

Das SAP Sperrkonzept (BC-CST-EQ)



Release 4.6C



Copyright

© Copyright 2001 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Software-Produkte können Software-Komponenten auch anderer Software-Hersteller enthalten.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] und SQL Server[®] sind eingetragene Marken der Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®] und OS/400[®] sind eingetragene Marken der IBM Corporation.

ORACLE[®] ist eine eingetragene Marke der ORACLE Corporation.

INFORMIX[®]-OnLine for SAP und Informix[®] Dynamic Server[™] sind eingetragene Marken der Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®] und Motif[®] sind eingetragene Marken der Open Group.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT[®] ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo und mySAP.com sind Marken oder eingetragene Marken der SAP AG in Deutschland und vielen anderen Ländern weltweit. Alle anderen Produkte sind Marken oder eingetragene Marken der jeweiligen Firmen.

Symbole

Symbol	Bedeutung
	Achtung
	Beispiel
	Hinweis
	Empfehlung
	Syntax

Inhalt

Das SAP Sperrkonzept (BC-CST-EQ)	5
Funktionsweise des R/3-Sperrkonzepts	6
Das Eigentümerkonzept	10
Die Sperrtabelle	13
Kollisionen von Sperren	16
Kumulation von Sperren.....	19
Fragen und Antworten zum Thema Sperren	21
Die wichtigsten Profilparameter zum Sperrkonzept.....	25
Sperreinträge verwalten	27
Sperreinträge auswählen und anzeigen	29
Sperreinträge löschen	32
Sperrverwaltung testen	35
Sperrstatistik anzeigen	36
Probleme analysieren und beheben.....	38

Das SAP Sperrkonzept (BC-CST-EQ)

Einsatzmöglichkeiten

Das SAP-System verfügt über einen eigenen Sperrmechanismus, um Zugriffe auf Daten in der Datenbank zu synchronisieren. Es muß verhindert werden, daß zwei Transaktionen parallel dieselben Daten in der Datenbank ändern können.

Einführungshinweise

Sperren sind in generischer Form im Data Dictionary als "Sperrobjekte" definiert. Ein *Sperreintrag* ist eine Ausprägung eines Sperrobjekts und sperrt ein bestimmtes Datenobjekt, beispielsweise einen Korrekturauftrag oder einen Tabelleneintrag.

Normalerweise werden Sperreinträge gesetzt und automatisch gelöscht, wenn Benutzerprogramme auf ein Datenobjekt zugreifen und es anschließend freigeben.

Integration

Der SAP-Sperrmechanismus hängt eng mit der [Verbuchung im R/3 \[Extern\]](#) zusammen. Der Umgang mit Sperrobjekten ist in der ABAP Dictionary Dokumentation unter [Sperrobjekte \[Extern\]](#) beschrieben.

Die ABAP-Dokumentation erläutert die wichtigsten Dinge über das Sperrkonzept im Hinblick auf die Programmierung von ABAP-Transaktionen im Kapitel [Das R/3-Sperrkonzept \[Extern\]](#).

Funktionsumfang

Sie können die [Sperrverwaltung \[Seite 27\]](#) (Transaktion SM12) einsetzen, um Sperreinträge zu prüfen und zu löschen, falls der SAP-Dispatcher, das Betriebssystem oder die Netzwerkverbindung vorzeitig abbricht und der Dispatcher Sperreinträge nicht löschen kann. In diesem Fall bleiben ungültige Sperreinträge wirksam und sperren beim Neustart des Systems den Zugriff auf die gesperrten Daten.

Hintergründe zum besseren Verständnis des R/3-Sperrkonzepts sind unter [Funktionsweise des SAP-Sperrkonzepts \[Seite 6\]](#) beschrieben.

Die wichtigsten Profilparameter zum R/3-Sperrkonzept finden Sie unter [hier \[Seite 25\]](#). Sie dienen dazu, die Ressourcen Ihres Systems optimal für Ihre Bedürfnisse nutzen zu können.

Funktionsweise des R/3-Sperrkonzepts

Funktionsweise des R/3-Sperrkonzepts

In diesem Kapitel wird erklärt, wie der R/3 Sperrmechanismus implementiert ist. Es bietet Hintergrundwissen, das dazu dient, die Sperrverwaltung besser zu verstehen und anwenden zu können. Die konkreten Möglichkeiten zum Umgang mit R/3-Sperren ist unter [Sperrereinträge verwalten \[Seite 27\]](#) beschrieben.

R/3 Sperrlogik

Der Programmierer einer R/3-Transaktion muß, wenn die Transaktion Datenbankobjekte verändern will, diese vorher gegen konkurrierenden Zugriff sperren und anschließend wieder freigeben.

Hierzu muß er im Data Dictionary ein Sperrobjekt *definieren* (siehe [Sperrobjekte \[Extern\]](#) in der Data Dictionary Dokumentation) und anschließend *aktivieren*. Das Aktivieren bewirkt, daß das System zwei Funktionsbausteine generiert: einen für das Sperren des Objekts, einen für das Freigeben. Dies ist im Detail unter [Sperrmechanismus \[Extern\]](#) in der ABAP Dictionary Dokumentation beschrieben.

Sperrserver

In einem verteilten R/3-System gibt es einen einzigen Sperrserver (auch *Enqueue-Server* genannt), der die [Sperrtabelle \[Seite 13\]](#) verwaltet. Dieser läuft auf der Zentralinstanz.

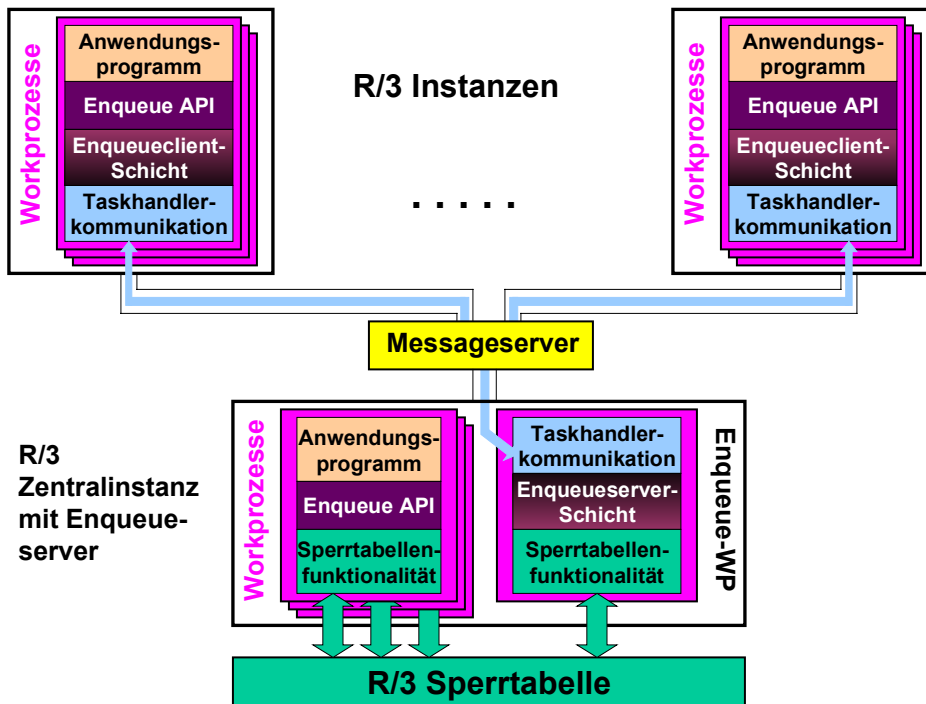
Soll in einer Applikation, die auf einer anderen als der Zentralinstanz (u. U. auf einem anderen Rechner) läuft, eine Sperre abgesetzt werden, so geht die Sperranforderung über den lokalen [Dispatcher \[Extern\]](#) und den [Message-Server \[Extern\]](#) zu dem Dispatcher der Zentralinstanz, der die Anfrage dann an den Enqueue-Workprozeß weitergibt. Dieser schaut in der Sperrtabelle nach, ob die Sperranforderung mit einer schon gesetzten Sperre kollidiert (siehe dazu [Kollisionen von Sperren \[Seite 16\]](#)). In diesem Fall wird die Anforderung zurückgewiesen, ansonsten wird die Sperre gesetzt, es wird der entsprechende Eintrag in der Sperrtabelle vorgenommen.



Die Workprozesse auf der Zentralinstanz haben direkt Zugriff auf die Sperrtabellenfunktionalität. Sie müssen ihre Sperranforderungen nicht via Dispatcher und Messageserver versenden.

Die folgende Grafik zeigt den Kommunikationsweg in einem verteilten R/3 System mit Zentralinstanz und weiteren Instanzen.

Enqueue-Kommunikation



Sperrungen und R/3 Verbuchung

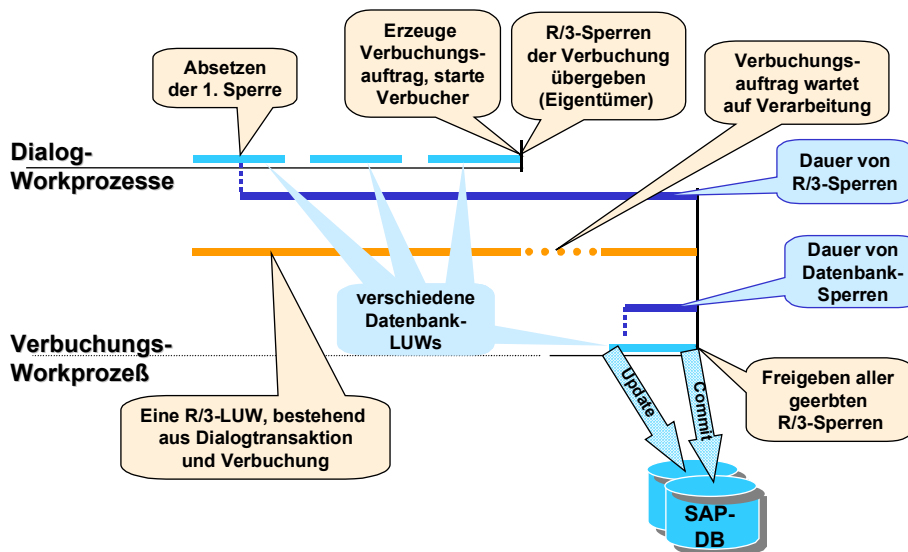
Die Sperre wird im Laufe der Transaktion an die [Verbuchung im R/3 \[Extern\]](#) übergeben. Wie das genau funktioniert, ist unter [Das Eigentümerkonzept \[Seite 10\]](#) und [Funktionsbausteine für Sperranforderungen \[Extern\]](#) beschrieben. Sperrungen, die an die Verbuchung übergeben wurden, werden auch zusätzlich zur Sperrtabelle in einer Backup-Datei gespeichert, so daß sie nicht verlorengehen, wenn der Enqueue-Server ausfällt. In der Sperrverwaltung ist dann das Backup-Flag gesetzt.

SAP-Sperrungen und Datenbanksperren

Die folgende Grafik zeigt den Zusammenhang zwischen SAP-Sperrungen und Datenbanksperren. Ziel ist natürlich, die Zeitspanne einer Datenbanksperre zu minimieren. Des weiteren kann eine SAP-Sperre über mehrere Datenbank-LUWs hinweg bestehen, eine Datenbanksperre hingegen nicht. Der Unterschied zwischen SAP-LUW und Datenbank-LUW ist unter [Funktionsweise der Verbuchung \[Extern\]](#) beschrieben.

Funktionsweise des R/3-Sperrkonzepts

R/3 Transaktionskonzept: Datenbank-Lock-Zeit minimieren



Die oberste (unterbrochene) blaue Linie deutet eine R/3 Dialogtransaktion an, die aus drei Dynpros (Eingabefenster) besteht; hierbei entspricht jedes Dynpro einer Datenbank-LUW (vgl. [Funktionsweise der Verbuchung \[Extern\]](#)). Nach einer Eingabe des Benutzers endet jeweils die Datenbank-LUW (hellblaue Linie).

Nach der Benutzereingabe übernimmt wieder ein Dialogworkprozeß die Bearbeitung. Nach der 2. Benutzereingabe wird die Bearbeitung abgeschlossen und der Dialogteil der SAP-LUW beendet.



Hierbei muß es nicht ein und derselbe Dialogworkprozeß sein, der die Transaktion abarbeitet. Bei jedem Dynpro wird vom Dispatcher ein freier Workprozeß gesucht, der die Bearbeitung durchführt.

In diesem Beispiel wurde im ersten Bild der Transaktion eine R/3-Sperre angefordert – diese besteht solange (obere dunkelblaue Linie), bis die Anwendungsdaten in der Datenbank geändert sind, d.h. in den meisten Fällen, bis die R/3 Verbuchung abgeschlossen ist. Dies ist keine Performance-Einbuße, da es sich nicht um eine Datenbanksperre handelt.

Die Datenbanksperre (durch die untere dunkelblaue Linie dargestellt) besteht nur während der Datenbank-LUW, in der die im R/3 vorgenommenen Änderungen tatsächlich verbucht werden (siehe [Funktionsweise der Verbuchung \[Extern\]](#)).

Details zur Funktionsweise des Sperrkonzepts finden Sie unter

[Das Eigentümerkonzept \[Seite 10\]](#)

[Die Sperrtabelle \[Seite 13\]](#)

[Kollisionen von Sperren \[Seite 16\]](#)

[Kumulation von Sperren \[Seite 19\]](#)

[Fragen und Antworten zum Thema Sperren \[Seite 21\]](#)

Das Eigentümerkonzept

Das Eigentümerkonzept

Zu Beginn einer R/3-Transaktion werden immer zwei *Eigentümer* angelegt, die Sperren anfordern können.

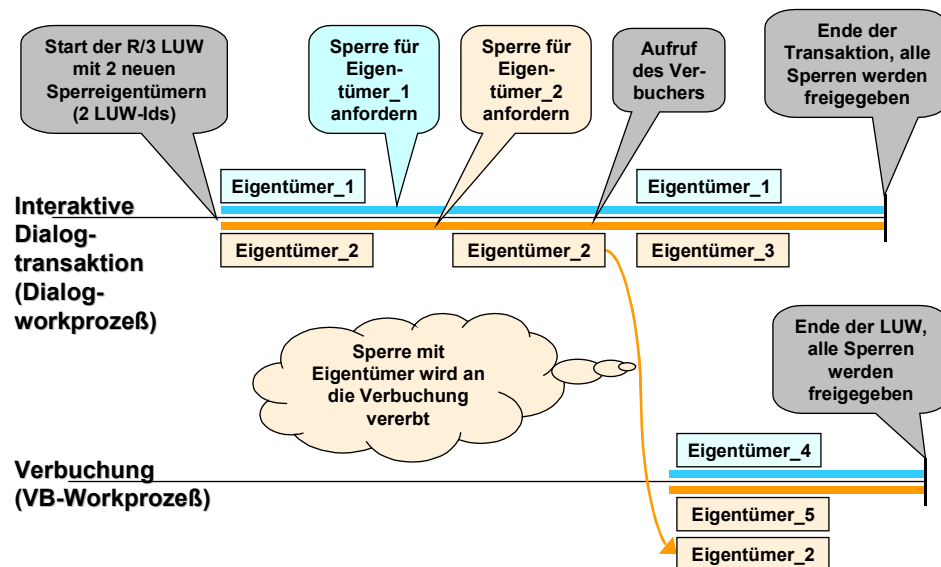
Ein Eigentümer wird durch seine Eigentümer-Id bestimmt, die im Kapitel [Die Sperrtabelle \[Seite 13\]](#) beschrieben ist.

Eine Sperre kann einen oder zwei Eigentümer haben. Dies wird vom ABAP-Programmierer mit dem `_SCOPE`-Parameter (siehe [Funktionsbausteine für Sperranforderungen \[Extern\]](#)) bestimmt.

Das folgende Diagramm zeigt beispielhaft eine SAP-LUW aus Sicht der Sperren.



SAP-LUW: Dialogtransaktion mit Verbuchung



Mit Beginn der Dialogtransaktion werden vom System zwei Sperreigentümer angelegt: der *Dialogeigentümer* Eigentümer_1 und der *Verbuchungseigentümer* Eigentümer_2.

Dann fordert im Laufe der Transaktion der Eigentümer_1 eine Sperre an, etwas später Eigentümer_2. Beim Aufruf des Verbuchers (siehe dazu [Funktionsweise der Verbuchung \[Extern\]](#)) wird die Sperre mit Eigentümer_2 an den Verbucher vererbt. Ein Verbuchungs-Workprozeß wird wie ein Dialogworkprozeß mit zwei Eigentümern gestartet und hat dann bis zum Ende der Verbuchung drei Eigentümer. Spätestens am Ende der Verbuchung werden alle Sperren mit einem impliziten `DEQUEUE_ALL` freigegeben.

Die folgende Abbildung zeigt die Sperren im Verlauf einer SAP-LUW im Zusammenhang mit dem `_SCOPE`-Parameter. Dieser dient dem Anwendungsprogrammierer dazu, festzulegen, welchem der zwei Eigentümer die Sperre wirklich gehören soll. Es wird dargestellt, wie lange die R/3-

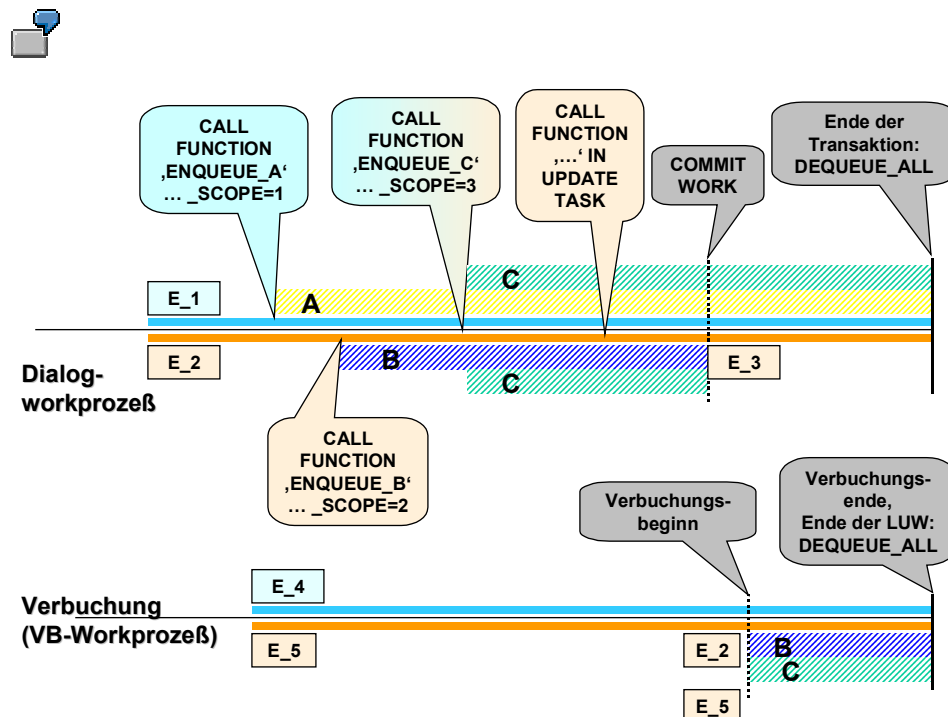
Das Eigentümerkonzept

Sperren aktiv sind (das sind nicht die tatsächlichen Datenbanksperren, wie unter [Funktionsweise des R/3-Sperrkonzepts \[Seite 6\]](#) beschrieben).

`_SCOPE` kann die Werte 1,2 oder 3 annehmen:

<code>_SCOPE = 1</code>	Die Sperre gehört nur Eigentümer_1, bleibt also nur in der Dialogtransaktion bestehen
<code>_SCOPE = 2</code>	Die Sperre gehört nur Eigentümer_2, wird also an die Verbuchung vererbt
<code>_SCOPE = 3</code>	Die Sperre gehört Eigentümer_1 und Eigentümer_2

Die Sperre besteht solange, bis entweder der entsprechende `DEQUEUE`-Funktionsbaustein aufgerufen wird oder (wie im Bild dargestellt) die Transaktion mit einem impliziten `DEQUEUE_ALL` endet.



In diesem Beispiel wird also im Laufe der Transaktion das Sperrojekt A (das der Programmierer vorher im ABAP Dictionary [angelegt \[Extern\]](#) hat) durch den Funktionsaufruf `CALL FUNCTION 'ENQUEUE_A'` gesperrt. Durch Setzen des `_SCOPE`-Parameters auf 1 ~~musste, daß die Sperre A~~ weitergegeben wird (sie gehört nur dem Dialogeigentümer E_1) und damit mit dem Funktionsaufruf `DEQUEUE_A` oder spätestens mit dem Ende der Dialogtransaktion aufgehoben wird.

Später werden die Sperre B, die E_2 (dem Verbuchungseigentümer) gehört (`_SCOPE=2`) und die Sperre C, die zwei Eigentümer hat (`_SCOPE=3`), angefordert. Mit dem Aufruf `CALL FUNCTION '...' IN UPDATE TASK` wird ein [Verbuchungsauftrag \[Extern\]](#) generiert. Beim `COMMIT WORK` wird der Verbucher aufgerufen, der die Sperren und den Verbuchungseigentümer der Sperren B und C

Das Eigentümerkonzept

erbt. Mit dem Ende der Verbuchung werden diese Sperren freigegeben. Es ist aber möglich, daß die Sperre **C** des Dialogeigentümers noch länger besteht (je nach Programmierung der Transaktion).

Die Dialogsperren **A** und **C** bestehen noch bis zum Ende der Dialogtransaktion.

Siehe auch:

[Die Sperrtabelle \[Seite 13\]](#)

[Kollisionen von Sperren \[Seite 16\]](#)

[Kumulation von Sperren \[Seite 19\]](#)

Die Sperrtabelle

Definition

Die Sperrtabelle ist eine Tabelle im Hauptspeicher des Enqueue-Servers, in der festgehalten wird, welche Sperren derzeit bestehen. Für jede bestehende Elementarsperre gibt sie die Eigentümer, den Sperrmodus, den Namen und die Felder der gesperrten Tabelle an.

Verwendung

Die Sperrtabelle dient zur Verwaltung der Sperren. Bei jeder Sperranfrage, die den Enqueue-Server erreicht, wird in der Sperrtabelle nachgesehen, ob die Sperranfrage mit einer bestehenden Sperre kollidiert (siehe [Kollisionen von Sperren \[Seite 16\]](#)). Wenn ja, wird die Anfrage zurückgewiesen, wenn nein, wird die neue Sperre in die Sperrtabelle geschrieben.

Struktur

Jede Elementarsperre, die derzeit gesetzt ist, entspricht einem Datensatz in der Sperrtabelle. Die folgende Grafik zeigt die Struktur der Sperrinträge.

Eigentümer_1	Eigentümer_2	Backup-Id	Elementarsperre		
			Sperrmodus	Name	Argument
•Eigentümer-Id •Kumulationszähler	•Eigentümer-Id •Kumulationszähler	•Backup-Id •Flag	•X, E oder S	•Name der gesperrten Tabelle	•Sperrargumente
⋮	⋮	⋮	⋮		

Hierbei haben die einzelnen Felder die folgende Bedeutung:

Feld	Inhalt und Bedeutung
------	----------------------

Die Sperrtabelle

Eigentümer_1	<i>Eigentümer-Id</i> und <i>Kumulationszähler</i> des Eigentümer_1: die Id enthält den Rechnernamen, den Workprozeß, einen Zeitstempel und wird auch verwendet, um die SAP-LUW zu identifizieren. Der Kumulationszähler gibt an, wie oft der Eigentümer diese Elementarsperre schon gesetzt hat.	
Eigentümer_2	analog für Eigentümer_2	
Backup-Id	Backup-Id (Index, wo der Sperreintrag in der Backup-Datei gespeichert ist) und Backup-Flag (0 (kein Backup) oder 1 (Backup)).	
Elementarsperre	Sperrmodus	S (Shared lock, Lesesperre) E (Exclusive lock, Schreibsperre) X (eXclusive lock, erweiterte Schreibsperre, nicht kumulierbar)
	Name	Name der Datenbanktabelle, in der Felder gesperrt werden sollen
14		April 2001

Die Sperrtabelle

	Argument	gesperrte Felder in der Datenbanktabelle (aneinandergehängte Schlüsselfelder, kann auch Wildcards enthalten)
--	----------	--

Integration

Zu Diagnosezwecken kann man sich die Sperreinträge ansehen. Dies ist unter [Sperreinträge verwalten \[Seite 27\]](#) beschrieben.

Kollisionen von Sperren

Kollisionen von Sperren

Die Überprüfung, ob eine Sperranforderung mit einer bestehenden Sperre kollidiert, erfolgt in zwei Schritten: zuerst wird überprüft, ob die Sperranfrage mit einer Elementarsperre in der [Sperrtabelle \[Seite 13\]](#) kollidiert. Ist dies der Fall, wird überprüft, ob eine *Eigentümer-Kollision* vorliegt. (Derselbe Eigentümer darf ja beispielsweise eine Schreibsperre mehrmals anfordern. Dies ist unter [Kumulation von Sperren \[Seite 19\]](#) beschrieben.)

Im Kollisionsfall erhält der Benutzer der Dialogtransaktion die Nachricht, daß das gewünschte Objekt gerade durch einen anderen Benutzer gesperrt ist. Bei Nicht-Dialog-Prozessen (etwa im Batch-Input) wird die Sperranforderung zu einem späteren Zeitpunkt wieder gestellt.

Kollision von Elementarsperren

Zwei Elementarsperren kollidieren, wenn alle folgenden Bedingungen erfüllt sind:

- *Name* der Elementarsperre (Tabelle, in der gesperrt werden soll) ist gleich
- *Sperrargument* ist gleich, genauer: die Buchstaben stimmen an jeder Position überein (der Wildcardbuchstabe, hier mit @ gekennzeichnet, ist zu allen Buchstaben gleich)
- mindestens eine Elementarsperre hat nicht den *Sperrmodus S* (Lesesperre)

in der folgenden Grafik sind einige Beispiele zusammengestellt, die zeigen, wann eine Sperranfrage (links) mit einer bestehenden Sperre (rechts) kollidiert.



Kollision von Elementarsperren – Beispiele

	Elementarsperre – Anfrage				bestehende Sperre			
	Sperrmodus	Name	Argument		Sperrmodus	Name	Argument	
1)	E	TAB1	ABCD	↔	E	TAB1	ABCD	
2)	E	TAB2	ABCD	↔	E	TAB1	ABCD	✓
3)	S	TAB1	ABCD	↔	S	TAB1	ABCD	✓
4)	E	TAB1	ABCD	↔	S	TAB1	ABCD	
5)	E	TAB1	AB@@	↔	E	TAB1	ABCD	
6)	E	TAB1	ABCD	↔	E	TAB1	AB@@	
7)	E	TAB1	@@CD	↔	E	TAB1	AB@@	
8)	E	TAB1	@@CDE	↔	E	TAB1	AB@@	✓

1. Schreibsperre **ABCD** auf Tabelle **TAB1** kollidiert mit vorhandener Schreibsperre **ABCD** auf Tabelle **TAB1**

Kollisionen von Sperren

2. Schreibsperre **ABCD** auf Tabelle **TAB2** kollidiert nicht mit vorhandener Schreibsperre **ABCD** auf Tabelle **TAB1**, wird also akzeptiert, weil Sperrname verschieden
3. Lesesperre **ABCD** auf Tabelle **TAB1** kollidiert nicht mit vorhandener Lesesperre **ABCD** auf Tabelle **TAB1**, wird also akzeptiert, weil beides Lesesperren sind
4. Schreibsperre **ABCD** auf Tabelle **TAB1** kollidiert mit vorhandener Lesesperre **ABCD** auf Tabelle **TAB1**
5. Schreibsperre **AB@@** auf Tabelle **TAB1** kollidiert mit vorhandener Schreibsperre **ABCD** auf Tabelle **TAB1**, weil @=C und @=D gilt
6. Schreibsperre **ABCD** auf Tabelle **TAB1** kollidiert mit vorhandener generischer Schreibsperre **AB@@** auf Tabelle **TAB1**, weil C=@ und D=@ gilt
7. Schreibsperre **@@CD** auf Tabelle **TAB1** kollidiert mit vorhandener Schreibsperre **AB@@** auf Tabelle **TAB1**, weil @=A und @=B und C=@ und D=@ gilt
8. Schreibsperre **@@CDE** auf Tabelle **TAB1** kollidiert nicht mit vorhandener Schreibsperre **AB@@** auf Tabelle **TAB1**, weil der 5. Buchstabe nicht übereinstimmt (E ungleich _)

Kollidieren die Elementarsperren nicht, so wird die Sperranfrage als neuer Eintrag in die Sperrtabelle aufgenommen. Kollidieren die Elementarsperren, so wird auf eine *Eigentümerkollision* geprüft (im folgenden beschrieben).

Eigentümerkollision

Im Fall einer Kollision von Elementarsperren erfolgt die Entscheidung, ob die Sperranfrage angenommen oder abgewiesen wird, also aufgrund der Eigentümer, die zu den Sperren gehören (siehe [Das Eigentümerkonzept \[Seite 10\]](#)).

Eine Eigentümerkollision liegt vor, wenn bei vorliegender Elementarsperrenkollision eine der folgenden Bedingungen zutrifft:

- mindestens ein Eigentümer unterscheidet sich
- die Eigentümer sind gleich, aber mindestens eine Sperre hat den Modus X (erweiterte Schreibsperre, keine Kumulation, siehe dazu [Sperrmodus \[Extern\]](#))

Die folgende Grafik zeigt beispielhaft einige Situationen: Hierbei ist O_x ein Eigentümer ungleich O_1 und O_2.



Kollisionen von Sperren

Eigentümer-Kollision – Beispiele

	Elementarsperre – Anfrage			↔	bestehende Sperre			
	Sperr-modus	Eigen-tümer_1	Eigen-tümer_2		Sperr-modus	Eigen-tümer_1	Eigen-tümer_2	
1)	E	O_1	O_2		E		O_2	✓
2)	E	O_1	O_2		E		O_x	✗
3)	E	O_1	O_2		E	O_x		✗
4)	E	O_1	O_3		E	O_1	O_2	✗
5)	E	O_1	O_3		E	O_1		✓
6)	X	O_1	O_2		E	O_1	O_2	✗
7)	S	O_1	O_2		X	O_1	O_2	✗
8)	E	O_1	O_2		E	O_x		✗
9)	E	O_1	O_2		E		O_1	✗

- Kein Eigentümer unterscheidet sich, keine Sperre im Modus X, daher keine Kollision
- Eigentümer_2 unterscheidet sich, also Kollision
- Eigentümer_1 unterscheidet sich, also Kollision
- Eigentümer_1 ist gleich, aber Eigentümer_2 unterscheidet sich, also Kollision
- Kein Eigentümer unterscheidet sich, keine Sperre im Modus X, daher keine Kollision
- Kein Eigentümer unterscheidet sich, aber die Sperranfrage hat den Modus X, daher Kollision
- Kein Eigentümer unterscheidet sich, aber die bestehende Sperre hat den Modus X, daher Kollision
- Eigentümer_1 unterscheidet sich, also Kollision
- Eigentümer_2 unterscheidet sich, also Kollision

Eigentümerkollision impliziert also Elementarsperrenkollision. Nur wenn eine Eigentümerkollision vorliegt, wird die Sperranfrage zurückgewiesen!

Siehe auch:

[Kumulation von Sperren \[Seite 19\]](#)

[Das Eigentümerkonzept \[Seite 10\]](#)

Kumulation von Sperren

Wie auch unter [Sperrmodus \[Extern\]](#) beschrieben, gibt es folgende drei Arten von Sperren:

- **Shared** (Lesesperre): mehrere Transaktionen können gleichzeitig eine Lesesperre halten (zum Lesen von Daten, die nicht geändert werden)
- **Exclusive** (Schreibsperre): Gleichzeitige Lese- oder Schreibsperren auf dieses Objekt werden zurückgewiesen; nur derselbe Eigentümer (siehe [Das Eigentümerkonzept \[Seite 10\]](#)) kann die Sperre noch einmal anfordern. Dies wird als *Kumulation* bezeichnet.
- **EXclusive**, nicht kumulierbare Sperre (erweiterte Schreibsperre): diese Sperre darf auch vom selben Eigentümer nur einmal angefordert werden.

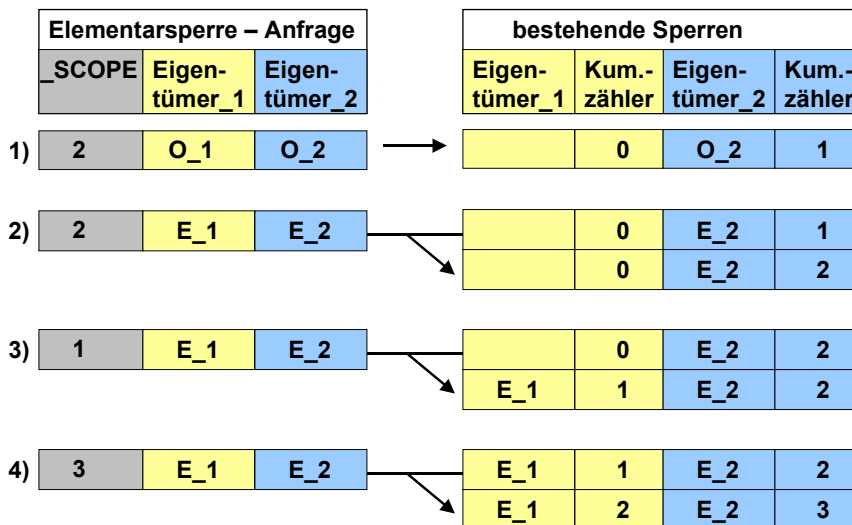
Welcher Typ von Sperre gewählt wird, entscheidet der Programmierer der Transaktion beim Anlegen des Sperrobjects. Sehen Sie hierzu [Das R/3-Sperrkonzept \[Extern\]](#) in der ABAP-Dokumentation.

Das mehrfache Absetzen einer Sperre (Kumulation) ist möglich, wenn Name, Argument und Sperrmodus der Elementarsperre gleich sind. Bei jeder weiteren Kumulation wird der Kumulationszähler um eins erhöht, bei jeder Freigabe einer Sperre um eins erniedrigt. Wird der Zähler Null, wird die Sperre aufgehoben.

Die folgende Grafik zeigt, wie Sperren kumulieren.



Kumulation von Sperren – Beispiel



1. Die Sperranfrage wird akzeptiert und in die [Sperrtabelle \[Seite 13\]](#) eingetragen. Der Kumulationszähler des Dialogeigentümers Eigentümer_1 ist 0, der des Verbuchungseigentümers Eigentümer_2 ist 1.

Kumulation von Sperren

2. Die Sperranfrage wird auch akzeptiert. Der Kumulationszähler des Dialogeigentümers Eigentümer_1 ist weiterhin 0 (weil `_SCOPE` auf 2 gesetzt war), der des Verbuchungseigentümers Eigentümer_2 ist nun 2.
3. Diesmal ist `_SCOPE` auf 1 gesetzt. Also erhöht sich der Kumulationszähler des Dialogeigentümers Eigentümer_1 auf 1, der des Verbuchungseigentümers Eigentümer_2 bleibt bei 2.
4. Diesmal ist `_SCOPE=3`, d.h. die Sperre hat zwei Eigentümer. Somit erhöht sich der Kumulationszähler des Dialogeigentümers Eigentümer_1 auf 2 und der des Eigentümer_2 auf 3.

Fragen und Antworten zum Thema Sperren

Im folgenden werden häufig gestellte Fragen zum Sperrkonzept mit zugehörigen Antworten aufgelistet, um schnell Information bereitzustellen. Das meiste ist unter [Funktionsweise des R/3-Sperrkonzepts \[Seite 6\]](#) genauer beschrieben.

Frage

Was passiert mit Sperren, wenn der Enqueue-Server durchgestartet wird?

Antwort

Sie gehen verloren, sofern sie nicht ins Backup-File auf Platte gesichert wurden. Auf Platte werden diejenigen Sperren gesichert, die bei **COMMIT WORK** nach **CALL FUNCTION .. IN UPDATE TASK** an den Verbucher vererbt werden. Die Sicherung auf Platte erfolgt zu dem Zeitpunkt, zu dem der Verbuchungsauftrag gültig wird, also im o.g. **COMMIT WORK**. Bei jedem Neustart des Enqueue-Servers werden die auf Platte gesicherten Sperreinträge in die [Sperrtabelle \[Seite 13\]](#) zurückgeladen.

Eine Sperre wird genau dann auf Platte gespeichert, wenn das *Backupflag* gesetzt ist.

Frage

Wo liegt die Sperrtabelle?

Antwort

Im Hauptspeicher, Shared Memory, des Enqueue-Servers. Zugriff haben alle Workprozesse des Enqueue-Servers. Fremde Applikationsserver lassen ihre Sperroperationen im Enqueue-Prozess des Enqueue-Servers ausführen. Die Kommunikation erfolgt dann über die betroffenen Dispatcher und den Message-Server.

Frage

Kann es nach Startup bereits Sperren geben?

Antwort

Ja, die gesicherten Sperren, die an den Verbucher vererbt wurden, werden beim Startup in die Sperrtabelle zurückgeladen (siehe erste Frage).

Frage

Wie schnell sind Sperroperationen?

Antwort

In Workprozessen des Enqueue-Servers einige 100 Mikrosekunden. In Workprozessen fremder Applikationsserver kommen die Message-Server Kommunikation und 8 Prozeßwechsel hinzu. Abhängig von CPU- und Netzlast zwischen ca. 20 (best case) und ca. 80 (typisch) Millisekunden.

Fragen und Antworten zum Thema Sperren

**Frage**

Was zuerst tun bei Problemen aller Art?

**Antwort**

Diagnosefunktionen ausführen:

sm12 Zusätze → *Diagnose* und danach

sm12 Zusätze → *Diagnose in VB*

Falls irgendwelche Probleme gemeldet werden, die Tracefiles *dev_w**, *dev_disp*, *dev_eq** sichern, im Syslog nachsehen.

**Frage**

Bei den Diagnoseeinzelheiten in SM12 bekommt man folgende Meldung:

```
Lock management operation mode
```

```
Internal lock management in same process
```

Was bedeutet diese Meldung, und was sind andere Optionen?

**Antwort**

"Internal lock management in same work process" in der Diagnosefunktion bedeutet, daß Sie am Enqueue-Server angemeldet sind und Ihr Workprozeß auf die Sperrtabelle sofort zugreifen kann. Sie muß Enqueue-Anforderungen nicht an einen Enqueue-Prozeß auf dem entfernten Enqueue-Server delegieren. Sind Sie an einem Applikationsserver angemeldet, der kein Enqueue-Server ist, teilt Ihnen die Diagnosefunktion den Namen des Enqueue-Servers mit.

In jedem R/3-System gibt es genau einen Applikationsserver, der als Enqueue-Server fungiert. Dieser pflegt die Sperrtabelle, die sich in einem Shared-Memory-Segment befindet. Alle Workprozesse auf dem Enqueue-Server können auf die Sperrtabelle zugreifen. Alle Workprozesse auf anderen Applikationsservern delegieren ihre Enqueue-Anforderungen an einen speziellen Enqueue-Workprozeß auf dem Enqueue-Server.



Das Verhalten ist selbstkonfiguriert. Im Default-Profil DEFAULT.PFL gibt die Parameterzeile "rdisp/enqname =<application server name>" Auskunft, welcher Applikationsserver als Enqueue-Server zur Verfügung steht. Stellt ein Applikationsserver fest, daß sein Name mit dem Namen des Enqueue-Servers übereinstimmt, legt er die Sperrtabelle an und alle ihre Workprozesse verarbeiten Enqueue-Anforderungen inline. Stellt ein Applikationsserver fest, daß sein Name nicht mit dem Namen des Enqueue-Servers übereinstimmt, sendet er alle Enqueue-Anforderungen an den Enqueue-Server.

Workprozesse vom Typ Enqueue garantieren die sofortige Verarbeitung eingehender Anforderungen. Normalerweise reicht ein Enqueue-Prozeß aus. In sehr großen R/3-Systemen mit sehr vielen Applikationsservern ist ein zweiter Prozeß vorteilhaft. Es ist jedoch nicht sinnvoll, mehr als zwei Enqueue-Prozesse zu konfigurieren. Wenn mit Transaktion SM50 -> [CPU] zu beobachten ist, daß nur der erste Enqueue-Prozeß benutzt wird, lag das Bottleneck nicht an dieser Stelle.

Fragen und Antworten zum Thema Sperren

**Frage**

Wieso benötigt man bei einem zentralen System überhaupt einen Enqueue-Workprozeß? Alle Workprozesse haben doch gleichermaßen Zugang auf das Shared Memory und damit auf die Sperrtabelle.

**Antwort**

Bei einem zentralen System wird der Enqueue-Prozeß nicht genutzt, er kostet aber auch nichts. Weil aber fast jeder Kunde früher oder später einen Applikationsserver dazunimmt, sind Fehler vorprogrammiert, wenn der Enqueue-Prozeß fehlt. Deshalb weist auch die Enqueue-Diagnose einen Fehler aus, wenn kein Enqueue-Prozeß konfiguriert ist.

**Frage**

Werden die Sperren in der Sperrtabelle auch auf Datenbankebene gesetzt? Wenn Nein, dann ist es möglich, in R/3 gesperrte Objekte mit Datenbank-Mitteln zu bearbeiten.

**Antwort**

Sperren werden nicht in der DB gesetzt, die Sperrtabelle liegt im Hauptspeicher des Enqueue-Servers.

**Frage**

Wird eine Sperrtabelle auch aufgebaut, wenn im Instanzenprofil kein Enqueue-Workprozeß auf dem Enqueue-Server gestartet wird?

**Antwort**

Ja, denn die Workprozesse auf dem Enqueue-Server arbeiten direkt auf die Sperrtabelle, ohne Umweg über den Enqueue-Prozeß. Dieser ist nur für Sperranforderungen aus fremden Applikationsservern zuständig.

**Frage**

Wie bekomme ich (programmatisch) heraus, wer gerade die nicht gewährte Sperre hält? D.h. wie kann ich nach einem ENQUEUE im Programm ermitteln, welcher Benutzer diese Sperre gerade hält, um es dem Benutzer mitzuteilen?

**Antwort**

Nach Return des Funktionsbausteins ENQUEUE_... mit steht der Name des Inhabers der Sperre in SY-MSGV1.

**Frage**

*Mit einem Single-Prozessor-System als Enqueue-Server haben wir X SD Benchmark Benutzer erreicht. Kann durch Einsatz eines Multiprozessor-Systems diese Zahl gesteigert werden (Message Server auf der gleichen Maschine wie Enqueue)? Ist die Skalierung linear zu erwarten (Anzahl CPUs * X SD user) ? Wieviele Prozessoren sind sinnvoll wenn Message-Server, Dispatcher, ein Dialog und zwei Enqueue Prozesse auf dem System laufen sollen ?*

Fragen und Antworten zum Thema Sperren



Antwort

Es ist zu erwarten, daß durch Einsatz von mehr Prozessoren der Durchsatz des Enqueue-Servers deutlich gesteigert werden kann. Die CPU-Last am Enqueue-Server verteilt sich ziemlich gleichmäßig auf Message-Server, Dispatcher und Enqueue-Workprozesse, so daß anzunehmen ist, daß bis zu 3 Prozessoren gleichzeitig beschäftigt werden können. Dispatcher und Message-Server bilden den Bottleneck beim Enqueue.

Eine lineare Skalierung ist bis 3 Prozessoren zu erwarten, und auch nur dann, wenn Sperranforderungen so häufig eintreffen, daß Message-Server, Dispatcher und Workprozesse gleichzeitig beschäftigt sind. Wegen asynchroner Systemprozesse (z.B. Syncer) kann der Einsatz von mehr Prozessoren den Durchsatz dennoch weiter verbessern.



Frage

Im Syslog finden sich öfters Meldungen wie z.B. die folgende: "Enqueue: Bisher aufgelaufene Wartezeit beim Sperren: 2500 Sekunden". Wie sollte vorgegangen werden, um das Problem zu analysieren? Oder ist dieser Eintrag nicht kritisch? (Abbrüche oder Timeouts wurden nicht bemerkt.)



Antwort

Die Meldung ist als Information zu verstehen, gibt aber u.U. Hinweise auf gestörten Parallelbetrieb von ABAP-Programmen. Es wird die Wartezeit angegeben, die seit Startup durch die Verwendung der WAIT-Parameters beim Aufruf von Enqueue-Funktionsbausteinen aufgelaufen ist.

Der WAIT-Parameter ermöglicht, daß ein Sperrversuch einige Male wiederholt werden kann, z.B. um im Verbucher nicht abbrechen zu müssen, wenn eine Sperre kurzzeitig von anderen Programmen gesetzt ist. Der Workprozeß bleibt zwischen den Sperrversuchen belegt.


Die wichtigsten Profilparameter zum Sperrkonzept

Die wichtigsten Profilparameter zum Sperrkonzept

Systemprofilparameter dienen dazu, das System optimal auf die Bedürfnisse und Ressourcen anzupassen. Sie sind allgemein unter [Profile \[Extern\]](#) in der CCMS-Dokumentation beschrieben.

Die wichtigsten Profilparameter zur Verbuchung sind unter [Hauptsystemprofilparameter für die Verbuchung \[Extern\]](#) beschrieben.

Die folgende Tabelle beschreibt dir wichtigsten Profilparameter, die mit dem Sperrmechanismus zu tun haben. Eine komplette Übersicht erhalten Sie, wenn Sie in der Transaktion `rz11` die Parameter mittels der Eingabe `*enq*` suchen.

Parameter	Bedeutung
enqueue/table_size	<p>Größe der Sperrtabelle [Seite 13], die vom Enqueue-Server im Hauptspeicher gehalten wird. Die Sperrtabelle enthält alle Information, welche Sperren derzeit von wem gehalten werden.</p> <p>Bei einem Überlauf der Sperrtabelle wird im Systemlog (vgl. Systemprotokolle [Extern]) ein entsprechender Vermerk gemacht.</p> <p>In diesem Fall sollten Sie überprüfen, ob der Verbuchungsserver ordnungsgemäß arbeitet, da bei einem Verbuchungsstopp die Sperrtabelle sehr schnell anwachsen kann. Liegen keine Verbuchungsprobleme zugrunde, können Sie mit diesem Parameter die Sperrtabelle vergrößern.</p> <p>Das Computing Center Management System (CCMS) überwacht den Zustand der Sperrtabelle ständig und gibt Warnungen heraus, wenn der Platz zur Neige geht.</p> <p>Sie können sich den Füllgrad der Sperrtabelle überprüfen, indem Sie sich die Sperrstatistik anzeigen [Seite 36] lassen.</p> <p>Einheit: Kilobyte Defaultwert: 4096</p>
rdisp/wp_no_enq	<p>Anzahl der Enqueue-Workprozesse, die auf dieser Instanz laufen sollen.</p>  <p>Natürlich ist dieser Parameter auf allen Applikationsservern, auf denen nicht der Enqueueserver läuft, auf 0 gesetzt, da nur die Instanz mit dem Enqueueserver Enqueue-Workprozesse enthalten kann.</p> <p>Einheit: ganze Zahl Defaultwert: 0</p> <p>Üblicherweise ist der Parameter auf der Enqueue-Instanz auf 1 gesetzt, d.h. es existiert im ganzen System nur ein Enqueue-Workprozeß. In großen Systemen kann der Parameter auf 2-4 gesetzt werden.</p>

Die wichtigsten Profilparameter zum Sperrkonzept

enque/backup_file	<p>Vollständiger Pfad zum Backupfile. Dieses dient dazu, nach einem Neustart des Enqueue-Servers die Sperren, die an die Verbuchung gegeben werden, wieder in die Sperrtabelle aufzunehmen (vgl. hierzu Funktionsweise des R/3-Sperrkonzepts [Seite 6]).</p> <p>Einheit: Zeichenfolge (Pfadname)</p> <p>Defaultwert: /usr/sap/<SID>/D<Instanznr>/log/ENQBCK</p>
rdisp/enqname	<p>Name des Applikationsservers, der den Enqueue-Service zur Verfügung stellt, d.h. auf dem der Enqueue-Server läuft (vgl. Funktionsweise des R/3-Sperrkonzepts [Seite 6]).</p> <p>Der Name sollte im Defaultprofil gepflegt sein, damit alle Applikationsserver den gleichen Namen verwenden.</p> <p>Einheit: Zeichenfolge (Instanz)</p> <p>Defaultwert: Name der Zentralinstanz</p>



Dies sind nicht alle Profilparameter, die mit der Sperrverwaltung zu tun haben. Die Defaultwerte sind für Ihr System optimal, und Änderungen sollten nur von Experten vorgenommen werden.

Sperrinträge verwalten

Verwendung

Die Sperrverwaltung dient dazu, Ihr System im Hinblick auf die Sperrlogik zu überwachen. Sie können ermitteln, was für Sperren momentan gehalten werden. Hierbei werden Sperren, bei denen das Backup-Flag gesetzt ist (weil sie schon dem Verbucher übergeben wurden), farblich hervorgehoben (siehe [Sperrinträge auswählen und anzeigen \[Seite 29\]](#)).

So können Sie einerseits Probleme erkennen, andererseits auch beseitigen, etwa durch das [Löschen \[Seite 32\]](#) nicht mehr erwünschter Sperren.

Integration

Das R/3 Sperrkonzept arbeitet eng mit der R/3-Verbuchung zusammen, die unter [Verbuchung im R/3 \[Extern\]](#) beschrieben ist.

Funktionsumfang

Das Einstiegsbild der Sperrverwaltung sieht so aus:

The screenshot shows the SAP R/3 'Sperrinträge selektieren' dialog box. The window title is 'Sperrinträge selektieren'. It features a menu bar with 'Sperrintrag', 'Bearbeiten', 'Springen', 'Zusätze', 'System', and 'Hilfe'. Below the menu is a toolbar with various icons. The main area contains a 'Auflisten' button and a form with the following fields: 'Tabellen-Name' (empty), 'Sperr-Argument' (empty), 'Mandant' (000), and 'Benutzer-Name' (MUSTER). At the bottom right, there is a status bar showing 'BIN (1) (000)', 'is0201', and 'INS'.

Mit der Sperrverwaltung können Sie

- [Sperrinträge auswählen und anzeigen \[Seite 29\]](#)
- [Sperrinträge löschen \[Seite 32\]](#)
- die [Sperrverwaltung testen \[Seite 35\]](#)
- die [Sperrstatistik anzeigen \[Seite 36\]](#)

Sperreinträge verwalten

Aktivitäten

Über *Werkzeuge* → *Administration*, *Monitor* → *Sperreinträge* oder durch Eingabe des Transaktionscodes **sm12** gelangen Sie in die Sperrverwaltung.

Sperrinträge auswählen und anzeigen

Sie können Sperrinträge für die Anzeige nach folgenden Kriterien auswählen (siehe hierzu [Funktionsweise des R/3-Sperrkonzepts \[Seite 6\]](#)):

Tabellen-Name	
Sperr-Argument	
Mandant	000
Benutzer-Name	MUSTERMANN

- Tabelle, in der Zeilen gesperrt sind
- Argument des Sperrintrags
- Mandant
- Benutzer, der die Sperre verhängt hat

Der Mandant und Ihr Benutzer werden automatisch eingefügt. Wenn Sie die Sperren eines anderen Benutzers oder Mandanten anzeigen wollen, müssen Sie das entsprechende Feld überschreiben. Lassen Sie das Feld leer oder geben Sie einen * ein, wird generisch nach allen möglichen Eingaben selektiert.

Der Anzeige der Sperrinträge können Sie folgende Informationen entnehmen:

Sperrereinträge auswählen und anzeigen

Man	Benutzer	Zeitpunkt	Shared	Tabelle	Sperrargument
000	BRAUCKHOFF	14:54:25		SCRP_ENQ	1300SAPMSLIC
000	BAREISST	14:56:17		TRDIR	VERDATA0
000	FANH	14:59:43		SCRP_ENQ	0100FANTEST_TABLECONTROL
000	BINDEWALD	15:21:54		TRDIR	RSSYSTDB
000	AGHADAVOODI	15:36:14		TFDIR	HTTP_TRFC_HANDLER
000	SCHIED	15:42:58		TRDIR	SCHIEDTMP
000	AGHADAVOODI	15:51:29		TRDIR	MATEST19
000	SANTA-CRUZ	15:56:04		DOKHL	RERSBDC_ARCHIVE
000	FUCHSS	15:59:05		TFDIR	RSPO_RCHANGE_SPOOLREQ_ATTR

Selektierte Sperrereinträge: 9

BIN (1) (000) ds0025 INS

- Der Spalte **Man** entnehmen Sie den Mandanten, in dem ein Sperrereintrag erzeugt wurde.
- Die Spalte **Benutzer** enthält den Benutzer, der die Sperre verhängt hat (d.h. der das ABAP-Programm ausgeführt hat, das die Sperre angelegt hat).
- In der Spalte **Zeitpunkt** sehen Sie, wann der Sperrereintrag erzeugt wurde.
- Der Spalte **Shared** entnehmen Sie, ob ein Sperrobjekt von mehreren Benutzern gemeinsam benutzt wird. Für diesen Fall enthält die Spalte ein Kreuz, ansonsten ist sie leer.
 - Ist ein Sperrobjekt mit mehreren Shared-Sperren belegt, können mehrere Benutzer dieselben Daten zum gleichen Zeitpunkt sperren. Siehe hierzu [Sperrmodus \[Extern\]](#) in der ABAP Dictionary Dokumentation.
- In der Spalte **Tabelle** zeigt das System die Tabelle an, in der Zeilen gesperrt sind.
- In der Spalte **Sperrargument** sehen Sie das Argument (Schlüsselwert) des Sperrereintrags.

Das entspricht den Einträgen der [Sperrtabelle \[Seite 13\]](#).

Die Sperrereinträge haben verschiedene Farben:

Blau bedeutet, daß die Sperre bereits an den Verbucher vererbt wurde (vgl. [Das Eigentümerkonzept \[Seite 10\]](#)) und damit das Backup-Flag gesetzt ist. Eine solche Sperre wird auch nach einem Neustart des Enqueue-Servers wieder in die Sperrtabelle aufgenommen.

Sperreinträge auswählen und anzeigen

Schwarz bedeutet, daß die Sperre (noch) dem Dialogeigentümer gehört. Das Backup-Flag ist nicht gesetzt.

Über *Bearbeiten* → *Sortieren nach* können Sie die Anzeige nach Benutzer, Zeit, Tabelle oder Hostsystem (Rechner) sortieren.

Sie können durch Doppelklick detaillierte Informationen zu jedem Sperreintrag anzeigen, einschließlich des Rechnernamens und der Nummer des R/3-Systems, in dem eine Sperre erzeugt wurde.

Sperrinträge löschen

Sperrinträge löschen

Verwendung

Bestimmte Probleme können zu Sperren führen, die nicht freigegeben werden, so daß Benutzer auf die gesperrten Objekte nicht zugreifen können.

Diese können mit dieser Funktion von Hand gelöscht werden.



Im allgemeinen sollten Sie Sperrinträge nicht direkt mit den Löschfunktionen der Sperrverwaltung löschen. Nicht freigegebene Sperren sind fast immer ein Symptom für ein anderes Problem. Wenn Sie das zugrundeliegende Problem lösen, wird der Sperrintrag automatisch freigegeben.

Siehe auch: [Fragen und Antworten zum Thema Sperren \[Seite 21\]](#)

Voraussetzungen

Die häufigsten Ursachen für nicht freigegebene Sperrinträge:

- **Probleme in der Verbuchung.** Sperren durch die Verbuchung werden vom System aufrechterhalten, bis die jeweilige Datenbankänderung verarbeitet oder vorzeitig beendet worden ist. Wenn ein Problem die Verbuchung verhindert, können die Sperren, die durch unverarbeitete Verbuchungssätze erzeugt werden, die Arbeit im System blockieren. (Vgl. hierzu [Verbuchungsfehler analysieren und beheben \[Extern\]](#).)

Sie können Sperren, die von Verbuchungssätzen gehalten werden, in der Liste der Sperrinträge identifizieren. Die Sperren, die vom V1-Teil eines Verbuchungssatzes gehalten werden, sind in der Liste hervorgehoben. In der Detailanzeige ist bei solchen Sperren auch das Backup-Kennzeichen gesetzt.

Behebung: So stellen Sie fest, ob ein Verbuchungsproblem vorliegt:

- a) Notieren Sie den Sperrschlüssel des betreffenden Sperrintrags. Markieren Sie diesen Eintrag, und wählen Sie *Details*, um den Transaktionscode anzuzeigen.
- b) Wählen Sie *Werkzeuge* → *Administration* → *Monitor* → *Verbuchung* ([Die Verbuchungsverwaltung \[Extern\]](#)), um zu prüfen, ob es noch unverarbeitete Verbuchungssätze (Status `init` oder `auto`) für den Benutzer gibt. Wenn Sie unverarbeitete Verbuchungssätze finden, identifizieren Sie Transaktionscode und Sperrschlüssel, um den Verbuchungssatz zu finden, der die Sperre hält.

Beachten Sie, daß Verbuchungssätze mit Status `err` keine Sperren halten. Sperren werden automatisch freigegeben, wenn eine Verbuchung vorzeitig beendet wird. Wenn Sie `err`-Verbuchungssätze finden, orientieren Sie sich an den Prozeduren zur Fehlerbehebung, die unter [Verbuchungsfehler analysieren und beheben \[Extern\]](#) beschrieben werden.

- c) Prüfen Sie, ob Ihr Verbuchungsserver normal läuft. Wählen Sie *Werkzeuge* → *Administration* → *Monitor* → *Systemüberwachung* → *Server* oder Transaktion `sm51`, positionieren den Cursor auf dem Applikationsserver, auf dem ein Verbuchungsserver läuft, und wählen dann *Prozesse*, um die Workprozesse auf dem Verbuchungsserver anzuzeigen. Wenn es offene Verbuchungssätze gibt und die

Sperreinträge löschen

Verbuchung normal läuft, sollten Sie Bewegung in den Verbuchungsprozessen feststellen können.

Falls erforderlich, starten Sie den Verbuchungsserver über *Werkzeuge* → *CCMS*.

- **vorzeitige SAPGUI-Beendigung.** Wenn ein Benutzer seinen PC ausschaltet, ohne sich vom R/3-System abzumelden, oder sein SAPGUI-Programm aus einem sonstigen Grund vorzeitig beendet wird, kann der Benutzer im R/3-System angemeldet bleiben. Die vom Benutzer bei Auftreten des Problems gehaltenen Sperren werden nicht freigegeben, weil der Benutzer nicht mehr im R/3-System aktiv ist.

Behebung: Sie können diese Sperren freigeben, indem Sie den Benutzer abmelden:

- a) Wählen Sie *Werkzeuge* → *Administration* → *Monitor* → *Systemüberwachung* → *Server* oder Transaktion sm51.

Zeigen Sie die Benutzer der SAP-Instanz an, die in der Detailanzeige der Sperreinträge genannt wird. Prüfen Sie, wann der Benutzer zuletzt im System aktiv war.

Wenn der Benutzer lange Zeit nicht aktiv gewesen ist, hat er vielleicht seinen Rechner verlassen, ohne die Arbeit abzuschließen, oder SAPGUI wurde vorzeitig beendet.

Sprechen Sie möglichst den Benutzer direkt an. Seine Arbeit geht verloren, wenn Sie ihn abmelden.

- b) Wenn der SAPGUI des Benutzers vorzeitig beendet wurde, können Sie ihn in der Benutzerübersicht abmelden. Der Benutzer kann die verlorenen Sperren nicht wieder anfordern, selbst wenn er sich wieder anmeldet. Er muß die Arbeiten an den zuvor gesperrten Objekten nochmals durchführen.
- c) Wenn der Benutzer lediglich noch nicht fertig ist, sollte er die Arbeit an dem gesperrten Objekt normal abschließen können. Wenn Sie ihn abmelden, geht seine Arbeit verloren.



Löschen Sie Sperreinträge nur dann, wenn Sie sicher sind, daß sie ungültig sind und gelöscht werden können, ohne einen aktiven Prozeß (Benutzermodus, Verbuchung usw.) zu beeinträchtigen. Wenn Sie eine Sperre löschen, die von einem aktiven Benutzer gehalten wird, kann es zu Datenverlusten oder Inkonsistenzen kommen.

Vorgehensweise

Wenn Sie einen einzelnen Sperreintrag löschen wollen, positionieren Sie den Cursor darauf und wählen *Sperreintrag* → *Löschen*.

Mit *Sperreintrag* → *Alle Löschen* können Sie alle Sperreinträge auf einmal löschen.

Sie erhalten zunächst einen Warntext, der Sie noch einmal auf die mit dem Löschen verbundenen Risiken hinweist. Danach kommt ein Popup, mit dem Sie den Sperreintrag endgültig löschen.



Für die laufende Transaktion können Sie den Warntext ausschalten, indem Sie *Bearbeiten* → *Warntext ausschalten* wählen.

Sperreinträge löschen

Ergebnis

Die Sperren sind wieder freigegeben und können neu angefordert werden.

Sperrverwaltung testen

Mit den Diagnosefunktionen im Menü *Zusätze* können Sie die Sperrverwaltung auf Fehler testen. Es gibt zwei Testfunktionen:

- *Diagnose*: prüft, ob der Sperr-Server auf Sperranforderungen reagiert. Wenn ein Benutzer ein Objekt sperrt, müssen der Workprozeß, in dem der Benutzer aktiv ist, und der Sperr-Server kommunizieren, um Sperren einzutragen und anschließend wieder aufzuheben. Siehe hierzu [Funktionsweise des R/3-Sperrkonzepts \[Seite 6\]](#).
- *Diagnose in VB*: prüft, ob eine für eine Transaktion ausgegebene Sperre erfolgreich an einen während der Verbuchung aktiven Verbuchungsserver weitergegeben und aufgehoben werden kann.

Im Kapitel [Fragen und Antworten zum Thema Sperren \[Seite 21\]](#) finden Sie noch weitere Hinweise zu den Diagnosefunktionen.

Über *Zusätze* → *Funktionshinweise* können Sie sich eine kurze Erklärung anzeigen, wie die Sperrverwaltung in einer verteilten Umgebung implementiert ist.

Sperrstatistik anzeigen

Sperrstatistik anzeigen

Verwendung

Zur Überwachung der in Ihrem System abgesetzten Sperren können Sie die Statistik verwenden. Sie kann nützlich sein, um Schwachstellen in der Konfiguration zu beheben.

Voraussetzungen

Sie müssen nicht auf dem Applikationsserver angemeldet sein, auf dem der Sperrserver läuft; es wird immer die globale Statistik angezeigt.

Vorgehensweise

Wählen Sie im Einstiegsbild der Sperrverwaltung (**sm12**) *Zusätze* → *Statistik*, um die Statistik anzuzeigen. Diese läuft seit dem letzten Neustart des Sperrservers.

Ergebnis

Die folgende Tabelle erklärt die wichtigsten Informationen, die die Enqueue-Statistik liefert.

Enqueue Statistics at 1999/02/16 09:53:50

Enqueue Requests.....: 110914	Anzahl der Sperranfragen
Rejects.....: 217	zurückgewiesene Sperranfragen
Errors.....: 0	Fehler
Dequeue Requests.....: 29149	Freigabeanforderungen (DEQUEUE)
Errors.....: 0	Fehler bei Freigabe
DequeueAll Requests...: 224832	Freigabe aller Sperren einer LUW
CleanUp Requests.....: 217	
Backup Requests.....: 732	
Reporting Requests...: 66	
Compress Requests....: 0	
Verify Requests.....: 0	
Writes to File.....: 12548	Schreibzugriffe auf Datei
Backup.....: 7091	
Owner Names.....: 5457	Maximale Anzahl von Eigentümernamen, die die Sperrtabelle aufnehmen kann
Peak Util.....: 5	bisher maximale Anzahl von Eigentümern, die gleichzeitig in der Sperrtabelle standen
Actual Util.....: 0	derzeitige Anzahl von Eigentümern in der Sperrtabelle

Sperrstatistik anzeigen

Granule Arguments.....: 5457	analog: Füllgrad der Sperrtabelle, Elementarsperren - Argumente betreffend
Peak Util.....: 65	
Actual Util.....: 0	
Granule Entries.....: 5457	analog: Füllgrad der Sperrtabelle, Elementarsperren - Namen betreffend
Peak Util.....: 65	
Actual Util.....: 0	
Update Queue Peak.....: 2	
Actual: 0	
Total Lock Time.....: 2407.752370s total	Zeit, die insgesamt für Sperroperationen gebraucht wurde
Total Lock Wait Time.: 1066.962126s total	davon Wartezeit



Wichtig zur Überwachung des Sperrmechanismus ist der **Füllgrad der Sperrtabelle**. Diese sollte auf keinen Fall überlaufen. Ein Überlauf droht, wenn bei Owner Names, Granule Arguments oder Granule Entries der Peak Util gleich oder nahe dem Maximum ist.

Probleme analysieren und beheben

Probleme analysieren und beheben

Sollten Sie Probleme mit der Sperrverwaltung haben (Syslogeinträge, Sperren, die nicht freigegeben werden, etc.), könnte einer der folgenden Hinweise die Lösung sein:

Hinweisnummer	Kurztext
0193864	Enqueue, tRFC wird mehrfach vergeben
0190985	Enqueue Patch Collection 46B 1999/12
0190701	Core Dump in Dequeue Rel. 45B ab Patchlevel 299
0187787	Enqueue Queue Überlauf
0170602	FM: LOG R1J=> ThAdXEnqState, reset mode <<username.
0149166	Enqueue: GE9, Akkumulierte Wartezeit für Sperre
0143774	Sperrtabelle nach Überlauf kaputt
0138559	Anzeige des Enqueue Logging Files ist fehlerhaft
0138542	Enqueue request schlägt fehl
0137500	Enqueue funktioniert nicht
0135971	Fehler im Enqueue-Reset
0120030	SM12 Sperreinträge aktualisieren geht nicht
0119705	Meldung LstRestore: restore from replication failed
0112075	Enqueue Workprozess Core-Dump
0110948	Transaktions-Reset nach Verlust der Sperrtabelle
0109473	Exclusive Lock Waits auf Tabelle USRBF, Deadlock
0079084	Syslog: Fehler beim Schreiben der Sperrhandler-Datei
0079001	Fehler beim lesen einer Sperrhandler-Datei.
0072301	Syslog: GE1 T Enqueue: OpCode 2, missing par. USER
0065972	Sperreinträge mit Zeit 00/00/00 nicht löschar
0043614	Enqueue, remaining lock entries
0013907	Systemfehler im Sperrhandler, Sperrtabelle Überlauf
0000651	Voraussetzungen fuer Enqueue-Handling