finden Sie unter ["Erstellen von Datenbanken"](#) auf Seite 17.
- 7. Konfigurieren von OMi-Datenbanken**
Weitere Informationen finden Sie unter ["Konfigurieren von Datenbanken"](#) auf Seite 34.
- 8. Überprüfen von Microsoft SQL Server und Datenbanken**

Weitere Informationen finden Sie unter ["Überprüfen und Ändern von Server- und Datenbankeinstellungen"](#) auf Seite 26.

9. Windows-Authentifizierung einrichten

Weitere Informationen finden Sie unter ["Verwenden der Windows-Authentifizierung für den Zugriff auf Microsoft SQL Server-Datenbanken"](#) auf Seite 49. Dieser Schritt ist nur relevant, wenn Sie die Windows-Authentifizierung anstelle der SQL Server-Authentifizierung ausgewählt haben.

Voraussetzungen für die Installation

• Systemanforderungen

Eine Liste der unterstützten Betriebssysteme finden Sie unter ["Systemanforderungen"](#) auf Seite 14.

• Software

Software	Benötigte Version
Microsoft Windows Installer	4.5 oder höher

• Speicherort der Datenbankdateien

Zur Verbesserung der Leistung wird empfohlen, die OMi-Datenbanken auf mehrere Festplatten oder RAID-Systeme zu verteilen.

• Datenspeicher

Mindestens 2 GB RAM.

• Benutzerkonten

- Wenn Sie planen, Aktivitäten außerhalb des lokalen Computers durchzuführen (wie das Kopieren von Dateien zu oder von einem anderen Server, Sicherung auf freigegebenen Remote-Repositorys usw.), müssen Sie über ein Domänenkonto für Microsoft SQL Server-Dienste verfügen.
- Wenn Sie OMi mithilfe des Installationsassistenten installieren möchten, müssen Sie ein Benutzerkonto angeben, das über Berechtigungen für die Datenbankerstellung verfügt. Wenn Datenbanken manuell erstellt werden sollen, müssen Sie für OMi ein Benutzerkonto angeben, das Teil von **db_datareader**, **db_datawriter** und **db_ddladmin** in jeder OMi-Datenbank ist.

• Sonstiges

- Stellen Sie sicher, dass die Festplatte, auf der Microsoft SQL Server installiert werden soll, nicht komprimiert ist.
- Deaktivieren Sie Antivirussoftware und andere Anwendungen, die bei der Installation mit Microsoft SQL Server in Konflikt geraten können.

Hinweise und Einschränkungen

- Microsoft SQL Server muss auf einem speziellen Computer für das Hosten der OMi-Datenbanken installiert werden.
- Der Standardwert für die Speichereinstellungen sollte nicht geändert werden. Aktivieren Sie für

Microsoft SQL Server die dynamische Speicherverwaltung, außer Sie konfigurieren die Option **AWE-fähig**. Die AWE-Funktion wurde aus SQL Server 2012 entfernt und wird nicht für die Anwendung ab SQL Server 2008 R2 empfohlen.

- Die 64-Bit-Versionen von Microsoft SQL Server umfassen Unterstützung für erweiterte Systeme, auch bekannt als WOW64 (Windows on Windows). WOW64 ist eine Funktion der 64-Bit-Editionen von Microsoft Windows, die es 32-Bit-Applikationen ermöglichen, systemintern im 32-Bit-Modus ausgeführt zu werden.
- Anwendungen funktionieren im 32-Bit Modus, obwohl das zugrundeliegende Betriebssystem auf der 64-Bit-Plattform ausgeführt wird.
- Bei der Arbeit mit Microsoft SQL Server müssen die RTSM-Sortierung und die SQL Server-Sortierung identisch sein, damit ein Upgrade ordnungsgemäß ausgeführt werden kann. Weitere Informationen finden Sie unter "[Konfiguration des Datenbankzeichensatzes](#)" auf Seite 11.

Prüfliste für Support und Zertifizierung

Die Informationen in diesem Abschnitt beziehen sich auf unterstützte und zertifizierte Microsoft SQL Server-Optionen. Die zertifizierten Optionen werden für das Arbeiten mit OMi empfohlen.

Betreff	Microsoft SQL Server	
	Unterstützt	Empfohlen
Instanzen	Standard, Einzelne	
Authentifizierungsmethode	Gemischt, Windows	
Sortierung	Groß-/Kleinschreibung ignorieren. OMi unterstützt keine binäre Sortierreihenfolge und die Beachtung der Groß-/Kleinschreibung. Es wird nur die Groß- und Kleinschreibung mit einer Kombination aus Akzent-, Kana- oder Breitereinstellung unterstützt.	Verwenden Sie die Sortierungsoptionen im Dialogfeld Einstellungen , um die Sortierung auszuwählen. Aktivieren Sie nicht das Kontrollkästchen für die binäre Einstellung. Akzent-, Kana- und Breitereinstellungen sollten gemäß der relevanten Anforderungen der Datensprache festgelegt werden. Die ausgewählte Sprache muss mit den Spracheinstellungen für das Gebietsschema im Windows-Betriebssystem übereinstimmen.
Netzwerkbibliotheken	Server: TCP/IP und Named Pipes Client: TCP/IP und Named Pipes	Server: TCP/IP Client: TCP/IP
Serverkonfigurationsoptionen	Standardeinstellungen, wenn nicht anders angegeben	
Eigenschaften der Datendatei	Manuelles Dateiwachstum oder FILEGROWTH muss kleiner als	FILEGROWTH: ~30 - 100 MB

Betreff	Microsoft SQL Server	
	Unterstützt	Empfohlen
	oder gleich 100 MB sein	
Datenbankeigenschaften für Sortierung	Serverstandard	
Datenbankoptionen	Standardeinstellungen, wenn nicht anders angegeben	
Wiederherstellungsmodell	Alle	Vollständig

Installieren von Microsoft SQL Server

Es ist wichtig, dass Sie sich mit allen Installationsdetails vertraut machen, damit Sie die geeigneten Optionen auswählen können. Das Auswählen der Standardoptionen kann sich in bestimmten Fällen negativ auf die Leistung von Microsoft SQL Server auswirken.

Wählen Sie die folgenden Optionen in den Dialogfeldern der Installation aus:

- **Konfiguration im Dialogfeld zur Funktionsauswahl.**
 - Entfernen Sie die Option **Volltextsuche** aus der Liste, da OMi diese Indextsuchfunktion nicht verwendet.
 - Stellen Sie unter **Zielordner** sicher, dass das Verzeichnis der Datendateien auf einem fehlertoleranten Datenträgersystem, z. B. RAID 1, gespeichert ist. Obwohl diese Systemdatenbanken relativ klein sind, sind sie wichtig für den Betrieb von Microsoft SQL Server.
- **Konfiguration des Dialogfelds "Instanzname".** Weitere Informationen zu benannte Instanzen finden Sie unter "[Portzuweisungsoptionen für benannte Instanzen](#)" auf Seite 28. Sie greifen auf eine Standardinstanz durch die Angabe des Servernamens oder der IP-Adresse zu.
- **Konfiguration des Dialogfelds "Dienstkonto".**
 - Wenn alle Microsoft SQL Server-Aktivitäten außerhalb des lokalen Computers durchgeführt werden (z. B. das Kopieren von Dateien zu oder von einem anderen Server, die Sicherung auf freigegebenen Remote-Repositorys, die Replikation mit anderen Servern, Jobschritte für ActiveX-Skripts, CmdExec-Jobschritte usw.), wählen Sie die Option zum Verwenden eines Domänenbenutzerkontos aus und geben Sie den Benutzernamen, das Kennwort und die Domäne eines Benutzers an, der Mitglied der Administratorgruppe des lokalen Computers ist und die entsprechenden Berechtigungen für Netzwerkressourcen hat.
 - Wenn alle Microsoft SQL Server-Aktivitäten auf die lokalen Computer beschränkt sind, wählen Sie die Option für das integrierte Systemkonto und dann **Lokales System** aus. Diese Auswahl gewährt nur SQL Server-Administratorberechtigungen auf dem lokalen Computer.
- **Dialogfeld "Authentifizierungsmodus".** OMi arbeitet sowohl mit der Windows-Authentifizierung als auch mit der Microsoft SQL Server-Authentifizierung (empfohlen). Zum Aktivieren der Microsoft SQL Server-Authentifizierung gehen Sie wie folgt vor:

- a. Wählen Sie **Gemischter Modus (Windows-Authentifizierung und SQL Server-Authentifizierung)**.
- b. Geben Sie das **sa**-Kennwort des Benutzers ein.

Hinweis: Aus Sicherheitsgründen ist es bei Microsoft SQL Server wichtig, ein Kennwort einzugeben.

- **Dialogfeld "Sortierungseinstellungen"**. Folgen Sie den Empfehlungen in der Checkliste unter "[Prüfliste für Support und Zertifizierung](#)" auf Seite 23.

Die oben beschriebenen Einstellungen wirken sich nur auf die Systemdatenbanken aus und dienen als Standardeinstellungen für Benutzerdatenbanken. Datenbanken können andere Sortierungseinstellungen als die Standardeinstellungen des Servers aufweisen, und eine Tabellenspalte kann andere Sortierungseinstellungen als die Standardeinstellungen der Datenbank aufweisen. Dies wird für dedizierte OMi-Datenbanken jedoch nicht empfohlen. Aufgrund der Flexibilität bei der Sortierungsverwaltung in Microsoft SQL Server können Sie eine Datenbank mit anderen Sortierungseinstellungen wiederherstellen oder anfügen.

Wichtig: Das Ändern der oben aufgeführten Einstellungen erfordert eine Skripterstellung für alle Systemobjekte und Routinen (Anmeldungen, benutzerdefinierte Systemmeldungen, in "master" gespeicherte Prozeduren und usw.), eine Neuinstallation von Microsoft SQL Server (oder das Ausführen des Dienstprogramms RebuildM.exe) mit den neuen Einstellungen, eine Neuerstellung aller Systemobjekte aus den gespeicherten Skripten und das Anfügen der Benutzerdatenbanken. Es wird daher empfohlen, die entsprechenden Optionen während der Installation auszuwählen.

- **Dialogfeld "Datenverzeichnisse"**. Geben Sie die Zielordner für die Daten- und Protokolldateien der Benutzerdatenbanken und temporären Datenbanken sowie das Sicherungsverzeichnis an. Es wird empfohlen, Daten- und Protokolldateien auf Hochleistungsspeichersystemen zu speichern.

Installieren Sie das neueste Service Pack für Microsoft SQL Server, das auf der Microsoft Download Center-Website unter folgender Adresse verfügbar ist:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=cb6c71ea-d649-47ff-9176-e7cac58fd4bc&DisplayLang=en>

Informationen zum aktuellsten Service Pack, das für OMi unterstützt wird, finden Sie unter "[Softwareanforderungen](#)" auf Seite 15.

Konfigurieren von Microsoft SQL Server

In diesem Abschnitt werden die Service- und Serveroptionen beschrieben, die Sie nach der Installation von Microsoft SQL Server konfigurieren können. Folgende Themen werden behandelt:

- "[Optionen für die Servicekonfiguration](#)" unten
- "[Serverkonfigurationsoptionen](#)" auf der nächsten Seite

Optionen für die Servicekonfiguration

Wenn Sie **Volltextsuche** installiert haben, stellen Sie sicher, dass die Option deaktiviert und auf den manuellen Modus festgelegt ist (suchen Sie den Dienst im Applet **Dienste** mit Microsoft Search), damit

keine Ressourcen verschwendet werden.

Wenn Sie verteilte Transaktionen verwenden, stellen Sie sicher, dass der Dienst **Distributed Transactions Coordinator** deaktiviert oder auf den manuellen Modus festgelegt ist.

Wenn die Option für dynamische Ports für Microsoft SQL Server-Instanzen verwendet wird, stellen Sie sicher, dass der Dienst **SQL Server-Browser** sowie alle SQL Server-Instanzdienste im automatischen Modus ausgeführt werden.

Serverkonfigurationsoptionen

Die meisten Serverkonfigurationsoptionen werden dynamisch durch Microsoft SQL Server konfiguriert. Für die OMi-Zertifizierung dürfen Sie die Standardoptionen nicht ändern, sofern Sie nicht durch den HP Software Support dazu aufgefordert werden.

In bestimmten Situationen kann eine Änderung der Standardeinstellungen wünschenswert sein. Sie können diese Einstellungen in der gespeicherten Prozedur **sp_configure** oder in den verschiedenen Dialogfeldern in Microsoft SQL Management Studio (vor allem im Dialogfeld **Servereigenschaften**) ändern.

Die Standardwerte sollten für alle Elemente mit Ausnahme der folgenden Elemente verwendet werden:

- Agent XPs (A) = 1
- awe aktiviert (A) = 1. Dieser Wert sollte nur geändert werden, wenn mehr als 4 GB auf einem 32-Bit-Computer erforderlich sind. Beachten Sie, dass diese Funktion ab Microsoft SQL Server 2012 nicht mehr verfügbar ist.

Eine vollständige Liste der Microsoft SQL Server-Konfigurationsoptionen finden Sie unter "[Microsoft SQL Server-Konfigurationsoptionen](#)" auf Seite 101.

Überprüfen und Ändern von Server- und Datenbankeinstellungen

Die folgende Tabelle enthält eine Übersicht über die Verfahren zur Prüfung oder zum Ändern von Server- und Datenbankeinstellungen:

Server-/Datenbankeinstellung	Überprüfen/Ändern der Einstellung
Standardinstanz	Im Dienste-Applet des Betriebssystems wird eine Standardinstanz von Microsoft SQL Server und eine benannte Instanz als SQL Server (Instanzname) angezeigt.
Authentifizierungsmethode	Klicken Sie in Microsoft SQL Server Management Studio mit der rechten Maustaste auf den Server, wählen Sie Eigenschaften aus und klicken Sie auf die Registerkarte Sicherheit . Wählen Sie SQL Server- und Windows-Authentifizierungsmodus (Gemischter Modus) aus.
Sortierungseinstellungen	Führen Sie den folgenden Befehl aus: <code>sp_helpsort</code>

Server-/Datenbankeinstellung	Überprüfen/Ändern der Einstellung
<p>Netzwerkbibliotheken</p>	<p>Wählen Sie auf dem Server Start > Programme > Microsoft SQL Server <Versionsnummer> > Konfigurationswerkzeuge > Konfigurations-Manager aus.</p> <p>Wählen Sie unter SQL Native Client die Option Clientprotokolle aus, und vergewissern Sie sich, dass das ausgewählte Protokoll aktiviert ist.</p> <ul style="list-style-type: none"> • Unterstützt Gemeinsamer Speicher, TCP/IP und Named Pipes für Server und Client • Empfohlen Nur TCP/IP für Server und Client
<p>Serverkonfigurationsoptionen anzeigen oder ändern</p>	<ul style="list-style-type: none"> • Führen Sie EXEC sp_configure 'show advanced options', 1 reconfigure with override aus, um die Anzeige aller Optionen zuzulassen. • Führen Sie EXEC sp_configure aus, um die aktuellen Werte anzuzeigen. • Führen Sie EXEC sp_configure '<Option>', <Wert> aus, um eine Einstellung zu ändern. <p>Einige Optionen werden erst wirksam, wenn Sie reconfigure mit override ausführen, während andere einen Neustart des Microsoft SQL Server-Dienstes erfordern. Weitere Informationen finden Sie in der Microsoft SQL Server-Dokumentation.</p>
<p>Prüfen Sie, ob der Benutzer, der OMi-Datenbanken erstellen möchte, über die Berechtigungen zum Erstellen von Datenbanken verfügt.</p>	<p>Melden Sie sich bei Microsoft SQL Server Management Studio mit der Anmeldung an, die geprüft werden soll, und führen Sie die folgenden Schritte aus:</p> <pre>USE master IF PERMISSIONS() & 1 = 1 PRINT 'User has CREATE DATABASE permissions' ELSE PRINT 'User does not have CREATE DATABASE permissions'</pre>
<p>Überprüfen Sie, ob der OMi-Datenbankbenutzer über ausreichende Berechtigungen in der Datenbank verfügt.</p>	<ul style="list-style-type: none"> • Melden Sie sich bei Microsoft SQL Server Management Studio mit dem Benutzernamen an, den Sie überprüfen möchten. • Ändern Sie den Kontext der Datenbank auf die erforderliche Datenbank. • Öffnen Sie eine neue Abfrage und führen Sie die folgenden Schritte in jeder Datenbank durch: <pre>select case when IS_MEMBER ('db_owner')=1 or IS_SRVROLEMEMBER ('sysadmin')=1 or (IS_MEMBER ('db_ddladmin') = 1 and IS_MEMBER ('db_datareader')=1 and IS_MEMBER ('db_datawriter')=1 and</pre>

Server-/Datenbankeinstellung	Überprüfen/Ändern der Einstellung
	<pre>IS_MEMBER ('db_denydatareader')=0 and IS_MEMBER ('db_denydatawriter')=0) then 'User has enough permissions' else 'User does not have enough permissions' end</pre>
Das Zielverzeichnis für Daten- und Protokolldatei ist nicht komprimiert (nur in NTFS)	Klicken Sie mit der rechten Maustaste auf das Verzeichnis und wählen Sie Eigenschaften anschließend Erweitert aus. Stellen Sie sicher, dass das Kontrollkästchen Komprimierung deaktiviert ist.
Eigenschaften für Datenbank- und Datenbankdatei (einschließlich Wiederherstellungsmodell und Sortierungseigenschaften)	<ul style="list-style-type: none"> Um die Eigenschaften von Datenbank- und Datenbankdatei anzuzeigen, führen Sie Folgendes aus: <pre>EXEC sp_helpdb <Datenbankname></pre> Um die Datenbankeigenschaften zu ändern, führen Sie Folgendes aus: <pre>ALTER DATABASE <Datenbankname> SET <Option> <Wert></pre> Um die Datendateieigenschaften zu ändern, führen Sie Folgendes aus: <pre>ALTER DATABASE <Datenbank> MODIFY FILE (name = <Dateiname>, <Eigenschaft> = <Wert>)</pre> <p>Sie können diese Eigenschaften auch über das Dialogfeld Datenbankeigenschaften im Enterprise Manager anzeigen oder ändern.</p>
Service Pack-Version und -Edition von Microsoft SQL Server	Melden Sie sich bei Management Studio an und führen Sie folgende Abfrage durch: <code>select @@version.</code>

Portzuweisungsoptionen für benannte Instanzen

Bei der Arbeit mit einer benannten Instanz in SQL Server gibt es zwei Möglichkeiten für die Portzuordnung der Instanz: Dynamischer Port und statischer Port.

- **Dynamischer Port.** Bei der Arbeit mit einem dynamischen Port wird ein neuer Port bei jedem Instanzstart zur Instanz zugewiesen. Damit Clients den Port beim Verbinden mit der Instanz erkennen, muss der Dienst **SQL Server-Browser** gestartet werden. Der Browser-Dienst hört Port 1434 ab und leitet Clients zu dem richtigen Port entsprechend dem gewünschten Instanznamen. Sie greifen auf die benannte Instanz durch die Angabe des Servernamens oder der IP-Adresse zu, gefolgt von `\<Instanzname>`. Beispiel: `Server1\inst1`.
- **Statischer Port.** Bei der Arbeit mit einem statischen Port wird ein bestimmter Port zur Instanz zugewiesen und der Dienst **SQL Server-Browser** ist nicht erforderlich, um die Verbindung zur Instanz herzustellen (Sie können ihn aber trotzdem verwenden).

Die Standardeinstellung für SQL Server ist ein dynamischer Port.

Zum Arbeiten mit einem statischen Port führen Sie die folgenden Schritte aus:

1. Wählen Sie **Start > Programme > Microsoft SQL Server > Konfigurationstools > SQL Server-Konfigurations-Manager** aus.
2. Erweitern Sie Protokolle für <Instanzname> und doppelklicken Sie auf **TCP/IP**.
3. Auf der Registerkarte **IP-Adressen** des Dialogfelds mit den TCP/IP-Eigenschaften werden mehrere IP-Adressen im Format IP1, IP2 bis zu IPAll angezeigt. Für jede Adresse gilt:
 - a. Wenn im Dialogfeld **Dynamische TCP-Ports** der Wert 0 festgelegt ist, der angibt, dass die Datenbank-Engine dynamische Ports abhört, löschen Sie die 0.
 - b. Geben Sie im Feld **TCP-Port** die Portnummer ein, die diese IP-Adresse abhören soll, und klicken Sie auf **OK**.
4. Klicken Sie im Konsolenbereich auf **SQL Server-Dienste**.
5. Im Bereich mit den Details klicken Sie mit der rechten Maustaste auf **SQL Server (<Instanznamen>)** und klicken Sie auf **Neu starten**, um den Vorgang zu beenden und SQL Server neu zu starten.

Sie greifen auf die benannte Instanz durch die Angabe des Servernamens oder der IP-Adresse zu, gefolgt von der Portnummer. Beispiel: **server1\1435**.

Kapitel 5: Erstellen und Konfigurieren von Microsoft SQL Server-Datenbanken

In diesem Kapitel wird die Erstellung und Konfiguration von OMi-Datenbanken unter Microsoft SQL Server beschrieben.

Dieses Kapitel umfasst die folgenden Themen:

- ["Erstellen von Datenbanken" unten](#)
- ["Konfigurieren von Datenbanken" auf Seite 34](#)

Erstellen von Datenbanken

Dieser Abschnitt umfasst die folgenden Themen:

- ["Datenbankberechtigungen" unten](#)
- ["Datenbankdateilayout" auf der nächsten Seite](#)
- ["Systemdatenbanken" auf Seite 33](#)

Datenbankberechtigungen

Zum Erstellen einer Datenbank müssen Sie über die Berechtigung **DATENBANK ERSTELLEN** verfügen. Damit die Berechtigung **DATENBANK ERSTELLEN** einem Benutzer erteilt werden kann, muss die Benutzeranmeldung zunächst einem Datenbankbenutzer in der Masterdatenbank zugeordnet werden.

Hinweis: OMi-Anmeldekonto sollten **dbo** in der Datenbank zugeordnet werden. Mitglieder der Serverrolle **sysadmin** verfügen automatisch über die Berechtigungen **DATENBANK ERSTELLEN** und werden auch in allen Datenbanken **dbo** zugewiesen. Ein Datenbankbesitzer wird automatisch der Rolle **dbo** in der Datenbank zugeordnet.

Um zu überprüfen, ob ein Benutzer verfügt über die Berechtigungen **DATENBANK ERSTELLEN** verfügt, melden Sie sich in Management Studio mit dem Anmeldekonto des Benutzers an, dessen Berechtigungen Sie prüfen möchten, und führen Sie die folgenden Schritte aus:

```
USE master
IF PERMISSIONS() & 1 = 1
PRINT 'User has CREATE DATABASE permissions.'
```

Um zu überprüfen, ob ein Benutzer über ausreichende Berechtigungen verfügt, melden Sie sich in Management Studio mit dem Anmeldekonto des Benutzers an, dessen Zuweisungen Sie prüfen möchten. Ändern Sie den Datenbankkontext auf die erforderliche Datenbank und führen Sie den folgenden Befehl aus:

```
select case when IS_MEMBER ('db_owner')=1
or IS_SRVROLEMEMBER ('sysadmin')=1
or (IS_MEMBER ('db_ddladmin') = 1 and
IS_MEMBER ('db_datareader')=1 and
```

```
IS_MEMBER ('db_datawriter')=1 and  
IS_MEMBER ('db_denydatareader')=0 and  
IS_MEMBER ('db_denydatawriter')=0 )  
then 'User has enough permissions'  
else 'User does not have enough permissions'  
end
```

Datenbankdateilayout

Wenn Sie eine Datenbank erstellen, muss sie mindestens eine Datendatei (mit einer .mdf-Erweiterung) und eine Transaktionsprotokolldatei (mit einer .ldf-Erweiterung enthalten). Optional können Sie zusätzliche Datendateien (.ndf) sowie zusätzliche Protokolldateien (.ldf) erstellen.

Zur Leistungsverbesserung können Sie mehrere Datendateien erstellen. Microsoft SQL Server verwendet Stripes für die Daten der Datendateien. Wenn keine RAID-Controller für das Stripen der Daten vorhanden sind, können Sie die Daten über mehrere physische Datenträger verteilen und auf diese Weise für die Daten den Stripevorgang durchführen. Das Protokoll wird jedoch sequenziell gelesen, sodass kein Leistungsgewinn durch das Hinzufügen mehrerer Protokolldateien erzielt wird. Eine zusätzliche Protokolldatei sollte auf einem anderen Datenträger erstellt werden, wenn für Ihr bestehendes Protokoll keine Speicherkapazitäten mehr vorliegen.

Speicherort für Daten und Protokoll

Wichtig:

- Es wird empfohlen, keine Daten- oder Protokolldateien auf der Festplatte zu speichern, auf der die Seitendatei (Auslagerung) gespeichert ist.
- Die Daten- und Protokolldateien sollten auf verschiedenen Festplattensubsystemen gespeichert werden.
- **Protokolldateien.** Es werden keine Änderungen in die Datenbank aufgenommen, bis sie in die Protokolldateien geschrieben wurden. Die Protokollarchitektur muss dabei serielle Schreibvorgänge vorschreiben. Es wird daher empfohlen, so wenig Störungen wie möglich bei der Protokollaktivität zu verursachen. Es ist normalerweise aufgrund der seriellen Schreibvorgänge im Protokoll ausreichend, das Protokoll auf einem RAID 1-System zu speichern. Wenn Prozesse aus dem Protokoll gelesen werden (z. B. Trigger, die auf eingefügte und gelöschte Ansichten zugreifen, die aus den Protokolldatensätzen oder der Transaktionsreplikation gebildet werden) oder mehrere Protokolldateien für verschieden Datenbanken vorliegen, sollten Sie die Protokolldatei(en) auf einem RAID 1+0-System (Striped Mirror) speichern.
- **Datendateien.** Datendateien sollten für eine optimale Leistung auf einem RAID 1+0-System gespeichert werden.

Datei- und Datenbankeigenschaften

Wenn Sie eine Datenbank erstellen, können Sie die folgenden Eigenschaften für jede Datei angeben (.mdf, .ndf, .ldf):

- **NAME.** Der logische Dateiname, den Sie später verwenden können, wenn Sie eine der Eigenschaften ändern möchten.
- **FILENAME.** Der physische Dateipfad und -name. Stellen Sie sicher, dass das Zielverzeichnis nicht komprimiert ist (klicken Sie mit der rechten Maustaste auf das Verzeichnis im Windows Explorer, wählen Sie die Option **Erweitert** und stellen Sie sicher, dass das Kontrollkästchen für die Komprimierung nicht aktiviert wurde).
- **SIZE.** Die anfängliche Dateigröße.
- **MAXSIZE.** Legt die maximale Größe fest, auf die die Datei anwachsen kann. Wenn dieses Argument weggelassen wird oder wenn Sie **Unbegrenzt** angeben, kann die Datei anwachsen, bis die Kapazität des Datenträgers erreicht ist.
- **FILEGROWTH:** Das Inkrement für das automatische Wachstum der Datei. Dieses Argument kann entweder als Prozentsatz der vorhandenen Dateigröße oder als eine feste Größe angegeben werden.

Eine Operation für das automatische Wachstum, die von einer Veränderung aufgerufen wurde, die wiederum von einem Client mit Zeitüberschreitung gesendet wurde, kann nicht erfolgreich abgeschlossen werden. Das nächste Mal, wenn ein Client eine Änderung sendet, startet die Operation für das automatische Wachstum daher am Anfang und unterliegt möglicherweise auch der Zeitüberschreitung.

Um dieses Problem zu vermeiden, wird empfohlen, dass Sie entweder die Dateien jedes Mal manuell erweitern, wenn die Datenbank fast die volle Kapazität erreicht hat (z. B. 20 Prozent frei), oder Sie legen das Inkrement für das Wachstum auf eine feste Größe fest, die weniger Zeit für die Zuweisung benötigt als die festgelegte Zeitüberschreitung für den Client. Ein Inkrement mit einem kleinen Wachstum wird nicht empfohlen, da es die Fragmentierung des Dateisystems erhöht. Wenn jedoch ein sehr großes Inkrement verwendet wird, können Änderungen, die von Clients gesendet werden, zu Verbindungszeitüberschreitungen führen, während auf die automatische Erweiterung gewartet wird. Bei großen Datenbanken kann ein prozentuales Inkrement für das Wachstum zu einem exponentiellen Wachstum der Datenbank führen und sollte vermieden werden.

Einstellungen für die tempdb-Datenbank

Die häufige Erweiterung der tempdb-Systemdatenbank kann Auswirkungen auf die Leistung der Datenbank haben, insbesondere bei großen Microsoft SQL Server-Installationen. Die tempdb-Datenbank sollte aus diesem Grund so groß sein, dass eine frühe Erweiterung vermieden werden kann. Das Inkrement für das Wachstum der tempdb-Datenbank sollte groß genug sein, um eine Fragmentierung zu vermeiden, jedoch nicht zu groß, um in einem angemessenen Zeitraum eine Erweiterung durchführen zu können. Erstellen Sie die tempdb-Datenbank mit einem minimalen anfänglichen Größe von 1 GB und mit einem Inkrement für das Wachstum von 50 MB. Die tempdb-Datenbank sollte per Striping auf mehrere Festplatten aufgeteilt werden, idealerweise auf einem RAID 1+0-Controller. Es wird empfohlen, die tempdb-Datenbank auf einen eigenen Satz von Datenträgern zu verschieben.

Um sicherzustellen, dass genügend Speicherplatz für das Wachstum der tempdb-Datenbank in Zeiten starker Auslastung vorliegt (z. B. beim Aggregieren oder Sortieren von Daten), wird empfohlen, mindestens 20 GB freien Speicherplatz auf dem Laufwerk vorzusehen, auf dem sich die tempdb-Datenbank befindet.

Dateigruppen

Dateigruppen sind logische Gruppierungen von Datendateien. Die folgenden Objekte können in eine eigene Dateigruppeneinheit platziert werden:

- Die Daten einer Tabelle
- Die großen Objekte einer Tabelle (text-, ntext-, image-Spalten)
- Ein Index

Daten werden, je nach den Speicherplatzkapazitäten in den Dateien, proportional in alle Dateien eingefügt, die zu der Dateigruppe gehören. Die Datei **.mdf** befindet sich in einer Dateigruppe namens **PRIMÄR**, die als **Standard** markiert wird, wenn die Datenbank erstellt wird (die Standarddateigruppe für Objekte wenn keine Datei angegeben ist). Wenn Sie keine anderen Datendateien (**.ndf** -Dateien) in ihre eigenen Dateigruppen platzieren, werden diese auch in der Dateigruppe **PRIMÄR** aufgenommen. Sie können die **Standard**-Dateigruppe zu einem späteren Zeitpunkt ändern.

Dateigruppen können zur Leistungsoptimierung oder zur Verwaltung verwendet werden. Weitere Informationen finden Sie der Microsoft SQL Server-Onlinedokumentation unter <http://www.microsoft.com/downloads>.

Es folgt ein Beispiel für die Verwendung von Dateigruppen für die Verwaltung:

- **Teilweises Wiederherstellen.** Microsoft SQL Server unterstützt nicht die Wiederherstellung einer einzelnen Tabelle. Auch wenn Sie eine einzelne Tabelle in eine Dateigruppe platzieren, können Sie eine Dateigruppe nicht auf einen Zeitpunkt wiederherstellen, der früher liegt als der Zeitpunkt, der für den Rest der Daten gilt. Stattdessen müssen Sie alle Protokollsicherungen verwenden, um die Dateigruppe mit dem Rest der Daten zu synchronisieren. Microsoft SQL Server unterstützt nur die teilweise Wiederherstellung auf eine Datenbank mit einem anderen Namen. Eine Teilwiederherstellung ermöglicht die Wiederherstellung einer einzelnen Dateigruppe und unterstützt die Zeitpunktwiederherstellung. Sie müssen jedoch die Dateigruppe **PRIMÄR** wiederherstellen, da sie die SYSTEM-Tabellen enthält.

Zum Wiederherstellen einer einzelnen Tabelle zu einem bestimmten Zeitpunkt, wenn ein logischer Fehler aufgetreten ist, müssen Sie Dateigruppen in Ihrer Datenbank wie folgt einrichten:

- Stellen Sie sicher, dass die **.mdf**-Datei die einzige Datei in der Dateigruppe **PRIMÄR** ist.
- Platzieren Sie alle großen Tabellen in eine eigene Dateigruppe.
- Platzieren Sie alle kleinen Tabellen in eine separate Dateigruppe.

Weitere Daten zum Erstellen von Dateigruppen finden Sie unter "[Empfehlungen zum Microsoft SQL Server-Datenspeicher](#)" auf Seite 98.

Systemdatenbanken

Die folgenden Systemdatenbanken sind besonders wichtig für die reibungslose Ausführung von Microsoft SQL Server:

- **tempdb.** Zahlreiche Microsoft SQL Server-Aktivitäten, wie das Erstellen von lokalen und globalen temporären Tabellen, das Erstellen von Arbeitstabellen im Hintergrund für zwischenzeitliche Spoolabfrageergebnisse und das Sortieren, verwenden implizit oder explizit die tempdb-

Systemdatenbank.

Wenn Ihr System nicht ordnungsgemäß konfiguriert ist, kann die tempdb-Datenbank zu einem Leistungsengpass führen, sodass es sehr wichtig ist, die ursprüngliche Größe der tempdb-Datenbank korrekt zu bestimmen.

Weitere Informationen zum Festlegen von Datenbankgrößen finden Sie unter ["Einstellungen für die tempdb-Datenbank" auf Seite 32](#)

Zum Verschieben der tempdb-Dateien verwenden Sie den Befehl **ALTER DATABASE tempdb MODIFY FILE** und starten Sie Microsoft SQL Server neu.

- **master, msdb, model.** Diese Datenbanken sind, obwohl von entscheidender Bedeutung für den Betrieb von Microsoft SQL Server, kleiner als die tempdb-Datenbank, da sie nur Metadaten speichern.

Es wird dringend empfohlen, einen fehlertoleranten Datenträger, im Idealfall RAID 1, für diese Datenbanken zu verwenden.

Hinweis: Speichern Sie für die OMi-Zertifizierung speichern Sie die Systemdatenbanken auf fehlertoleranten Datenträgern. Die Verwendung von RAID 1-Datenträgern wird empfohlen.

Führen Sie den folgenden Befehl aus, um die Eigenschaften Ihrer Datenbank zu überprüfen:

```
EXEC sp_helpdb <Datenbankname>
```

Konfigurieren von Datenbanken

Nachdem Sie die erforderlichen Datenbanken erstellt haben, können Sie neue Dateien zu den Datenbanken hinzufügen, einige der vorhandenen Datenbankeigenschaften ändern und die Optionen der Datenbankkonfiguration entsprechend auswählen.

In diesem Abschnitt wird Folgendes behandelt:

- ["Datenbankdateikonfiguration" unten](#)
- ["Datenbankkonfigurationsoptionen" auf der nächsten Seite](#)

Wichtig: Wenn Sie OMi für die Erstellung von OMi-Datenbanken aktiviert haben, wird dringend empfohlen, die Konfiguration mithilfe der Anleitungen in diesem Abschnitt durchzuführen.

Datenbankdateikonfiguration

Sie können mit einer der folgenden Methoden bestimmte Datenbankdateieigenschaften ändern und Dateien hinzufügen oder entfernen:

- Verwenden Sie das Eigenschaften-Dialogfeld in Management Studio.
- Verwenden Sie den Befehl **ALTER DATABASE** (weitere Informationen finden Sie in der Microsoft SQL Server-Onlinedokumentation).

Hinzufügen von Dateien

Datendateien können zu einer bestehenden Dateigruppe in einer Datenbank oder zu einer neuen Dateigruppe hinzugefügt werden. Es sind keine besonderen Beschränkungen oder Anforderungen

erforderlich.

Löschen von Dateien

Zum Löschen einer Datei müssen Sie diese zunächst mithilfe des Befehls `DBCC SHRINKFILE` und der Option `EMPTYFILE` leeren, sodass die Dateidaten an die andere Datei in der Dateigruppe übertragen werden. Nach dem Leeren der Datei können Sie den Befehl `ALTER DATABASE <Datenbankname> DROP FILE` zum Löschen der Datei verwenden.

Ändern der Dateieigenschaften

Sie können die Eigenschaften für die Größe der Datenbanken sowie die Eigenschaft für den Dateinamen der tempdb-Datenbank ändern (die Änderung wird nach dem Neustart von Microsoft SQL Server wirksam). Die Eigenschaften `SIZE`, `MAXSIZE` und `FILEGROWTH` können mit dem Befehl `ALTER DATABASE tempdb MODIFY FILE` geändert werden. Beachten Sie, dass mit der Eigenschaft `SIZE` nur vergrößert werden kann.

Um eine Datei zu verkleinern, verwenden Sie den Befehl `DBCC SHRINKFILE`. Weitere Informationen und Empfehlungen zu Dateieigenschaften finden Sie unter ["Erstellen von Datenbanken" auf Seite 30](#).

Datenbankkonfigurationsoptionen

Jede Datenbank enthält einen Satz konfigurierbarer Optionen, die ihr Verhalten bestimmen. Sie können die Datenbankoptionen mit einer der folgenden Methoden anzeigen oder ändern:

- Über die Registerkarte **Optionen** im Dialogfeld **Eigenschaften** in Management Studio
- Mit der gespeicherten Prozedur `EXEC sp_dboptions`
- Mit dem Befehl `ALTER DATABASE <Datenbankname> SET`

Hinweis: Nicht alle der Datenbankkonfigurationsoptionen stehen in diesem Dialogfeld zur Verfügung.

Die folgende Tabelle enthält, in alphabetischer Reihenfolge, die Standardkonfigurationsoptionen sowie die erforderlichen Konfigurationseinstellungen für die OMi-Zertifizierung:

Konfigurationsoption	Beschreibung	Standardwert	OMi-Zertifizierung in Microsoft SQL Server
ANSI NULL als Standard (siehe Hinweis unten)	Gibt an, ob die Datenbankspalten standardmäßig als NULL oder NICHT NULL definiert sind.	Nicht festgelegt	Nicht festgelegt
ANSI PADDING	Steuert die Art, wie die Spalte Werte speichert, die kürzer sind als die definierte Größe für die Spalte, und Werte, die über nachgestellten Leerzeichen in den char-, varchar-, binary- und	AUS	EIN

Konfigurationsoption	Beschreibung	Standardwert	OMi-Zertifizierung in Microsoft SQL Server
	varbinary-Daten verfügen.		
Automatisch schließen	Gibt an, ob die Datenbank beendet wird, nachdem die Ressourcen freigegeben wurden und die Benutzer Ihre Aktivitäten beendet haben.	Nicht festgelegt	Nicht festgelegt Wird diese Option festgelegt, kann es eine Weile dauern, bis die Ressource von der Datenbank bei jeder Benutzerverbindung zugeordnet wird, nachdem die Datenbank geschlossen wurde.
Statistiken automatisch erstellen	Gibt an, ob fehlende Statistiken, die von einer Abfrage für die Optimierung angefordert werden, während der Optimierung automatisch erstellt werden.	Festgelegt	Festgelegt
Automatisch verkleinern	Gibt an, ob die Datenbank automatisch einmal pro Stunde verkleinert wird und so 25 % freien Speicherplatz erzeugt.	Nicht festgelegt	Nicht festgelegt Hinweis: Wenn diese Option festgelegt wird, kann die kontinuierliche Vergrößerung / Verkleinerung zur Dateisystem-fragmentierung führen.
Statistiken automatisch aktualisieren	Gibt an, ob veraltete Statistiken, die von einer Abfrage für die Optimierung angefordert werden, während der Optimierung automatisch erstellt werden.	Festgelegt	Festgelegt
Kompatibilitätsgrad	Die Version von Microsoft SQL Server, die die Datenbank (für die Applikation) aufweist.	Dieselbe Version wie das installierte Release	Dieselbe Version wie das installierte Release
Schreibgeschützt	Die Datenbank ist schreibgeschützt.	Nicht festgelegt (READ_WRITE)	READ_WRITE

Konfigurationsoption	Beschreibung	Standardwert	OMi-Zertifizierung in Microsoft SQL Server
Wiederherstellen	Das Datenbankwiederherstellungsmodell bestimmt die Wiederherstellungsmöglichkeiten, indem der Umfang der Massenvorgangsprotokollierung (z. B. Select into, Bulk, Insert, Create Index , LOB-Manipulation) gesteuert wird. Je entwickelter das Wiederherstellungsmodell, desto entwickelter die Wiederherstellungsfunktionen. Der Protokollierungsumfang nimmt jedoch auch zu, was sich auf die Leistung auswirken kann.	Vollständig	Vollständig (es sein denn, Sie sind sicher, dass die geringeren Wiederherstellungsfunktionen für Ihr System ausreichend sind)
Rekursive Trigger	Gibt an, ob rekursive Trigger unterstützt werden.	Nicht festgelegt	Nicht festgelegt
Zugriff beschränken	Nur einzelne Benutzer oder Mitglieder der db_owner-, dbcreator- oder sysadmin-Gruppen können auf die Datenbank zugreifen.	Nicht festgelegt (MULTI_USER)	MULTI_USER
Seitenüberprüfung	Gibt an, ob unvollständige Seiten erkannt werden können.	Festgelegt	Festgelegt
Bezeichner in Anführungszeichen aktiviert	Gibt an, ob Microsoft SQL Server ANSI-Regeln in Hinblick auf Anführungszeichen festlegt. Wählen Sie diese Option aus, um anzugeben, dass doppelte Anführungszeichen für Bezeichner verwendet werden, wie z. B. Spalten- und Tabellennamen. Beachten Sie, dass Zeichenfolgen in einfache Anführungszeichen eingeschlossen sein müssen.	Nicht festgelegt	Nicht festgelegt

Hinweis: Nicht alle ANSI-Optionen können mithilfe von Management Studio festgelegt werden. Zu

den Optionen für die ANSI-Datenbankkonfiguration gehören: ANSI_NULLS, ANSI_NULL_DEFAULT, ANSI_PADDING, ANSI_WARNINGS, ARITHABORT, CONCAT_NULL_YIELDS_NULL, NUMERIC_ROUNDABORT und QUOTED_IDENTIFIER.

Die festgelegten Optionen sind möglicherweise nicht wirksam, da diese Optionen auch auf einer höheren Ebene festgelegt werden können.

Wenn beispielsweise die Sitzungsoption **QUOTED_IDENTIFIER** eingeschaltet wurde, ist die entsprechenden Option der Datenbankkonfiguration nicht relevant. Einige Werkzeuge oder Datenbankschnittstellen schalten bestimmte Sitzungsoptionen ein oder aus, sodass die Optionen der Datenbankkonfiguration nie wirksam werden.

Die folgende Tabelle enthält eine Übersicht über die Merkmale der einzelnen Wiederherstellungsmodelle:

Modell/Unterstützung	Ermöglicht Protokoll-sicherung	Ermöglicht Zeitpunkt-/Protokoll-markierungs-wiederher-stellung	Ermöglicht Sicherungs-protokollierung bei Datenabsturz (speichert die Änderungen bis zum Absturz)	Umfang der Massenvorgangs-protokollierung (kann sich auf die Leistung von Massen-vorgängen auswirken)
Einfach	Nein	Nein	Nein	Minimal
Massenprotokollierung	Ja	Nein	Nein	Minimal
Vollständig	Ja	Ja	Ja	Vollständig

Führen Sie den folgenden Befehl aus, um die Eigenschaften Ihrer Datenbank zu überprüfen:

```
EXEC sp_helpdb <Datenbankname>
```

Weitere Informationen zu SQL-Datenbanken finden Sie in der Microsoft SQL Server- Onlinedokumentation.

Kapitel 6: Warten von Microsoft SQL Server-Datenbanken

Dieses Kapitel beschreibt die verschiedenen Wartungsaufgaben, die für OMi-Datenbanken unter Microsoft SQL Server empfohlen werden, wie z. B. Sichern von Datenbanken, Überprüfen der Datenbankintegrität, Verarbeiten von Fragmentierung und Überwachen von Datenbanken.

Dieses Kapitel umfasst die folgenden Themen:

- ["Sichern von Datenbanken" unten](#)
- ["Datenbankintegrität und -fragmentierung" auf Seite 42](#)

Sichern von Datenbanken

Microsoft SQL Server unterstützt drei Arten der Datenbanksicherung: Vollständige, differenziell und protokollierte Sicherung. Darüberhinaus wird die Sicherung von Dateien/Dateigruppen unterstützt, die in einem separaten Abschnitt weiter unten beschrieben wird. Für die Entwicklung von Sicherungsrichtlinien mit den erforderlichen Wiederherstellungsoptionen müssen alle Sicherungstypen und die Datenbankkonfiguration des Wiederherstellungsmodells aus dem vorherigen Abschnitt umfassend verstanden werden.

Sie können die Sicherungsvorgänge mithilfe von Microsoft SQL Agent-Jobs automatisieren. Der Microsoft SQL-Agent (vertreten durch den SQLServerAgent-Service) wird automatisch installiert, wenn Sie Microsoft SQL Server installieren. Stellen Sie sicher, dass der Microsoft SQL-Agent im Dienste-Applet des Betriebssystems so konfiguriert ist, dass er automatisch beim Start des Servers gestartet wird.

Die folgenden Punkte gelten für alle Sicherungstypen:

- Die Sicherung beinhaltet alle Änderungen, die bis zum Abschluss der Sicherung durchgeführt wurden.
- Die Sicherung kann online durchgeführt werden, aber es wird empfohlen, die Datenbank während Zeiten geringer Aktivität zu sichern, da sich die Sicherungsverfahren negativ auf die Leistung Ihres Systems auswirken können.
- Die folgenden Operationen sollten nicht während eines Sicherungsverfahrens durchgeführt werden:
 - Hinzufügen oder Entfernen von Dateien
 - Verkleinern der Datenbank
- Das Sicherungsziel kann ein Datenträger sein (lokal oder auf einer Netzwerkfreigabe, da das Dienstkonto von Microsoft SQL Server eine Berechtigung für den Zugriff benötigt) oder ein Band sein (nur lokal).

Dieser Abschnitt umfasst die folgenden Themen:

- ["Vollständige Sicherung" auf der nächsten Seite](#)
- ["Differenzielle Sicherung" auf der nächsten Seite](#)
- ["Protokollsicherung" auf der nächsten Seite](#)

- ["Datei und Dateigruppensicherung" auf der nächsten Seite](#)
- ["Wartungsplan" auf der nächsten Seite](#)
- ["Probleme mit dem Transaktionsprotokoll" auf der nächsten Seite](#)

Vollständige Sicherung

Wenn Sie eine vollständige Sicherung der Datenbank durchführen, sind alle Informationen über die gesicherte Datenbank wie Daten, Metadaten und Dateinformationen in der Sicherung enthalten. Eine vollständige Sicherung ist die Basis für differenzielle und Protokollsicherungen. Bei kleinen Datenbanken wird empfohlen eine vollständige Sicherung jeden Tag durchzuführen (z. B. bei Systemdatenbanken, die hauptsächlich Metadaten speichern). Für große Datenbanken wird allgemein empfohlen, längere Intervalle zwischen den vollständigen Sicherungen einzuhalten (z. B. einmal pro Woche).

Die Speicheranforderungen für eine vollständige Sicherung sind identisch mit den Speicheranforderungen für den ausgelasteten Datenteil der Dateien. Beispiel: Wenn die Gesamtgröße der Datendateien 20 GB beträgt, aber nur 15 GB verwendet werden (es gibt 5 GB freien Speicherplatz), sollte die Größe für die vollständige Sicherung der Datenbank ca. 15 GB betragen.

Differenzielle Sicherung

Verwenden Sie eine differenzielle Sicherung für das Sichern von Blöcken (Blöcke von acht zusammenhängenden Seiten mit 8 KB), die seit der letzten vollständigen Sicherung geändert wurden. Bei der Wiederherstellung einer Datenbank müssen Sie nur die letzte differenzielle Sicherung wiederherstellen, die nach der vollständigen Sicherung durchgeführt wurde.

Nach dem Durchführen aller Operationen, die sich auf große Teile von Daten beziehen, wie zum Beispiel Indexneuerstellungen oder Defragmentierungen, wird empfohlen, eine vollständige Sicherung durchzuführen. Andernfalls können die differenziellen Sicherungen sehr groß werden. Weitere Informationen zur Neuerstellung und Defragmentierung von Indizes finden Sie unter ["Datenbankintegrität und -fragmentierung" auf Seite 42](#).

Die differenzielle Sicherung wird normalerweise in Intervallen zwischen den vollständigen Sicherungen geplant. Wenn Sie beispielsweise eine vollständige Sicherung einmal pro Woche durchführen, möchten Sie eine differenzielle Sicherung möglicherweise täglich oder sogar mehrmals täglich durchführen.

Die Speicheranforderung für eine differenzielle Sicherung wird durch die Gesamtgröße der Blöcke (64 KB-Blöcke) bestimmt, die seit der letzten vollständigen Sicherung geändert wurden.

Protokollsicherung

Bei einer Protokollsicherung werden im Gegensatz zu vollständigen und differenziellen Sicherungen, bei denen eher ein Bild der Erweiterungen gesichert wird, Transaktionen aus dem Transaktionsprotokoll gesichert und bei er Wiederherstellung erneut wiedergegeben. Um eine Protokollsicherung durchzuführen, muss die Datenbank für das Wiederherstellungsmodell der vollständigen Protokollierung oder Massenprotokollierung eingerichtet werden. Wenn Sie eine Zeitpunkt-/Protokollmarkierungswiederherstellung durchführen möchten oder Änderungen sichern möchten, die bei einem Absturz im Protokoll aufgezeichnet wurden, müssen Sie für die Datenbank das Modell für eine

vollständige Wiederherstellung einrichten. Andernfalls gehen alle Änderungen, die Sie seit der letzten Sicherung durchgeführt haben, verloren.

Bei der Protokollsicherung handelt es sich um eine inkrementelle Sicherung, bei der nur die Transaktionen seit der vorherigen Sicherung gesichert werden. Beim Wiederherstellen einer Datenbank müssen Sie alle Protokollsicherungen nach der letzten differenziellen (oder vollständigen) Sicherungskopie wiederherstellen.

Eine Protokollsicherung markiert auch den Teil des Protokolls, der als verfügbar für die Wiederverwendung gesichert wurde. In einer Datenbank, die für das Wiederherstellungsmodell der vollständigen Protokollierung oder Massenprotokollierung eingerichtet wurde, können Protokollteile, die nicht gesichert wurden, nicht wiederverwendet werden. Wenn die Protokolldatei voll ist und Microsoft SQL Server nicht an den Anfang wechseln kann, um den Speicherplatz erneut zu verwenden, muss es erweitert werden. Die Häufigkeit für die Protokollsicherungen ist daher wichtig für die Bestimmung der erforderlichen Größe des Transaktionsprotokolls. Häufige Protokollsicherungen ermöglichen kleinere Transaktionsprotokolle. Es wird empfohlen, das Protokoll so häufig wie möglich zu sichern, z. B. alle 30 Minuten.

Datei und Dateigruppensicherung

Anstelle der Sicherung der gesamten Datenbank können Sie eine Datei oder Dateigruppe sichern. Wenn Sie jedoch eine einzelne Datei oder Dateigruppe wiederherstellen, müssen Sie alle Protokollsicherungen bis einschließlich des Fehlerzeitpunkts verwenden, um die Datei/Dateigruppe mit dem Rest der Datenbank zu synchronisieren (derselbe Zeitpunkt). Diese Art der Sicherung ist im Allgemeinen bei sehr großen Datenbanken sinnvoll, für die Sie nicht häufig eine vollständige Sicherung durchführen können.

Wartungsplan

In Microsoft SQL Server Management Studio befindet sich unterhalb der Verwaltungsstruktur das grafische Werkzeug "Datenbankwartungspläne". Dieses Werkzeug ermöglicht Ihnen das Definieren und Automatisieren von Wartungsaufgaben (vollständige Sicherung und Protokollsicherungen, Integritätstests, Indexwiederherstellungen und Statistiksammlung).

Probleme mit dem Transaktionsprotokoll

Die Wartung ist in Hinblick auf das Protokoll ein sensibler Vorgang. Wenn das Protokoll voll ist, wird versucht, einen Zyklus zu durchlaufen und inaktiv gesicherten Protokollspeicher zu verwenden. Ist dieser Platz nicht vorhanden, versucht das Protokoll, die Datei zu erweitern. Wenn es keinen Platz für die Dateierweiterung gibt, lehnt Microsoft SQL Server die Anforderung zur Datenänderung ab.

Um die Protokollexplosion zu vermeiden, stellen Sie sicher, dass das Protokoll groß genug ist und dass es häufig gesichert wird (im Idealfall nach Zeitplan). Darüber hinaus beginnt der aktive Teil der Protokolldatei mit der ältesten offenen Transaktion und wird bis zum aktuellen Zeiger im Protokoll weitergeführt. Der aktive Teil kann nicht wiederverwendet oder abgeschnitten werden. Ist eine Transaktion für eine lange Zeit geöffnet, führt dies zwangsläufig an einem bestimmten Zeitpunkt zu einer Protokollexplosion, auch wenn das Protokoll gesichert wurde.

Um festzustellen, ob dieses Problem vorliegt, führen Sie `DBCC OPENTRAN` aus, um die Transaktion zu erhalten, die für den längsten Zeitraum geöffnet war. Zum Beenden des Vorgangs, der die Transaktion ausführt, und zum Zurücksetzen der Transaktionsaktivität verwenden Sie folgenden Befehl:

KILL <Prozess-ID>

Hinweis: In Microsoft SQL Server sollte der Befehl DBCCSHRINKFILE immer erfolgreich sein.

Datenbankintegrität und -fragmentierung

Es ist wichtig, regelmäßig die physische Integrität der Datenbankobjekte zu überprüfen und die Indexfragmentierung zu beachten, da es sich hierbei um die Hauptursache für eine Leistungsminderung handelt.

Dieser Abschnitt umfasst die folgenden Themen:

- ["Datenbankintegrität" unten](#)
- ["Grundlegende Informationen zur Dateisystemfragmentierung" unten](#)
- ["Grundlegende Informationen zur internen Fragmentierung" auf der nächsten Seite](#)
- ["Grundlegende Informationen zur externen Fragmentierung" auf der nächsten Seite](#)
- ["Ermitteln und Verarbeiten der Indexfragmentierung" auf Seite 44](#)
- ["Bereitgestellte Dienstprogramme zur Überwachung und Wiederherstellung von Indizes" auf Seite 45](#)
- ["Verteilungsstatistik" auf Seite 46](#)
- ["Sammeln von Statistiken für RTSM" auf Seite 47](#)
- ["Dienstprogramm für das Aktualisieren von Statistiken" auf Seite 47](#)

Datenbankintegrität

Es wird empfohlen, dass Sie DBCC CHECKDB in regelmäßigen Abständen ausführen, um die Zuordnung und die strukturelle Integrität der Objekte in der Datenbank zu überprüfen. Sie können den DBCC CHECKDB mithilfe der Microsoft SQL Agent-Jobs automatisieren und planen. Verwenden Sie die folgende Befehlsyntax:

```
DBCC CHECKDB ('Datenbankname')
```

Hinweis: Sie können die Option WITH NO_INFOMSGS verwenden, um die Verarbeitungszeit und die tempdb-Verwendung zu reduzieren. Sie können auch einen schnellen, nur physischen Test (Seitenstruktur und Datensatzheader) mithilfe der Option PHYSICAL_ONLY durchführen.

Da die Microsoft SQL Server-Datenbank nur Schemasperrungen enthält (die Schemaänderungen verhindern) und keine Datenänderungen, kann der Befehl DBCC CHECKDB online ausgeführt werden. Es wird jedoch empfohlen, den Befehl DBCC CHECKDB während Zeiten geringer Aktivität auszuführen, da er sich negativ auf die Systemleistung auswirken kann (DBCC CHECKDB ist CPU- und Festplatten-intensiv und verwendet die tempdb-Datenbank für die Sortierung).

Grundlegende Informationen zur Dateisystemfragmentierung

Die Dateisystemfragmentierung betrifft alle Datenträgerdateien und nicht nur Datenbankdateien. Damit ist die Streuung von Teilen der gleichen Datenträgerdatei über verschiedene Bereiche des Datenträgers gemeint, wenn neue Teile der Datei hinzugefügt und vorhandene Teile gelöscht werden.

Durch die Dateisystemfragmentierung wird der Zugriff auf den Datenträger verlangsamt und die Leistung der Datenträgeroperationen insgesamt beeinträchtigt, wenn auch nicht schwerwiegend.

Zum Defragmentieren eines Dateisystems schreiben Sie Teile einer Datei auf benachbarte Bereiche auf einem Datenträger. Dies erhöht die Geschwindigkeit für den Datenzugriff und den Abruf. Um die Fragmentierung der Datenbankdateien zu vermeiden, erstellen Sie Dateien mit einer möglichst großen Anfangsgröße (sodass Änderungen in der Zukunft berücksichtigt werden können) und erweitern Sie diese mit großen Inkrementen für das Wachstum, wenn Platzprobleme auftreten.

Wenn Sie die zukünftige Größe einer Datenbankdatei nicht absehen können, verwenden Sie einen hohen Wert für das Inkrement des Wachstums, um kleine fragmentierte Teile zu vermeiden. Verwenden Sie keinen zu hohen Wert, da dies zu Zeitüberschreitungen für Clientanforderungen führt, wenn die Datei automatisch wächst. (Weitere Informationen finden Sie unter ["Datenbankberechtigungen" auf Seite 30.](#)) Darüber hinaus vermeiden Sie die Option für die automatische Verkleinerung, da sich die Wahrscheinlichkeit einer Fragmentierung erhöht, wenn die Datenbankdateien fortlaufend verkleinert und vergrößert werden.

Hinweis: Es wird empfohlen, regelmäßig für die Datenbank ein Defragmentierungsprogramm auszuführen.

Grundlegende Informationen zur internen Fragmentierung

Die interne Fragmentierung bezieht sich auf den Prozentsatz der Daten, die in den Seiten enthalten sind. In Umgebungen wie dem OMi-System, das sich durch Transaktionen charakterisieren lässt, die häufig Daten einfügen, wird die interne Fragmentierung manchmal gestartet, bevor neue Daten in Indizes aufgenommen werden. Dies kann positive Auswirkungen haben. Wenn Sie einen bestimmten Prozentsatz der Indexseiten frei lassen, können Sie Seitenteilungen für einen bestimmten Zeitraum vermeiden. Dies ist besonders für Cluster-Indizes relevant, da sie die tatsächlichen Datenseiten enthalten. Sie können dabei in regelmäßigen Abständen eine interne Fragmentierung erzielen, indem Sie Ihre Indizes über den Befehl `CREATE INDEX` neu erstellen, wobei die Optionen `DROP_EXISTING` und `FILLFACTOR` verwendet werden. Oder Sie verwenden den Befehl `ALTER INDEX REBUILD` (online oder offline) und die Option `FILLFACTOR`. Mit der Option `FILLFACTOR` wird die Seitenauslastung bei Indexseiten auf Blattebene angegeben.

Grundlegende Informationen zur externen Fragmentierung

Wenn in Ihren Indizes Seitenteilungen auftreten, werden neu zugeordnete Seiten aus der Datenbankdatei erfasst. Im Idealfall sollte bei einer Seitenteilung eine Seitenzuordnung zu der Seite führen, die an die geteilte Seite angrenzt. In der Praxis ist jedoch der Platz neben der geteilten Seite normalerweise bereits belegt. Je mehr Seitenteilungen auftreten, desto weniger spiegelt die verknüpfte Liste des Index das physische Layout der Seiten auf dem Datenträger wider und desto größer ist der Anteil der externen Fragmentierung.

Externe Fragmentierung wirkt sich negativ auf die Leistung der sortierten Indexscans aus, da der Festplattenarm vor- und zurückbewegt werden muss, damit die Seiten von der Festplatte abgerufen werden können. Die verknüpfte Liste sollte das physische Layout der Seiten auf einem Datenträger wiedergeben, sodass bei einem sortierten Indexscan der Festplattenarm in eine Richtung bewegt wird, wenn die Seiten vom Datenträger abgerufen werden.

Die externe Fragmentierung kann proaktiv durchgeführt werden, indem die interne Fragmentierung gestartet wird und ein bestimmter Prozentsatz der Indexseiten auf Blattebene freigelassen wird, um so Seitenteilungen für eine bestimmte Zeitdauer zu vermeiden. Wie bereits erwähnt, kann die interne Fragmentierung in regelmäßigen Abständen erreicht werden, indem die Option `FILLFACTOR` verwendet wird. Sie können auch eine externe Fragmentierung verarbeiten, indem Sie den Status der externen Fragmentierung Ihrer Indizes überprüfen und die Indizes wiederherstellen.

Ermitteln und Verarbeiten der Indexfragmentierung

Verwenden von "sys.dm_db_index_physical_stats" zum Erkennen der Fragmentierung

Die dynamische Verwaltungsfunktion **sys.dm_db_index_physical_stats** dient zur Bestimmung des Fragmentierungsgrads eines Index. Sie können die Fragmentierung in einem bestimmten Index, in allen Indizes einer Tabelle oder in einer indizierten Ansicht, in allen Indizes in einer bestimmten Datenbank oder in allen Indizes in allen Datenbanken ermitteln. Für partitionierte Indizes stellt **sys.dm_db_index_physical_stats** außerdem Fragmentierungsinformationen für jede Partition bereit.

Die Fragmentierung einer Tabelle erfolgt über den Prozess der Datenänderungen (`INSERT`, `UPDATE` und `DELETE`-Anweisungen), die in der Tabelle und für die in der Tabelle definierten Indizes durchgeführt werden. Da diese Änderungen normalerweise nicht gleichmäßig auf die Zeilen der Tabelle und Indizes verteilt werden, kann die Seitenauslastung im Zeitverlauf variieren. Für Abfragen, die alle oder einen Teil der Indizes einer Tabelle scannen, kann diese Art der Fragmentierung zu langsameren Abfrageergebnissen führen.

Die Tabellensyntax von `sys.dm_db_index_physical_stats` lautet:

```
sys.dm_db_index_physical_stats (  
{ database_id | NULL }  
, { object_id | NULL }  
, { index_id | NULL | 0 }  
, { partition_number | NULL }  
, { mode | NULL | DEFAULT }  
)
```

Führen Sie diesen Befehl aus der Tabelle als reguläre `select`-Anweisung aus. Beispiel:

```
select * from sys.dm_db_index_physical_stats (DB_ID('<BAC-Datenbank>'),  
object_id('<Tabellenname>'), NULL, NULL, 'SAMPLED')
```

Die Spalte **Avg_fragmentation_in_percent**, die von **sys.dm_db_index_physical_stats** zurückgegeben wird, ist die logische und Blockfragmentierung des Index.

Weitere Informationen zu **sys.dm_db_index_physical_stats** finden Sie in der Microsoft SQL Server-Dokumentation.

Fragmentierungsbehandlung

Die Tabellenspalte **Avg_fragmentation_in_percent**, die von **sys.dm_db_index_physical_stats** zurückgegeben wird, gibt den Fragmentierungsgrad pro Index an. Basierend auf diesem Wert können

Sie bestimmen, ob die Fragmentierung verarbeitet wird und welche Methode dabei verwendet werden soll.

Verwenden Sie die folgenden groben Anhaltspunkte zur Bestimmung der besten Methode zur Korrektur der Fragmentierung:

- **Zwischen 5 % und 30 %.** Verwenden Sie den Befehl `ALTER INDEX REORGANIZE` zum erneuten Organisieren des Index. Die Neuorganisation des Index wird immer online ausgeführt.
- **Mehr als 30 %.** Verwenden Sie den Befehl `ALTER INDEX REBUILD` zum erneuten Erstellen des Index. Die Indexneuerstellung kann online oder offline ausgeführt werden. Erstellen Sie die Indizes online, um eine Verfügbarkeit zu erzielen, die der Option zur Neuorganisation ähnelt.

Ein sehr niedriges Fragmentierungsniveau (kleiner als 5 %) sollte nicht mit diesen Befehlen behandelt werden, da nur ein geringer positiver Effekt durch das Entfernen einer so geringen Fragmentierung erreicht werden kann.

Weitere Informationen zum Defragmentieren von Indizes finden Sie im Abschnitt zum Neuorganisieren und Neuerstellen von Indizes des Themas zum Entwerfen und Implementieren von strukturiertem Speicher in der SQL Server-Onlinedokumentation: <http://technet.microsoft.com/en-us/library/ms189858.aspx>.

Hinweis: Es wird nachdrücklich empfohlen, eine automatische Aufgabe zur Neuerstellung eines Index für jede der OMi-Datenbanken zu erstellen. Führen Sie diese Aufgabe in den RTSM- und Ereignisdatenbanken mindestens einmal pro Tag aus.

Bereitgestellte Dienstprogramme zur Überwachung und Wiederherstellung von Indizes

OMi bietet zwei Dienstprogramme, die für das Ermitteln und die Wiederherstellung von fragmentierten Indizes verwendet werden können. Das Dienstprogramm **rebuild_fragmented_indexes.bat** verwendet die logische Scanfragmentierung und Scandichte als Kriterien für die Erkennung und – sofern angewiesen – Wiederherstellung von fragmentierten Indizes. Der Vorgang der Auflistung von fragmentierten Tabellen hat eine sehr geringe Auswirkung auf die Systemleistung und kann gleichzeitig ausgeführt werden. Die Wiederherstellung von Indizes wirkt sich normalerweise auf die Leistung aus, da Tabellen dann teilweise gesperrt werden und CPU- und E/A-Vorgänge stark ausgelastet werden. Es wird empfohlen, Indizes in einem Wartungsfenster wiederherzustellen. Die Dienstprogramme sollten von einem Datenbankadministrator ausgeführt werden.

Die Dienstprogramme für Microsoft SQL Server befinden sich im Verzeichnis **<Stammverzeichnis des Gateway-Servers>\AppServer\webapps\site.war\DataBases\SQL_Svr_DB_Utills** auf dem OMi-Gateway-Server.

Dienstprogramm für das Wiederherstellen aller Indizes in der Datenbank

Das Dienstprogramm **rebuild_indexes.bat** wird für alle Tabellen in der Datenbank ausgeführt und stellt zugehörige Indizes wieder her.

So führen Sie das Dienstprogramm "rebuild_indexes.bat" aus:

Führen Sie **rebuild_indexes.bat** mit den folgenden Parametern aus:

- SQL Server-Name
- Datenbankname
- SA-Kennwort

Beispiel:

```
rebuild_indexes.bat SQL_SRVR_3 OMi_DB_3 ad%min52.
```

Die Ausgabe des Verfahrens befindet sich in der Datei **rebuild_indexes.log** in demselben Verzeichnis.

Dienstprogramm für das Wiederherstellen von Indizes basierend auf dem Fragmentierungslevel für jeden Index

Das Dienstprogramm **rebuild_fragmented_indexes.bat** verfügt über zwei Betriebsmodi:

- Auflisten von fragmentierten Tabellen. In diesem Modus wird eine Liste der fragmentierten Tabellen (das heißt, Tabellen mit mehr als 30 % Fragmentierung) zusammen mit den Befehlen zurückgegeben, die erforderlich sind, um die Tabelle zu einem späteren Zeitpunkt wiederherzustellen.
- Wiederherstellen von fragmentierten Tabellen. In diesem Modus werden alle fragmentierten Tabellen (Tabelle mit mehr als 30 % Fragmentierung) neu erstellt.

So führen Sie das Dienstprogramm "rebuild_fragmented_indexes.bat" aus:

Führen Sie **rebuild_fragmented_indexes.bat** mit den folgenden Parametern aus:

- SQL Server-Name
- Datenbankname
- SA-Kennwort
- Betriebsmodus - 0 zum Bereitstellen eines wiederhergestellten Skripts für die spätere Verwendung; 1 für das automatische Wiederherstellen von Statistiken.

Beispiel:

```
rebuild_fragmented_indexes.bat SQL_SRVR_3 OMi_DB_3 ad%min52 1
```

Die Ausgabe des Verfahrens (eine Liste von fragmentierten Tabellen und Wiederherstellungsbefehle) befindet sich in der Datei **rebuild_indexes.log** im selben Verzeichnis.

Verteilungsstatistik

Microsoft SQL Server ermöglicht das Erstellen von statistischen Informationen hinsichtlich der Verteilung der Werte in einer Spalte. Diese statistischen Informationen können vom Abfrageprozessor zum Ermitteln der optimalen Strategie für die Auswertung einer Abfrage verwendet werden. Wenn ein Index erstellt wird, speichert SQL Server automatisch statistische Informationen hinsichtlich der Verteilung der Werte in den indizierten Spalten. Der Abfrageoptimierer in SQL Server verwendet die Statistik zur Schätzung der Kosten für das Verwenden des Index für eine Abfrage. Wenn die Daten in einer Spalte geändert werden, sind die Index- und Spaltenstatistiken nicht mehr aktuell und können zu weniger optimalen Entscheidungen darüber führen, wie eine Abfrage durchgeführt werden soll.

Es wird empfohlen, Indexstatistiken täglich zu aktualisieren, um den Abfrageoptimierer mit aktuellen Informationen über die Verteilung der Datenwerte in den Tabellen zu versorgen. Dadurch kann der Abfrageoptimierer bessere Entscheidungen über die optimale Methode für den Datenzugriff treffen, da mehr Informationen über die in der Datenbank gespeicherten Daten vorliegen.

Unabhängig davon, ob die Option für die automatische Aktualisierung von Statistiken aktiviert oder deaktiviert ist, wird dringend empfohlen, eine automatische Aufgabe für das tägliche Aktualisieren der OMi-Datenbanken zu erstellen, da sich die Daten häufig ändern. Der Job sollte die `sp_updatestats`-API für die spezifische Datenbank durchführen.

Sammeln von Statistiken für RTSM

Im Gegensatz zu einigen Datenbanken, bei denen Abfragen vordefiniert sind und entsprechend den erwarteten Datenbankgröße geändert werden können, erstellt die RTSM-Datenbank Abfragen dynamisch nach Patternansichten, die für das Datenmodell definiert wurden. Hierzu sind jederzeit genaue Statistiken erforderlich. Zusätzlich zum Ausführen einer täglichen Aktualisierung von Statistiken, wird empfohlen manuell Statistiken zu aktualisieren, wenn größere Änderungen an den RTSM-Schemaobjekten, normalerweise als Ergebnis von Masseneinfügetransaktionen, aufgetreten sind.

Ein Beispiel für ein Szenario, das eine manuelle Aktualisierung der RTSM-Statistiken garantiert, sind die automatisierten DFM-Jobs (Data Flow Management). DFM ist der verantwortliche Prozess für das automatische Erkennen von Konfigurationselementen und das Einfügen dieser Elemente in RTSM.

Dienstprogramm für das Aktualisieren von Statistiken

Das Dienstprogramm **update_statistics.bat** verfügt über zwei Betriebsmodi:

- Auflisten von Tabellen mit veralteten Statistiken. In diesem Modus wird eine Liste der Tabellen zusammen mit den Befehlen zurückgegeben, die für das Aktualisieren der Statistiken in den Tabellen zu einem späteren Zeitpunkt benötigt werden.
- Aktualisieren von Statistiken zu den Tabellen. In diesem Modus werden alle Tabellen mit veralteten Statistiken aktualisiert.

So führen Sie das Dienstprogramm "update_statistics.bat" aus:

Führen Sie unter **<Stammverzeichnis des Gateway-Servers>\AppServer\webapps\site.war\DataBases\SQL_Svr_DB_Utills** das Programm **update_statistics.bat** mit den folgenden Parametern aus:

- SQL Server-Name.
- Name der Datenbank.
- SA-Kennwort.
- Betriebsmodus - 0 zum Bereitstellen eines Skripts für die spätere Verwendung; 1 für das automatische Aktualisieren von Statistiken.

Beispiel:

```
update_statistics.bat SQL_SRVR_3 RTSM_DB_3 ad%min52 1
```

Die Ausgabe des Verfahrens (eine Liste von Tabellen und der Aktualisierungsbefehle) befindet sich in der Datei **update_statistics.log** im selben Verzeichnis.

Kapitel 7: Verwenden der Windows-Authentifizierung für den Zugriff auf Microsoft SQL Server-Datenbanken

Sofern nicht anders konfiguriert, verwendet OMi die Microsoft SQL Server-Authentifizierung für den Zugriff auf Microsoft SQL Server-Datenbanken. Die Windows-Authentifizierung kann jedoch auch verwendet werden.

In diesem Kapitel wird beschrieben, wie OMi für die Verwendung der Windows-Authentifizierung aktiviert wird, um auf die Microsoft SQL Server-Datenbanken zugreifen zu können.

Dieses Kapitel umfasst die folgenden Themen:

["Aktivieren von OMi für die Arbeit mit der Windows-Authentifizierung" unten](#)

Aktivieren von OMi für die Arbeit mit der Windows-Authentifizierung

Sie können OMi für die Verwendung der Windows-Authentifizierung anstelle der Microsoft SQL Server-Authentifizierung für den Zugriff auf die OMi-Datenbanken aktivieren.

Sie müssen folgendermaßen vorgehen, um OMi für die Verwendung der Windows-Authentifizierung für den Zugriff auf eine Microsoft SQL-Datenbank zu aktivieren:

- Konfigurieren Sie Microsoft SQL Server für die Verwendung der Windows-Authentifizierung.
- Starten Sie den OMi-Serverdienst auf allen OMi-Servern mit einem Windows-Benutzer, der über die erforderlichen Berechtigungen für den Zugriff auf die Microsoft SQL-Datenbank verfügt.
- Verwenden Sie den Konfigurations-Assistenten zum Erstellen oder Verbinden zu einer Microsoft SQL-Datenbank und geben Sie die Verwendung der Windows-Authentifizierung an.

Dieser Abschnitt umfasst die folgenden Themen:

- ["Konfigurieren von Microsoft SQL Server zur Verwendung der Windows-Authentifizierung" unten](#)
- ["Starten des OMi-Dienstes mit einem Windows-Benutzer" auf der nächsten Seite](#)
- ["Erstellen einer Verbindung zu einer Microsoft SQL-Datenbank mit Windows Authentifizierung oder Herstellen der Verbindung" auf der nächsten Seite](#)

Konfigurieren von Microsoft SQL Server zur Verwendung der Windows-Authentifizierung

Wählen Sie in SQL Server Enterprise Manager **Sicherheit > Anmeldungen** aus und klicken Sie mit der rechten Maustaste auf **Neue Anmeldung**. Geben Sie das gewünschte Domänenkonto, einschließlich des Domänennamens, im folgenden Muster an: `DOMÄNE\BENUTZER` (z. B. `MEINE_DOMÄNE\MEIN_KONTO`).

Auf der Registerkarte **Serverrollen** wählen Sie **Systemadministratoren** und klicken Sie auf **OK**.

Starten des OMi-Dienstes mit einem Windows-Benutzer

Standardmäßig wird der OMi-Dienst als Systemdienst ausgeführt. Wenn Sie Microsoft SQL Server für die Verwendung der Windows-Authentifizierung konfiguriert haben, müssen Sie den Benutzer, der den OMi-Dienst ausführt, in denselben Windows-Benutzer ändern, den Sie für Microsoft SQL Server definiert haben, damit der Benutzer des Dienstes Zugriff auf die Datenbank hat.

Hinweis: Weitere Informationen über zusätzliche Berechtigungen, über die der Benutzer des Dienstes verfügen muss, finden Sie im OMi-Installations- und Upgrade-Handbuch.

So ändern Sie den Benutzer des OMi-Dienstes:

1. Deaktivieren Sie OMi (**Start > Programme > OMi > Verwaltung > OMi deaktivieren**).
2. Doppelklicken Sie im Microsoft-Fenster **Dienste** auf **HP Operations Manager i**. Das Dialogfeld **Servereigenschaften** (Lokaler Computer) von OMi wird geöffnet.
3. Klicken Sie auf die Registerkarte **Anmelden**.
4. Wählen Sie Option für dieses Konto aus und wählen Sie den Benutzer aus, den Sie zuvor für Microsoft SQL Server definiert haben.
5. Geben Sie das Windows-Kennwort des ausgewählten Benutzers ein und bestätigen Sie dieses Kennwort.
6. Klicken Sie auf **Übernehmen**, um die Einstellungen zu speichern, und auf **OK**, um das Dialogfeld zu schließen.
7. Aktivieren Sie OMi (**Start > Programme > OMi > Verwaltung > OMi aktivieren**).

Erstellen einer Verbindung zu einer Microsoft SQL-Datenbank mit Windows Authentifizierung oder Herstellen der Verbindung

Sie erstellen eine Datenbank oder eine Verbindung zu einer Datenbank mithilfe des Konfigurations-Assistenten. Zum Erstellen von oder Verbinden mit einer Microsoft SQL-Datenbank mit der Windows-Authentifizierung müssen Sie diese Option innerhalb des Konfigurations-Assistenten auswählen. Weitere Informationen zum verwenden des Konfigurationsassistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.

Teil III: Bereitstellen und Verwalten der Oracle Server-Datenbank

Kapitel 8: Übersicht über die Oracle Server-Bereitstellung

Sie können OMi-Schemas auf einem Oracle-Server festlegen. In diesem Kapitel werden die folgenden Themen zur Bereitstellung von Oracle-Servern für die Verwendung mit OMi beschrieben:

- ["Informationen über die Oracle Server-Bereitstellung"](#) unten
- ["Systemanforderungen"](#) auf der nächsten Seite
- ["Zeichensatzunterstützung in Oracle"](#) auf der nächsten Seite

Informationen über die Oracle Server-Bereitstellung

Führen Sie die folgenden Schritte aus, um Oracle Server für die Verwendung mit OMi bereitzustellen:

- **Installieren Sie Oracle Server.**

Weitere Informationen zur Installation der Oracle-Software finden Sie im Installationshandbuch in der Dokumentation für Ihre jeweilige Oracle-Plattform. Informationen zu Softwareinstallationsoptionen finden Sie unter ["Richtlinien für die Oracle Server-Konfiguration und -Dimensionierung"](#) auf Seite 59 und ["Oracle-Prüfliste"](#) auf Seite 79.

Hinweis: Wenn Sie Oracle 12c in einer typischen Installation installieren, empfiehlt es sich, die Option **Create as Container database** zu deaktivieren. Hierdurch wird sichergestellt, dass die Datenbank ohne CDB erstellt wird und mit der Datenbank in Oracle 11 identisch ist.

Verwenden Sie bei einer *unbeaufsichtigten* Installation (oder einer Installation mit *Antwortdatei*) die Operation **createDatabase** (mit dem entsprechenden Konfigurationsabschnitt) anstelle von **createPluggableDatabase** in Ihrer DBCA-Antwortdatei.

```
OPERATION_TYPE = "createDatabase"  
[CREATEDATABASE]
```

- **Erstellen Sie eine Datenbank unter Oracle Server, um OMi-Daten zu speichern.**

Informationen zur Instanzkonfiguration und Richtlinien zur Festlegung der Größe finden Sie unter ["Richtlinien für die Oracle Server-Konfiguration und -Dimensionierung"](#) auf Seite 59. Weitere Informationen zur Installation der Datenbankinstanz finden Sie im Installationshandbuch in der Dokumentation für Ihre jeweilige Oracle-Plattform.

- **Erstellen Sie einen oder mehrere Oracle-Tablespaces zum Speichern von OMi-Daten.**

Weitere Informationen finden Sie unter ["Oracle-Tablespaces"](#) auf Seite 62.

- **Erstellen Sie ein Oracle-Benutzerschema für OMi-Schemas.**

Sie können OMi-Benutzerschemas manuell erstellen oder den Konfigurations-Assistenten zum Erstellen der Schemas verwenden. Weitere Informationen zum Erstellen eines Oracle Benutzerschemas für OMi finden Sie unter ["Manuelles Erstellen eines OMi-Schemas"](#) auf Seite 55.

Systemanforderungen

In diesem Abschnitt werden die Systemanforderungen für das Arbeiten mit Oracle Server in Kombination mit OMi beschrieben.

Dieser Abschnitt umfasst die folgenden Themen:

- "Hardwareanforderungen" unten
- "Softwareanforderungen" unten
- "Oracle-Instanzen" unten

Hardwareanforderungen

Richtlinien für die OMi-Hardware-Dimensionierung finden Sie unter "[Anforderungen](#)" auf Seite 11.

Informationen zu den Oracle-Hardwareanforderungen finden Sie im Installationshandbuch für Ihre jeweilige Oracle-Plattform. Zusätzliche Informationen stehen auch über die Medien der Oracle-Softwareverteilung sowie über die Oracle-Dokumentation zur Verfügung. Informationen zur Oracle-Dokumentation finden Sie unter:

<http://otn.oracle.com/documentation/index.html>

Softwareanforderungen

Eine Liste der unterstützten Oracle-Datenbankversionen finden Sie in der Tabelle zur OMi-Unterstützung unter:

<http://support.openview.hp.com/selfsolve/document/KM323488>

Oracle-Instanzen

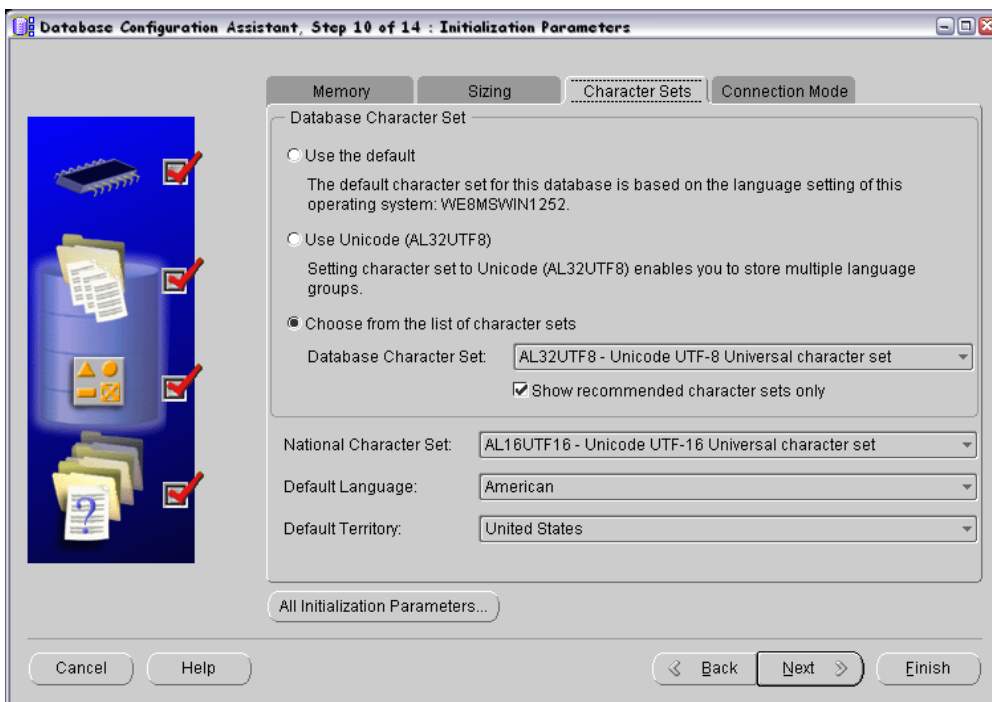
Sie können mehrere Oracle-Instanzen auf einem Computer installieren, wenn Sie dieselbe Oracle-Datenbank-Engine verwenden.

Verwenden Sie für die OMi-Zertifizierung nicht mehrere Oracle-Instanzen. Wenn Sie mehr als eine Instanz für OMi-Datenbanken verwenden möchten, stellen Sie sicher, dass alle Instanzen in diesem Dokument wie beschrieben konfiguriert sind und dass sie alle dieselben Merkmale haben (wie zum Beispiel den gleichen Zeichensatz).

Zeichensatzunterstützung in Oracle

Wenn Sie in einer Oracle-Datenbank eine neue Instanz erstellen, müssen Sie den Zeichensatz für die Instanz angeben. Sämtliche Zeichendaten, einschließlich der Daten im Datenwörterbuch, werden in dem Zeichensatz der Instanz gespeichert.

Während der Datenbankerstellung können Sie aus der Liste der verfügbaren Zeichensätze auswählen.



Wenn Sie nur grundlegende lateinische Zeichen verwenden, können Sie den Zeichensatz `we8ISO8859P1` verwenden, der die meisten westeuropäischen Zeichen unterstützt. Wenn Sie jedoch nicht-lateinische Zeichen oder mehrere Sprachen verwenden, müssen Sie einen UTF-8-Zeichensatz für die Instanz verwenden. Es gibt zwei Oracle-Zeichensätze, die die UTF-8-Codierung implementieren, `UTF8` und `AL32UTF8`.

Es wird auch empfohlen, das Clientgebietsschema durch Einrichten der Umgebungsvariablen `NLS_Lang` auf geeignete Werte festzulegen, sodass Datenbankclientapplikationen mit dem Datenbankserver kommunizieren können.

Sprache	NLS_LANG Wert
Englisch	<code>american_america.WE8ISO8859P1</code> oder <code>american_america.AL32UTF8</code>
Französisch	<code>french_france.AL32UTF8</code>
Spanisch	<code>spanish_spain.AL32UTF8</code>
Deutsch	<code>german_germany.AL32UTF8</code>
Russisch	<code>russian_russia.AL32UTF8</code>
Japanisch	<code>japanese_japan.AL32UTF8</code>
Koreanisch	<code>korean_korea.AL32UTF8</code>
Vereinfachtes Chinesisch	<code>simplified_chinese_china.AL32UTF8</code>

Kapitel 9: Manuelles Erstellen der Oracle-Schemas

In diesem Kapitel wird beschrieben, wie die OMi-Benutzerschemas manuell erstellen und aufgefüllt werden. Es enthält außerdem Informationen zu Oracle-Berechtigungen für OMi-Benutzerschemata.

Dieses Kapitel umfasst die folgenden Themen:

- ["Oracle Server-Datenbanken - Übersicht" unten](#)
- ["Manuelles Erstellen eines OMi-Schemas" unten](#)
- ["OMi – Berechtigungen für Oracle-Schema" auf Seite 58](#)

Oracle Server-Datenbanken - Übersicht

OMi verwendet die Management-, RTSM- und Ereignisdatenbanken.

Während des OMi-Setups können neue Datenbanken automatisch über das Setupverfahren eingerichtet werden oder vorhandene Datenbanken genutzt werden. Die vorhandenen Datenbanken können entweder manuell im Voraus (z. B. aufgrund von Sicherheitsbeschränkungen im Unternehmen) oder durch eine frühere Installation der gleichen OMi-Version erstellt werden.

Weitere Informationen zum Installieren des OMi-Servers finden Sie im OMi-Installationshandbuch.

Manuelles Erstellen eines OMi-Schemas

Die OMi-Benutzerschemas können manuell oder automatisch mithilfe des Konfigurationsassistenten erstellt werden. Weitere Informationen zum Konfigurations-Assistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.

Der Konfigurationsassistent fordert Sie zur Eingabe der folgenden Parameter auf:

- **Hostname.** Der Name des Hostcomputers, auf dem Oracle Server installiert ist.
- **Port.** Der Oracle-Listenerport. In OMi wird automatisch der Standardport **1521** angezeigt.
- **SID.** Der Oracle-Instanzname, mit dem die von OMi verwendete Oracle-Datenbankinstanz eindeutig identifiziert wird.
- **Kennwort.** Das Kennwort eines Benutzerkontos, das von OMi verwendet wird, um sich mit den OMi-Datenbanken zu verbinden und Daten abzurufen.
- **Administratorbenutzername und -kennwort.** (Um als Administrator eine Verbindung herzustellen) Der Name und das Kennwort eines Benutzers mit Administratorberechtigungen für Oracle Server (z. B. ein Systembenutzer).
- **Temporärer Tablespace.** Der Name des temporären Tablespaces, den Sie den Benutzerschemas zugewiesen haben. Der standardmäßige temporäre Oracle-Tablespace ist **temp**.
- **Standard-Tablespace.** Der Name des speziellen Standard-Tablespace, den Sie für die Benutzerschemas erstellt haben. Weitere Informationen zum Erstellen eines dedizierten OMi-

Tablespaces finden Sie unter ["Oracle-Tablespaces"](#) auf Seite 62.

- **Datenbanknamen (Management, RTSM und Event).** Die Namen der vorhandenen Benutzerschemas oder die Namen, die Sie für die neuen Benutzerschemas auswählen (z. B. OMi_Management).

Hinweis: Wenn Sie den Oracle-Client SQL* Plus verwenden, stellen Sie sicher, dass Sie die wichtigen Einschränkungen kennen. Ein Befehl für eine Datenbankerstellung kann die maximale Befehlszeilenlänge überschreiten, die von SQL*Plus unterstützt wird. Weitere Informationen zu den Einschränkungen von SQL*Plus finden Sie in der Oracle-Dokumentation.

Dieser Abschnitt umfasst die folgenden Themen:

- ["Manuelles Erstellen eines Management-Benutzerschemas"](#) unten
- ["Manuelles Erstellen eines RTSM-Benutzerschemas"](#) auf der nächsten Seite
- ["Manuelles Erstellen eines Ereignisschemas"](#) auf der nächsten Seite

Manuelles Erstellen eines Management-Benutzerschemas

Das Management-Benutzerschema sollte von einem zertifizierten Datenbankadministrator manuell erstellt werden (mit der datenbankspezifischen Methodik Ihrer Organisation). Sie können dann später über den Konfigurations-Assistenten eine Verbindung zu dem Management-Benutzerschema herstellen. Weitere Informationen zum Konfigurations-Assistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.

Hinweis: Wenn keine Datenbankadministrator-Verbindungsparameter über eine nicht sichere Netzwerkverbindung übertragen werden können, müssen Sie ein Management-Benutzerschema möglicherweise manuell konfigurieren.

So erstellen Sie ein SQL-Skript für ein Management-Schema und führen es aus:

1. Erstellen des Benutzerschemas:

Erstellen Sie ein Benutzerschema mit den unter ["OMi – Berechtigungen für Oracle-Schema"](#) auf Seite 58 beschriebenen Berechtigungen.

2. Erstellen des SQL-Skripts für das Management-Schema:

Führen Sie die folgende Skriptdatei im Verzeichnis `<OMi_GW_Stammverzeichnis>/dbverify/bin` aus:

Für Windows:

```
generate_create_script.bat [Bereitstellungstyp] [Name des generierten Skripts]  
[DB-Typ] [Sitzungstyp]
```

Für Linux:

```
./generate_create_script.sh [Bereitstellungstyp] [Name des generierten Skripts]  
[DB-Typ] [Sitzungstyp]
```

Die folgenden Parameter müssen für das Management-Schema verwendet werden:

- **deployment_type.** EE (Enterprise Edition)
- **Name des generierten Skripts.** Der Name der zu generierenden SQL-Datei, z. B.


```
management.sql.
```

- **DB-Typ.** Oracle
- **Sitzungstyp.** Management

Beispiel:

```
/opt/HP/BSM/dbverify/bin/generate_create_script.sh EE Management.sql Oracle  
Management
```

3. Ausführen des SQL-Skripts für das Management-Schema:
 - a. Stellen Sie die Verbindung zu dem Management-Schema her und führen Sie das in Schritt 2 erstellte Skript zum Bereitstellen der Management-Objekte aus.
 - b. Führen Sie das folgende SQL-Skript unter `<OMi_GW_Stammverzeichnis>/AppServer/webapps/site.war/DataBases/ORACLE_DB_Utills` aus:

```
management_ora_inserts.sql
```
 - c. Wenn das Skript beendet wurde, führen Sie die Änderung durch.

Manuelles Erstellen eines RTSM-Benutzerschemas

Das RTSM-Benutzerschema sollte von einem zertifizierten Datenbankadministrator manuell erstellt werden (mit der entsprechenden Datenbankmethode Ihrer Organisation). Sie können dann später über den Konfigurationsassistenten eine Verbindung zu dem RTSM-Benutzerschema herstellen. Weitere Informationen zum Konfigurations-Assistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.

So erstellen Sie ein separates RTSM-Benutzerschema manuell:

Erstellen Sie ein Benutzerschema mit den unter "[OMi – Berechtigungen für Oracle-Schema](#)" auf der [nächsten Seite](#) beschriebenen Berechtigungen.

Manuelles Erstellen eines Ereignisschemas

Erstellen Sie ein Ereignisschema mit den unter "[OMi – Berechtigungen für Oracle-Schema](#)" auf der [nächsten Seite](#) beschriebenen Berechtigungen.

So erstellen Sie Ereignisschema-Skripts:

Führen Sie eine der folgenden Skriptdateien aus:

Für Windows:

```
<OMi_STAMMVERZEICHNIS>\opr\bin\opr-schema-script-generator.bat
```

Für Linux:

```
<OMi_STAMMVERZEICHNIS>/opr/bin/opr-schema-script-generator.sh
```

Die folgenden Parameter müssen für das Ereignisschema verwendet werden:

DB-Typ. ORACLE

Geben Sie den folgenden Befehl ein, um beispielsweise Datenbanktabellen für eine Oracle-Datenbank für die Datei `oprGenerateDbTablesForOracle.sql` zu erstellen:

```
<OMi_HOME>\opr\bin\opr-schema-script-generator.bat -d ORACLE -o  
C:\temp\oprGenerateDbTablesForOracle.sql
```

So führen Sie das SQL-Skript für die Ereignisdatenbank aus:

1. Erstellen Sie die Ereignisdatenbank.
2. Stellen Sie die Verbindung zu der Ereignisdatenbank her und führen Sie das im vorherigen Abschnitt generierte Skript zum Bereitstellen der Operations-Objekte aus.

Hinweis: Führen Sie die oben genannten Verfahren nur durch, wenn Sie ein erfahrener Datenbankadministrator für Oracle sind.

OMi – Berechtigungen für Oracle-Schema

Bei der Verwendung eines Oracle-Benutzerschemas können Sie alle Datenbankaktionen durchführen, die im Rahmen der Oracle-Berechtigung für den Benutzer zulässig sind. In der folgenden Liste (aus dem Skript `oracle_user_create.sql` unter **<Stammverzeichnis des Gateway-Servers>\AppServer\webapps\site.war\DataBases\ORA_DB_Utills**) sind die erforderlichen Datenbankberechtigungen beschrieben, die dem OMi-Benutzer erteilt werden müssen. Diese Berechtigungen werden auch von den OMi-Plattformkomponenten zum Erstellen eines neuen Oracle-Benutzers verwendet:

- GRANT "CONNECT" TO <HP Operations Manager i-Oracle-Benutzerschema>
- GRANT CREATE SEQUENCE TO <HP Operations Manager i-Oracle-Benutzerschema>
- GRANT CREATE TABLE TO <HP Operations Manager i-Oracle-Benutzerschema>
- GRANT CREATE TRIGGER TO <HP Operations Manager i-Oracle-Benutzerschema>
- GRANT UNLIMITED TABLESPACE TO <HP Operations Manager i-Oracle-Benutzerschema>
- GRANT CREATE VIEW TO <HP Operations Manager i-Oracle-Benutzerschema>
- GRANT CREATE PROCEDURE TO <HP Operations Manager i-Oracle-Benutzerschema>
- ALTER USER <HP Operations Manager i-Oracle-Benutzerschema> DEFAULT ROLE ALL

Wird das Schema für die RTSM-Datenbank verwendet, ist die folgende Berechtigung ebenfalls erforderlich:

```
GRANT CREATE TYPE TO <HP Operations Manager i-Oracle-Benutzerschema für RTSM>
```

Hinweis: OMi unterstützt alle Benutzer mit höheren Berechtigungen. Verwenden Sie für die OMi-Zertifizierung einen Oracle-Benutzer, der über die exakten, oben beschriebenen Oracle-Berechtigungen verfügt.

Kapitel 10: Richtlinien für die Oracle Server-Konfiguration und -Dimensionierung

Dieses Kapitel enthält Richtlinien für die Einstellungen der Oracle-Datenbankkonfiguration und -Speicherung, die bei der Arbeit mit Oracle Server und OMi verwendet werden sollten. Beachten Sie, dass sich die empfohlenen Einstellungen je nach Größe Ihrer OMi-Bereitstellung ändern.

Dieses Kapitel umfasst die folgenden Themen:

- "Oracle-Parametereinstellungen" unten
- "Oracle-Tablespaces" auf Seite 62
- "Einstellungen für Oracle-Tablespace" auf Seite 64
- "Verwenden der RAID-Konfiguration" auf Seite 67

Oracle-Parametereinstellungen

Die folgende Tabelle beschreibt die empfohlenen Werte für eine Reihe von Oracle-Datenbankinitialisierungsparametern, die bei der Arbeit mit dem OMi-Datenbankserver verwendet werden können:

Parametername	OMiDeployment		Anmerkungen
	Klein	Groß	
DB_BLOCK_SIZE	8K	8K–16K	Sollte ein Vielfaches der Betriebssystem-Blockgröße sein.
DB_CACHE_ADVICE	EIN	EIN	Für die Statistikerfassung, wenn eine Abstimmung erforderlich ist.
SGA_TARGET	1 GB	Mindestens 4 GB	<p>Wenn Sie diesen Parameter festlegen, legt Oracle die Größe des Puffer-Cache (db_cache_size), des Shared Pool (shared_pool_size), des großen Pools (large_pool_size), des Java-Pools (java_pool_size) und des Streams-Pools (streams_pool_size) automatisch fest.</p> <p>Der für SGA_TARGET konfigurierte Wert legt die Gesamtgröße der SGA-Komponenten fest.</p> <p>Wenn SGA_TARGET eingestellt ist (d. h. der Wert ist nicht 0), und einer der oben genannten Pools ist auch ungleich Null, wird der Pool-Wert als</p>

Parametername	OMiDeployment		Anmerkungen
	Klein	Groß	
			Mindestwert für den Pool verwendet.
MEMORY_TARGET	2 GB	Mindestens 5 GB	Bei der automatischen Arbeitsspeicherverwaltung wird der gesamte Instanz-Arbeitsspeicher automatisch von der Instanz verwaltet und abgestimmt. Der Instanz-Arbeitsspeicher enthält die System Global Area (SGA) und die Programm Global Area (PGA). MEMORY_TARGET ist der einzige erforderliche Arbeitsspeicherparameter, der festgelegt werden muss. Es wird jedoch empfohlen, SGA_TARGET oder PGA_AGGREGATE_TARGET festzulegen sowie das häufige Ändern der Größe der SGA- und PGA-Komponenten zu vermeiden. Die eingegebenen Werte für SGA_TARGET und PGA_AGGREGATE_TARGET dienen als Mindestwerte.
LOG_BUFFER	1 MB	5 MB	
DB_FILE_MULTIBLOCK_READ_COUNT	Oracle-Standardwert	Oracle-Standardwert	
PROCESSES	200	500	
SESSIONS	225	445	(1,1 * PROCESSES) + 5
OPTIMIZER_INDEX_COST_ADJ	100	100	Ist der Parameter auf 100 festgelegt, wertet der Optimierer den Indexzugriffspfad mit dem normalen Aufwand aus. Wird der Parameter auf einen anderen Wert festgelegt (z. B. 50) wird der Indexzugriffspfad mit diesem Prozentsatz des normalen Aufwands ausgewertet.
TIMED_STATISTICS	True	True	
LOG_CHECKPOINT_INTERVAL	0	0	
LOG_CHECKPOINT_TIMEOUT	0 oder ≥ 1800 (größer als	0 oder ≥ 1800 (größer als	

Parametername	OMiDeployment		Anmerkungen
	Klein	Groß	
	oder gleich 1800)	oder gleich 1800)	
OPTIMIZER_ MODE	ALL_ROWS	ALL_ROWS	
CURSOR_ SHARING	Exact	Exact	
OPEN_ CURSORS	2000	2000	
COMPATIBLE	Identisch mit dem installierten Release des Oracle-Servers	Identisch mit dem installierten Release des Oracle-Servers	Beispiel: 11.2.0 für Oracle 11g R2 oder 12.0.0 für Oracle 12c.
SQL_ TRACE	False	False	
UNDO_ MANAGEMENT	Auto	Auto	
UNDO_ RETENTION	Oracle-Standardwert	Oracle-Standardwert	Automatische Abstimmung wird durchgeführt.
RECYCLEBIN	Aus	Aus	
NLS_ LENGTH_ SEMANTICS	BYTE	BYTE	Dieser Parameter steuert die Längendefinition der varchar-Spalten.
WORKAREA_ SIZE_ POLICY	AUTO	AUTO	
PGA_ AGGREGATE_ TARGET	400 MB	Mindestens 1 GB	
STATISTICS_ LEVEL	TYPICAL	TYPICAL	Aktiviert die Abstimmung, falls erforderlich.
STAR_ TRANSFORMATION_ ENABLED	TRUE	TRUE	Bestimmt, ob die Star-Transformation vom Optimizer auf Abfragen angewendet werden kann. Das Festlegen des Parameters auf `False` wirkt sich auf die Leistung der End User Management-Reports aus.
ALWAYS STAR TRANSFORMATION	TRUE	TRUE	Dieser ausgeblendete Parameter weist den Optimizer an, immer die Star-Transformation zu bevorzugen, sofern zutreffend.

Parametername	OMiDeployment		Anmerkungen
	Klein	Groß	
OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES	FALSE	FALSE	Steuert die automatische Planerfassung als Teil von Oracle SQL Management Base (SMB).
AUDIT_TRAIL	NONE	NONE	Vordefiniertes Out-of-the-box-Auditing wird in die SSYS.AUD\$-Audit-Trail-Tabelle geschrieben. Es ist ratsam, diesen Wert auf `None` zu ändern, um ein Anwachsen des System-Tablespace zu vermeiden.
CURSOR_SPACE_FOR_TIME	False	False	
USE_STORED_OUTLINES	False	False	Oracle-Standardwert
BLANK_TRIMMING	False	False	
FIXED_DATE	Nicht festgelegt	Nicht festgelegt	OMi verwendet die SYSDATE-Funktion, um die Systemzeit als Teil des Applikationsprozesses zu generieren.
_PARTITION_LARGE_EXTENTS	FALSE	FALSE	Relevant für Oracle 11.2.0.2 und Oracle 12c. Wenn dieser ausgeblendete Parameter auf TRUE festgelegt ist, wirkt sich dies auf die Größe von Partitionen in systemeigenen partitionierten Tabellen aus. Die anfängliche Speicherbereich, der jeder Partition zugewiesen ist, ist sehr groß, wodurch ein unerwünschtes Wachstum der Datenbankdateien entsteht.
DEFERRED_SEGMENT_CREATION	FALSE	FALSE	Nur relevant für Oracle 11.2.0.*

Oracle-Tablespaces

Ein Oracle-Tablespace ist ein Oracle-Objekt, das einen logischen Container der Datenbankobjekte darstellt, z. B. Tabellen, Indizes usw. Bei der Arbeit mit OMi müssen Sie einen oder mehrere spezielle Tablespaces für Ihre OMi-Benutzerschemata erstellen. Sie können auch einen speziellen temporären

Tablespace für OMi erstellen. Zum Erstellen eines Tablespace müssen Sie bestimmte Dateien des Betriebssystems, die den Tablespace physisch darstellen, sowie Erweiterungsparameter bereitstellen.

Beim Zuordnen von Betriebssystemdateien ist eine Option zum automatischen Erweitern der Datei verfügbar. Diese Funktion wird von OMi unterstützt, ist aber nicht für die Verwendung mit OMi zertifiziert, da sie zu einem erhöhten Festplattenspeicherverbrauch durch das System führen kann.

Hinweis: Die folgenden Begriffe beziehen sich alle auf dieselbe Datenbank: Ereignisschema = OPR-Schema = Ereignisdatenbank.

In diesem Abschnitt wird Folgendes behandelt:

- ["Lokal verwaltete Tablespaces" unten](#)
- ["Erstellen eines Oracle-Tablespace" unten](#)

Lokal verwaltete Tablespaces

Lokal verwaltete Tablespaces können einheitliche Erweiterungsgrößen oder variable Erweiterungsgrößen aufweisen, die automatisch vom System ermittelt werden. Beim Erstellen des Tablespace wird mit der Systemverwaltungsoption **uniform** oder **autoallocate** der Zuordnungstyp angegeben.

Für die systemverwalteten Erweiterungen, bestimmt Oracle die optimale Größe der Erweiterungen mit einer Mindesterweiterungsgröße von 64 KB. Dies ist die Standardeinstellung für die Erweiterungsgröße bei permanenten Tablespaces.

Bei einheitlichen Erweiterungen können Sie eine Erweiterungsgröße angeben oder die Standardgröße von 1 MB verwenden. Temporäre Tablespaces, die ihre Erweiterungen lokal verwalten, dürfen nur diese Art der Zuweisung verwenden.

Beachten Sie, dass die Speicherparameter NEXT, PCTINCREASE, MINEXTENTS, MAXEXTENTS und DEFAULT STORAGE nicht für Erweiterungen gelten, die lokal verwaltet werden.

Alle Daten und temporären Tablespaces sollten bei der Arbeit mit OMi lokal verwaltet werden.

Weitere Informationen zur lokalen Verwaltung von temporären Tablespaces mit TEMPFILE finden Sie unter ["Einstellungen für temporäre Tablespaces" auf Seite 65](#).

Erstellen eines Oracle-Tablespace

Sie erstellen in OMi einen Oracle-Tablespace mithilfe des Skripts **oracle_tablespace_create.bat** (für eine Windows-Installation) oder **oracle_tablespace_create.sh** (für eine Linux-Installation). Beide Skripts finden Sie unter **<HP Operations Manager i-Server>AppServer\webapps\site.war\Databases\ORA_DB_Utils**.

Hinweis: Dieses Skript ist Basisskript für die Tablespace-Erstellung und enthält eine Datendatei. Sie können diese Datei entsprechend der Größe Ihrer OMi-Installation bearbeiten. Weitere Informationen finden Sie unter ["Einstellungen für den Daten-Tablespace" auf der nächsten Seite](#).

So erstellen Sie einen Oracle-Tablespace in OMi:

Führen Sie den folgenden Befehl auf dem Verzeichnis aus, in dem sich die Skripts **oracle_tablespace_create.bat** oder **oracle_tablespace_create.sh** befinden:

```
oracle_tablespace_create [Admin-Benutzer] [Admin-Kennwort] [tns-Eingabename]
[Tablespace-Name] [Dateiname] [Dateigröße]
```

Beispiel:

```
oracle_tablespace_create system manager topaz omi_tablespace
h:\oracle\oradata\topaz01.ora 50M
```

- **[Admin-Benutzer].** Name des Benutzers mit Administratorberechtigungen für den Oracle-Server.
- **[Admin-Kennwort].** Kennwort für den angegebenen Benutzer.
- **[tns-Eingabename].** Der TNS-Name, der in der Datei **tnsnames.ora** auf dem lokalen Oracle-Client angegeben wird.
- **[Tablespace-Name].** Name des zu erstellenden Tablespace.
- **[Dateiname].** Name der zu erstellenden Datei, einschließlich des vollständigen Pfads zur Datei.
- **[Dateigröße].** Dateigröße; verwenden Sie M für MB und K für KB. Beachten Sie, dass die tatsächliche Größe je nach Datenmenge, die in OMi erzeugt wird, und nach Zeitraum, für den historische Daten beibehalten werden, variiert.

Einstellungen für Oracle-Tablespace

In diesem Abschnitt werden die Speichereinstellungen und Richtlinien der Dateidimensionierung für Daten-Tablespaces, temporäre Tablespaces, Wiederholen-Protokolle und Rückgängigmachen-Tablespaces beschrieben.

Dieser Abschnitt umfasst die folgenden Themen:

- ["Einstellungen für den Daten-Tablespace" unten](#)
- ["System-Tablespace-Einstellungen" auf der nächsten Seite](#)
- ["Einstellungen für temporäre Tablespaces" auf der nächsten Seite](#)
- ["Einstellungen für das Wiederholen-Protokoll" auf Seite 66](#)
- ["Einstellungen für Rückgängigmachen-Segmente" auf Seite 66](#)

Einstellungen für den Daten-Tablespace

Die folgende Tabelle gibt die empfohlene Größe für OMi-Tablespaces an:

Tablespace	OMiDeployment		Anmerkungen
	Klein	Groß	
Management	5 GB	10 GB	Die angegebene Größe ist eine Mindestanforderung.
RTSM	1 GB	20 GB	Die angegebene Größe ist eine Mindestanforderung.
Ereignis	5 GB	20 GB	Die angegebene Größe ist eine Mindestanforderung.

Standardspeichereinstellungen für den Daten-Tablespace

Die Speichereinstellungen für Daten-Tablespaces sollten wie folgt lauten:

- Lokal verwalteter Tablespace
- Automatic Segment Space Management (ASSM)
- Automatische Verwaltung der lokalen Erweiterung

System-Tablespace-Einstellungen

Die folgende Tabelle gibt die empfohlenen Einstellungen für System-Tablespaces an:

Tablespace	OMiDeployment	
	Klein	Groß
SYSTEM	2 GB	5 GB
SYSAUX	2 GB	5 GB

Hinweis: Stellen Sie sicher, dass Sie die SYSTEM- und SYSAUX-Tablespaces für die OMi-Schemata verwenden.

Die standardmäßigen System-Tablespace-Speichereinstellungen sollten wie folgt lauten:

- Lokal verwalteter Tablespace
- Verwaltung der Segmentgröße:
 - **SYSAUX:** Automatisch
 - **SYSTEM:** Manuell
- Automatische Verwaltung der lokalen Erweiterung

Einstellungen für temporäre Tablespaces

Die folgende Tabelle gibt die empfohlenen Einstellungen für temporäre Tablespaces an:

Tablespace	OMiDeployment		Anmerkungen
	Klein	Groß	
TEMP	1 GB	10 GB	Verwenden mehrerer Dateien mit großen Tablespaces.
TEMP-Speichereinstellungen	Uniform-Zuordnung: 2 MB	Uniform-Zuordnung: 2 MB	<ul style="list-style-type: none"> • Sollte lokal verwaltet werden (Uniform-Zuordnung). • Tablespaces sollten einen temporären Typ aufweisen (Verwendung von TEMPFILE).

Tablespace	OMiDeployment		Anmerkungen
	Klein	Groß	
			<ul style="list-style-type: none"> Die Segmentgrößenverwaltung in temporären Tablespaces erfolgt manuell.

Einstellungen für das Wiederholen-Protokoll

Die folgende Tabelle gibt die empfohlenen Einstellungen für die Wiederholen-Protokolldateien an:

Einstellung	OMiDeployment	
	Klein	Groß
Wiederholen-Protokolldateigröße	200 MB	500 MB - 1 GB
Minimale Anzahl von Gruppen	4	4
Minimale Anzahl der Mitglieder pro Gruppe	2	2

Einstellungen für Rückgängigmachen-Segmente

Die folgende Tabelle gibt die empfohlenen Rückgängigmachen-Einstellungen an:

Einstellung	OMi-Systemprofile		Anmerkungen
	Klein	Groß	
Rückgängigmachen-Tablespace-Größe	1 GB	10 GB	Die Anzahl der Segmente, die minimale Anzahl an Erweiterungen und die Rollback-Segmentgröße (anfängliche, nächste) werden alle automatisch von Oracle festgelegt.
UNDO_ MANAGEMENT- Parameter	AUTO		Oracle-Standardwerte.
UNDO_ RETENTION- Parameter	Oracle-Standardwert		

Die standardmäßigen Speichereinstellungen für Rückgängigmachen-Tablespaces sollten wie folgt lauten:

- Lokal verwalteter Tablespace
- Automatische Verwaltung der lokalen Erweiterung
- Die Segmentgrößenverwaltung in Rückgängigmachen-Tablespaces erfolgt manuell.

Verwenden der RAID-Konfiguration

Die Verwendung von RAID ist transparent für Oracle. Alle speziellen Funktionen für die RAID-Konfigurationen werden durch das Betriebssystem und nicht durch Oracle verarbeitet.

Die Verwendung von RAID-Geräten unterscheidet sich je nach Oracle-Dateityp. Datendateien und Archivierungsprotokolle können auf RAID-Geräten platziert werden, da auf sie nach dem Zufallsprinzip zugegriffen wird. Wiederholen-Protokolle sollten nicht auf RAID-Geräten abgelegt werden, da auf sie sequenziell zugegriffen wird, und die Leistung daher optimiert wird, wenn der Laufwerks-Lesekopf sich in der Nähe der letzten Schreibposition befindet. Die Spiegelung von Wiederholen-Protokolldateien wird von Oracle dagegen dringend empfohlen.

RAID ist viel einfacher zu verwenden als die Oracle-Methoden für die Datenplatzierung und Striping.

Beachten Sie die folgenden RAID-Konfigurationsempfehlungen:

- RAID wirkt sich normalerweise mehr auf Leseoperationen als auf Schreiboperationen aus. Dies gilt insbesondere, wenn die Parität berechnet werden muss (RAID 3, RAID 5 usw.).
- Sie können online verfügbare oder archivierte Wiederholen-Protokolldateien auf RAID 1-Geräten ablegen. Verwenden Sie nicht RAID 5. Platzieren Sie darüber hinaus TEMP-Tablespace-Datendateien auf RAID 1-Geräten, und nicht auf RAID 5, da die gestreamte Schreibleistung der verteilten Parität (RAID 5) nicht so gut ist wie eine einfache Spiegelung (RAID 1).
- Auslagerungsbereiche können auf RAID-Geräten verwendet werden, ohne dass dies Auswirkungen auf Oracle hat.

Die folgende Tabelle beschreibt die RAID-Geräte und -Typen, die mit den einzelnen Oracle-Dateitypen verwendet werden sollten:

RAID	RAID-Typ	Steuerungsdatei	Datenbankdatei	Wiederholen-Protokolldatei/Temporär	Archivdatei
0	Striping	Vermeiden	OK	Vermeiden	Vermeiden
1	Spiegeln	Empfohlen	OK	Empfohlen	Empfohlen
1+0	Striping und Spiegelung	OK	Empfohlen	Vermeiden	Vermeiden
3	Striping mit statischer Parität	OK	OK	Vermeiden	Vermeiden
5	Striping mit rotierender Parität	OK	Empfohlen, wenn RAID 1+0 nicht verfügbar ist.	Vermeiden	Vermeiden

Hinweis:

- RAID 0 bietet keinen Schutz vor Ausfällen. Es erfordert eine gute Sicherheitsstrategie.

- RAID 1+0 wird für Datenbankdateien empfohlen, da Hotspots vermieden werden und es die optimale Leistung während eines Datenträgerfehlers liefert. Der Nachteil von RAID 1+0 ist die teure Konfiguration.
- Verwenden Sie Festplatten mit sehr hohen Umdrehungen pro Minute für temporäre/Wiederholen-Protokolle. Verwenden Sie so viele Controller im Array wie möglich und stellen Sie sicher, dass Sie die Wiederholen-Protokollgruppen auf verschiedenen Controllern platzieren.

ASM-Speicher

Zusätzlich zu den herkömmlichen Dateisystemoptionen kann Oracle Automatic Storage Management (ASM) als Speicheroption für Oracle-Datenbankdateien verwendet werden.

Weitere Informationen zum Konfigurieren und Verwalten von ASM finden Sie in der Dokumentation "Oracle Database Storage Administrator's Guide".

Eine Übersicht zu ASM, bewährte Methoden und technische Dokumente finden Sie unter:

<http://www.oracle.com/technology/products/database/asm/index.html>

Achten Sie auf die folgenden Aufgaben:

- Vorbereiten und Zuordnen der ASM-Datenträger, Konfigurieren der ASM-Datenträgergruppen und Einrichten der richtigen Redundanz.
- Verwenden des Oracle ASMLIB-Treibers für eine bessere Erkennung und Verwaltung von ASM-Datenträgern und für erweiterte E/A-Verarbeitung (nur Linux).
- Ordnungsgemäße Einrichtung von ASM_DISKSTRING entsprechend des Pfads zur Datenträgererkennung.
- Ausgleichsoperationen für manuelle und automatische Steuerungsoptionen.

Kapitel 11: Warten einer Oracle Server-Datenbank

Dieses Kapitel beschreibt die verschiedenen Wartungs- und Abstimmungsverfahren, die für OMi-Datenbanken empfohlen werden, die auf Oracle-Servern erstellt wurden, sowie einige der verfügbaren Datenbanksicherungs- und Wiederherstellungsmethoden.

Dieses Kapitel umfasst die folgenden Themen:

- ["Datenbankwartung und -abstimmung" unten](#)
- ["Oracle-Datenbank - Sicherung und Wiederherstellung" auf Seite 74](#)

Datenbankwartung und -abstimmung

Schlechte Datenbankleistung kann durch die fehlerhafte Konfiguration der Instanz und der Datenbank verursacht werden oder durch anormalen Ressourcenverbrauch von Oracle-Transaktionen, Benutzern oder Prozessen. Der Datenbankadministrator muss unbedingt proaktiv den Ressourcenverbrauch überwachen und eventuelle Anormalitäten korrigieren, bevor die Leistung betroffen ist.

Hinweis: Speicher, CPU und E/A sind die drei wichtigsten Systemressourcen auf die Oracle zugreift.

Es gibt eine Reihe von Drittanbieterwerkzeugen, die Sie für die Überwachung des Datenbankverhaltens verwenden können und die Ihnen beim Ermitteln von Leistungsengpässen im System helfen. Verwenden Sie dabei die folgenden Richtlinien.

Dieser Abschnitt umfasst die folgenden Themen:

- ["System Global Area \(SGA\)" unten](#)
- ["Datenbankauslastung - Verhalten" auf der nächsten Seite](#)
- ["CPU und E/A" auf der nächsten Seite](#)
- ["Oracle-Warnungsdatei" auf der nächsten Seite](#)
- ["Archivprotokoll - Dateisystem" auf der nächsten Seite](#)
- ["Tablespace-Speicherplatz" auf Seite 71](#)
- ["Speicherplatzverwaltung" auf Seite 71](#)
- ["Sammeln von Statistiken für Datenbanken" auf Seite 71](#)
- ["Oracle - Automatisierte Statistiksammlung" auf Seite 71](#)
- ["Sammeln von Statistiken für RTSM" auf Seite 72](#)
- ["Indexfragmentierung" auf Seite 72](#)

System Global Area (SGA)

Konfigurieren Sie Ihre SGA immer in Hinblick auf den physischen Speicher und vermeiden Sie Swapping. Es wird empfohlen, dass Sie den SGA nicht auf mehr als 70 Prozent des physischen Systemspeichers

festlegen. So verbleibt noch ausreichend Speicher für zusätzliche System- und Clientprozesse. Die Parameter `MEMORY_TARGET` oder `MEMORY_MAX_TARGET` dürfen nicht größer sein als das Shared Memory-Dateisystem (**/dev/shm**) unter Ihrem Betriebssystem.

Datenbankauslastung - Verhalten

Oracle AWR (Automatic Workload Repository)-Reports können zur Überwachung von Leistungsengpässen und Datenbankverhalten verwendet werden. Weitere Informationen finden Sie im Oracle Metalink-Artikel 276103.1: PERFORMANCE TUNING USING ADVISORS AND MANAGEABILITY FEATURES.

Es wird auch empfohlen, die E/A-Auslastung im System zu überwachen, um E/A-Konflikte zu ermitteln. Nachdem Sie den am stärksten belasteten Datenträger ermittelt haben, können Sie die AWR-Ausgabe verwenden, um festzustellen, welche Oracle-Datendatei die Ursache des Konflikts ist, und gegebenenfalls die E/A-Speicherkonfiguration ändern.

CPU und E/A

Es wird empfohlen, die CPU und das Dateisystem zu überwachen, die die wichtigsten Ressourcen darstellen, auf die der Datenbankserver zugreift. Die CPU-Auslastung darf nicht mehr als 70 Prozent und die E/A-Wartezeit nicht mehr als 10 Prozent betragen.

Die Überwachung der oben genannten Ressourcen können Sie mit **perfmon** unter Windows oder **top** unter Linux sowie den Verwaltungswerkzeugen für das Speichersystem durchführen.

Oracle-Warnungsdatei

Oracle registriert anormale Ereignisse in der Datei **alert.log**, deren Speicherort durch den Parameter `BACKGROUND_DUMP_DEST` definiert wird. Ab Oracle 11g wird der Parameter `BACKGROUND_DUMP_DEST` von `DIAGNOSTIC_DEST` als Teil von Oracle Automatic Diagnostic Repository (ADR) ersetzt.

Es wird empfohlen, diese Datei regelmäßig zu überprüfen, um Anormalitäten zu identifizieren, die korrigiert werden müssen, z. B. ORA-XXXXX-Fehler.

Archivprotokoll - Dateisystem

Bei der Verwendung des Archivprotokoll-Modus überwachen Sie den Speicherort `ARCHIVE_DUMP_DEST` in Hinblick auf die Datenträgernutzung. Diese Dateien müssen regelmäßig gesichert und gelöscht werden, damit ausreichend freier Speicherplatz für neue Archivdateien zur Verfügung steht.

Die Archivdatei hat normalerweise die gleiche Größe wie die Wiederholen-Protokolldatei. Zur Bestimmung der Größe einer Wiederholen-Protokolldatei verwenden Sie die Betriebssystembefehl oder die folgende Abfrage:

```
SQL> select GROUP#, BYTES  
       from V$LOG;
```

Um die Anzahl der Archivdateien festzulegen, die in einem bestimmten Zeitraum erzeugt werden, z. B. in einem Tag, können Sie die folgende Abfrage verwenden, wenn das System stabil ist:

```
SQL> alter session set NLS_DATE_FORMAT = 'DD-MON-YYYY';  
SQL> select FIRST_TIME as "Day",
```

```
COUNT(*) as "Number of files"  
from V$LOG_HISTORY  
group by FIRST_TIME  
order by 1 asc;
```

Tablespace-Speicherplatz

Um Speicherplatzprobleme aufgrund von Datenwachstum zu verhindern, sollten Sie die Auslastung des Tablespace regelmäßig überprüfen.

Wenn in einem der Tablespaces der Speicherplatz knapp wird, können Sie ein oder mehrere Datendateien hinzufügen, indem Sie den Befehl `ALTER TABLESPACE <Tablespace-Name> ADD DATAFILE...` verwenden.

Speicherplatzverwaltung

Freier Speicherplatz in Oracle-Tablespaces besteht aus den neu erstellten Erweiterungen oder dem zuvor belegten Speicherplatz, der durch Operationen wie Aktualisieren und Löschen gewonnen wurde. Wenn ein Teil des freien Speicherplatzes in einem Tablespace aus Erweiterungen besteht, die zuvor belegt waren und dann frei wurden, kann der Tablespace möglicherweise fragmentiert werden. Sie können mit dem Oracle Segment Advisor feststellen, ob Objekte über nicht verwendeten Speicherplatz verfügen, der freigegeben werden kann. Objekte, die fragmentiert wurden, können neu organisiert werden, um verbrauchten Speicher zurückzugewinnen und um ein kompakteres Segment zu bilden. Einen Überblick über die Rückgewinnung von belegtem Speicherplatz finden Sie in der Dokumentation "Oracle Database Administrator's Guide" für Ihre Oracle-Version.

Sammeln von Statistiken für Datenbanken

Die OMi-Plattform wurde für die Zusammenarbeit mit dem Oracle Cost Base Optimizer geplant und erstellt. Damit der Optimizer ordnungsgemäß arbeitet, müssen Sie regelmäßig Statistiken für alle Schematabellen erfassen.

in der ersten Phase der OMi-Bereitstellung wird empfohlen, Statistiken für alle OMi-Objekte (Tabellen und Indizes) zu sammeln.

Beim Arbeiten mit großen OMi-Umgebungen wird empfohlen, Statistiken nur für Objekte zu erfassen, bei denen die Menge der Daten sich im Tagesverlauf erheblich ändert, oder für neu erstellte Objekte (wie die Tabellen und Indizes, die vom Manager für Partitionierung und Purging erstellt wurden). Wenn Ihr OMi-System stabil ist, sollten Sie die Sammlung von Statistiken einmal pro Tag durchführen.

Oracle - Automatisierte Statistiksammlung

Der Job für das automatisierte Sammeln von Optimizer-Statistiken, `GATHER_STATS_JOB`, wird vom Automatic Maintenance Tasks-Framework verwaltet und nicht vom Oracle Scheduler. Automatic Maintenance Tasks sind vordefinierte Aufgaben, die Wartungsarbeiten in der Datenbank ausführen. Diese Aufgaben werden in einem Wartungsfenster ausgeführt, bei dem es sich um ein Zeitintervall handelt, das in einer Periode mit niedriger Systemauslastung auftreten soll. Sie können die

Wartungsfenster manuell basierend auf den Ressourcenauslastungsmustern Ihrer Datenbank anpassen oder bestimmte Standardfenster von der Ausführung ausschließen.

Gemäß der Oracle-Dokumentation ist die bevorzugte Methode zum Verwalten oder Ändern der Konfiguration des Jobs für das automatisierte Sammeln von Optimizer-Statistiken die Verwendung von Automatic Maintenance Tasks-Bildschirmen von Enterprise Manager Database Control und Grid Control. Eine Übersicht über die Automatic Maintenance Tasks finden Sie im "Oracle Database Administrator's Guide".

So erfassen Sie manuell Statistiken für alle OMi-Objekte auf einem Datenbankschema:

1. Melden Sie sich bei dem OMi-relevanten Benutzerschema mit SQL*Plus (einem Drittanbieter Oracle-Client) an.
2. Führen Sie den folgenden Befehl aus:

```
Exec DBMS_STATS.GATHER_SCHEMA_STATS (ownname => '<Name des Oracle-Schemas>',  
options => 'GATHER AUTO');
```

So erfassen Sie Statistiken für bestimmte Schematabellen und deren Indizes:

1. Melden Sie sich bei dem Schema mit SQL*Plus an.
2. Erfassen Sie für jede Tabelle Statistiken durch Ausführen des folgenden Befehls:

```
Exec DBMS_STATS.GATHER_TABLE_STATS (ownname => '<Name des Oracle-Schemas>',  
tabname => '<Name der Tabelle, für die Statistiken erfasst werden sollen>',  
estimate_percent => 5, cascade => TRUE);
```

Hinweis:

- "Cascade => True" weist die Oracle-Datenbank an, alle Indizes in der Tabelle zu analysieren.
- Das Sammeln von Statistiken ist eine ressourcenintensive Operation, die eine Weile dauern kann. Es wird daher empfohlen, die Statistiken während besonderer Wartungsstunden zu sammeln.

Sammeln von Statistiken für RTSM

Im Gegensatz zu einigen Datenbanken, bei denen Abfragen vordefiniert sind und entsprechend der erwarteten Datenbankgröße optimiert werden können, erstellt die RTSM-Datenbank Abfragen dynamisch nach Patternansichten, die für das Datenmodell definiert wurden. Hierzu sind jederzeit genaue Statistiken erforderlich. Zusätzlich zum Ausführen eines täglichen Aktualisierungsjobs für RTSM sollten Sie die Statistiken manuell aktualisieren, wenn größere Änderungen an RTSM-Schemaobjekten durchgeführt wurden, wie Masseneinfügungstransaktionen durch automatisierte DFM-Jobs. DFM ist der verantwortliche Prozess für das automatische Erkennen von Konfigurationselementen und das Einfügen dieser Elemente in RTSM.

Indexfragmentierung

OMi-Schemata bestehen aus B-Tree-Indizes von Oracle zur Verbesserung der Suche in Tabellenspalten.

Es ist empfehlenswert, die Struktur der Schemaindizes in regelmäßigen Abständen (mindestens wöchentlich für aktive Systeme) zu überprüfen und, wenn erforderlich, die fragmentierten Indizes wiederherzustellen.

Die wichtigsten Gründe für die Fragmentierung von Indizes sind:

- **Zeilenlöschungen.** Wenn Zeilen in einer Tabelle gelöscht werden, werden Oracle-Indexknoten nicht physisch gelöscht und die Einträge werden nicht aus dem Index entfernt. Oracle löscht den Indexeintrag eher logisch und es verbleiben "tote" Knoten in der Indexstruktur, wo sie wiederverwendet werden können, wenn ein anderer angrenzender Eintrag erforderlich ist. Wenn jedoch eine große Anzahl an benachbarten Zeilen gelöscht wird, ist es höchst unwahrscheinlich, dass Oracle die gelöschten Blattzeilen weiterverwenden kann. Zusätzlich zur Speicherplatzverschwendung verursachen große Mengen an gelöschten Blattknoten eine Verlangsamung der Indexscans.
Im Lauf der Zeit kann es nach Zeilenlöschungen aus Schematabellen erforderlich sein, einige des Schemaindizes wiederherzustellen.
- **Indexhöhe.** Die Höhe eines Index bezieht sich auf die maximale Anzahl von Ebenen, die im Index festgestellt wurden. Da die Anzahl der Ebenen in einem Index ansteigt, werden mehr Blocklesevorgänge benötigt, wenn der Index durchsucht wird. Wenn einer Tabelle eine große Anzahl an Zeilen hinzugefügt wird, erstellt Oracle möglicherweise weitere Indexebenen für die neue Zeilen. Auf diese Weise kann ein Index vier Ebenen erreichen, allerdings nur in den Bereichen der Indexstruktur, in denen massive Einfügungen aufgetreten sind. Während Oracle-Indizes viele Millionen Einträge in drei Ebenen unterstützen, können Oracle-Indizes, die mindestens vier Ebenen aufweisen, von der Wiederherstellung profitieren.

Bei OMi-Tabellen wird empfohlen, alle Indizes wiederherzustellen, die mehr als drei Ebenen aufweisen.

Dienstprogramm zur Indexwartung

Das OMi-Dienstprogramm zur Indexwartung (**maintain_indexes.bat**) kann verwendet werden, um Indizes zu identifizieren und neu zu erstellen, die mehr als drei Ebenen oder mindestens 100.000 Werte mit einem Anteil von 10 % an gelöschten Werten aufweisen.

Sie können ein Kennzeichen beim Ausführen des Dienstprogramms setzen, mit dem das Dienstprogramm angewiesen wird, Indizes automatisch wiederherzustellen, die fragmentiert wurden. Es wird jedoch empfohlen, Indizes manuell wiederherzustellen.

Bei der Ausführung erzeugt das Dienstprogramm eine Protokolldatei (**index_stats.log**), die die folgenden Einträge enthält:

- Eine alphabetische Liste der Indizes, die als Kandidaten für die Neuerstellung ermittelt wurden. Für jeden aufgeführten Index werden Statistiken wie die Höhe des Index und der Prozentanteil der gelöschten Zeilen angezeigt.
- Befehle zur Wiederherstellung für jeden aufgeführten Index können für das manuelle Wiederherstellen von Indizes verwendet werden.

Das Dienstprogramm erzeugt auch eine Tabelle namens **TEMP_STATS** im Zielschema, die alle Indizes und die zugehörigen Statistiken enthält (nicht nur die Indizes, die als Kandidaten für das Wiederherstellen aufgelistet werden). Der Tabelle verbleibt im Schema, bis sie manuell gelöscht wird, damit die Ergebnisse zu einem späteren Zeitpunkt überprüft werden können.

Warnung: Das Dienstprogramm zur Indexwartung ist ressourcenintensiv, da alle Indizes im Schema analysiert werden. Darüber hinaus kann es zu Sperren von Datenbankobjekten führen oder Indizes überspringen, die von anderen Sitzungen gesperrt werden. Es empfiehlt sich, das Dienstprogramm für die Indexwartung stündlich während der Wartungszeiten auszuführen.

So führen Sie das Dienstprogramm zur Indexwartung aus:

1. Kopieren Sie die folgenden Dateien aus dem Verzeichnis **\<Stammverzeichnis des Gateway-Servers>\AppServer\webapps\site.war\DataBases\ORA_DB_Utills** auf einen Windows-Computer, auf dem der Datenbankclient installiert ist:

- **maintain_indexes.bat**
- **maintain_indexes.sql**

2. Auf dem Computer, auf dem Sie die Dateien kopiert haben, öffnen Sie ein DOS-Befehlsfenster und wechseln in das Verzeichnis, in das Sie die Dateien kopiert haben.
3. Führen Sie das Dienstprogramm zur Indexwartung mit dem folgenden Befehl aus:

```
maintain_indexes.bat <Schema> <Kennwort><DB-Alias> (rebuild flag)
```

Hierbei gilt Folgendes:

- **Schema.** Der Name des Datenbankschemabenedutzers des Schemas, für das Sie das Dienstprogramm ausführen.
- **Kennwort.** Das Kennwort für den Benutzer des Datenbankschemas.
- **DB-Alias.** Der DB-Alias für die Verbindung zur Zieldatenbank wie in der Datei **tnsnames.ora** angegeben. Stellen Sie sicher, dass es in der Datei **tnsnames.ora** einen Eintrag für den Zielservers gibt.
- **rebuild flag.** Das Kennzeichen, mit dem das Dienstprogramm angewiesen wird, Indizes automatisch zu erstellen. Setzen Sie das Kennzeichen auf 0, wenn das Dienstprogramm nicht automatisch Indizes neu erstellen soll, und auf 1, wenn sie wiederhergestellt werden sollen. Die Standardeinstellung ist 0.

Wenn das Dienstprogramm zur Indexwartung beendet ist, überprüfen Sie die Datei **index_stats.log** in dem Verzeichnis, in das Sie die Dateien in Schritt 1 kopiert haben, in Hinblick auf die Liste der Indizes, die Kandidaten für die Wiederherstellung sind, und in Hinblick auf die verwendeten Wiederherstellungsbefehle.

Hinweis: Die Ausführungszeit des Dienstprogramms für die Indexwartung hängt von der Größe der Indizes und der Auslastung auf dem System bei der Ausführung ab.

Oracle-Datenbank - Sicherung und Wiederherstellung

Ihre Sicherungsstrategie wird geprüft, wenn ein Fehler auftritt und Daten verloren gehen. Daten können aus unterschiedlichen Gründen verloren gehen oder beschädigt werden. Beispielsweise bei einem Fehler einer logischen Applikation, bei einem Instanzfehler, der den Start von Oracle verhindert, oder einem Medienfehler, der von einem Festplattenabsturz verursacht wurde. Zusätzlich zu Ihren geplanten Sicherungen, ist es wichtig, eine Sicherung durchzuführen, wenn sich die Datenbankstrukturen ändern (z. B., wenn eine Datendatei zur Datenbank hinzugefügt wurde) oder bevor Sie Software oder Hardware aktualisieren.

Bei der Auswahl einer Sicherungsstrategie sollten mehrere Faktoren berücksichtigt werden, wie z. B. Systembelastung, der Auslastungszeitplan, die Bedeutung der Daten und die Hardwareumgebung der Datenbank.

Oracle-Sicherungen können mithilfe von Skripten durchgeführt werden, die SQL-Befehle kombiniert mit Betriebssystembefehlen zum Kopieren von Dateien ausführen, oder über Oracle RMAN (Recovery Manager)-Befehle.

Die aktualisierten Datensätze von Sicherungen Ihrer Datenbank sollten behalten werden, sodass Sie diese bei Bedarf verwenden können. Wenn Sie RMAN verwenden, stehen Kataloginformationen aus dem Katalog zur Verfügung.

Dieser Abschnitt umfasst die folgenden Themen:

- ["Verfügbare Sicherungsmethoden" unten](#)
- ["Oracle Recovery Manager - RMAN" auf der nächsten Seite](#)

Verfügbare Sicherungsmethoden

In diesem Abschnitt werden die verschiedenen verfügbaren Sicherungsmethoden beschrieben.

Cold Backup

Beim Cold Backup oder der Offline-Sicherung handelt es sich um eine Sicherung auf Datenbankebene. Normalerweise muss die Datenbank heruntergefahren werden, bevor die Sicherung gestartet wird. Die Ausfallzeit hängt von der Datenbankgröße, dem Sicherungsmedium (Datenträger oder Band), der Sicherungssoftware und der verwendeten Hardware ab.

Nachdem die Instanz heruntergefahren wurde, müssen alle Datendateien, Protokolldateien, Steuerungsdateien und Konfigurationsdateien auf eine Festplatte oder ein anderes Medium kopiert werden. Nach dem Kopiervorgang kann die Instanz neu gestartet werden.

Bei dieser Sicherungsmethode kann die Wiederherstellung für einen Zeitpunkt in der Vergangenheit durchgeführt werden, an dem der Datenbank-Snapshot aufgenommen wurde.

Weitere Informationen hierzu finden Sie im "Oracle Backup and Recovery Guide" für Ihre Oracle-Version.

Hot Backup

Das Hot Backup, auch Online-Sicherung genannt, ermöglicht es Ihnen, eine Sicherung durchzuführen, während die Instanz ausgeführt wird und Benutzer mit der Datenbank verbunden sind. Bei dieser Sicherungsmethode handelt es sich um eine Sicherung auf Tablespace-Ebene. Die Datenbank muss im Archivprotokoll-Modus ausgeführt werden, damit Oracle die Änderungen im Zeitverlauf nachverfolgen kann, indem Kopien der Wiederholen-Protokolldateien, sogenannte Archivdateien, erstellt werden. Die erstellten Archivdateien werden in das Archivziel geschrieben, das mit dem LOG_ARCHIVE_DEST (oder LOG_ARCHIVE_DEST_NN)-Parameter in den Instanzparameterdateien angegeben wurde.

Hinweis: Wenn die Oracle Flash Recovery Area verwendet wird, dann wird das Standardziel für die Archivierung mit dem Wert im Parameter `db_recovery_file_dest` angegeben.

Nachdem die Sicherung gestartet wurde, müssen alle Datendateien, Steuerungsdateien, Archivdateien und Konfigurationsdateien auf eine Festplatte oder ein anderes Medium kopiert werden. Diese Methode ermöglicht die Wiederherstellung auf einen beliebigen Zeitpunkt. Beachten Sie, dass bei der Arbeit im Archivprotokoll-Modus zusätzlicher Speicherplatz für die Aufnahme der inkrementellen Archivdateien erforderlich ist, wodurch die Datenbankleistung beeinträchtigt werden kann. Während der Sicherung

kann es in OMi möglicherweise auch zu einer Leistungsver schlechterung aufgrund der Festplattenbelastung kommen.

Weitere Informationen hierzu finden Sie im "Oracle Backup and Recovery Guide" für Ihre Oracle-Version.

Data Pump - Export/Import

Zusätzlich zu den physischen Offline- und Online-Sicherungsmethoden können Sie eine Sicherungsmethode verwenden, die Data Pump genannt wird. Die Data Pump-APIs zum Verschieben von Daten und Metadaten zwischen Datenbanken und Schemas haben die ursprünglichen Export-/Import-Dienstprogramme von Oracle ersetzt.

Mit dem Oracle Data Pump-Dienstprogramm für den Export werden die Schemastruktur und Inhalte in eine oder mehrere binäre Oracle-Dateien, sogenannte Dump-Dateien, gesichert. Mit dieser Methode können Daten zwischen zwei Schemas in derselben Datenbank oder zwischen zwei separaten Oracle-Datenbanken transferiert werden. Um exportierte Daten wieder zurück in die Datenbank zu laden, verwenden Sie das Data Pump-Dienstprogramm für den Import. Weitere Informationen zu den Oracle Data Pump-Dienstprogrammen finden Sie unter "Oracle Database Utilities" in der Oracle-Dokumentation für Ihre Version.

Hinweis: OMi setzt nicht voraus, dass Sie eine bestimmte Methode für die Sicherung verwenden. Ihre Sicherungsmethode sollte jedoch die Verwendung von mehreren OMi-Datenbankbenutzerschemata unterstützen.

Oracle Recovery Manager - RMAN

Recovery Manager (RMAN) ist ein generisches Oracle-Tool zum Sichern und Wiederherstellen Ihrer Zieldatenbank. Wenn Sie RMAN verwenden, können Sie mit dem RMAN-Katalogschema arbeiten. Der Katalog wird innerhalb des Oracle-Schemas verwaltet und speichert Informationen zur registrierten Datenbankstruktur und zu den mit RMAN durchgeführten Sicherungen. Er kann abgefragt werden, um Sicherungs-Reports zu erzeugen und die Verfügbarkeit zu kopieren. Ein einzelner Katalog kann Informationen von einer oder mehreren Zieldatenbanken verwalten.

Der RMAN-Katalog befindet sich im Allgemeinen in einer anderen Datenbankinstanz als die operationale Datenbank und verfügt über eine eigene Sicherungsstrategie. Er muss nur während des Sicherungs- oder Wiederherstellungsprozesses verfügbar sein.

Das RMAN-Tool kann in Verbindung mit Sicherungssoftware von Drittanbietern verwendet werden, um eine vollständige Sicherungs- und Wiederherstellungslösung zu erzielen.

RMAN weist u. a. folgende Vorteile auf:

- Reduziert die gesicherten Daten, indem sie durch das Ausschließen leerer Datenblöcke komprimiert werden; dies spart Zeit und Speicherplatz ein.
- Unterstützt inkrementelle Sicherungen.
- Enthält Funktionen für die Berichterstellung zum Sicherungsstatus.
- Unterstützt parallele Sicherungs- und Wiederherstellungsprozesse, sofern möglich.
- Kann mit einem Sicherungsmedien-Tool von einem Drittanbieter verwendet werden.

Weitere Informationen zu RMAN finden Sie im Oracle Recovery Manager User's Guide für Ihre Version.

Kapitel 12: Konfigurieren des Oracle-Clients für OMi

In diesem Kapitel wird beschrieben, wie Sie den Oracle-Client für die Offline-Dienstprogramme der OMi-Datenbank konfigurieren. Die Installation eines Oracle-Clients auf dem Applikationsserver ist nicht unbedingt erforderlich. Es wird jedoch empfohlen, einen Client für das Ausführen der verschiedenen Offline-Datenbankskripts, wie das manuelle Erstellen von Benutzern oder das Sammeln von Statistiken in OMi-Datenbanken, bereitzustellen.

Oracle-Clientversionen und Betriebssystemplattformen

Die Oracle-Clientversionen und die Betriebssystemplattformen, die für die Zusammenarbeit mit OMi unterstützt und empfohlen werden, sind mit den Oracle-Serverversionen und -Plattformen identisch. Weitere Informationen finden Sie in der Tabelle mit den Oracle Server-Anforderungen unter "[Softwareanforderungen](#)" auf Seite 53.

Oracle-Clientinstallation

Informationen zur Oracle-Clientinstallation finden Sie in der Oracle-Dokumentation.

Wenn Sie die benutzerdefinierte Installation während der Installation auswählen, stellen Sie sicher, dass Sie die folgenden Komponenten (unter **Oracle Client**) installieren:

- Oracle Net (einschließlich TCP/IP-Adapter)
- Oracle-Datenbankdienstprogramme
- SQL*Plus
- Oracle Call Interface (OCI)

Oracle-Clientkonfiguration

Um Omi nutzen zu können, müssen Sie die Datei **tnsnames.ora** konfigurieren, die im Verzeichnis **<ORACLE_HOME>\network\admin** gespeichert ist. Stellen Sie sicher, dass Sie den Namen oder die IP des Oracle Server-Hostcomputers, den Oracle Server-Listener-Port (standardmäßig 1521) und die SID (standardmäßig ORCL) oder den Dienstnamen angeben. Nachfolgend finden Sie ein Beispiel der Datei **tnsnames.ora**.

```
# TNSNAMES.ORA Network Configuration File: D:\oracle\ora81\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

LONDON.MERCURY.CO.IL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = london)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = london)
    )
  )
```

Es wird empfohlen, das Werkzeug **Oracle Net Configuration Assistant** zum Konfigurieren der Datei **tnsnames.ora** zu verwenden. Weitere Informationen finden Sie in der Oracle-Dokumentation.

Stellen Sie sicher, dass die Oracle-Clientkonfiguration, wie z. B. SID und Porteinstellungen mit der Oracle-Serverkonfiguration übereinstimmt. Verwenden Sie zum Testen der Verbindung zwischen dem Oracle-Client- und dem Oracle-Servercomputer das Dienstprogramm **tnsping**.

Wenn Sie eine RAC-Bereitstellung verwenden, können Sie den tnsnames-Eintrag so konfigurieren, dass Ihre RAC unterstützt wird. Ein Beispiel für einen RAC-Eintrag finden Sie unter "[Unterstützung für Oracle Real Applikation Cluster](#)" auf Seite 107.

Hinweis:

- Für den Zugriff auf den Oracle-Server verwenden die OMI-Server den JDBC Thin-Treiber. Der Name des JDBC Thin-Treibers unterstützt keine net*8/9-kompatible Firewall-Verbindung und ermöglicht daher nur die SQL-Datenübertragung.
- Mithilfe der Easy Connect Naming-Methode müssen Namen nicht mehr in den tnsnames.ora-Dateien für TCP/IP-Umgebungen gesucht werden. Weitere Informationen finden Sie im "Oracle Database Net Services Administrator's Guide" in der Oracle-Dokumentation Ihrer Version.

Kapitel 13: Oracle-Prüfliste

Dieses Kapitel enthält eine Prüfliste, die die Anforderungen für den OMi-Support und die -Zertifizierung zusammenfasst.

Weitere Informationen zu den Einstellungen für die Oracle-Datenbankkonfiguration, die bei der Arbeit mit Oracle Server und OMi verwendet werden sollten, finden Sie unter ["Richtlinien für die Oracle Server-Konfiguration und -Dimensionierung"](#) auf Seite 59.

Prüfliste für OMi-Support und Zertifizierung

Die Informationen in diesem Abschnitt beziehen sich auf unterstützte und zertifizierte Oracle-Optionen. Die zertifizierten Optionen werden für das Arbeiten mit OMi empfohlen.

Option	Unterstützt	Empfohlen	Anmerkungen	Weitere Informationen finden Sie unter
Oracle Edition	Enterprise	Enterprise		
Dedizierter OMi-Server	Nicht erforderlich	Nicht erforderlich Es wird empfohlen, eine Instanz für OMi zu benennen.		
Verwendung von mehreren Oracle-Instanzen	Ja	Nein	Die Konfiguration aller Instanzen muss einer zertifizierten Umgebung entsprechen.	"Oracle-Instanzen" auf Seite 53
Verwendung von nicht standardmäßigen Ports	Ja	Ja		
Management rückgängig machen	Automatisch; manuell	Automatisch	Legen Sie den Parameter UNDO_ MANAGEMENT in einer zertifizierten Umgebung auf AUTO fest.	
Verbindungsmethode für gemeinsamen	Ja	Nein	OMi verwendet eine Verbindungspool-Architektur. Verwenden Sie die dedizierten	

Option	Unterstützt	Empfohlen	Anmerkungen	Weitere Informationen finden Sie unter
Oracle-Server			Verbindungsmethode für Server in einer zertifizierten Umgebung.	
Oracle-Replikation	Keine volle Unterstützung	Nein		
Betriebssystem-Dateikomprimierung	Nein	Nein	Nicht von Oracle unterstützt; verursacht abweichendes Verhalten und beeinträchtigt die Leistung.	
Datenbanksteuerungsdateien sind erforderlich	Größer als oder gleich 2	3	Empfiehl sich auf verschiedenen Datenträgern.	
Wiederholen-Protokollgruppen	Größer als oder gleich 3	4	Oracle aktiviert die Software-Spiegelung von Wiederholen-Protokolldateien. Dies wird durch das Erstellen von mindestens zwei Mitgliedern bei dem Wiederholen-Protokoll in jeder Gruppe erzielt. Mitglieder derselben Gruppe sollten sich auf unterschiedlichen Datenträgern befinden.	
Zeichensatz	WE8ISO8859P1; UTF8,AL32UTF8	AL32UTF8		
OPEN_CURSORS	2000	2000		
Arbeiten im Archivprotokollmodus	True; False	True		
Option zum automatischen Erweitern in Tablespace-Dateien	Ja	Nein		
Lokal verwalteter	Ja	Ja		"Lokal"

Option	Unterstützt	Empfohlen	Anmerkungen	Weitere Informationen finden Sie unter
Daten-Tablespace				verwaltete Tablespaces auf Seite 63
Tablespace-Erweiterungsverwaltung	Lokale automatische Erweiterung für Applikations-schemas; lokale einheitliche Erweiterung für TEMP-Tablespace	Lokale automatische Erweiterung für Applikations-daten-Tablespaces; lokale einheitliche Erweiterung für TEMP-Tablespace		"Einstellungen für Oracle-Tablespace" auf Seite 64
Automatic Segment Space Management (ASSM)-Tablespace	Ja	Ja		

Oracle-Clientkonfiguration

Um OMI nutzen zu können, müssen Sie die Datei **tnsnames.ora** konfigurieren, die im Verzeichnis **<ORACLE_HOME>\network\admin** gespeichert ist. Stellen Sie sicher, dass Sie den Namen oder die IP des Oracle Server-Hostcomputers, den Oracle Server-Listener-Port (standardmäßig 1521) und die SID (standardmäßig ORCL) oder den Dienstenamen angeben. Nachfolgend finden Sie ein Beispiel der Datei **tnsnames.ora**.

```
# TNSNAMES.ORA Network Configuration File: D:\oracle\ora81\network\admin\tnsnames.ora
# Generated by oracle configuration tools.

LONDON.MERCURY.CO.IL =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = london)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = london)
    )
  )
```

Es wird empfohlen, das Werkzeug **Oracle Net Configuration Assistant** zum Konfigurieren der Datei **tnsnames.ora** zu verwenden. Weitere Informationen finden Sie in der Oracle-Dokumentation.

Stellen Sie sicher, dass die Oracle-Clientkonfiguration, wie z. B. SID und Porteinstellungen mit der Oracle-Serverkonfiguration übereinstimmt. Verwenden Sie zum Testen der Verbindung zwischen dem Oracle-Client- und dem Oracle-Servercomputer das Dienstprogramm **tnsping**.

Wenn Sie eine RAC-Bereitstellung verwenden, können Sie den tnsnames-Eintrag so konfigurieren, dass Ihre RAC unterstützt wird. Ein Beispiel für einen RAC-Eintrag finden Sie unter "[Unterstützung für Oracle Real Applikation Cluster](#)" auf Seite 107.

Hinweis:

- Für den Zugriff auf den Oracle-Server verwenden die OMi-Server den JDBC Thin-Treiber. Der Name des JDBC Thin-Treibers unterstützt keine net*8/9-kompatible Firewall-Verbindung und ermöglicht daher nur die SQL-Datenübertragung.
- Mithilfe der Easy Connect Naming-Methode müssen Namen nicht mehr in den tnsnames.ora-Dateien für TCP/IP-Umgebungen gesucht werden. Weitere Informationen finden Sie im "Oracle Database Net Services Administrator's Guide" in der Oracle-Dokumentation Ihrer Version.

Oracle-Server und -Clientanforderungen

Informationen zu den Oracle Server- und Client-Versionen und den Betriebssystemplattformen, die für die Arbeit mit OMi unterstützt werden, finden Sie unter "[Softwareanforderungen](#)" auf Seite 53.

Festlegen der Oracle-Initialisierungsparameter

Eine Liste der empfohlenen Einstellungen für Oracle Server-Initialisierungsparametern finden Sie unter "[Oracle-Parametereinstellungen](#)" auf Seite 59.

Teil IV: Bereitstellen und Verwalten der PostgreSQL Server-Datenbanken

Kapitel 14: Übersicht über die PostgreSQL-Bereitstellung

Sie können PostgreSQL für die Bereitstellung von OMi-Datenbanken verwenden. In diesem Kapitel werden die folgenden Themen zur Bereitstellung von PostgreSQL für die Verwendung mit OMi beschrieben:

- ["Informationen zur PostgreSQL-Bereitstellung" unten](#)
- ["Systemanforderungen" auf der nächsten Seite](#)
- ["Zeichensatzunterstützung in PostgreSQL" auf Seite 86](#)

Hinweis: Beim Konfigurieren von OMi können Sie zwischen einer externen PostgreSQL-Datenbank (**Postgres**) und einer eingebetteten PostgreSQL-Datenbank (**PostgresEmbedded**) wählen. Sie können jedoch jederzeit Ihre eingebettete Datenbank in eine externe migrieren. Weitere Informationen finden Sie unter ["Migrieren einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank" auf Seite 95](#).

Wenn Sie die eingebettete PostgreSQL-Datenbank verwenden, können Sie nur einen Datenverarbeitungsserver einrichten. Keine weiteren Datenverarbeitungsserver können einer OMi-Bereitstellung hinzugefügt werden.

Informationen zur PostgreSQL-Bereitstellung

Führen Sie die folgenden Schritte aus, um PostgreSQL für die Verwendung mit OMi bereitzustellen:

- **Installieren und konfigurieren Sie PostgreSQL.**

Weitere Informationen zur Installation der PostgreSQL-Software finden Sie im Installationshandbuch in der Dokumentation für Ihre jeweilige PostgreSQL-Version.

- **Erstellen Sie Datenbanken auf dem PostgreSQL-Server.**

Sie können OMi-Datenbanken manuell erstellen oder den Konfigurations-Assistenten zum Erstellen der Datenbanken verwenden.

Der Konfigurationsassistent fordert Sie zur Eingabe der folgenden Parameter auf:

- **Hostname.** Der Name des Hostcomputers, auf dem PostgreSQL installiert wird.
- **Port.** Der PostgreSQL-Listenerport. OMi zeigt automatisch den Standardport für die externe PostgreSQL-Datenbank, 5432, an.

Hinweis: Wenn Sie die eingebettete PostgreSQL-Datenbank verwenden, verwendet OMi den Port 5433, um die Verbindung zur Datenbank herzustellen.

- **Benutzername und Kennwort.** Der Name und das Kennwort eines Benutzers, der von OMi verwendet wird, um sich mit den OMi-Datenbanken zu verbinden und Daten abzurufen.
- **SSL verwenden?** Wenn diese Option ausgewählt ist, verwendet OMi eine verschlüsselte

Verbindung für die Kommunikation mit dem Datenbankserver.

Hinweis: Wenn Sie SSL verwenden, stellen Sie sicher, dass der PostgreSQL-Server mit aktiviertem SSL gestartet wird und dass ein Clientzertifikat für die Verwendung durch OMi vorhanden ist. Weitere Informationen zum Konfigurieren von PostgreSQL für SSL finden Sie in der PostgreSQL-Dokumentation.

- **Administratorbenutzername und -kennwort.** (Nur beim Erstellen einer neuen Datenbank.) Der Name und das Kennwort eines Benutzers mit Administratorberechtigungen für den PostgreSQL-Server.
- **Datenbanknamen (Management, RTSM und Ereignis).** Die Namen der Datenbanken (z. B. OMi_Management).

Hinweis: Wenn Sie die eingebettete PostgreSQL-Datenbank verwenden, werden Sie aufgefordert, nur das Kennwort eines Benutzers anzugeben, der von OMi zum Verbinden mit den OMi-Datenbanken und zum Abrufen von Daten verwendet wird.

Weitere Informationen zum manuellen Erstellen von Datenbanken für OMi finden Sie unter "[Manuelles Erstellen der OMi-Datenbanken](#)" auf Seite 87. Weitere Informationen zum Konfigurations-Assistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.

Systemanforderungen

In diesem Abschnitt werden die Systemanforderungen für das Arbeiten mit PostgreSQL in Kombination mit OMi beschrieben.

Dieser Abschnitt umfasst die folgenden Themen:

- "[Hardwareanforderungen](#)" unten
- "[Softwareanforderungen](#)" unten

Hardwareanforderungen

Richtlinien für die OMi-Hardware-Dimensionierung finden Sie unter "[Anforderungen](#)" auf Seite 11.

Informationen zu PostgreSQL-Hardwareanforderungen finden Sie in der PostgreSQL-Dokumentation unter:

<http://www.postgresql.org/docs/manuals/>

Softwareanforderungen

Eine Liste der unterstützten PostgreSQL-Datenbankversionen finden Sie in der Tabelle zur OMi-Unterstützung unter folgender Adresse:

<http://support.openview.hp.com/selfsolve/document/KM323488>

Zeichensatzunterstützung in PostgreSQL

Wenn Sie einen neuen PostgreSQL-Cluster in einer PostgreSQL-Datenbank erstellen, können Sie den Zeichensatz während der Erstellung des Clusters durch die Verwendung des Parameters `-E` mit dem Befehl `initdb` angeben:

```
initdb -E UTF8
```

Zum Erstellen einer Datenbank innerhalb eines Clusters mit einem anderen Gebietsschema verwenden Sie einen der folgenden Befehle:

- Über ein Terminal: `createdb -E UTF8 <Datenbankname>`
- In SQL: `CREATE DATABASE <dbname> WITH ENCODING 'UTF8';`

Hinweis: Wenn Sie die Codierung des Datenbankservers überprüfen möchten, führen Sie den Befehl `psql -l` aus oder geben `\l` im `psql`-Terminal ein.

Wenn Sie die PostgreSQL-Clientapplikation oder eine andere Clientapplikation zum manuellen Verbinden der Datenbank verwenden, sollten Sie den richtigen Zeichensatz mit dem folgenden SQL-Befehl angeben:

```
SET NAMES "UTF8";
```

Kapitel 15: Manuelles Erstellen der OMi-Datenbanken

In diesem Kapitel wird die manuelle Erstellung von OMi-Datenbanken auf einem PostgreSQL-Server beschrieben.

Dieses Kapitel umfasst die folgenden Themen:

- ["Übersicht über die PostgreSQL-Datenbank" unten](#)
- ["Manuelles Erstellen von OMi-Datenbanken" unten](#)

Übersicht über die PostgreSQL-Datenbank

OMi verwendet die Management-, RTSM- und Ereignisdatenbanken.

Während der OMi-Konfiguration können neue Datenbanken automatisch durch den Konfigurationsassistenten erstellt werden, oder es können bereits bestehende Datenbanken verwendet werden. Die vorhandenen Datenbanken können entweder manuell im Voraus (z. B. aufgrund von Sicherheitsbeschränkungen im Unternehmen) oder durch eine frühere Installation der gleichen OMi-Version erstellt werden.

Weitere Informationen zum Installieren des OMi-Servers finden Sie im OMi-Installations- und Upgrade-Handbuch.

Manuelles Erstellen von OMi-Datenbanken

In diesem Abschnitt wird beschrieben, wie jede der OMi-Datenbanken manuell erstellt wird:

- ["Manuelles Erstellen einer Management-Datenbank" unten](#)
- ["Manuelles Erstellen einer RTSM-Datenbank" auf der nächsten Seite](#)
- ["Manuelles Erstellen einer Ereignisdatenbank" auf der nächsten Seite](#)

Hinweis: Es wird empfohlen, dass ein zertifizierter Datenbankadministrator die erforderliche Datenbank in Übereinstimmung mit den Datenbankrichtlinien des Unternehmens erstellt.

Manuelles Erstellen einer Management-Datenbank

Zum manuellen Erstellen einer Management-Datenbank führen Sie folgende Schritte aus:

1. Erstellen Sie die Management-Datenbank zum Speichern von Managementobjekten.
2. Registrieren Sie die Datenbankerweiterungen durch Ausführen des folgenden Befehls:

```
<OMi_HOME>/AppServer/webapps/site.war/DataBases/Postgres_DB_Utills/postgres_database_extensions.sql
```

Hinweis: Sie müssen diesen Schritt ausführen, da die Management-Datenbank erfordert, dass die Erweiterung `uuid-oss` registriert wird.

- Erstellen Sie die Managementobjekte wie folgt:
 - Generieren Sie das SQL-Skript der Management-Datenbank mit folgender Skriptdatei:

- Für Windows:**

- `<OMi_STAMMVERZEICHNIS>\dbverify\bin\generate_create_script.bat`

- Für Linux:**

- `/opt/HP/BSM/dbverify/bin/generate_create_script.sh`

Verwenden Sie die folgenden Parameter für die Management-Datenbank:

- deployment_type.** EE (Enterprise Edition)
- Name des generierten Skripts.** Der Name der zu generierenden SQL-Datei, z. B. `management.sql`.
- DB-Typ.** Postgres
- Sitzungstyp.** Management

Beispiel:

```
/opt/HP/BSM/dbverify/bin/generate_create_script.sh EE management.sql  
Postgres Management
```

- Verbinden Sie sich mit der Management-Datenbank, und führen Sie dann das generierte Skript aus, um die Managementobjekte bereitzustellen.
- Führen Sie das folgende SQL-Skript unter `<OMi_Stammverzeichnis>/AppServer/webapps/site.war/DataBases/Postgres_DB_Utills` aus:
`management_postgres_inserts.sql`

Manuelles Erstellen einer RTSM-Datenbank

Erstellen Sie eine Datenbank zum Speichern von RTSM-Objekten. Anders als bei der Management- und der Ereignisdatenbank ist es für die RTSM-Datenbank nicht erforderlich, Skripts auszuführen, um die Objekte zum Füllen der Datenbank zu erstellen.

Manuelles Erstellen einer Ereignisdatenbank

Zum manuellen Erstellen einer Ereignisdatenbank führen Sie folgende Schritte aus:

- Erstellen Sie die Ereignisdatenbank zum Speichern von Ereignisobjekten.
- Registrieren Sie die Datenbankerweiterungen durch Ausführen des folgenden Befehls:

```
<OMi_HOME>/AppServer/webapps/site.war/DataBases/Postgres_DB_Utills/postgres_  
database_extensions.sql
```

Hinweis: Sie müssen diesen Schritt ausführen, da die Ereignisdatenbank erfordert, dass die

Erweiterung uuid-ossd registriert wird.

3. Erstellen Sie die Ereignisobjekte wie folgt:

a. Generieren Sie das SQL-Skript der Ereignisdatenbank mit folgender Skriptdatei:

o **Für Windows:**

```
<OMi_STAMMVERZEICHNIS>\opr\bin\opr-schema-script-generator.bat
```

o **Für Linux:**

```
<OMi_STAMMVERZEICHNIS>/opr/bin/opr-schema-script-generator.sh
```

Legen Sie den Parameter **dbType** auf **Postgres** fest.

Um zum Beispiel für Windows ein Skript mit dem Namen `oprGenerateDbTablesForPsqlServer.sql` zu generieren, das zum Erstellen der erforderlichen Tabellen in der PostgreSQL-Datenbank verwendet werden kann, führen Sie den folgenden Befehl aus:

```
<OMi_STAMMVERZEICHNIS>\opr\bin\opr-schema-script-generator.bat -d Postgres -o C:\temp\oprGenerateDbTablesForPsqlServer.sql
```

b. Verbinden Sie sich mit der Ereignisdatenbank, und führen Sie dann das generierte Skript aus, um die Ereignisobjekte bereitzustellen.

Kapitel 16: Warten einer PostgreSQL-Datenbank

Dieses Kapitel beschreibt die Wartungsverfahren, die für auf PostgreSQL-Servern erstellte OMi-Datenbanken empfohlen werden.

Dieses Kapitel umfasst die folgenden Themen:

- ["Datenbankwartung" unten](#)
- ["Sichern und Wiederherstellen der PostgreSQL-Datenbank" auf der nächsten Seite](#)
- ["Zurücksetzen oder Ändern des Benutzerkennworts für die eingebettete PostgreSQL-Datenbank" auf Seite 93](#)

Datenbankwartung

Für eine optimale Datenbankleistung müssen bestimmte Wartungsaufgaben regelmäßig durchgeführt werden. Eine der erforderlichen Wartungsaufgaben für die PostgreSQL-Datenbank wird **Bereinigung** (Vacuuming) genannt und mit dem Autovacuum-Dämon durchgeführt, einem integrierten PostgreSQL-Werkzeug, das in OMi aktiviert ist und die folgenden sich wiederholenden Wartungsaufgaben durchführt:

- Wiederherstellen des Speicherplatzes nach dem Ausführen der VACUUM-Befehle
- Aktualisieren von Planungsstatistiken

Sie können die Bereinigung der Datenbank auch manuell durchführen. Dies ist besonders nach der Ausführung von zahlreichen INSERT, UPDATE oder DELETE-Anweisungen empfehlenswert. Zu diesem Zweck verwenden Sie das integrierte PostgreSQL-Werkzeug `vacuumdb`, das Sie im folgenden Verzeichnis finden:

```
<OMi_Stammverzeichnis>/pgsql/bin/
```

Hinweis: Das integrierte Werkzeug ist möglicherweise nicht vollständig kompatibel mit der Version des externen PostgreSQL-Datenbankservers. Wenn Sie eine externe PostgreSQL-Datenbank verwenden, sollten Sie daher stattdessen das `vacuumdb`-Werkzeug des Datenbankservers verwenden.

Obwohl Indizes regelmäßig von PostgreSQL gewartet werden, bietet sich manchmal auch eine manuelle Wiederherstellung eines Index an, um die Abfrageleistung zu verbessern (z. B. nach der Ausführung von zahlreichen INSERT, UPDATE oder DELETE-Anweisungen). Dazu verwenden Sie `reindexdb`, ein integriertes Werkzeug zur Neuerstellung von Indizes in einer PostgreSQL-Datenbank.

Hinweis: Durch die Verwendung von `reindexdb` wird möglicherweise der Zugriff auf die Tabelle blockiert. Führen Sie dieses Werkzeug daher nur aus, wenn der Datenbankserver gering belastet wird.

Sichern und Wiederherstellen der PostgreSQL-Datenbank

Eine klar definierte Sicherungs- und Wiederherstellungsstrategie für jede Datenbank ist unabdingbar, da Daten aus verschiedenen Gründen verloren gehen oder beschädigt werden können. Ungeachtet der Ursache für den Datenverlust oder die Beschädigung wird Ihre Sicherungsstrategie, der Schlüssel zur erfolgreichen Wiederherstellung Ihrer Daten, in diesem Fall auf die Probe gestellt. Zusätzlich zu Ihren geplanten Sicherungen ist es wichtig, eine Sicherung jedes Mal durchzuführen, wenn die Datenbankstruktur geändert wird oder die Software oder Hardware aktualisiert wird.

Je nachdem, ob Sie eine integrierte oder externe PostgreSQL-Datenbank verwenden, finden Sie in den folgenden Abschnitten Informationen zum Sichern und Wiederherstellen Ihrer Daten:

- "[Sichern und Wiederherstellen einer eingebetteten PostgreSQL-Datenbank](#)" unten
- "[Sichern und Wiederherstellen einer externen PostgreSQL-Datenbank](#)" auf der nächsten Seite

Sichern und Wiederherstellen einer eingebetteten PostgreSQL-Datenbank

OMi bietet Ihnen das Tool `opr-pgctl`, mit dem Sie Ihre eingebettete PostgreSQL-Datenbank verwalten, sichern und wiederherstellen können. Sie finden Sie dieses Tool an folgendem Speicherort:

- **Windows:** `<OMi_STAMMVERZEICHNIS>\bin\opr-pgctl.bat`
- **Linux:** `<OMi_STAMMVERZEICHNIS>/bin/opr-pgctl.sh`

Das Tool `opr-pgctl` steht nur auf dem Datenverarbeitungsserver zur Verfügung.

Sicherung der eingebetteten PostgreSQL-Datenbank

Für eine vollständige Sicherung der eingebetteten PostgreSQL-Datenbank führen Sie den folgenden Befehl aus:

```
opr-pgctl -backup <Ziel> [-z]
```

In diesem Beispiel ist `<Ziel>` der Name der Sicherungsausgabedatei und Sie müssen sicherstellen, dass er nicht bereits vorhanden ist. Wenn Sie den Parameter `-z` verwenden, wird die Ausgabedatei mit `gzip` komprimiert.

Die Datei, die während des Sicherungsvorgangs erstellt wird, enthält Informationen zur Neuerstellung aller Datenbanken, Benutzerrollen und Gruppen, Zugriffsberechtigungen und Tablespace des eingebetteten PostgreSQL-Datenbankclusters. Sie enthält jedoch keine Informationen zur Konfigurationsdatei (`postgresql.conf`). Wenn Sie das vollständige Cluster-Datenverzeichnis sichern möchten, einschließlich aller Konfigurationsinformationen der Datenbank, können Sie die kontinuierliche Archivierungsmethode verwenden. Informationen hierzu finden Sie in der PostgreSQL-Dokumentation, die unter folgender Adresse verfügbar ist:

<http://www.postgresql.org/docs/manuals/>

Hinweis: Das Sichern der integrierten PostgreSQL-Datenbank erzeugt keine Ausfallzeit für OMi.

Wiederherstellung der eingebetteten PostgreSQL-Datenbank

Um eine eingebettete PostgreSQL-Datenbank mithilfe der Sicherungsdatei wiederherzustellen, die Sie während des Sicherungsvorgangs erstellt haben, führen Sie den folgenden Befehl aus:

```
opr-pgctl -restore <Quelle> [-z]
```

In diesem Beispiel ist <Quelle> der Name der Sicherungsdatei. Stellen Sie sicher, dass Sie den Parameter `-z` angeben, wenn Sie die Sicherungsdatei mit `gzip` komprimiert haben.

Das Wiederherstellen der eingebetteten PostgreSQL-Datenbank wird in der folgenden Reihenfolge durchgeführt:

- Das bestehende Cluster-Datenverzeichnis der eingebetteten PostgreSQL-Datenbank wird entfernt.
- Das Cluster-Datenverzeichnis wird mit den im Konfigurations-Assistenten eingegebenen Anmeldeinformationen neu initialisiert.
- Datenbanken werden mit der angegebenen Sicherungsdatei neu erstellt.

Achtung: Sie müssen OMi beenden, bevor Sie die integrierte PostgreSQL-Datenbank wiederherstellen.

Sichern und Wiederherstellen einer externen PostgreSQL-Datenbank

Beim Sichern einer externen PostgreSQL-Datenbank können Sie eine der folgenden Sicherungsmethoden wählen:

- SQL-Dump
- Dateisystemebene
- Kontinuierliche Archivierung

Dieser Abschnitt umfasst eine kurze Beschreibung aller drei Methoden. Weitere Informationen sowie Vor- und Nachteile der einzelnen Sicherungsmethoden finden Sie in der PostgreSQL-Dokumentation.

SQL-Dump

Bei der Methode SQL-Dump wird eine Datei mit den erforderlichen SQL-Befehlen zum Neuerstellen der Datenbank generiert. Sie können zwischen den folgenden Dienstprogrammen zum Generieren der Sicherungsdatei wählen:

- `pg_dump`

Die Sicherungsausgabedatei enthält Informationen zu den Tabellen einer einzelnen Datenbank. Sie enthält jedoch nicht die Informationen zu allen Datenbanken, Benutzerrollen und Gruppen, Zugriffsberechtigungen und Tablespace, was bedeutet, dass Sie diese Informationen vor der Wiederherstellung Ihrer Datenbank manuell neu erstellen müssen.

- `pg_dumpall`

Die Sicherungsausgabedatei enthält alle Informationen des Datenbankclusters, einschließlich der Informationen zu allen Datenbanken, Benutzerrollen und Gruppen, Zugriffsberechtigungen und Tablespace.

Hinweis: Bei der Sicherungsmethode SQL-Dump werden keine Konfigurationsdateien gesichert. Deshalb müssen Sie sie manuell neu erstellen oder sichern.

Dateisystemebene

Bei dieser Sicherungsmethode kopieren Sie die Dateien, die Ihre PostgreSQL-Datenbank zum Speichern der Daten verwendet. Zu diesem Zweck können Sie Tools wie `copy` oder `tar` verwenden.

Achtung: Stellen Sie sicher, dass Sie den Datenbankserver herunterfahren, bevor Sie die zum Speichern der Daten verwendeten Dateien kopieren.

Kontinuierliche Archivierung

Diese Methode umfasst die kontinuierliche Archivierung der Write Ahead Logs (WALs), in denen alle Änderungen erfasst werden, die an den Datendateien in einer PostgreSQL-Datenbank vorgenommen wurden. Standardmäßig werden diese Protokolldateien in regelmäßigen Abständen mit neuen Informationen überschrieben. Sie können PostgreSQL jedoch auch so konfigurieren, dass Kopien der alten WALs beibehalten werden, sodass es ist möglich, die Datenbank durch Einspielen der entsprechenden Protokolleinträge auf einen ganz bestimmten Zeitpunkt wiederherzustellen.

Zurücksetzen oder Ändern des Benutzerkennworts für die eingebettete PostgreSQL-Datenbank

Führen Sie die folgenden Schritte auf dem Datenverarbeitungsserver aus, um ein verloren gegangenes Benutzerkennwort für die eingebettete PostgreSQL-Datenbank zurückzusetzen oder ein Benutzerkennwort für die Datenbank zu ändern:

1. Beenden Sie OMi.
2. Deaktivieren Sie vorübergehend die Kennwortanforderungen für den PostgreSQL-Administratorbenutzer `postgres`. Bearbeiten Sie hierfür die Datei `<OMi_Stammverzeichnis>/db/pgdata/pg_hba.conf`, indem Sie die folgende Zeile oder Zeilen am Anfang der Datei hinzufügen:
 - **Windows:**

```
host all postgres ::1/128 trust
host all postgres 127.0.0.1/32 trust
```
 - **Linux:**

```
local all postgres trust
```
3. Starten Sie die eingebettete PostgreSQL-Datenbank, indem Sie folgenden Befehl ausführen:

```
<OMi_Stammverzeichnis>/bin/opr-pgctl -start
```

- Führen Sie den folgenden Befehl aus, um eine Verbindung zur Datenbank herzustellen:

- **Windows:**

```
<OMi_Stammverzeichnis>\pgsql\bin\psql.exe -U postgres -p 5433
```

- **Linux:**

```
<OMi_Stammverzeichnis>/pgsql/bin/psql -U postgres -p 5433
```

Hinweis: Sie werden nicht zur Angabe eines Kennworts aufgefordert, da die Kennwortanforderung für den Benutzer `postgres` vorübergehend deaktiviert wurde.

- Ändern Sie das Kennwort für die Datenbankbenutzer `postgres` und `hpbsm` (beide Benutzer müssen das gleiche Kennwort verwenden), indem Sie die folgenden SQL-Befehle in einer offenen PostgreSQL-Clientsitzung ausführen:

```
\password postgres
```

```
\password hpbsm
```

Sie müssen ein neues Kennwort zweimal für jeden Befehl eingeben.

- Beenden Sie die PostgreSQL-Clientsitzung, indem Sie `\q` eingeben.
- Stoppen Sie die eingebettete PostgreSQL-Datenbank, indem Sie folgenden Befehl ausführen:

```
<OMi_Stammverzeichnis>/bin/opr-pgctl -stop
```

- Aktivieren Sie die Kennwortanforderung für den Benutzer `postgres`, indem Sie Folgendes aus der Datei `pg_hba.conf` entfernen:

- **Windows:**

```
host all postgres ::1/128 trust
```

```
host all postgres 127.0.0.1/32 trust
```

- **Linux:**

```
local all postgres trust
```

Nach dem Zurücksetzen des verloren gegangenen Benutzerkennworts müssen Sie OMi neu konfigurieren, damit das neue Kennwort verwendet wird. Führen Sie hierfür auf allen Datenverarbeitungsservern und Gateway-Servern die folgenden Schritte aus:

- Führen Sie den OMi-Konfigurations-Assistenten aus. Informationen zum Konfigurations-Assistenten finden Sie im OMi-Installations- und Upgrade-Handbuch.
- Wählen Sie **Verbindung zu vorhandenem Datenbank- oder Benutzerschema herstellen** aus.
- Wählen Sie **Eingebettete Postgres** als Datenbanktyp aus.
- Geben Sie Ihr neues Kennwort ein.

Kapitel 17: Migrieren einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank

In diesem Kapitel wird beschrieben, wie die Migration der eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank durchgeführt wird.

Dieses Kapitel umfasst die folgenden Themen:

- ["Gründe für die Migration Ihrer eingebetteten PostgreSQL-Datenbank" unten](#)
- ["Verfahren zum Migrieren einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank" unten](#)

Gründe für die Migration Ihrer eingebetteten PostgreSQL-Datenbank

Der eingebettete PostgreSQL-Datenbankserver, der mit OMi bereitgestellt wird, ermöglicht eine schnelle und einfache Konfiguration und reicht für kleine Umgebungen vollkommen aus. Wenn die verwaltete Umgebung jedoch komplexer wird, kann es nötig werden, zu einem dedizierten externen PostgreSQL-Datenbankserver zu wechseln.

Die externe PostgreSQL-Datenbank eignet sich besser für komplexe Umgebungen, da der Administrator eine bessere Kontrolle über die Konfiguration des Datenbankservers hat, sodass die Datenbank besser an die Bedürfnisse der OMi-Umgebung angepasst werden kann. Bei einer externen Datenbank ist es auch möglich, Replikationen und den Lastenausgleich zu konfigurieren, wodurch sich die Leistung und die Verfügbarkeit des Datenbankservers stark erhöht und Ausfallzeiten bei Fehlern reduziert werden.

Hinweis: Bei einer externen PostgreSQL-Datenbank ist es die Aufgabe des Datenbankadministrators, die Datenbank zu warten und für eine gute Reaktionsfähigkeit und Leistung zu sorgen.

Verfahren zum Migrieren einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank

Um von einer eingebetteten PostgreSQL-Datenbank in eine externe PostgreSQL-Datenbank zu migrieren, führen Sie folgende Schritte aus:

1. Erstellen Sie eine Sicherung der bestehenden eingebetteten PostgreSQL-Datenbank gemäß der Beschreibung unter ["Sichern und Wiederherstellen einer eingebetteten PostgreSQL-Datenbank"](#)

auf Seite 91.

2. Erstellen Sie die OMi-Datenbanken auf dem externen PostgreSQL-Datenbankcluster neu, indem Sie folgenden Befehl ausführen:

```
psql -f <Sicherungsdatei> -h <DB-Host> -p <DB-Port> -U <Admin-Benutzer>
```

In diesem Fall ist <Sicherungsdatei> der Name der Sicherungsdatei, die Sie während des Sicherungsverfahrens erstellt haben. Wenn Sie den Befehl `opr-pgctl` mit dem Parameter `-z` verwendet haben, stellen Sie sicher, dass Sie die Sicherungsdatei manuell mithilfe von Tools wie 7-Zip oder `gzip` entpacken.

Hinweis: Das Werkzeug `psql` wird mit OMi bereitgestellt und ist im Verzeichnis `<OMi_Stammverzeichnis>/pgsql/bin` verfügbar. Wenn Sie jedoch eine externe PostgreSQL-Datenbank verwenden, wird nachdrücklich empfohlen, stattdessen das Tool des Datenbanksservers zu verwenden.

3. Konfigurieren Sie OMi neu, damit eine Verbindung zur bereits bestehenden PostgreSQL-Datenbank hergestellt werden kann.

Die alten OMi-Datenbanken werden entfernt und die folgenden Datenbanken werden auf dem angegebenen externen PostgreSQL-Datenbankcluster erstellt:

- `mgmt`
- `rtsm`
- `event`

Teil V: Anhänge

Anhang A: Empfehlungen zum Microsoft SQL Server-Datenspeicher

Diese Anlage enthält Empfehlungen für das Erstellen von Microsoft SQL Server-Dateigruppen, in denen OMi-Daten gespeichert werden sollen. Diese Informationen gelten für nur an erfahrene Benutzer.

Dieser Anhang enthält:

["Erstellen von Microsoft SQL Server-Dateigruppen" unten](#)

Erstellen von Microsoft SQL Server-Dateigruppen

Wenn OMi-Datenbanken automatisch vom Konfigurations-Assistenten erstellt werden, speichert OMi Datenbankobjekte in zwei Dateigruppen: In der Dateigruppe **PRIMÄR** für Systemtabellen und in der **userdata001**-Dateigruppe für Benutzertabellen. Wenn Sie eine große Bereitstellung von OMi einrichten, wird empfohlen zusätzliche Dateigruppen zu erstellen, in denen OMi-Datenbankobjekte gespeichert werden.

Vorteile von mehreren Dateigruppen

Das Platzieren des Systemkatalogs in einer eigenen Dateigruppe und das Erstellen von mehreren benutzerdefinierten Dateigruppen für die Datenspeicherung hat die folgenden Vorteile:

- **Wartung.** Das Sichern und Wiederherstellen von Datenbanken kann auf Dateigruppenniveau durchgeführt werden.
- **Teilweise Wiederherstellung.** Sie können eine Datenbank auf Dateigruppenniveau teilweise wiederherstellen. In diesem Fall müssen Sie jedoch auch die Dateigruppe **PRIMÄR** wiederherstellen, da sie den Systemkatalog enthält. Wenn Sie nur die **.mdf**-Datei in der Dateigruppe **PRIMÄR** behalten und eine andere Benutzerdateigruppe als Standarddateigruppe einrichten, bleibt die Dateigruppe **PRIMÄR** klein und stellt kein Problem bei einer teilweisen Wiederherstellung dar. Die vorgeschlagene Dateigruppenstruktur erstellt eine einzelne Benutzerdateigruppe. Hierbei handelt es sich um einen vorbereitender Schritt für mehrere Dateigruppen, die vom Datenbankadministrator oder von OMi zukünftig erstellt werden können.
- **Schreibgeschützt.** Eine Dateigruppe kann als schreibgeschützt markiert werden. Da die Daten nicht geändert werden können, muss Microsoft SQL Server keine freigegebenen Sperren erwerben und die Leistung wird verbessert.
- **Wiederherstellen.** Wenn der Datenteil einer Datenbank ausfällt, können Sie die Änderungen, die vor dem Absturz durchgeführt wurden, speichern, auch wenn die **.mdf**-Datei beschädigt wurde. Voraussetzung dafür ist, dass die Protokolldatei und die Masterdatenbank unbeschädigt sind.
- **Leistung.** Es stehen zahlreiche Optionen zur Leistungsoptimierung zur Verfügung, wie z. B. schnell wachsende Tabellen auf schnellen Festplatten, die Trennung von Tabellendaten und Indizes in unterschiedlichen Dateigruppen usw.
- **Flexibilität und Kontrolle.** Die Verwendung von mehreren Dateigruppen bietet dem Datenbankadministrator mehr Flexibilität und Kontrolle. Darüber hinaus verhindert der Einsatz einer

festen Größe für das Datenbankwachstum Situationen, in denen das standardmäßige Inkrement von 10 Prozent für automatisches Wachstum in großen Datenbanken nicht vor der Ausfallzeiteinstellung von OMi-Komponenten beendet werden kann und die Datenbankaktivität rückgängig gemacht wird, wodurch eine Endlosschleife von Einfügungen verursacht wird.

Sie können eine zusätzliche Dateigruppe erstellen (zusammen mit einer weiteren Microsoft SQL Server-Datenbank), indem Sie die folgenden Skripts ausführen.

So erstellen Sie eine Datenbank mit Standardeinstellungen:

```
CREATE DATABASE [testdb]
```

So ändern Sie den SIZE-Parameter für die .mdf-Datei (Katalog):

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb',
    SIZE = 5MB
)
```

So ändern Sie den FILEGROWTH-Parameter für die .mdf-Datei (Katalog):

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb',
    FILEGROWTH = 5MB
)
```

So ändern Sie den SIZE-Parameter für die .ldf-Datei (Protokoll):

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb_log',
    SIZE = 10MB
)
```

So ändern Sie den FILEGROWTH-Parameter für die .ldf-Datei (Protokoll):

```
ALTER DATABASE [testdb]
MODIFY FILE
(
    NAME = N'testdb_log',
    FILEGROWTH = 50MB
)
```

So erhalten Sie den Pfad für das Standarddatenverzeichnis der Datenbank:

```
SELECT LEFT(filename,
    LEN(filename) - CHARINDEX(N'\', REVERSE(filename))) AS path
FROM master.dbo.sysdatabases
WHERE name = N'testdb'
```

Verwenden Sie den Namen von **'testdb'**, der im obigen Skript zurückgegeben wurde, im folgenden Skript.

So erstellen Sie eine Benutzerdateigruppe, fügen eine Datendatei hinzu und legen diese als Standarddateigruppe fest:

```
ALTER DATABASE [testdb]
ADD FILEGROUP USERDATA001

ALTER DATABASE [testdb]
ADD FILE
(
    NAME = N'testdb_Data001',
    FILENAME = N'<Pfad zum Datenverzeichnis>',
    SIZE = 10MB,
    FILEGROWTH = 50MB
)
TO FILEGROUP USERDATA001
```

So legen Sie die Dateigruppe userdata001 als Standarddateigruppe fest:

```
ALTER DATABASE [testdb] MODIFY FILEGROUP USERDATA001 DEFAULT
```

Hinweis:

Beim Erstellen von Dateigruppen sind folgende Konfigurationsregeln wichtig für die ordnungsgemäße Wiederherstellung von Dateien, für die Fehlertoleranz, die Leistung und die Archivierung:

- Es wird empfohlen, zwei eigene Datendateien in zwei getrennten Dateigruppen zu erstellen: Die **.mdf**-Datei in der Dateigruppe **PRIMÄR** und die **.ndf** in der **userdata001**-Gruppe.
- Die Gruppe **userdata001** sollte als Standarddateigruppe festgelegt werden.
- Die Datei **.mdf** sollte die einzige Datei in der Dateigruppe **PRIMÄR** sein, die den Systemkatalog enthält. Die Datei **.mdf** muss mit einem ursprünglichen **SIZE**-Parameterwert von 5 MB und einem **FILEGROWTH**-Parameterwert von 5 MB erstellt werden. Die andere Datendatei (**.ndf**) und Protokolldatei (**.ldf**) sollte mit einem anfänglichen **SIZE**-Parameterwert von 10 MB und einem **FILEGROWTH**-Parameterwert von 50 MB erstellt werden. In allen Fällen muss der **FILEGROWTH**-Parameterwert eine feste Größe (kein Prozentsatz) sein.

Anhang B: Microsoft SQL Server-Konfigurationsoptionen

Die meisten Serverkonfigurationsoptionen werden dynamisch durch Microsoft SQL Server konfiguriert. Für die OMi-Zertifizierung dürfen Sie die Standardoptionen nicht ändern, sofern Sie nicht durch den HP Software Support dazu aufgefordert werden.

In bestimmten Situationen kann eine Änderung der Standardeinstellungen wünschenswert sein. Sie können diese Einstellungen in der gespeicherte Prozedur **sp_configure** oder in den verschiedenen Dialogfeldern in Microsoft SQL Management Studio (vor allem im Dialogfeld **Servereigenschaften**) ändern .

Die Standardwerte sollten für alle Elemente mit Ausnahme der folgenden Elemente verwendet werden:

- Agent XPs (A) = 1
- awe aktiviert (A) = 1. Dieser Wert sollte nur geändert werden, wenn mehr als 4 GB auf einem 32-Bit-Computer erforderlich sind. Beachten Sie, dass diese Funktion ab Microsoft SQL Server 2012 nicht mehr verfügbar ist.

Konfigurationsoption	Standardwert für SQL Server 2012 und 2014	Standardwert für SQL Server 2008 und 2008 R2	OMi-Zertifizierung
access check cache bucket count (A)	0	0	Standardwert
access check cache quota(A)	0	0	Standardwert
ad hoc distributed queries(A)	0	0	Standardwert
affinity I/O mask (A, RR)	0	0	Standardwert
affinity64 I/O mask (A, nur auf der 64-Bit-Version von SQL Server verfügbar)	0	0	Standardwert
affinity mask (A)	0	0	Standardwert
affinity64 mask (A, RR), nur auf der 64-Bit-Version von SQL Server verfügbar)	0	0	Standardwert
Agent XPs (A)	0 (Wird beim Start von SQL Server Agent auf 1 geändert. Der Standardwert ist 0, wenn für den SQL Server Agent	0 (Wird beim Start von SQL Server Agent auf 1 geändert. Der Standardwert ist 0, wenn für den SQL Server Agent	1

Konfigurationsoption	Standardwert für SQL Server 2012 und 2014	Standardwert für SQL Server 2008 und 2008 R2	OMi-Zertifizierung
	beim Setup der automatische Start festgelegt ist)	beim Setup der automatische Start festgelegt ist)	
allow updates (Veraltet. Nicht verwenden. Führt zu einem Fehler während der erneuten Konfiguration.)	0	0	Standardwert
awe enabled (A, RR)		0	Standard, sofern nicht mehr als 4 GB Arbeitsspeicher auf einem 32 Bit-Computer erforderlich ist. Diese Funktion in Microsoft SQL Server 2012 nicht mehr unterstützt.
backup compression default	0	0	Standardwert
blocked process threshold(A)	0	0	Standardwert
c2 audit mode (A, RR)	0	0	Standardwert
clr enabled	0	0	Standardwert
common criteria compliance enabled (A, RR)	0	0	Standardwert
contained database authentication	0	n/v	Standardwert
cost threshold for parallelism (A)	5	5	Standardwert
cross db ownership chaining	0	0	Standardwert
cursor threshold (A)	-1	-1	Standardwert
Database Mail XPs (A)	0	0	Standardwert
default full-text language(A)	1033	1033	Standardwert
default language	0	0	Standardwert
default trace enabled (A)	1	1	Standardwert
disallow results from triggers (A)	0	0	Standardwert

Konfigurationsoption	Standardwert für SQL Server 2012 und 2014	Standardwert für SQL Server 2008 und 2008 R2	OMi-Zertifizierung
EKM provider enabled	0	0	Standardwert
filestream_access_level	0	0	Standardwert
fill factor (A, RR)	0	0	Standardwert
ft crawl bandwidth (max), siehe ft crawl bandwidth(A)	100	100	Standardwert
ft crawl bandwidth (min), siehe ft crawl bandwidth(A)	0	0	Standardwert
ft notify bandwidth (max), siehe ft notify bandwidth(A)	100	100	Standardwert
ft notify bandwidth (min), siehe ft notify bandwidth(A)	0	0	Standardwert
index create memory (A, SC)	0	0	Standardwert
in-doubt xact resolution (A)	0	0	Standardwert
lightweight pooling (A, RR)	0	0	Standardwert
locks (A, RR, SC)	0	0	Standardwert
max degree of parallelism(A)	0	0	Standardwert
max full-text crawl range(A)	4	4	Standardwert
max server memory (A, SC)	2147483647	2147483647	Standardwert
max text repl size	65536	65536	Standardwert
max worker threads (A)	0 Null legt automatisch die Anzahl der maximalen Arbeitsthreads abhängig von den Prozessoren fest. Grundlage ist die Formel $(256 + (<Prozessoren> - 4) * 8)$ für 32-Bit-SQL Server. Für 64-Bit-SQL Server wird der doppelte Wert	0 Null legt automatisch die Anzahl der maximalen Arbeitsthreads abhängig von den Prozessoren fest. Grundlage ist die Formel $(256 + (<Prozessoren> - 4) * 8)$ für 32-Bit-SQL Server. Für 64-Bit-SQL Server wird der doppelte Wert	Standardwert

Konfigurationsoption	Standardwert für SQL Server 2012 und 2014	Standardwert für SQL Server 2008 und 2008 R2	OMi-Zertifizierung
	verwendet.	verwendet.	
media retention (A, RR)	0	0	Standardwert
min memory per query (A)	1024	1024	Standardwert
min server memory (A, SC)	0	0	Standardwert
nested triggers	1	1	Standardwert
network packet size (A)	4096	4096	Standardwert
Ole Automation Procedures(A)	0	0	Standardwert
open objects (A, RR, veraltet)	0	0	Standardwert
optimize for ad hoc workloads (A)	0	0	Standardwert
PH_timeout (A)	60	60	Standardwert
precompute rank (A)	0	0	Standardwert
priority boost (A, RR)	0	0	Standardwert
query governor cost limit(A)	0	0	Standardwert
query wait (A)	-1	-1	Standardwert
recovery interval (A, SC)	0	0	Standardwert
remote access (RR)	1	1	Standardwert
remote admin connections	0	0	Standardwert
remote login timeout	10	20	Standardwert
remote proc trans	0	0	Standardwert
remote query timeout	600	600	Standardwert
Replication XPs Option (A)	0	0	Standardwert
scan for startup procs (A, RR)	0	0	Standardwert
server trigger recursion	1	1	Standardwert
set working set size (A, RR, veraltet)	0	0	Standardwert
show advanced options	0	0	Standardwert

Konfigurationsoption	Standardwert für SQL Server 2012 und 2014	Standardwert für SQL Server 2008 und 2008 R2	OMi-Zertifizierung
SMO und DMO XPs (A)	1	1	Standardwert
SQL Mail XPs (A)	n/v	0	Standardwert
transform noise words (A)	0	0	Standardwert
two digit year cutoff (A)	2049	2049	Standardwert
user connections (A, RR, SC)	0	0	Standardwert
User Instance Timeout (A, nur in SQL Server 2008 Express.)	n/v	60	Standardwert
user instances enabled (A, nur in SQL Server 2008 Express.)	n/v	0	Standardwert
user options	0	0	Standardwert
xp_cmdshell (A)	0	0	Standardwert

Anhang C: Ändern des Datenbank-Hosts

In diesem Abschnitt werden die Verfahren für die Migration des OMi-Datenbankschemas auf einen anderen Server beschrieben.

1. Halten Sie alle OMi-Server an. Dies ist erforderlich, um Datenverlust und falsche Zuordnungen der Schemastruktur in der Zielumgebung zu vermeiden.

2. Duplizieren der OMi-Schemas, die Sie migrieren möchten.

Dies kann über eine Offlinesicherungskopie der Datenbank oder einen Datenbankexport ausgeführt werden. Weitere Informationen finden Sie je nach Datenbanktyp unter den folgenden Abschnitten:

- Microsoft SQL: "[Sichern von Datenbanken](#)" auf Seite 39
- Oracle: "[Oracle-Datenbank - Sicherung und Wiederherstellung](#)" auf Seite 74
- PostgreSQL: "[Sichern und Wiederherstellen der PostgreSQL-Datenbank](#)" auf Seite 91

3. Ändern Sie die Sitzungstabelle in der Verwaltungsdatenbank gemäß der neuen Konfiguration.
4. Führen Sie den Konfigurationsassistenten aus und geben Sie den Speicherort des neuen Datenbankhosts auf der Seite **Datenbankeinstellungen** an.
5. Starten Sie alle OMi-Server.

Anhang D: Unterstützung für Oracle Real Applikation Cluster

In diesem Anhang wird die Konfiguration beschrieben, die für die Zusammenarbeit von HP Operations Manager i mit Oracle Real Applikation Cluster erforderlich ist. Diese Informationen gelten für nur an erfahrene Benutzer.

Dieser Anhang enthält:

- ["Informationen über Oracle Real Applikation Cluster \(RAC\)" unten](#)
- ["Single Client Access Name" auf der nächsten Seite](#)
- ["Clientseitige Konfiguration für Oracle RAC" auf der nächsten Seite](#)
- ["Serverseitige Konfiguration" auf Seite 111](#)
- ["Erstellen von oder Verbinden mit einer der OMi-Datenbanken" auf Seite 112](#)

Informationen über Oracle Real Applikation Cluster (RAC)

Ein Cluster ist eine Sammlung von miteinander verbundenen Servern, die für Endbenutzer und Applikationen wie ein Server angezeigt werden. Oracle Real Applikation Cluster (RAC) ist die Oracle-Lösung für hohe Verfügbarkeit, Skalierbarkeit und Fehlertoleranz. Es basiert auf Clusterservern, die denselben Speicher verwenden.

Oracle RAC ist eine einzelne Oracle-Datenbank, die auf einem Cluster von Hardwareserver installiert wurde. Jeder Server führt eine Instanz der Datenbank aus und alle Instanzen greifen auf dieselben Datenbankdateien zu.

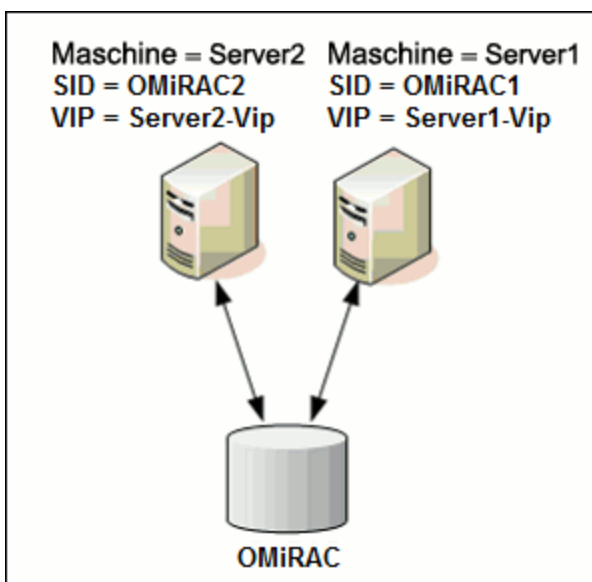
Weitere Informationen über Oracle RAC finden Sie im "Oracle Clusterware Guide" und im "Oracle Real Applikation Clusters Administration and Deployment Guide" in der Oracle-Dokumentation Ihrer Version.

In diesem Anhang wird das folgende Oracle RAC-Beispiel verwendet:

- Oracle RAC-Datenbankname: OMiRAC
- Computernamen: Server1, Server2
- Auf jedem Computer gibt es eine Oracle-Instanz von OMiRAC:
 - SID auf Server1: OMiRAC1
 - SID auf Server2: OMiRAC2
- Auf jedem Computer ist eine virtuelle IP vorhanden (Server1-VIP und Server2-VIP):
 - Server1-VIP wird Server1 zugewiesen
 - Server2-VIP wird Server2 zugewiesen

Die virtuelle IP wird dem Computer zusätzlich zur statischen IP zugewiesen.

- Die Listener auf beiden Servern hören den Standardport 1521 ab und unterstützen den Datenbankservice OMiRAC.



Single Client Access Name

In Release 11g hat Oracle den Single Client Access Name (SCAN) als bevorzugte Methode des Zugriffs für Clientverbindungen mit dem RAC eingeführt. Bei dieser Methode müssen Clients nicht einzelne Knoten im RAC konfigurieren, sondern sie verwenden eine einzelne virtuelle IP, die SCAN oder SCAN VIP genannt wird.

SCAN ist ein einzelner Netzwerkname, der für das Cluster im DNS (Domain Name Server) Ihrer Organisation oder im GNS (Grid Naming Service) definiert wird, der zwischen verschiedenen IP-Adressen rotiert und mehrere Listener im Cluster berücksichtigt. Mit dem SCAN erübrigt es sich, den Client zu ändern, wenn Knoten zum Cluster hinzugefügt oder von dort entfernt werden.

Der SCAN und die zugehörigen IP-Adressen bieten einen stabilen Namen für Clients, die für Verbindungen verwendet werden, und zwar unabhängig von den Knoten, aus denen sich der Cluster zusammensetzt. Die SCAN-Adressen des Datenbankservers, virtuelle IP-Adressen und öffentliche IP-Adressen müssen sich auf demselben Teilnetz befinden.

Hinweis: Bei der Verwendung von Oracle 11g RAC wird empfohlen, die SCAN-Methode zu verwenden.

Clientseitige Konfiguration für Oracle RAC

OMi verwendet die JDBC-Treiber von DataDirect für die Verbindung zu regulären Oracle-Datenbanken und RAC-Datenbanken von Oracle.

Führen Sie die folgenden Änderungen an den OMi-Konfigurationsdateien durch, bevor Sie die Verwaltungsdatenbank erstellen oder eine Verbindung zu einer RAC-Datenbank von Oracle herstellen:

1. Erstellen Sie auf allen Servern die Datei `<OMI_HOME>\conf\OMI-tnsnames.ora`.

Das Format der `omi-tnsnames.ora` ist identisch mit `tnsnames.ora`:

```
<DB-Servicename> =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <Virtuelle IP der ersten Instanz> )
    (PORT = <Listener-Port der ersten Instanz>))
    (ADDRESS = (PROTOCOL = TCP)(HOST = <Virtuelle IP der zweiten Instanz> )(PORT
= <Listener-Port der zweiten Instanz>))
    (... entry for each instance...)
    (LOAD_BALANCE = on)
    (FAILOVER = on)
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = <DB-Servicename>)
  )
)
```

Hierbei gilt Folgendes:

- **<DB-Servicename>** ist der Name eines Service, der den Listener unterstützt. Es ist derselbe Name, der im `CONNECT_DATA`-Teil verwendet wird.
- **ADDRESS_LIST** enthält einen Adresseintrag für jeden Knoten in der RAC-Umgebung. Wenn Oracle SCAN verwendet, enthält es nur die virtuelle SCAN-IP. Die Adresse enthält alle erforderlichen Details für die Verbindung mit dem Knoten:
 - **HOST** enthält die virtuelle-IP für diese Instanz. Es ist für eine schnellere Fehlererkennung wichtig, die virtuelle IP zu verwenden und nicht die statische IP des Knotens.
 - **PORT** ist der Port auf dem der Listener für das Abhören des speziellen Knotens konfiguriert ist.
- Ist **FAILOVER** auf **on** gesetzt, kann der Treiber nach dem Fehlschlagen einer Verbindung versuchen, eine Verbindung zu einem anderen Knoten herzustellen. Die Verbindungsversuche werden fortgesetzt, bis eine Verbindung zu einem der Knoten erfolgreich hergestellt ist oder einer der Knoten erreicht werden kann.

Ein Versuch, eine Verbindung zu einem anderen Knoten herzustellen erfolgt erst, wenn die Verbindung zum aktuellen Knoten fehlschlägt. Wenn die Verbindung vom Knoten verweigert wird (z. B. wurde die Kommunikation mit dem Knoten hergestellt, aber die Verbindung wurde zurückgewiesen), wird nicht versucht, eine Verbindung zu einem anderen Knoten herzustellen.

Wichtig: Failover ist nur für Verbindungsversuche gültig. Wenn eine Verbindung während einer Datenbanktransaktion fehlschlägt, gibt es keinen Failover der Transaktion auf einen anderen Computer, der die Transaktion fortsetzen soll.

- Wird **LOAD_BALANCE** auf **on** gesetzt, muss der Treiber Verbindungsanforderungen zwischen den Knoten verteilen, um eine Überlastung von einzelnen Knoten zu vermeiden. Die Reihenfolge,

in der der Treiber auf die Knoten zugreift, ist zufällig.

- **SERVER** ist der zu verwendende Verbindungsmodus. Es gibt zwei mögliche Verbindungsmodi: **dedicated** und **shared**. Konfigurieren Sie dies entsprechend der unterstützten Serverkonfiguration.
- **SERVICE_NAME** ist der Name eines Services, den die Listener unterstützen. Es ist derselbe Name, den Sie bei **<DB-Servicename>** angegeben haben.

Im obigen Beispiel würde **omi-tnsnames.ora** folgendermaßen konfiguriert:

```
OMIRAC =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = Server1-Vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = Server2-Vip)(PORT = 1521))
    (LOAD_BALANCE = on)
    (FAILOVER = on)
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = OMIRAC)
  )
)
```

Hinweis: Wenn OMI-Benutzerschemas in mehr als einer Oracle-Datenbank oder einem Service bereitgestellt werden, darf die Datei **omi-tnsnames.ora** nur einen Eintrag für jede Datenbank oder jeden Datenbankservice enthalten. Wenn sich alle Schemata unter demselben RAC-Service befinden, ist nur ein Eintrag erforderlich.

2. Bearbeiten Sie auf allen Servern die Datei **<OMI_STAMMVERZEICHNIS>\conf\jdbc.drivers.properties**.

- a. Suchen Sie den Abschnitt **#ddoracle**. In diesem Abschnitt, gibt es eine Zeile für **ddoracle.url**, die der folgenden Zeile ähnelt:

```
ddoracle.url=jdbc:mercury:oracle://${host}:${port};sid=${sid}
```

- b. Ersetzen Sie diese Zeile mit einer der folgenden Zeilen (abhängig vom Betriebssystem):

- **Windows.**

```
ddoracle.url=jdbc:mercury:oracle:TNSNamesFile=<OMI_
STAMMVERZEICHNIS>\\conf\\omi-tnsnames.ora;TNSServerName=${sid}
```

Dabei korreliert **\$(sid)** mit dem Servicennamen, der im Schritt der Service- und Datenbankkonfiguration eingegeben wurde. Dieser Parameter wird pro OMI-Schema gespeichert. Wenn mehrere Servicennamen im System vorliegen, müssen Sie überprüfen, ob **omi-tnsnames.ora** einen Eintrag für jeden Servicennamen enthält.

Beachten Sie, dass jeder umgekehrte Schrägstrich (\) im Pfad von **<OMI_STAMMVERZEICHNIS>** verdoppelt wird.

- **Linux.**

```
ddoracle.url=jdbc:mercury:oracle:TNSNamesFile=<OMi_
STAMMVERZEICHNIS>/conf/omi-tnsnames.ora;TNSServerName=${sid}
```

3. **Öffnen Sie auf allen Servern das Verzeichnis <OMi_STAMMVERZEICHNIS>\odb\conf. Suchen Sie die Datei jdbc.properties.**

- a. Suchen Sie die Zeile beginnend mit **cmdb.url**.
- b. Ersetzen Sie diese mit der folgenden Zeile:

```
cmdb.url = jdbc:mercury:oracle:TNSNamesFile=<OMi_HOME>\\conf\\omi-
tnsnames.ora;TNSServerName=<SERVICENAME>
```

wobei <SERVICE NAME> der Eintrag in der Datei **omi-tnsnames.ora** ist, der dem Namen des RAC-Dienstes entspricht.

- c. Wenn die Datei nicht vorhanden ist, erstellen Sie eine leere **jdbc.properties**-Datei im oben genannten Ordner und fügen Sie den folgenden Eintrag hinzu:

```
Oracle = ddoracle
cmdb.url =
jdbc:mercury:oracle:TNSNamesFile=<OMi_HOME>\\conf\\omi-
tnsnames.ora;TNSServerName=<SERVICENAME>
```

wobei <SERVICE NAME> der Eintrag in der Datei **omi-tnsnames.ora** ist, der dem Namen des RAC-Dienstes entspricht.

- d. Wenn Ihr Server unter dem Linux-Betriebssystem ausgeführt wird, ersetzen Sie alle doppelten umgekehrten Schrägstriche mit einzelnen Schrägstrichen.

Serverseitige Konfiguration

In Oracle RAC gleichen die Oracle-Listener immer die Verbindung zwischen den Knoten nach den folgenden Algorithmen aus:

- **Lastbasiert (Standard).** Der Listener leitet die Verbindung entsprechend der Länge der Warteschlange auf den Knoten um. Der am wenigsten CPU-weise belastete Knoten wird verbunden.
- **Sitzungsbasiert.** Der Listener gleicht die Anzahl der Sitzungen zwischen den Knoten aus.

Der erste Algorithmus wird für kurzlebige Verbindungen und weniger für langlebige Verbindungen optimiert (wie die mit den Verbindungspools verwendeten).

Der gesamte Lastenausgleich wird während der Verbindungszeit und nicht danach durchgeführt. Dies bedeutet, dass eine hergestellte Verbindung nicht auf einen anderen Knoten verschoben wird.

Empfehlungen für OMi-Datenbanken

Es wird empfohlen, den sitzungsbasierten Algorithmus zu verwenden, da OMi Verbindungspools verwendet. Eine Verbindung kann zu unterschiedlichen Zwecken verwendet werden und hat eine lange Lebensdauer.

Damit ein Oracle-Listener den sitzungsbasierten Algorithmus verwenden kann, muss ein Listenerparameter zur Parameterdatei **Listener.ora** jedes Listeners hinzugefügt werden (jeder Knoten hat einen eigenen Listener, sodass diese Änderung für alle Listener durchgeführt werden muss):

```
PREFER_LEAST_LOADED_NODE_<LISTENERNAME> =OFF
```

Hierbei gilt Folgendes:

<LISTENERNAME> ist der Name des Listeners. Der Standardwert ist LISTENER_<Knotenname>.

Beispiel:

Die standardmäßige Listenername auf **Server1** ist **LISTENER_SERVER1**. In diesem Fall müssen Sie Folgendes zur Datei **listener.ora** von **Server1** hinzufügen:

```
PREFER_LEAST_LOADED_NODE_LISTENER_SERVER1=OFF
```

Erstellen von oder Verbinden mit einer der OMi-Datenbanken

Beim Erstellen eines neuen Datenbankschemas oder beim Herstellen einer Verbindung mit einem vorhandenen Schema im Konfigurations-Assistenten legen Sie alle Parameter wie gewohnt fest. Eine Ausnahme bilden die folgenden Parameter:

Parameter	Benötigter Wert
Oracle-Computer	Eine der virtuelle IP-Adressen. Die virtuelle SCAN-IP kann verwendet werden.
Port	Der lokale Listenerport auf dem Oracle-Computer oder der Port des SCAN-Listeners.
SID	Der Servicename der Datenbank. Hinweis: Der Name muss identisch mit <SERVICENAME> sein, den Sie in der Datei omitsnames.ora eingegeben haben.

In diesem Beispiel lauten die Parameter wie folgt:

Parameter	Wert
Oracle-Computer	Server1-Vip
Port	1521
SID	OMiRAC

Senden von Feedback zur Dokumentation

Wenn Sie Anmerkungen zu diesem Dokument haben, können Sie sich per E-Mail [an das Dokumentationsteam wenden](#). Ist auf Ihrem System ein E-Mail-Client installiert, klicken Sie auf den Link oben und ein E-Mail-Fenster mit folgendem Betreff wird geöffnet:

Feedback zu Datenbankhandbuch (Operations Manager i 10.00)

Geben Sie Ihr Feedback ein und klicken Sie auf **Senden**.

Wenn kein E-Mail-Client verfügbar ist, kopieren Sie die oben genannten Informationen in eine neue Nachricht in einem Web-E-Mail-Client und senden Sie Ihr Feedback an ovdoc-asm@hp.com.

Ihr Feedback ist uns willkommen!



[Gehe zu OMi!](#)