

---

# 1 Einleitung

## 1.1 Zielsetzung des Handbuchs

System Managed Storage bezeichnet ein Konzept für Datenadministration. Es ist charakterisiert durch eine bestimmte Aufgabenverteilung zwischen Systembetreuer, Benutzer und System, welche für die Dateiverwaltung eine hohe Effizienz ermöglicht. Zur Unterstützung des Systembetreuers werden in BS2000/OSD im Pubset-Management und in den Produkten der Software-Konfiguration, insbesondere in HSMS, umfangreiche Funktionen bereitgestellt. Diese sind in den jeweiligen Handbüchern im Detail beschrieben. Das vorliegende Handbuch verfolgt das Ziel, das Gesamtkonzept, die Bedeutung der einzelnen Funktionen innerhalb des Gesamtkonzepts, ihr Zusammenspiel und ihre gegenseitigen Abhängigkeiten zu verdeutlichen. Es soll damit dem Systembetreuer als Leitfaden durch die umfangreiche und breit gestreute SMS-Funktionalität dienen.

## 1.2 Zielgruppe des Handbuchs

Das Handbuch wendet sich an die Systembetreuung des Betriebssystems BS2000/OSD.

## 1.3 Aufbau des Handbuchs

Das Handbuch besteht aus 11 Kapiteln.

### *Kapitel 1 „Einleitung“*

beschreibt die Zielsetzung des Handbuchs, nennt die Handbuch-Zielgruppe und erläutert den Aufbau des Handbuchs.

### *Kapitel 2 „System Managed Storage – Idee“*

stellt die zentrale Idee dar, die dem Konzept des System Managed Storage zu Grunde liegt.

### *Kapitel 3 „System Managed Storage im BS2000 – Übersicht“*

gibt einen Überblick, wie das Konzept des System Managed Storage im BS2000 verwirklicht ist.

### *Kapitel 4 „Merkmale eines SM-Pubsets“*

beschreibt den Aufbau und die Eigenschaften von SM-Pubsets. Die SM-Pubsets sind für die Realisierung von System Managed Storage im BS2000 von zentraler Bedeutung.

### *Kapitel 5 „Neueinrichten eines SM-Pubsets“*

zeigt, wie der Systembetreuer neue SM-Pubsets einrichten kann.

### *Kapitel 6 „Erzeugen eines SM-Pubsets aus bestehenden SF-Pubsets“*

zeigt, wie sich bereits vorhandene SF-Pubsets unter Beibehaltung ihres Dateibestands in SM-Pubsets überführen lassen. Dabei werden zunächst die grundsätzlichen Probleme dargestellt, die bei der Migration zu einem SM-Pubset beachtet werden müssen, dann werden allgemeine Vorgehensweisen besprochen, und schließlich werden für bestimmte Anwendungsszenarien die erforderlichen Schritte skizziert.

### *Kapitel 7 „Auskunftsaktionen für SM-Pubsets“*

gibt eine Zusammenstellung über Auskunftsaktionen, die für SM-Pubsets relevant sind.

### *Kapitel 8 „Änderung der Merkmale von SM-Pubsets“*

beschreibt, welche Möglichkeiten zur Verfügung stehen, um die Konfiguration und die Merkmale eines bestehenden SM-Pubsets zu ändern. Insbesondere wird die Vergrößerung und Verkleinerung der Konfiguration eines SM-Pubsets im laufenden Pubset-Betrieb erläutert.

### *Kapitel 9 „Management des Lebenszyklus einer Datei“*

beschreibt die Schnittstellen zwischen Systembetreuer und Benutzer, die für SM-Pubsets zur Verfügung stehen, um die verschiedenen innerhalb des Lebenszyklus einer Datei erforderlichen Aktivitäten der Dateiverwaltung effizient und Kosten sparend durchführen zu können (Anlegen der Datei, Verlagern auf Hintergrundebenen, Zurückholen in die Verarbeitungsebene, Sichern, Wiedereinspielen). Außerdem wird gezeigt, wie der Systembetreuer selbst konzipierte Strategien der Speichernutzung zum Einsatz bringen kann und welche Möglichkeiten für die Automatisierung der Datenadministration zur Verfügung stehen. Ferner wird dargestellt, wie Benutzer, die Systemautomatismen bewusst nicht nutzen wollen (z.B. Benutzer der physikalischen Allokierung), in SM-Pubsets integriert werden können, und wie sich damit bei SM-Pubsets eine attraktive Alternative zur Nutzung von Privatplatten ergibt.

### *Kapitel 10 „Pubset-Überwachung und Pubset-Pflege“*

skizziert verschiedene Aufgabengebiete der Pubset-Pflege und gibt Hinweise, wie die für SM-Pubsets verfügbaren Funktionen den Systembetreuer bei der Durchführung dieser Aufgaben unterstützen.

### *Kapitel 11 „Ressourcenkontrolle für SM-Pubsets“*

gibt einen Überblick über die Möglichkeiten, mit welchen der Systembetreuer den Umfang der von den einzelnen Benutzern beanspruchten Pubset-Ressourcen kontrollieren und beschränken kann.

Am Ende des Handbuchs befinden sich das Literatur- und Stichwortverzeichnis. Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift ist im Literaturverzeichnis aufgeführt.

## 1.4 Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen BS2000/OSD-Version zu diesem Handbuch entnehmen Sie bitte ggf. der entsprechenden Readme-Datei.

Sie finden die Readme-Datei zu BS2000/OSD V6.0 auf Ihrem BS2000-Rechner unter dem Namen SYSRME.BS2CP.150.D. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Den vollständigen Pfadnamen können Sie auch mit folgendem Kommando ermitteln:

```
/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=BS2CP,LOGICAL-ID=SYSRME.D
```

Die Readme-Datei können Sie mit dem Kommando SHOW-FILE oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
/PRINT-DOCUMENT SYSRME.BS2CP.150.D,LINE-SPACING=*BY-EBCDIC-CONTROL
```

## 2 System Managed Storage – Idee

**System Managed Storage (SMS)** bezeichnet ein Konzept für Dateiverwaltung, das eine bestimmte Rollenverteilung zwischen den Instanzen Systembetreuer, Benutzer und System vorsieht. Es bietet insbesondere bei großen Datenträgerkonfigurationen die Grundlage für die effiziente Administration und Nutzung von Speicherressourcen. Die Benutzer verwenden dabei **Datei-Services** auf logischer Ebene. Deren physikalische Realisierung bleibt ihnen weitgehend verborgen. Die Durchführung der Maßnahmen, welche die Services realisieren, ist ein Aufgabengebiet des Systembetreuers, das als **Datenadministration** bezeichnet wird. Der Systembetreuer erhält für diese Aufgabe umfangreiche Unterstützung durch das System.

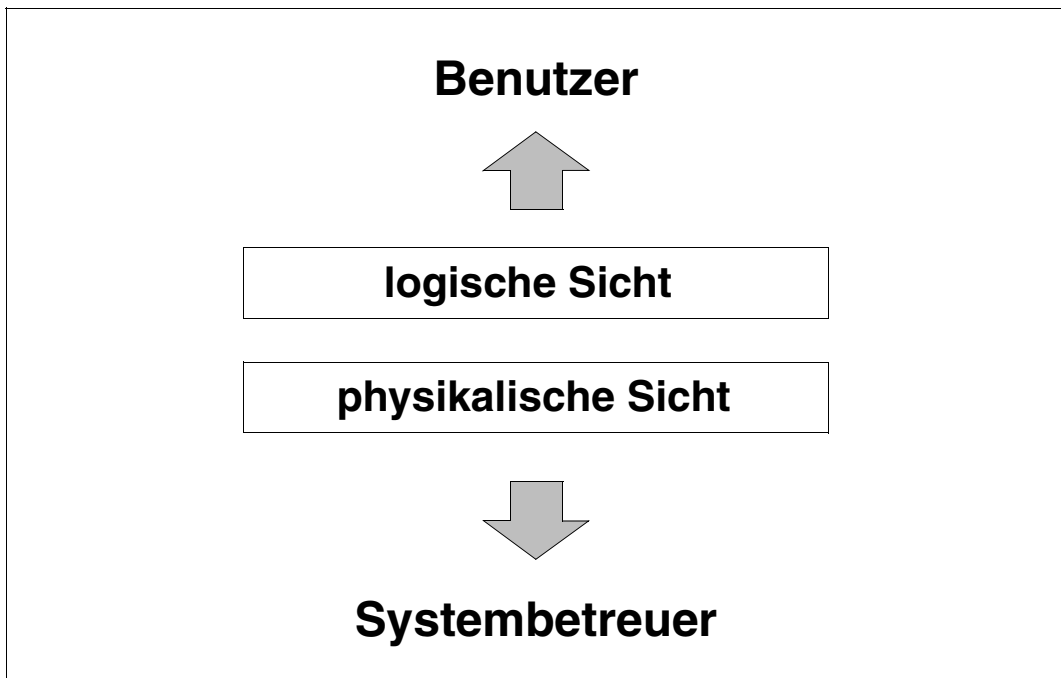


Bild 1: Trennung von logischer Sicht und physikalischer Sicht

Die Trennung von logischer und physikalischer Sicht betrifft alle Aktivitäten der Dateiverwaltung:

- das Anlegen einer Datei auf einem Medium, das ihren Anforderungen während der Verarbeitung am besten entspricht,
- das Verlagern der Datei auf ein besser geeignetes Verarbeitungsmedium, wenn sich die Anforderungen für die Verarbeitung der Datei ändern oder neue Datenträger eingesetzt werden,
- das Verdrängen einer Datei auf eine kostengünstige Hintergrundebene während inaktiver Phasen, in denen auf die Datei nicht zugegriffen wird (Migrate), sowie das Zurückholen der Datei in die Verarbeitungsebene, wenn sie wieder verarbeitet werden soll (Recall),
- das Erstellen von Sicherungen und deren Verwaltung (Backup); das Wiedereinspielen von Sicherungen (Restore),
- das Aufräumen nicht mehr benötigter Dateien und Dateisicherungen.

Die verschiedenen Aktivitäten im Rahmen der Dateiverwaltung sind in [Bild 2](#) illustriert.

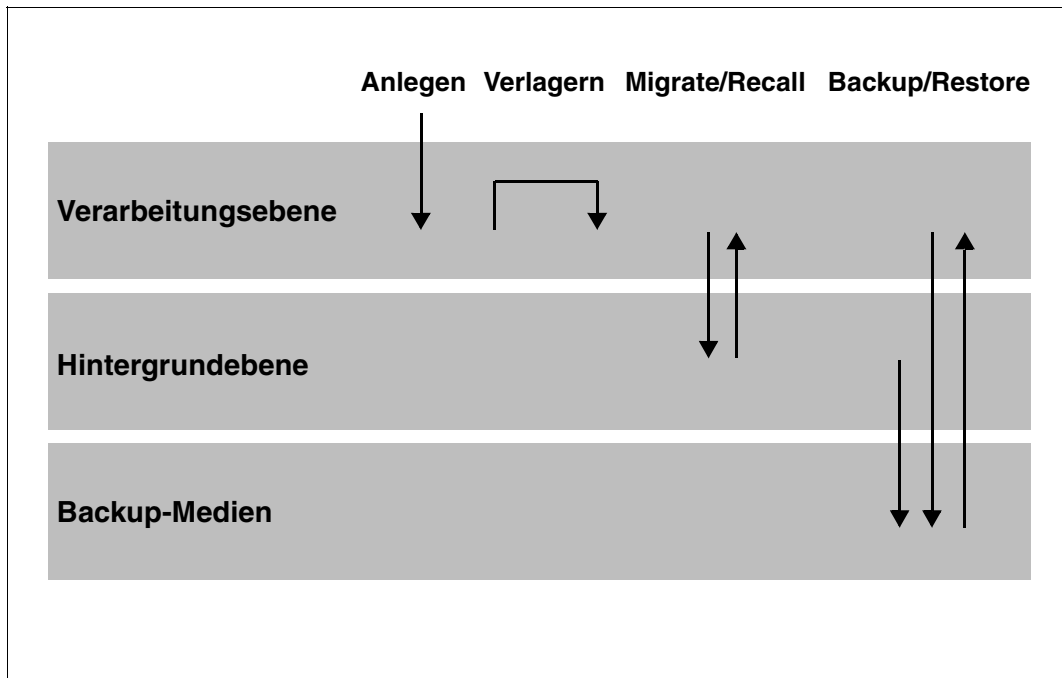


Bild 2: Aktivitäten im Rahmen der Dateiverwaltung

Die Datei-Services, die sich auf die Verarbeitungseigenschaften einer Datei beziehen und mit dem Ablageort der Datei innerhalb der Verarbeitungsebene in Zusammenhang stehen, werden als **Storage-Services** bezeichnet. Sie erlauben dem Benutzer beispielsweise die gewünschte Performance für eine Datei zu spezifizieren. Die Datei-Services, welche die Verdrängung auf Hintergrundebenen, das Erstellen von Sicherungen und das Aufräumen nicht mehr benötigter Dateien und Dateisicherungen steuern, heißen **HSMS-Management-Services**.

Die durch System-Managed-Storage begünstigte Aufgabenverteilung zwischen Systembetreuer, Benutzer und System bietet die Grundlage für eine effiziente und Kosten sparende Dateiverwaltung und schafft damit wirtschaftlichen Nutzen:

- Vereinfachung der Benutzerschnittstelle  
Da die Benutzer die Anforderungen hinsichtlich Dateiverwaltung auf logischer Ebene spezifizieren, sich aber um die Abwicklung nicht kümmern müssen, werden sie für ihre eigentlichen Aufgaben entlastet. Die Einarbeitung in Aufgabengebiete, welche den Benutzern im Allgemeinen fremd sind, z.B. die Beschäftigung mit den Eigenschaften verschiedener Speichermedien zur Bestimmung des geeigneten Ablageorts der Dateien, entfällt. Auch das Problem, dass die Benutzer auf Grund von unzureichenden Kenntnissen die Datenadministration nicht optimal durchführen und dadurch Ressourcen nicht effizient nutzen, wird reduziert.
- Effiziente Abwicklung der Datenadministration  
Die Durchführung der Datenadministration erfolgt (weitgehend) zentral durch den Systembetreuer, der auf diesem Gebiet spezielle Kompetenz besitzt. Er kann dabei versuchen, die Erfüllung der verschiedenen Benutzeranforderungen unter globalen Aspekten zu optimieren. Dies erlaubt sowohl eine effiziente Organisation der Datenadministration als auch eine gute Nutzung der vorhandenen Datenträger. Voraussetzung für die Zentralisierung der Datenadministration ist, dass die Eingriffsmöglichkeiten der Benutzer beschränkt werden können. Ermöglicht wird dies durch die logische Schnittstelle, welche den Benutzern zwar das Formulieren ihrer Wünsche erlaubt, sie aber von der Abwicklung der Datenadministration abschottet.
- Flexible Anpassbarkeit der Konfiguration  
Durch die Abschottung der Benutzer von der Durchführung der Datenadministration können die Datenträgerkonfiguration, die Verteilung der Dateien auf die Datenträger und die Verfahren der Datenadministration geändert werden, ohne dass Anpassungen vonseiten der Benutzer notwendig sind. Da sie den genauen Ablageort einer Datei nicht kennen und dieser nicht in Kommando-prozeduren eingeht, können Dateien bei Bedarf auf andere Datenträger verlagert werden, ohne dass die Kommando-prozeduren ungültig werden. Dies bietet die Grundlage für die dynamische Modifizierbarkeit der Dateiverteilung auf die einzelnen Datenträger. Zusammen mit den in SMS vorgesehenen Möglichkeiten, die Datenträgerkonfiguration im laufenden Betrieb ändern zu können, eröffnet sich damit der Weg, die Verteilung der Dateien flexibel an sich ändernde Anforderungen und Belastungen anzupassen. Auch lassen sich damit vorhandene Speichermedien durch Speichermedien einer neuen Technologie austauschen und ohne

Anpassungsaufwand vonseiten der Benutzer für bereits bestehende Dateien nutzbar machen. Weil Dateien problemlos verlagert werden können, wird auch die kontinuierliche Reorganisation der Datenträger im laufenden Betrieb ermöglicht.

- Systemunterstützung für den Systembetreuer  
In komplexen Konfigurationen kann der Umfang der Datenadministration sehr groß werden. Um den Arbeitsaufwand in Grenzen zu halten, ist es erforderlich, insbesondere Routinetätigkeiten des Systembetreuers weitgehend zu automatisieren. Die Unterstützung durch das System kann auf verschiedene Arten erfolgen:
  - a) durch Automatismen, die in das System integriert sind und durch geeignete Parametrisierung den spezifischen Anwendungsumgebungen angepasst werden können
  - b) durch Schnittstellen, welche die Entwicklung automatisch ablauffähiger Prozeduren oder Programme für die Durchführung der Datenadministration ermöglichen.

Das folgende Bild zeigt, wie das Zusammenwirken der aufgezählten Faktoren zu wirtschaftlichem Nutzen führt.

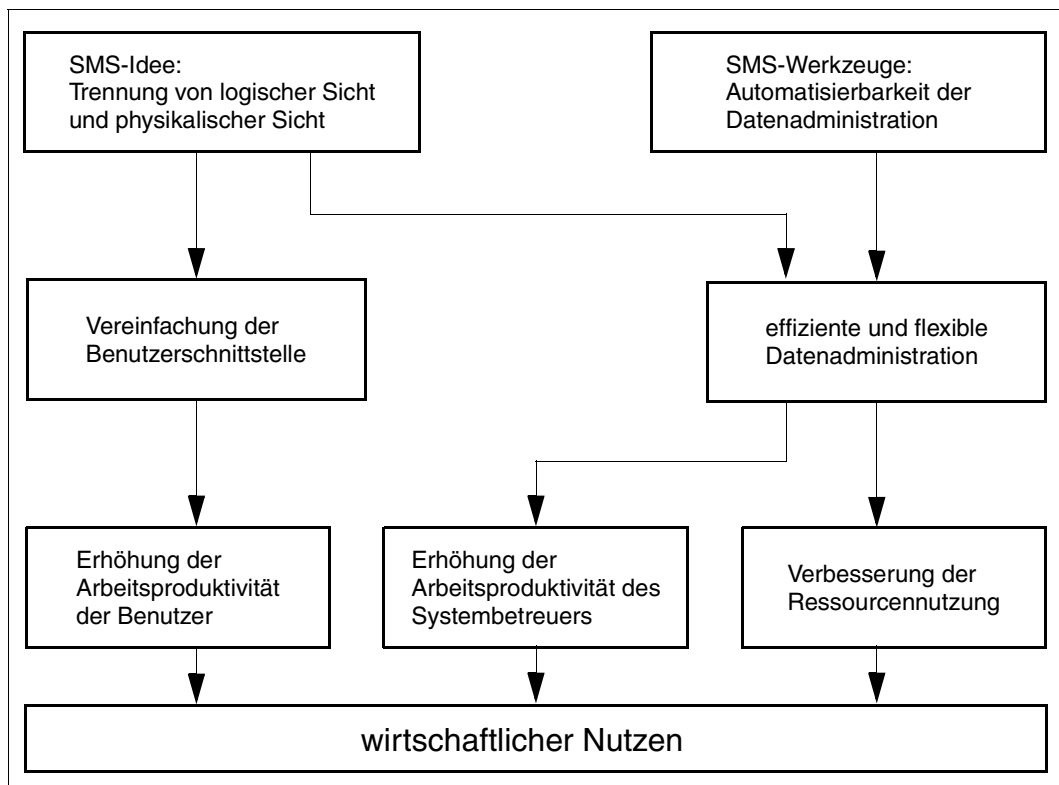


Bild 3: Wirtschaftlicher Nutzen durch System Managed Storage



---

## 3 System Managed Storage im BS2000 – Übersicht

### 3.1 Datenadministration und Pubset-Management

Im BS2000 stellt der Systembetreuer Pubsets mit zugeordneten Hintergrundebenen und Backup-Konfigurationen als Umgebung zur Verfügung, in der die Benutzer Dateien anlegen, bearbeiten und sichern können. Datenadministration als Aufgabengebiet des Systembetreuers ist im BS2000 daher - sowohl lokal als auch in einem Shared-Pubset-Verbund - weitgehend mit **Pubset-Management** gleichzusetzen. Dieses umfasst folgende Bereiche:

- Einrichten der Pubsets und der ihnen zugeordneten Hintergrundebenen und Backup-Konfigurationen;
- In- und Außerbetriebnahme von Pubsets;
- Zuteilung der Pubset-Ressourcen an verschiedene Benutzer;
- Unterstützung der Benutzer bei der Dateiverwaltung in den verschiedenen Phasen des Lebenszyklus einer Datei (Anlegen von Dateien, Sichern, Wiedereinspielen, Verdrängen auf Hintergrundebenen, etc.);
- Verhindern, Erkennen und Beheben kritischer Pubset-Zustände;
- Anpassen der Pubset-Konfiguration an sich ändernde Anforderungen.

Das Pubset-Management als Aufgabengebiet der Systembetreuung betrifft im BS2000 die Instanzen Systemverwalter, Operator und HSMS-Administrator. In diesem Handbuch differenzieren wir zwischen diesen Instanzen im Allgemeinen nicht.

## 3.2 Pubset-Konzept und SMS

Für die Realisierung der SMS-Idee im BS2000 bietet das Pubset-Konzept einen idealen Rahmen. Der Systembetreuer kann zwei verschiedene Typen von Pubsets einrichten: **Single-Feature-Pubsets (SF-Pubsets)** und **System-Managed-Pubsets (SM-Pubsets)**. SM-Pubsets sind Erweiterungen von SF-Pubsets. Sie tragen der Idee des System Managed Storage durch spezielle Eigenschaften besonders Rechnung.

Auch SF-Pubsets verwirklichen bereits viele SMS-Gedanken. Beispielsweise müssen sich die Benutzer nicht um den Ablageort einer Datei innerhalb eines Pubsets kümmern. Der physikalische Aufbau der einzelnen Volumes bleibt ihnen verborgen. Allerdings sind SF-Pubsets bezüglich der Eigenschaften Performance, Verfügbarkeit, Format, etc. homogene Einheiten. Der Benutzer muss selbst Sorge dafür tragen, dass die Dateien auf solchen SF-Pubsets angelegt werden, welche die gewünschten Dateieigenschaften am besten erfüllen können.

Folgende Erweiterungen bei SM-Pubsets dienen der konsequenten Fortführung der SMS-Idee:

- Im Unterschied zu SF-Pubsets können in SM-Pubsets Ressourcen verschiedener Eigenschaften zusammengefasst werden. Damit eignen sich SM-Pubsets als Ablageort von Dateien mit sehr unterschiedlichen Anforderungen. Auch wenn sich für eine in einem SM-Pubset liegende Datei im Laufe ihres Bestehens die Anforderungen ändern, können diese durch die Verlagerung der Datei innerhalb des Pubsets erfüllt werden. Damit besteht keine Notwendigkeit, dass eine Datei den SM-Pubset verlässt. Solange sich eine Datei innerhalb eines SM-Pubsets bewegt, bleibt ihre Adressierung unverändert (gleiche Catid).
- Auf einem SM-Pubset werden spezielle Möglichkeiten bereitgestellt, damit die Benutzer ihre Anforderungen auf logischer Ebene formulieren können, ohne den internen Aufbau des SM-Pubsets kennen zu müssen.
- Die Datenträgerkonfiguration eines SM-Pubsets kann im laufenden Pubset-Betrieb vergrößert oder verkleinert werden. Damit lässt sich seine Speicherkapazität einfach an schwankende Anforderungen anpassen. Auch der Austausch fehlerhafter oder veralteter Datenträger ist im laufenden Pubset-Betrieb möglich. Den Änderungen der Pubset-Konfiguration kann durch Umverteilung der Benutzerdateien innerhalb des SM-Pubsets Rechnung getragen werden. Zwar können auch SF-Pubsets im laufenden Betrieb vergrößert und verkleinert werden, doch bieten SM-Pubsets den Vorteil, dass sie dem Systembetreuer speziell für das Verkeinern bessere Voraussetzungen bieten. Bei SF-Pubsets dürften die Bedingungen, die für das Verkleinern erfüllt sein müssen, oft nur sehr schwer herzustellen sein, was den flexiblen Einsatz der Pubset-Ressourcen erheblich einschränkt.
- SM-Pubsets bieten im Vergleich zu SF-Pubsets bessere Voraussetzungen, um die Tätigkeiten für die Datenadministration automatisieren zu können.

- Innerhalb eines SM-Pubsets können kleinere Ausfalleinheiten gebildet werden. Fällt ein Volume eines SM-Pubsets aus, ist nur die Ausfalleinheit betroffen, der dieses Volume angehört.
- Wie SF-Pubsets sind SM-Pubsets abgeschlossene Dateibehälter, die neben den Dateien auch alle für den Zugriff auf diese Dateien relevanten Informationen und Datenträger enthalten. Im Vergleich zu den SF-Pubsets sind in SM-Pubsets aber auch die Hintergrundebenen und die Backup-Konfiguration des Pubsets konsequent der Behälteridee untergeordnet. Die Behältereigenschaft der SM-Pubsets ist insbesondere in einem Rechnerverbund von Bedeutung. Ein SM-Pubset kann dadurch mit geringem administrativen Aufwand von einem Rechner an einen anderen Rechner umgeschaltet werden, auch wenn er eine komplexe und umfangreiche Konfiguration enthält.
- Die Benutzerverwaltung auf SM-Pubsets ist gegenüber SF-Pubsets differenzierter, um den Verbrauch der unterschiedlichen Ressourcen durch die Benutzern kontrollieren zu können.
- Ein SM-Pubset erlaubt innerhalb des Pubsets die Koexistenz von Benutzern, die SMS-Features nutzen wollen, und Benutzern, die selbst Datenadministrationsaufgaben wahrnehmen wollen. Für spezielle Benutzer, welche die physikalischen Gegebenheiten kennen und ausnutzen wollen, können pubset-intern „benutzerkontrollierte“ Domänen vorgesehen werden. Benutzerkontrollierte Domänen sind beispielsweise für die bisherigen Privatplattennutzer attraktiv, da sie in die Pubset-Welt integriert sind und daher bestimmte Pubset-Funktionen genutzt werden können, die für Privatplatten nicht verfügbar sind (wie Security-Funktionen, F5-Rekonstruktion, etc.). Wünscht ein Benutzer des SM-Pubsets, der bisher seine Dateiverwaltung benutzerkontrolliert durchgeführt hat, SMS-Features zu nutzen, kann der Übergang innerhalb des SM-Pubsets mit einfachen administrativen Maßnahmen durchgeführt werden.

Die zentralen Pubset-Eigenschaften wie Behälterkonzept und Adressierungseinheit sind SF-Pubsets und SM-Pubsets gemeinsam. Damit ist für den Anwender der Übergang von SF-Pubsets auf SM-Pubsets SMS möglich, ohne einen harten Bruch vollziehen zu müssen.

### 3.3 Aufbau eines SM-Pubsets

Die oben aufgezählten Eigenschaften der SM-Pubsets werden durch einen im Vergleich zu SF-Pubsets komplexeren Aufbau ermöglicht. Dieser wird in [Bild 4](#) veranschaulicht. Obligatorischer Bestandteil eines SM-Pubsets ist seine **Verarbeitungsebene**. Sie besteht aus einem oder mehreren **Volume-Sets**. Ein Volume-Set besteht aus einem oder mehreren Volumes. Jede Datei liegt innerhalb eines einzigen Volume-Sets und kann sich nicht über mehrere Volume-Sets erstrecken.

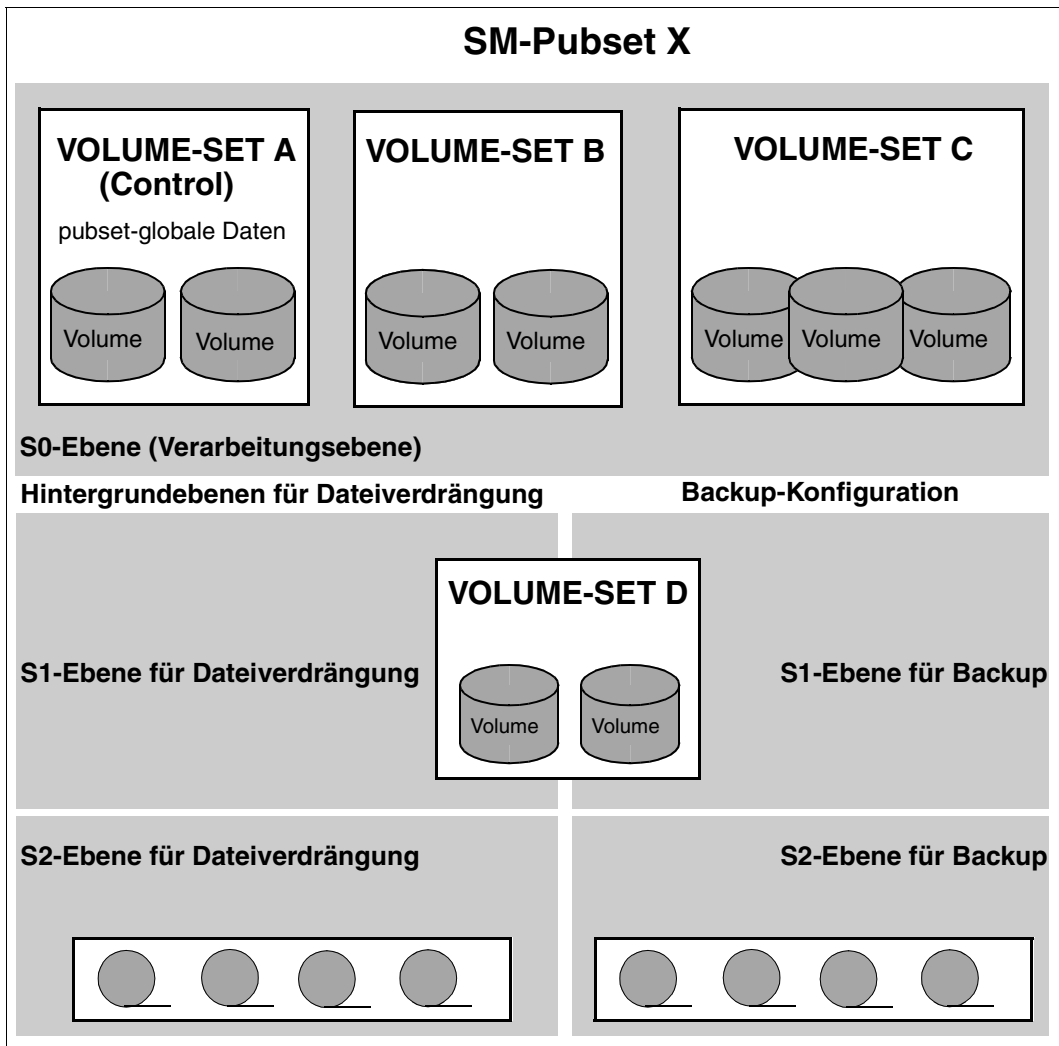


Bild 4: Aufbau eines SM-Pubsets

Unter den Volume-Sets nimmt der **Control-Volume-Set** eine Sonderrolle ein. Er ist ein unverzichtbarer Bestandteil des SM-Pubsets und enthält alle pubset-globalen Metadaten wie beispielsweise den Benutzerkatalog. Als Kern des SM-Pubsets sollte er vor allem für große SM-Pubsets, die aus vielen Volume-Sets bestehen, besonders ausfallsicher sein, beispielsweise indem er auf Volume-Ebene gespiegelt wird. Der Benutzer muss den Aufbau des SM-Pubsets nicht kennen. Insbesondere ist die Adressierung einer Datei unabhängig von ihrer Lage innerhalb des Pubsets.

Die Volume-Sets eines SM-Pubsets haben mehrere Funktionen:

### 1. Ressourcentyp

Volume-Sets bilden Einheiten, in welche die unterschiedlichen Ressourcentypen innerhalb eines SM-Pubsets gruppiert werden können. Entsprechend den Datenträgern, aus denen ein Volume-Set besteht, ist er durch bestimmte Eigenschaften charakterisiert. Diese werden berücksichtigt, wenn für eine Datei ein geeigneter Ablageort ermittelt wird. Ein Volume-Set muss homogen sein, damit er einen bestimmten Ressourcentyp repräsentieren kann, d.h. die verschiedenen in ihm zusammengefassten Volumes müssen in ihren Eigenschaften übereinstimmen. Es ist aber nicht notwendig, dass alle in einem SM-Pubset vorhandenen eigenschaftsgleichen Volumes zu einem einzigen Volume-Set gehören. Oft ist es sinnvoll, sie in mehrere eigenschaftsgleiche Volume-Sets zu gruppieren (siehe Volume-Set als Ausfalleinheit).

### 2. Rekonfigurationseinheit

Mit Ausnahme des Control-Volume-Sets sind die Volume-Sets Rekonfigurationseinheiten, d.h. ein Pubset kann im laufenden Pubset-Betrieb durch Aufnehmen von Volume-Sets vergrößert und durch Entfernen von Volume-Sets verkleinert werden. Bei diesen Rekonfigurationsmaßnahmen dürfen sich auf den involvierten Volume-Sets keine Dateien befinden. Um den Volume-Set von Dateien frei zu machen (Leeren eines Volume-Sets), müssen darauf liegende Dateien gelöscht oder auf andere Volume-Sets bzw. Hintergrundebenen verlagert werden. Hierfür werden besondere Funktionen zur Verfügung gestellt. Eine indirekte Unterstützung besteht darin, dass die pubset-globalen Metadatenfiles, welche nicht gelöscht und nicht verlagert werden können, auf den Control-Volume-Set konzentriert sind. Die Volume-Sets (einschließlich dem Control-Volume-Set) können ihrerseits durch Hinzunahme von Volumes vergrößert und durch Entfernen von Volumes verkleinert werden, wobei die involvierten Volumes keine Dateien oder Datei-Extents enthalten dürfen. Das Leeren einzelner Volumes eines Volume-Sets wird ebenfalls durch Systemfunktionen unterstützt, allerdings nicht in dem Umfang wie das Leeren von Volume-Sets.

### 3. Ausfalleinheit

Mit Ausnahme des Control-Volume-Sets sind die Volume-Sets in sich abgeschlossene Ausfalleinheiten innerhalb eines SM-Pubsets. Fällt ein Volume aus, gehen höchstens die Dateien des Volume-Sets verloren, dem dieses Volume angehört. Der betroffene Volume-Set kann als defekter Volume-Set aus dem SM-Pubset entfernt werden. Das System erstellt dabei eine Liste der Dateien, die der Volume-Set enthalten hat. Der übrige SM-Pubset wird durch den Ausfall des Volumes nicht beeinträchtigt. Auch der Pubset-Betrieb muss nicht unterbrochen werden.

Beim Ausfall des Control-Volume-Sets wird der komplette SM-Pubset zerstört. Er lässt sich aber aus den übrigen, noch intakten Volume-Sets wiederherstellen. Dabei müssen die auf dem ausgefallenen Control-Volume-Set gelegenen Daten, insbesondere die Metadaten des SM-Pubsets selbst, aus Sicherungen wiederhergestellt werden. Die Dateien auf den übrigen Volume-Sets bleiben in ihrem aktuellen Zustand erhalten. Damit bildet auch der Control-Volume-Set hinsichtlich der Ausfallsicherheit der Daten keinen „single point of failure“.

Die Eigenschaft der Volume-Sets, innerhalb des SM-Pubsets jeweils abgegrenzte Ausfalleinheiten zu bilden, steht in direktem Zusammenhang damit, dass sich eine Datei nicht über mehrere Volume-Sets erstrecken kann. Beim Festlegen der Volume-Set-Konfiguration für einen SM-Pubset, der eine größere Anzahl eigenschaftsgleicher Volumes enthält, ergibt sich daraus folgender Zielkonflikt:

- Ein SM-Pubset aus wenigen großen Volume-Sets hat den Nachteil, dass der Ausfall eines Volumes einen großen Teil des Pubsets betrifft und der Schaden (Anzahl der betroffenen Dateien) größer ist als beim Ausfall eines Volumes in einem SM-Pubset mit mehreren kleinen Volume-Sets.
- Ein SM-Pubset aus vielen kleinen Volume-Sets hat gegenüber einem SM-Pubset aus wenigen großen Volume-Sets den Nachteil, dass der vorhandene Platz nicht so flexibel genutzt werden kann. Ist z.B. die Kapazität eines kleinen Volume-Sets erschöpft, kann eine darauf befindliche Datei nicht vergrößert werden, auch wenn ein Volume eines anderen gleich gut geeigneten Volume-Sets ausreichend Platz bietet.

SF- und SM-Pubsets können mithilfe des Produkts HSMS mit **Hintergrundebenen für Dateiverdrängung** ausgestattet werden. Diese bilden zusammen mit der Verarbeitungsebene bezüglich Zugriffszeit, Verfügbarkeit und Kosten eine dreistufige Speicherhierarchie. Die Position innerhalb der Hierarchie wird durch die Bezeichnungen **S0-Ebene** (für die Verarbeitungsebene), **S1-Ebene für Hintergrundverdrängung** (für die online-verfügbare Hintergrundebene) und **S2-Ebene für Hintergrundverdrängung** (für die aus offline zugreifbaren Datenträgern gebildete Hintergrundebene) verdeutlicht. Wenn Dateien längere Zeit nicht bearbeitet werden, ist es empfehlenswert, sie auf eine kostengünstigere Hintergrundebene zu verdrängen. Bei SM-Pubsets wird die S1-Ebene durch einen zum Pubset gehörenden Volume-Set realisiert, der ausschließlich für HSMS reserviert sein muss. Als Datenträger für die S2-Ebene dient ein aus Magnetbändern, Magnetbandkassetten oder optischen Platten bestehender Band-Pool. Die Hintergrundebenen sind einem SM-Pubset

exklusiv zugeordnet und können im Unterschied zu den Hintergrundebenen von SF-Pubsets nicht durch mehrere Pubsets genutzt werden. Die für die Benutzung der Hintergrundebenen benötigten Metadaten, wie z.B. das Directory des Migrationsarchivs, befinden sich in dem SM-Pubset selbst und liegen dort auf dem Control-Volume-Set.

HSMS ermöglicht für SF-Pubsets und SM-Pubsets auch das Einrichten von **Backup-Archiven** für Dateisicherung, wobei dem **System-Backup-Archiv** besondere Bedeutung zukommt. Für SM-Pubsets bestehen gegenüber SF-Pubsets folgende Unterschiede:

- Bei SF-Pubsets wie bei SM-Pubsets sind als Sicherungsdatenträger Bänder und Platten möglich (Anmerkung: Im Folgenden wird der Begriff „Band“ für Magnetbänder, Magnetbandkassetten und optische Platten verwendet). Bei SM-Pubsets müssen die Plattensicherungen innerhalb des Pubsets auf einem speziell dafür vorgesehenen Volume-Set abgelegt werden, der nur von HSMS genutzt wird und auf dem sich keine normalen Benutzerdateien befinden. Dadurch wird sichergestellt, dass Originaldaten und Sicherungskopien sich zwar in dem gleichen SM-Pubset befinden (Behältereigenschaft), aber trotzdem auf unabhängigen Ausfalleinheiten abgelegt sind. Der aus Platten bestehende Teil der Backup-Konfiguration wird als **S1-Ebene für Backup** bezeichnet, während die Bänder als Sicherungsmedien die **S2-Ebene für Backup** bilden.
- Die Backup-Archive eines SM-Pubsets können nur Dateisicherungen des SM-Pubsets enthalten, dem sie zugeordnet sind.
- Die für die Benutzung der Backup-Archive benötigten Metadaten liegen zwingend auf dem SM-Pubset selbst. Innerhalb des SM-Pubsets werden sie durch das System bevorzugt auf dem Control-Volume-Set angelegt. Für die Directories der Backup-Archive kann der als Ablageort gewünschte Volume-Set durch den Systembetreuer vorgegeben werden. Für das System-Backup-Archiv sollte immer der Control-Volume-Set gewählt werden.

## 3.4 Dateiverwaltung in einem SM-Pubset

Die Dateiverwaltung in einem SM-Pubset wird in [Bild 5](#) illustriert. Bei SM-Pubsets - wie bei SF-Pubsets - wird der Pubset, in dem eine Datei angelegt wird, von den Benutzern implizit durch den Pfadnamen der Datei spezifiziert (Catid). Zwischen dem pubset-internen Ablageort einer Datei und ihrer Adressierung besteht keine Korrelation. Über die interne Struktur benötigen die Benutzer keine Kenntnisse. Die von einem SM-Pubset gebotenen Möglichkeiten werden über eine logische Schnittstelle sichtbar gemacht. Aus Benutzersicht erscheint ein SM-Pubset als ein Dateibehälter, der bestimmte Services für das Dateimanagement bereitstellt. Diese werden als **Datei-Services** bezeichnet. Sie lassen sich in Storage-Services (gebotene Performance, Verfügbarkeit,...) und HSMS-Management-Services (Steuerung der Dateiverdrängung auf Hintergrundebenen, Steuerung der Erstellung von Dateisicherungen, etc.) unterteilen. Das Anfordern der gewünschten Datei-Services durch die Benutzer erfolgt mithilfe von Dateiattributen, für welche den Dateien passende Werte zugewiesen werden müssen.

### 3.4.1 Storage-Services

Die Anforderung der Storage-Services kann durch folgende alternative Möglichkeiten erfolgen:

1. Direktattributierung

Bei der **Direktattributierung** spezifiziert der Benutzer für eine Datei die Werte für die ablageort-relevanten Attribute Performance, Verfügbarkeit, etc. einzeln.

2. Zuordnung einer Storage-Klasse

Der Systembetreuer kann **Storage-Klassen** einrichten, welche für die ablageortrelevanten Dateiattribute eine bestimmte Kombination von Attributwerten repräsentieren. Der Benutzer kann für eine Datei aus den vorhandenen Storage-Klassen diejenige auswählen, welche den jeweiligen Anforderungen am besten entspricht, und sie der Datei zuordnen.

Der von dem Benutzer für eine Datei gewünschte Storage-Service wird berücksichtigt, wenn für die Datei der Ablageort in der Verarbeitungsebene bestimmt wird. Dies ist der Fall, wenn eine Datei neu angelegt wird, wenn eine Datei von einer Hintergrundebene in die Verarbeitungsebene zurückgeholt wird, oder wenn der Benutzer für eine bestehende Datei die ablageortrelevanten Dateiattribute so ändert, dass sie mit dem bisherigen Ablageort unverträglich sind. Dabei ermittelt das System automatisch den am besten geeigneten Volume-Set. Das Systemverhalten kann dabei durch den Systembetreuer beeinflusst werden, um bestimmte rechenzentrumsspezifische Strategien zu realisieren. Das Instrument hierfür sind **Volume-Set-Listen**. Sie können durch den Systembetreuer eingerichtet und mit Storage-Klassen verknüpft werden.



Ordnet ein Benutzer einer Datei eine Storage-Klasse zu, der eine Volume-Set-Liste zugeordnet ist, wird die Datei durch das System bevorzugt auf einem Volume-Set angelegt, der dieser Volume-Set-Liste angehört. Dateien, denen keine Storage-Klasse oder eine Storage-Klasse ohne Volume-Set-Liste zugewiesen ist, werden bevorzugt auf solchen Volume-Sets angelegt, die keiner der von dem Systembetreuer eingerichteten Volume-Set-Listen angehören.

### 3.4.2 HSMS-Management-Services

Das Anfordern der Management-Services erfolgt mithilfe von **HSMS-Management-Klassen**. Sie müssen von dem Systembetreuer eingerichtet werden. Die HSMS-Management-Klassen repräsentieren bestimmte Verfahren für Dateisicherung (z.B. Sicherungshäufigkeit, Lebensdauer der Sicherungsversionen, etc.) sowie Regeln, welche die Verdrängung auf die Hintergrundebenen steuern (z.B. Verdrängbarkeit in Abhängigkeit der seit dem letzten Zugriff vergangenen Zeit). Indem der Benutzer einer Datei eine HSMS-Management-Klasse zuweist, unterwirft er sie den Sicherungs- bzw. Verdrängungsverfahren, welche durch diese HSMS-Management-Klasse repräsentiert werden. Für den speziellen Benutzerwunsch, bestimmte Dateien generell oder unter bestimmten Randbedingungen von der Verdrängung auf Hintergrundebenen auszuschließen, stehen zusätzlich **Migrationssperren** zur Verfügung.

Die Realisierung des Services, den eine HSMS-Management-Klasse repräsentiert, unterscheidet sich von der Realisierung der Storage-Services dadurch, dass sie nicht durch Automatismen erfolgt, die bereits in das System integriert sind. Dem Systembetreuer werden an Stelle dessen automatisierungsfreundliche Schnittstellen zur Verfügung gestellt, mit denen er Kommandoprozeduren oder Programme entwickeln kann, welche die vorgesehenen Verfahren abwickeln.

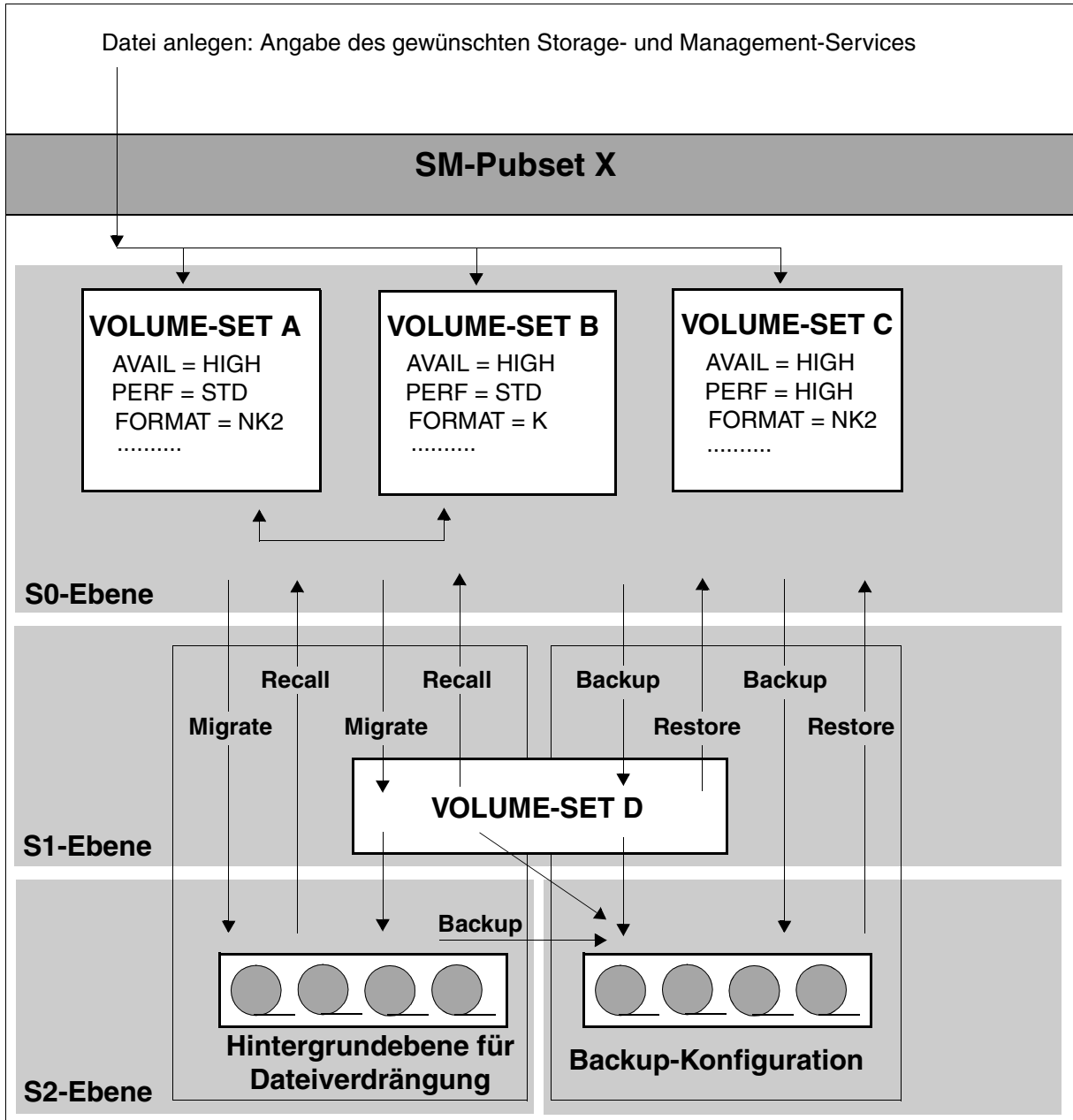


Bild 5: Dateiverwaltung in einem SM-Pubset

---

## 4 Merkmale eines SM-Pubsets

In diesem Kapitel erfolgt eine Aufstellung der Merkmale eines SM-Pubsets. Zu diesen gehören

- der **Aufbau** des SM-Pubsets,
- die **Zustände** des SM-Pubsets bzw. seiner einzelnen Volumes,
- die von dem Systembetreuer für die Pubset-Nutzung getroffenen **Einstellungen**.

### 4.1 Pubset-Kennung und Dateinamensraum

SM-Pubsets werden wie SF-Pubsets durch die Pubset-Kennung identifiziert. Sie geht als Catid in die Pfadnamen aller in dem Pubset katalogisierten Dateien und Jobvariablen ein und bestimmt dadurch den (Datei-) Namensraum, der dem Pubset zugeordnet ist. Der Namensraum umfasst damit auch auf Privatplatten oder Bändern liegende Dateien, die in dem Pubset katalogisiert sind. Die Pubset-Kennung wird in dem Pubset selbst vermerkt.

### 4.2 Aufbau der Verarbeitungsebene eines SM-Pubsets

[Bild 6](#) zeigt ein Beispiel für die Konfiguration der Verarbeitungsebenen eines SM-Pubsets. Sie besteht aus einem oder mehreren Volume-Sets (Maximalzahl 255). Jeder Volume-Set ist ein Behälter für ganze Dateien. In den Pfadnamen einer Datei geht nicht ein, auf welchem Volume-Set sie liegt. Unter den verschiedenen Volume-Sets eines SM-Pubsets kommt dem Control-Volume-Set als Behälter für pubset-globale Metadateien eine Sonderrolle zu. Es sollte daher besonders ausfallsicher sein. Die zu einem SM-Pubset gehörenden Volume-Sets und ihre Merkmale sind in der Pubset-Konfigurationsdatei `:<catid>: $TSOS.SYS.PUBSET.CONFIG` vermerkt. Sie wird implizit bei der Pubset-Generierung erzeugt und liegt immer auf dem Control-Volume-Set.

Im Normalfall besteht ein Volume-Set aus einem oder mehreren Volumes (Maximalzahl 255). Im Rahmen der Pubset-Rekonfiguration kann sich aber auch der spezielle Zustand ergeben, dass ein Volume-Set in der Konfigurationsdatei des SM-Pubsets eingetragen ist, ohne dass er über eine physikalische Konfiguration verfügt. Volume-Set C in [Bild 6](#) illustriert einen solchen Fall. Dem Control-Volume-Set muss immer eine intakte physikalische Konfiguration zugewiesen sein. Unter den Volumes eines Volume-Sets (mit einer physikalischen Konfiguration) hat die Volres eine spezielle Rolle. In ihrem SVL (Standard Volume Label) ist der Volume-Katalog hinterlegt, in dem alle zu dem Volume-Set gehörenden Volumes vermerkt sind. Jedes Volume wird durch seine VSN (Volume Serial Number) identifiziert. Die VSN ist durch eine Namenskonvention mit der Kennung des Volume-Sets, dem das Volume angehört, korreliert. Diese Namenskonvention erlaubt, dass sowohl aus der VSN die Volume-Set-Kennung bestimmt werden kann, als auch umgekehrt aus der Volume-Set-Kennung die VSN der Volres abgeleitet werden kann. Zwischen der VSN und der Kennung des SM-Pubsets besteht kein Zusammenhang.

Volume-Sets, die keinem SM-Pubset angehören, heißen freie Volume-Sets. Sie dienen ausschließlich als Elemente der Pubset-Rekonfiguration und sind nicht eigenständig für Dateiverarbeitung nutzbar (Anmerkung: Freie Volume-Sets dürfen nicht mit SF-Pubsets verwechselt werden!). Im Normalfall sind freie Volume-Sets leer, d.h. sie enthalten - abgesehen von dem Dateikatalog - keine Dateien. Erst nach ihrer Einbindung in einen SM-Pubset können Benutzerdateien auf ihnen angelegt werden. Freie Volume-Sets können mithilfe von SIR durch den Systembetreuer erzeugt werden. Sie entstehen auch dann, wenn Volume-Sets mit den Möglichkeiten der dynamischen Pubset-Rekonfiguration (physikalisch) aus einem SM-Pubset entfernt werden. Um mithilfe der Pubset-Rekonfiguration einen freien Volume-Set in einen SM-Pubset aufzunehmen, sind folgende Einzelschritte durch den Systembetreuer erforderlich:

- Eintragen des Volume-Sets in die Konfigurationsdatei,
- Bereitstellen eines freien Volume-Sets,
- Aufnahme des freien Volume-Sets in den SM-Pubset.

Freie Volume-Sets, die nicht leer sind, treten nur in Sondersituationen auf, z.B. wenn der Control-Volume-Set ausfällt und der SM-Pubset als Einheit zerstört ist, oder wenn ein Volume-Set aus einem SM-Pubset zwangsentefernt wird. Freie Volume-Sets, die nicht leer sind, können nicht unmittelbar genutzt werden; sie können im Unterschied zu leeren freien Volume-Sets auch nicht direkt durch Pubset-Rekonfiguration in SM-Pubsets aufgenommen werden. Damit auf die Dateien von freien Volume-Sets, die nicht leer sind wieder zugegriffen werden kann, muss zunächst im Rahmen einer speziellen Inbetriebnahme eine Konversion in SF-Pubsets erfolgen. Ein wichtiger Anwendungsfall dieser Funktion ist die Wiederherstellung eines SM-Pubsets nach einem Ausfall des Control-Volume-Sets.

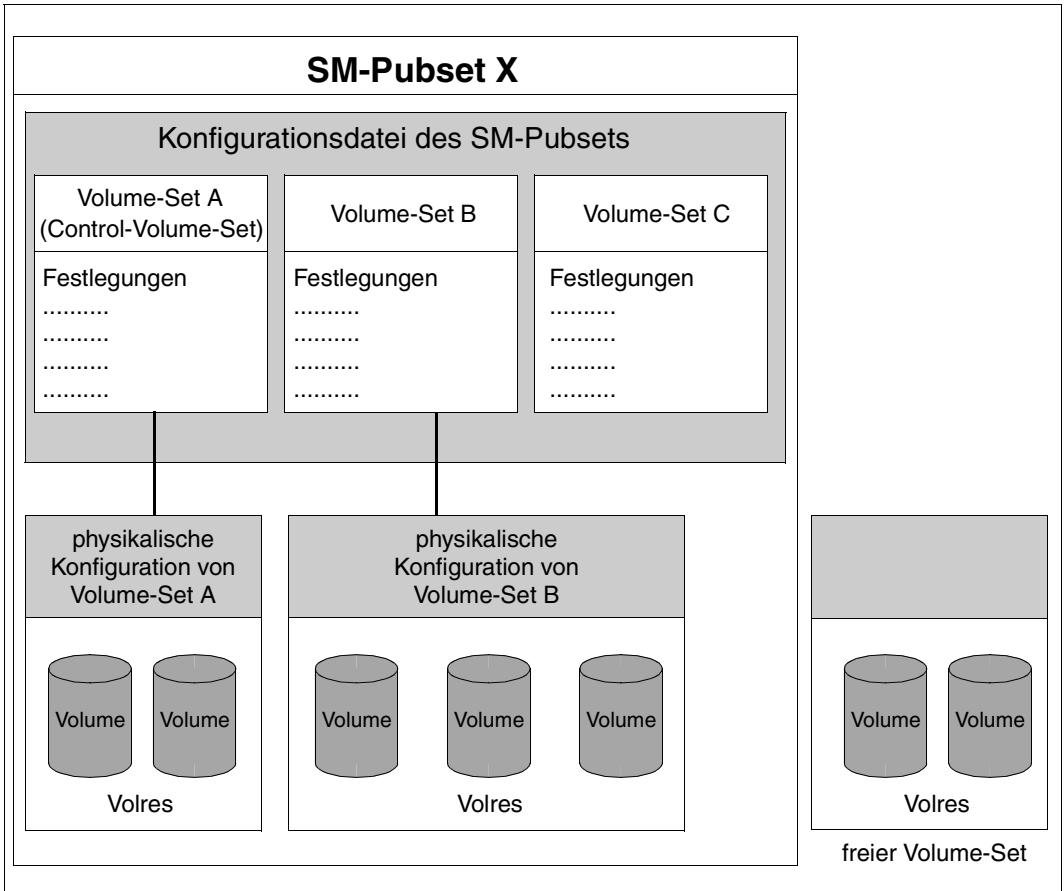


Bild 6: Konfiguration eines SM-Pubsets (ohne Hintergrundebenen und Backup-Medien)

## 4.3 Merkmale der einzelnen Volume-Sets eines SM-Pubsets

### 4.3.1 Konfigurationszustand eines Volume-Sets

Im Normalfall verfügen die Volume-Sets eines SM-Pubsets über eine intakte physikalische Volume-Set-Konfiguration, d.h. sie bestehen aus einem oder mehreren Volumes, auf welche ohne Probleme zugegriffen werden kann. Allerdings können sich für die Konfiguration eines Volume-Sets auch spezielle Zustände ergeben, wenn beispielsweise als Folge von Plattendefekten oder des Ausfalls von Datenpfaden die Zugriffe auf einzelne Volumes nicht oder nur eingeschränkt möglich sind. Ein weiterer Sonderzustand ergibt sich im Rahmen der Pubset-Rekonfiguration. Die Aufnahme eines Volume-Sets in einen SM-Pubset gliedert sich in die Schritte

- logische Volume-Set-Hinzunahme
- Hinzunahme eines freien Volume-Sets.

Nach dem ersten Schritt ist der Volume-Set in die Pubset-Konfigurationsdatei eingetragen, es sind ihm aber noch keine Volumes zugeordnet. Entsprechendes gilt in umgekehrter Reihenfolge für das Entfernen eines Volume-Sets aus einem SM-Pubset. Insgesamt wird hinsichtlich der physikalischen Konfiguration eines Volume-Sets zwischen folgenden Konfigurationszuständen unterschieden:

normal use	Der Volume-Set verfügt über eine intakte physikalische Konfiguration. Der Control-Volume-Set muss immer im Zustand „normal use“ sein.
defective	Als Folge von Defekten kann die physikalische Konfiguration des Volume-Sets nicht mehr genutzt werden.
in hold	Auf die Volumes des Volume-Sets kann vorübergehend nur eingeschränkt zugegriffen werden (z.B. wegen Problemen bei den Zugriffspfaden, welche aber behebbar sind). Während ein Volume-Set des Pubsets sich in diesem Zustand befindet, ist auch die Nutzung des restlichen SM-Pubsets beeinträchtigt.
defined only	Der Volume-Set ist in der Konfigurationsdatei eingetragen, ihm ist aber keine physikalische Konfiguration zugeordnet. Die Volume-Set-Merkmale, die sich aus der physikalischen Konfiguration ergeben, sind in diesem Zustand undefiniert. Ein freier Volume-Set gehört nicht zur Konfiguration eines SM-Pubsets, auch wenn in der Konfigurationsdatei des Pubsets ein Eintrag für die gleiche Kennung vorhanden ist

Die genaue Bedeutung der einzelnen Zustände sowie die möglichen Zustandsübergänge werden im Rahmen der dynamischen Pubset-Rekonfiguration erläutert.

Die Konfigurationszustände eines Volume-Sets sind nicht rechner-spezifisch, d.h. sie sind unabhängig davon definiert, ob und an welchen Rechnern der SM-Pubset, dem der Volume-Set angehört, in Betrieb ist. Die Konfigurationszustände werden in der Pubset-Konfigurationsdatei vermerkt. Über freie Volume-Sets enthält die Pubset-Konfigurationsdatei keine Informationen. Sie werden nicht als einem SM-Pubset zugehörig betrachtet.

### 4.3.2 Eigenschaftsprofile für die Volume-Set-Selektion

Die einzelnen Volume-Sets eines SM-Pubsets können sich in den Eigenschaften Format, Größe der Allocation-Unit, Performance-Verhalten und Verfügbarkeit unterscheiden. Diese Eigenschaften werden von dem System (Volume-Set-Selektion) bei der Auswahl desjenigen Volume-Sets berücksichtigt, der sich für eine Datei jeweils am besten eignet.

- Die Größe der Allocation-Unit und das Format werden einem Volume-Set bei der Volume-Set-Initialisierung/Formatierung aufgeprägt und haften ihm für die gesamte Dauer seines Bestehens fest an. Insbesondere sind sie auch für freie Volume-Sets definiert.
- Das Performance- und das Verfügbarkeitsprofil der zu dem Pubset gehörenden Volume-Sets werden durch den Systembetreuer festgelegt, wenn der SM-Pubset gebildet oder wenn ein Volume-Set im Rahmen einer Pubset-Rekonfiguration in den Pubset aufgenommen wird. Sie werden in der Konfigurationsdatei des SM-Pubsets vermerkt. Für freie Volume-Sets sind sie undefiniert.

Die Beschreibung des Performance- und Verfügbarkeitsprofils eines Volume-Sets ist nur in dem Detaillierungsgrad möglich, der von der Volume-Set-Selektion berücksichtigt wird. Weitergehende Unterschiede (z.B. zwischen der Verfügbarkeit von DRV-, oder REMOTE-COPY-Platten) können nicht spezifiziert werden. Im Folgenden werden die einzelnen Eigenschaften genauer betrachtet.

#### Format

Mögliche Formate eines Volume-Sets sind K, NK2, NK4. Jeder Volume-Set ist bezüglich des Formats homogen (wie bei SF-Pubset), d.h. das Format aller zu dem Volume-Set gehörenden Volumes ist gleich.

#### Größe der Allocation-Unit

Für die Allocation-Unit eines Volume-Sets sind die Werte 4KB, 6KB, 64KB möglich. Ein Volume-Set ist bezüglich der Allocation-Unit homogen, d.h. alle seine Volumes haben die gleiche Allocation-Unit.

## Verfügbarkeit

Die Verfügbarkeit eines Volume-Sets ist durch das Attribut AVAILABILITY charakterisiert. Mögliche Werte sind STD und HIGH. Zum Erreichen der erhöhten Verfügbarkeit eines Volume-Sets sind verschiedene Mittel, wie z.B. DRV oder REMOTE-COPY gegeben. Bei der Wertzuweisung für das Volume-Set-Attribut AVAILABILITY wird jedoch nicht überprüft, ob die tatsächliche Eigenschaft des Volume-Sets den zugeordneten Werten entspricht. Es obliegt dem Systembetreuer, der Realität entsprechende Werte zu wählen. Dass die Zuordnung des AVAILABILITY-Attributs in den Verantwortungsbereich des Systembetreuers gelegt wird und nicht automatisch durch das System erfolgt, trägt folgenden Gesichtspunkten Rechnung:

- Dem Systembetreuer muss es möglich sein, eigene Erfahrungen in die Bewertung der Verfügbarkeit einfließen zu lassen. Eine starr durch das System vorgenommene Bewertung könnte mit der Einschätzung des Systembetreuers nicht übereinstimmen.
- Es ist sinnvoll, die Bewertung eines Mediums in Relation zu den Eigenschaften der übrigen in einem Pubset vorhandenen Datenträger vorzunehmen. Bei der Aufnahme neuer Hardware in einen Pubset kann eine Anpassung der bisherigen Bewertungen empfehlenswert sein.
- Der von dem Systembetreuer zugewiesene Verfügbarkeitswert entspricht der langfristigen Positionierung, die dem Volume-Set innerhalb des SM-Pubsets zgedacht ist. Eine Änderung des AVAILABILITY-Werts von HIGH zu STD ist mit erheblichem organisatorischem Aufwand verbunden, da zuvor insbesondere alle auf dem Volume-Set liegenden Dateien mit dem Dateiattribut AVAILABILITY=HIGH auf einen anderen hoch verfügbaren Volume-Set gebracht werden müssen. Der Verfügbarkeitswert wird sinnvollerweise nur dann modifiziert, wenn sich die Positionierung des Volume-Sets langfristig ändert, nicht jedoch um auf vorübergehende Schwankungen (z.B. bei kurzfristiger Änderung des Betriebszustands eines Volumens von „Dual“ nach „Mono“ bei DRV) zu reagieren. Die explizite Zuordnung bei der Pubset-Generierung bzw. Pubset-Rekonfiguration soll dem Systembetreuer den langfristigen Charakter dieser Festlegung bewusst machen.



## Performance

Die Beschreibung des Performance-Profiles eines Volume-Sets erfolgt durch die Angabe eines Performance-Spektrums und die Spezifikation von Einschränkungen bezüglich Schreibsicherheit. Diese komplexe Struktur ist notwendig, um für die Volume-Set-Selektion das Verhalten von solchen Volume-Sets ausreichend charakterisieren zu können, denen über PFA (Performant File Access) Caches zugeordnet sind. Bei PFA steuern die Benutzer über die Dateiattribute für Performance sowohl die Auswahl eines geeigneten Volume-Sets als auch die aktive Cache-Nutzung bei der Bearbeitung der Dateien.

- Ein mit einem Cache ausgestatteter Volume-Set kann im Allgemeinen ein Spektrum unterschiedlicher Performance-Anforderungen vonseiten der Benutzer abdecken (STD, HIGH, VERY-HIGH). Der Volume-Set ist damit gleichzeitig als Ablageort für Dateien mit der Anforderung PERFORMANCE=\*STD, Dateien mit der Anforderung PERFORMANCE=\*HIGH und für Dateien mit der Anforderung PERFORMANCE=\*VERY-HIGH geeignet.

### *Hinweis*

Eine gleichzeitige Zuordnung mehrerer unterschiedlicher Caches zu einem Volume-Set ist nicht möglich. Die unterschiedlichen Performance-Werte korrespondieren mit verschiedenen Arten der Cache-Nutzung, welche bei der Bearbeitung einer Datei möglich sind:

PERFORMANCE=*STD	keine Cache-Nutzung,
PERFORMANCE=*HIGH	Caching mit Verdrängung,
PERFORMANCE=*VERY-HIGH	Caching ohne Verdrängung

Die Art der Cache-Nutzung kann innerhalb des gleichen Volume-Sets individuell den spezifischen Performance-Anforderungen der einzelnen Dateien angepasst werden.

- Die verschiedenen Cache-Medien (Globalspeicher, Hauptspeicher) unterscheiden sich im Hinblick auf die Schreibsicherheit. Ein Cache gilt dann als schreibsicher, wenn für Ausgabedaten, die in ihm zwischengepuffert werden, kein höheres Ausfallrisiko besteht, als wenn sie unmittelbar auf Platte geschrieben würden. Nicht-schreibsichere Caches dürfen generell für die Performance-Verbesserung bei lesenden Dateizugriffen verwendet werden, für Schreibzugriffe sind sie aber nur dann nutzbar, wenn der Benutzer Einschränkungen hinsichtlich der Schreibsicherheit akzeptiert und dies durch das Dateiattribut DISK-WRITE=\*BY-CLOSE zum Ausdruck bringt. Nur wenn diese Voraussetzungen erfüllt sind, können Volume-Sets mit einem nicht-schreibsicherem Cache das volle Performance-Spektrum (STD, HIGH, VERY-HIGH) zur Verfügung stellen. Für Schreibzugriffe, bei denen eine eingeschränkte Schreibsicherheit nicht in Kauf genommen wird, darf ein nicht-schreibsicherer Cache nicht genutzt werden. Das Performance-Spektrum eines Volume-Sets mit einem nicht-schreibsicheren Cache reduziert

sich in diesem Fall auf den Wert PERFORMANCE=\*STD. Volume-Sets, welche mit einem schreibsicheren Cache ausgestattet sind, bieten uneingeschränkt das volle Performance-Spektrum (STD, HIGH, VERY-HIGH).

- Wenn der Systembetreuer für einen Volume-Set die Vorgabe WRITE CONSISTENCY=\*BY-CLOSE trifft, geht das System davon aus, dass auf dem Volume-Set liegende Dateien hinsichtlich ihrer Verfügbarkeit nicht eingeschränkt sind, wenn ihnen keine erhöhten Performance geboten wird (d.h. auf Caching verzichtet wird). Damit ist beispielsweise die Angabe PERFORMANCE=\*VERY-HIGH und WRITE-CONSISTENCY=\*BY-CLOSE für das Performance-Profil eines Volume-Sets mit der Angabe PERFORMANCE=(STD,VERY-HIGH) und WRITE-CONSISTENCY=\*BY-CLOSE gleichbedeutend.
- Performance-Erhöhungen lassen sich neben der Nutzung von Caches auch durch die Verwendung von Volume-Sets erreichen, die aus speziellen hochperformanten Volumens wie im Globalspeicher emulierten Volumens gebildet sind. Das Performance-Spektrum solcher Volume-Sets lässt sich für die Volume-Set-Selektion durch einen einzigen Performance-Wert ausreichend charakterisieren. Einen speziellen Fall stellen über ADM-PFA (Administrator Performant File Access) genutzte Caches dar. Bei diesen steuern nicht die Benutzer über Dateiattribute das Caching, sondern der Systembetreuer (Cache-Administrator) trifft für das Caching bestimmter Dateien gezielte Vorkehrungen. In solche Caches muss auch der Ablageort der Dateien miteinbezogen werden. Der Ablageort kann über physikalische Allokierung (siehe [Seite 197](#)) durch den Systembetreuer festgelegt werden. Die performance-bezogenen Dateiattribute und das Performance-Profil eines Volume-Sets sind bei Caching über ADM-PFA ohne Bedeutung.

*Anmerkung zur Terminologie*

Das Performance-Profil eines Volume-Sets kann durch das Kommando MODIFY-PUBSET-DEFINITION-FILE eingestellt werden. Zwischen den in dieser Darstellung verwendeten Begriffen und den in der Kommandosyntax definierten Operanden bzw. Operandenwerten zum Erstellen eines Performance-Profils besteht folgende Relation:

<b>In dieser Darstellung verwendete Terminologie</b>	<b>Kommandosyntax</b>
Performance-Profil	PERFORMANCE-ATTRIBUTES
Performance-Spektrum	PERFORMANCE
Schreibsicherheit bei erhöhter Performance ja (gegeben) nein (nicht gegeben)	WRITE-CONSISTENCY IMMEDIATE BY-CLOSE

*Empfehlungen für Performance-Profil*

Analog wie das Verfügbarkeitsprofil wird das Performance-Profil eines Volume-Sets nicht automatisch aus der physikalischen Konstellation ermittelt. Es obliegt dem Systembetreuer, das Performance-Profil der Volume-Sets so festzulegen, dass es die realen Verhältnisse richtig beschreibt. Bei der Beschreibung des Performance-Spektrums muss vor allem darauf geachtet werden, dass die einzelnen Volume-Sets im Vergleich zu den übrigen im Pubset vorhandenen Volume-Sets korrekt eingestuft werden. Die in [Tabelle 1](#) vorgeschlagenen Einstellungen für die verschiedenen Cache-Medien können folglich hinsichtlich des Performance-Spektrums nur als Orientierungshilfe dienen, da sie die Gegebenheiten konkreter Konfigurationen nicht berücksichtigen. Für die Festlegung der Schreibsicherheit bei erhöhter Performance sind hingegen Empfehlungen möglich, die in jeder Konfiguration anwendbar sind.

Volume-Set-Konfiguration	empfohlenes Performance-Profil	
	Performance-Spektrum	Schreibsicherheit bei erhöhter Performance (HIGH und VERY-HIGH) WRITE-CONSISTENCY
Volume-Set, der aus Volumes ohne besondere Performance-Eigenschaften besteht und dem kein Cache zugeordnet ist	STD	irrelevant
Volume-Set, dem ein Hauptspeicher als Cache-Medium zugeordnet ist	STD, HIGH, VERY-HIGH	BY-CLOSE (Schreibsicherheit ist nicht gegeben)
Volume-Set, dem ein flüchtiger Globalspeicher zugeordnet ist (VOLATILITY=*YES; Kommando MODIFY-PUBSET-CACHE-ATTRIBUTES)	STD, HIGH, VERY-HIGH	BY-CLOSE (Schreibsicherheit ist nicht gegeben)
Volume-Set, dem ein nicht-flüchtiger Globalspeicher zugeordnet ist (VOLATILITY=*NO; Kommando MODIFY-PUBSET-CACHE-ATTRIBUTES)	STD, HIGH, VERY-HIGH	IMMEDIATE (Schreibsicherheit ist gegeben)
Volume-Set, der aus im Globalspeicher emulierten Volumes besteht	VERY-HIGH	IMMEDIATE (Schreibsicherheit ist gegeben)

Tabelle 1: Empfehlungen für Performance-Profile

### 4.3.3 Physikalische Volume-Set-Merkmale

Wie bereits erwähnt, muss der Systembetreuer dafür sorgen, dass die Eigenschaftsprofile der Volume-Sets durch die physikalische Volume-Set-Konfiguration erfüllt werden. Hierfür stehen zurzeit auf „physikalischer Ebene“ eine Vielzahl von Möglichkeiten zur Verfügung, die sich in Zukunft noch vergrößern werden. Wir beschränken uns hier auf die Darstellung einzelner exemplarischer Aspekte:

#### Physikalische Eigenschaften der einzelnen Volumes

Die physikalischen Eigenschaften eines Volume-Set sind durch die Eigenschaften der ihm zugehörigen Volumes sowie deren I/O-Konfiguration (Controller, Zugangspfade, etc.) bestimmt. Damit den Volume-Sets sinnvolle Eigenschaftsprofile zugeordnet werden können, sollte die Volume-Konfiguration der Volume-Sets homogen sein, d.h. der Systembetreuer sollte nur Volumes zu einem Volume-Set zusammenfassen, deren Eigenschaften mit dem vorgesehenen Eigenschaftsprofil harmonieren. Bei bestimmten Eigenschaften wird die Homogenität des Volume-Sets durch das System erzwungen (Größe der Allocation-Unit, Format, im Globalspeicher emulierte Volumes, etc.).

#### Maximale I/O-Länge eines Volume-Sets

Eine der physikalischen Eigenschaften, die ein Volume charakterisieren, ist seine maximale I/O-Länge, d.h. die maximale Anzahl von Blöcken, die bei einem I/O-Auftrag transferiert werden können. Der kleinste Wert aller einem Volume-Set angehörenden Volumes bestimmt die maximale I/O-Länge des Volume-Sets. Wird in einen Volume-Set mithilfe der Pubset-Rekonfiguration ein weiteres Volume hinzugenommen, kann sich die maximale I/O-Länge des Volume-Sets verkleinern. Die maximale I/O-Länge eines Volume-Sets wird durch bestimmte privilegierte Anwendungen (ARCHIVE, CCOPY) zur Optimierung der Ein-/Ausgaben herangezogen.

## Volume-Sets als unabhängige Ausfalleinheiten

Das Pubset-Management unterstützt die Möglichkeit, dass die einzelnen Volume-Sets eines SM-Pubsets voneinander unabhängige Ausfalleinheiten darstellen, d.h. bei dem Ausfall eines Volumes ist nur der Volume-Set betroffen, dem dieses angehört. Falls es sich dabei nicht um den Control-Volume-Set handelt, kann der restliche SM-Pubset weiterhin in Betrieb bleiben. Allerdings sind dafür auch Maßnahmen auf physikalischer Ebene erforderlich: Volume-Sets, die bezüglich des Ausfalls unabhängig sein sollen, müssen auf physikalische Konfigurationen abgebildet werden, die hinsichtlich Ausfall unabhängig voneinander sind. Beispielsweise müssen die Zugangspfade (Kanäle, Controller, etc.) zu den Volumes von Volume-Sets, die bezüglich Ausfallsicherheit unabhängig sein sollen, voneinander entkoppelt sein. Auch die den Volume-Sets zugeordneten Cache-Konfigurationen sind zu berücksichtigen. Der Control-Volume-Set hat auf Grund seiner Eigenschaft als zentraler Metadatenbehälter des SM-Pubsets eine Sonderrolle, aber auch er bildet keinen „single point of failure“. Bei seinem Ausfall ist es möglich, zunächst die übrigen nicht beschädigten Volume-Sets im Rahmen einer speziellen Inbetriebnahme in SF-Pubsets umzuwandeln und diese dann anschließend mit SMPGEN wieder zu einem SM-Pubset zusammenfassen. Die auf dem Control-Volume-Set selbst befindlichen Daten, die bei der Bildung des SM-Pubsets nicht automatisch im gewünschten Zustand rekonstruiert werden können, wie z.B. die HSMS-Metadaten, die Kataloge für Storage-Klassen und Volume-Set-Listen oder der Benutzerkatalog, muss der Systembetreuer aus Sicherungen wiederherstellen.

## Ausfallsicherheit

Zum Bereitstellen besonders ausfallsicherer Volume-Sets können DRV-, RAID oder REMOTE-COPY Funktionen genutzt werden. REMOTE-COPY-Funktionalität wird beispielsweise von Symmetrix-Systemen mit Symmetrix Remote Data Facility (SRDF) geboten.

## Volume-Sets mit zugeordnetem Cache

Voraussetzung für die Nutzung von Caches ist das Bereitstellen bestimmter Cache-Konfigurationen. Den einzelnen Volume-Sets des SM-Pubsets können individuell verschiedene Cache-Bereiche zugeordnet werden. Die für die Nutzung von Caches notwendigen Vorkehrungen hängen stark von dem jeweiligen Cache-Medium ab. Im Folgenden werden exemplarisch Volume-Sets betrachtet, denen über PFA Globalspeicher-Caches zugeordnet sind.

- *Zuordnen der Cache-Konfiguration*

Für jeden Volume-Set ist durch den Systembetreuer festzulegen, ob es mit einem Cache betrieben werden soll, welches Cache-Medium gewünscht wird, und wie groß der Cache-Bereich für das Volume-Set sein soll. Bei Cache-Nutzung sind ferner bestimmte Cache-Betriebsparameter zu spezifizieren. Der Umfang und die Bedeutung der Betriebsparameter sind für die einzelnen Cache-Medien unterschiedlich. Die Angaben des Systembetreuers bestimmen die defined-Werte für die Cache-Konfiguration der Volume-Sets. Sie werden für die einzelnen Volume-Sets in der Konfigurationsdatei des SM-Pubsets vermerkt. Änderungen der defined-Werte sind (im Unterschied zu SF-Pubsets!) nur während einer Pubset-Session möglich.

Die defined-Werte für die Volume-Set-Cache-Konfiguration kommen zum Tragen,

- wenn der SM-Pubset, dem der Volume-Set angehört, an einem Rechner in Betrieb genommen wird,
- wenn einem SM-Pubset durch dynamische Pubset-Rekonfiguration ein neuer Volume-Set physikalisch hinzugefügt wird, d.h. wenn einem Volume-Set, der bisher im Zustand „defined only“ war, ein freier Volume-Set zugewiesen wird,
- wenn durch das Kommando START-PUBSET-CACHING der Cache für einen im Zustand „normal use“ befindlichen Volume-Set aktiviert wird.

Dabei versucht das System, den Volume-Set entsprechend den eingestellten defined-Werten mit einem neuen Cache-Bereich zu konnektieren. In bestimmten Situationen können sich Abweichungen in der Größe ergeben. Die Größe des aktuell zugeordneten Cache-Bereichs sowie die für den Volume-Set aktuell wirksamen Cache-Betriebsparameter werden durch die current-Werte der (Volume-Set-) Cache-Konfiguration beschrieben.

Das Diskonnektieren des Cache-Bereichs eines Volume-Sets erfolgt:

- wenn der SM-Pubset, dem der Volume-Set angehört, außer Betrieb genommen wird,
- wenn dem Volume-Set durch dynamische Pubset-Rekonfiguration die physikalische Volume-Konfiguration entzogen wird,
- wenn der Cache eines Volume-Sets durch das Kommando STOP-PUBSET-CACHING deaktiviert wird

Die Volume-Sets von nicht in Betrieb befindlichen Pubsets sind im Normalfall nicht mit Cache-Bereichen konnektiert. In Sonderfällen, z.B. nach einem System-Crash, können auch den Volume-Sets von exportierten SM-Pubsets Cache-Bereiche zugeordnet sein.

- *Einrichten der GS-Partition*

Als Voraussetzung dafür, dass für Volume-Sets eines SM-Pubsets GS-Cache-Bereiche bereitgestellt werden können, müssen durch den Systembetreuer GS-Partitionen für den Pubset eingerichtet worden sein. Diese sind dem SM-Pubset exklusiv zugeordnet, auch wenn er nicht in Betrieb ist. Sie dienen als Reservoir für alle GS-Cache-Bereiche, die mit einem der Volume-Sets des SM-Pubsets konnektiert werden. Die Korrelation zwischen GS-Partitionen und SM-Pubset erfolgt durch eine Namenskonvention für GS-Partitionen.

- *Berücksichtigung der Verfügbarkeit des Cache-Mediums*

Caches sind mit einem bestimmten Ausfallrisiko behaftet. Bei einer nur lesenden Dateiverarbeitung können Caches unabhängig von ihrer Zuverlässigkeit als Zwischenpuffer verwendet werden, ohne dass die Ausfallsicherheit der Datei beeinflusst wird. Bei der Zwischenpufferung von schreibenden Zugriffen hingegen führt der Ausfall des Caches zu Datenverlusten, welche die Zerstörung der gesamten Datei zur Folge haben können. In die Beurteilung der Ausfallsicherheit eines Volume-Sets/SF-Pubsets, der mit einem Cache betrieben wird, muss daher neben der Ausfallsicherheit der Volumes und ihrer Zugangspfade auch die Verfügbarkeit eines zugeordneten Caches einbezogen werden.

Im Fall von GS-Caches sind folgende Aspekte zu beachten:

- Ausfallsicherheit der Stromversorgung für GS-Caches (VOLATILITY)

GS-Caches können durch spezielle Vorkehrungen (Batteriebetrieb, unterbrechungsfreie Stromversorgung etc.) gegen Stromausfall abgesichert werden. Ohne diese Maßnahmen sind die GS-Caches flüchtig, d.h. bei Stromausfall gehen die in ihnen abgelegten Daten verloren. Dateien, für welche Ausgabedaten in den betroffenen GS-Bereichen zwischengepuffert waren, sind anschließend nicht mehr in ihrem aktuellen Bearbeitungszustand erhalten.

- Ausfallsicherheit der Cache-Medien selbst

Neben der Stromversorgung für GS-Caches sind auch die Cache-Medien selbst mit einem bestimmten Ausfallrisiko behaftet. Durch die Verwendung von redundanter Datenaufzeichnung kann das Ausfallrisiko für Daten, die in einem GS-Cache gepuffert werden, reduziert werden.

– Ausfallsicherheit bei System-Crash

Nicht flüchtige wie auch flüchtige GS-Caches bieten generell Ausfallsicherheit bei einem System-Crash, da alle für die Nutzung der Cache-Bereiche erforderlichen Metadaten in dem Cache selbst abgelegt sind. Für die Situation nach einem System-Crash oder nach einer erzwungenen Außerbetriebnahme des Pubsets ist zu unterscheiden, ob die dem Pubset zugeordnete GS-Cache-Konfiguration rechnerlokal oder rechnerübergreifend genutzt wird. Bei einer rechnerlokalen Nutzung ist die Wiederherstellung von Dateien, von denen sich noch Schreibdaten in dem GS-Cache befinden, erst dann möglich, wenn der SM-Pubset wieder am gleichen Rechner in Betrieb genommen wird. Bei einem rechnerübergreifenden GS-Cache kann die Wiederherstellung auch von anderen Rechnern aus erfolgen, an denen der Pubset in Betrieb genommen wird, falls von ihnen aus auf den GS-Cache zugegriffen werden kann. Im Shared-Pubset-Betrieb erfolgt bei einem rechnerübergreifend genutzten GS-Cache nach dem System-Crash eines am Verbund teilnehmenden Rechners die Wiederherstellung der betroffenen Dateien am aktuellen Master-Rechner oder im Rahmen des Master-Wechsels.

In einen GS-Cache, der nicht verfügbar genug ist, um als schreibsicher zu gelten, dürfen keine Ausgabedaten von Dateien zwischengepuffert werden, für welche der Benutzer Schreibsicherheit fordert (Dateiattribut `DISK-WRITE=*IMMEDIATE`). Im Unterschied zu dem Cache-Medium Hauptspeicher kann das System bei GS-Caches nicht selbst entscheiden, ob Schreibsicherheit gegeben ist. Es benötigt daher Informationen durch den Systembetreuer, welche dieser über die Festlegung der `VOLATILITY` (Flüchtigkeit) bereitstellen muss (im Kommando `MODIFY-PUBSET-CACHE-PROCESSING`).

Die Einstellung `VOLATILITY=*YES` bewirkt, dass der GS-Cache nur zur Pufferung von Dateien verwendet wird, für welche der Benutzer eine eingeschränkte Schreibsicherheit akzeptiert (Dateiattribut `DISK-WRITE=*BY-CLOSE`). Sie ist für einen GS-Cache unbedingt dann zu empfehlen, wenn er über keine ausfallsichere Stromversorgung verfügt. Aber auch wenn ein GS-Cache aus anderen Gründen nicht zuverlässig genug erscheint, um als schreibsicher zu gelten, sollte `VOLATILITY=*YES` gewählt werden.

Die Schreibsicherheit eines Caches muss auch bei der Auswahl eines geeigneten Volume-Sets beachtet werden, die aber nicht anhand der `VOLATILITY`-Festlegung, sondern unter Berücksichtigung des Performance-Profiles erfolgt, das vom Systembetreuer für die Volume-Sets passend festgelegt sein muss.



### 4.3.4 Nutzungsarten für Volume-Sets

Für die Volume-Sets eines SM-Pubsets sind die alternativen Nutzungsarten STD, WORK, HSMS-CONTROLLED möglich:

- Die Nutzungsart STD stellt den Normalfall dar. Für den Control-Volume-Set ist sie die einzig mögliche Nutzungsart.
- Volume-Sets mit der Nutzungsart WORK - und nur diese - dienen als Ablageort für Arbeitsdateien.
- Volume-Sets mit der Nutzungsart HSMS-CONTROLLED werden benötigt, um mithilfe von HSMS die S1-Ebene einrichten zu können. Sie dienen ausschließlich diesem Zweck.

Die Nutzungsart der Volume-Sets wird in der Pubset-Konfigurationsdatei vermerkt. Sie ist für einen Volume-Set auch dann definiert, wenn ihm aktuell keine physikalische Konfiguration zugeordnet ist, d.h. wenn er sich im Konfigurationszustand „defined only“ befindet.

### 4.3.5 Nutzungsrestriktionen für Volume-Sets

Die Nutzung eines Volume-Sets kann durch das Kommando MODIFY-PUBSET-RESTRICTIONS eingeschränkt werden. Die Nutzungsrestriktionen können im Gegensatz zu der Nutzungsart auch kurzfristig modifiziert werden.

#### a) Restriktionen für das Neuanlegen von Dateien

Das Anlegen neuer Dateien auf einem Volume-Set kann durch die Restriktion NEW-FILE-ALLOCATION eingeschränkt werden, die folgende Einstellungen zulässt:

NOT-RESTRICTED	keine Einschränkung
PHYSICAL-ONLY	Auf dem Volume-Set werden neue Dateien nur angelegt, wenn dies explizit durch physikalische Allokierung angefordert wird.
NOT-ALLOWED	Es können generell keine neuen Dateien angelegt werden. Für den Control-Volume-Set ist diese Sperre nicht erlaubt.

Die Einstellung NOT-ALLOWED ist ein Hilfsmittel, um einen Volume-Set vor dem Entfernen aus dem Pubset leeren zu können. Für den Control-Volume-Set ist sie nicht möglich, da sie zu Behinderungen von Systemfunktionen führen würde.

#### b) Restriktionen für das Bearbeiten von Dateien

Das Bearbeiten von Dateien, die auf einem bestimmten Volume-Set liegen, kann durch die Restriktion VOLUME-SET-ACCESS eingeschränkt werden. Sie ist nur erlaubt, wenn die Sperre für Neuallokierung den Wert NOT-ALLOWED hat. Folgende Einstellungen sind möglich:

ADMINISTRATOR-ONLY	TSOS-privilegierte Tasks dürfen Dateien uneingeschränkt eröffnen (oder durch SECURE reservieren). Tasks, welche nicht das TSOS-Privileg haben, sollen laufende Bearbeitungen noch geordnet beenden können, auch wenn sie dazu noch weitere Dateien eröffnen müssen, der Beginn neuer Bearbeitungen soll aber nach Möglichkeit verhindert werden. Tasks ohne TSOS-Privileg dürfen deshalb neue Dateien nur dann eröffnen, wenn bereits Dateien für sie eröffnet sind.
NOT-RESTRICTED	Es besteht keine Einschränkung.

#### c) Temporäres Stilllegen eines Volume-Sets

Auf einen Volume-Set, der als defekt erkannt worden ist, sollten keine Ein-/Ausgaben erfolgen. Gleiches gilt auch für einen Volume-Set, der nur vorübergehend funktionsunfähig ist, aber mit großer Wahrscheinlichkeit zu einem späteren Zeitpunkt ohne Verlust von Daten wieder benutzt werden kann.

Indem der Volume-Set mit der Restriktion PROCESSING-STATE=\*HOLD versehen wird, werden alle Ein-/Ausgaben für die Ausführung von Systemfunktionen unterbunden. Dem Benutzer wird dieser Zustand durch eine Einschränkung der Systemfunktionalität sichtbar (z.B. eingeschränkte Ausgabeinformation von SHOW-FILE-ATTRIBUTES, kein Neueröffnen von Dateien, keine Sekundärallokierung, etc.). Laufende Bearbeitungen von Benutzerdateien, die auf dem betroffenen Volume-Set liegen, werden nicht automatisch angehalten. Probleme bei den Zugriffen führen zu entsprechenden Return-Codes durch die jeweilige Zugriffsmethode.

Der Zustand „in hold“ wird automatisch durch das System oder explizit durch den Systembetreuer (durch das Kommando MODIFY-PUBSET-RESTRICTIONS oder beim Importieren des Pubsets) herbeigeführt. Für den Control-Volume-Set ist er nicht möglich. Er wird aufgehoben, indem der Systembetreuer den Volume-Set physikalisch aus dem SM-Pubset entfernt, oder indem er die Restriktion „in hold“ explizit rücksetzt, wenn ein vorübergehend funktionsunfähiger Volume-Set wieder nutzbar ist. Auch bei der Pubset-Inbetriebnahme versucht das System implizit, den in hold-Zustand aufzuheben. Der Systembetreuer sollte darauf achten, dass der Zustand „in hold“ nur kurz andauert, da dadurch auch die DMS-Funktionalität für den übrigen SM-Pubset beeinträchtigt wird (z.B. Nebeneffekte beim Anlegen neuer Dateien).

d) Nutzungsrestriktionen für einzelne Volumes eines Volume-Sets

Für die einzelnen Volumes eines Volume-Sets können Allokierungsrestriktionen erlassen werden (ALLOCATION-ON-VOLUME). Folgende Einstellungen sind möglich:

NOT-RESTRICTED	keine Einschränkung
PHYSICAL-ONLY	auf dem Volume wird nur Platz vergeben, wenn er explizit durch physikalische Allokierung angefordert wird; es erfolgen auch keine Sekundärallokierungen
NOT-ALLOWED	es erfolgen generell keine Allokierungen

Durch das Setzen der volume-bezogenen Restriktionen wird die Anzahl der frei verfügbaren Volumes und damit des frei verfügbaren Platzes auf dem Volume-Set reduziert. Es wird nicht erlaubt, wenn auf dem Volume-Set ein gravierender Speicherengpass bereits vorliegt (Überschreitung des Saturation-Levels 4) oder durch das Setzen der Restriktion herbeigeführt würde.

Die Nutzungsrestriktionen NEW-FILE-ALLOCATION, VOLUME-SET-ACCESS und PROCESSING-STATE werden in der Pubset-Konfigurationsdatei des SM-Pubsets vermerkt. Sie haben auch dann definierte Werte, wenn dem Volume-Set keine physikalische Volume-Set-Konfiguration zugewiesen ist (Konfigurationszustand „defined only“). Die Nutzungsrestriktionen auf Volume-Ebene (ALLOCATION-ON-VOLUME) werden in dem Volume-Katalog des SVLs der Volres abgelegt. Bei dem Entfernen des Volume-Sets werden sie nicht verändert. Sie haften damit auch den Volumes freier Volume-Sets an.

### 4.3.6 Sättigungsschwellwerte für Volume-Sets

Den einzelnen Volume-Sets eines SM-Pubsets sind durch den Systembetreuer festlegbare Schwellwerte für die Überwachung von Speicherengpässen zugeordnet. Dabei wird zwischen defined- und current-Werten unterschieden. Die eigentliche Sättigungsüberwachung für die Volume-Sets erfolgt anhand der current-Werte. Der höchste überschrittene (current-) Sättigungslevel bestimmt den aktuellen Sättigungszustand des Volume-Sets. Dieser dient dem System als ein Kriterium für die Bestimmung des Ablageorts der Dateien. Beispielsweise wird von zwei Volume-Sets, die sich hinsichtlich ihrer übrigen Eigenschaften für eine Datei gleich gut eignen, derjenige mit der geringeren Sättigung bevorzugt. Wenn durch das Vergrößern oder Neuanlegen einer Datei auf dem Volume-Set, auf dem die Datei bisher liegt, oder den das System als am besten geeigneten Ablageort für die neue Datei ermittelt hat, ein höheres Sättigungslevel überschritten würde als bisher, hat dies für die Benutzeranforderung entsprechende Auswirkungen wie bei SF-Pubsets (Meldungsausgabe, Annahme bzw. Zurückweisung in Abhängigkeit von dem TSOS-Privileg, etc.).

Die current-Werte werden bei der Pubset-Inbetriebnahme im Modus „exclusive“ oder „shared-master“ für jeden Volume-Set aus den in der Konfigurationsdatei vermerkten defined-Werten bestimmt, sofern es sich nicht um einen Volume-Set im Zustand „defined only“ handelt. Dabei werden im allgemeinen die defined-Werte unverändert übernommen. Einen Sonderfall stellt die „logische“ Einstellung BY-SYSTEM dar, mit der die Sättigungsschwellwerte des Volume-Sets nach der Erzeugung des SM-Pubsets initialisiert sind. Sie bewirkt, dass bei der Pubset-Inbetriebnahme die current-Sättigungsschwellwerte nach einem bestimmten Verfahren berechnet werden, in welches der Systemparameter L4SPDEF und die Kapazität des Volume-Sets eingehen.

Für einen Volume-Set im Zustand „defined only“ sind nur die defined-Werte von Relevanz. Die current-Werte werden erst dann bestimmt, wenn die physikalische Volume-Set-Konfiguration zugewiesen wird. Das dabei angewandte Verfahren ist analog wie bei der Pubset-Inbetriebnahme.

Sowohl die defined- als auch die current-Werte können durch den Systembetreuer modifiziert werden, während der Pubset in Betrieb ist. Eine Änderung der defined-Werte für Volume-Sets exportierter SM-Pubsets ist nicht möglich (Unterschied zu SF-Pubsets!).

## 4.4 Volume-Set-übergreifende Merkmale eines SM-Pubsets

### 4.4.1 Umfang der gebotenen Storage-Services

Die Eigenschaftsprofile, Nutzungsart und Nutzungsrestriktionen der einzelnen Volume-Sets bestimmen den Umfang der Storage-Services, der den Benutzern durch den SM-Pubset geboten werden kann. Die Storage-Services treten für die Benutzer als Kombinationen von Dateiattributwerten für Performance, Verfügbarkeit, Format, etc. in Erscheinung, die für in dem SM-Pubset liegende Dateien möglich sind bzw. optimal unterstützt werden. Durch das Einrichten von Storage-Klassen durch den Systembetreuer können Storage-Services mit Namen versehen werden.

### 4.4.2 Maximale I/O-Länge des SM-Pubsets

Die maximale I/O-Länge ergibt sich für einen SM-Pubset wie für einen SF-Pubset aus der kleinsten maximalen I/O-Länge aller zu dem Pubset gehörenden Volumes.

### 4.4.3 Voreinstellungen für Pubset-Space

Über die Voreinstellungen der Pubset-Space-Werte kann die Auswahl der Voreinstellungen für SM-Pubsets gesteuert werden. Die Pubset-Space-Voreinstellungen haben Einfluss auf:

- Voreinstellungen der Größe von Primär-, Sekundär- und Maximalallokierung (Diese Voreinstellungen sind auch bei SF-Pubsets vorhanden.)
- Voreinstellung des Dateiformats

Dabei gibt es eine defined- und eine current-Ausprägung. Die defined-Werte werden pubset-session-übergreifend in der Pubset-Konfigurationsdatei verwaltet und dienen zur Initialisierung der current-Werte bei der Pubset-Inbetriebnahme im Modus „exclusive“ oder „shared-master“. Während dabei die defined-Voreinstellungen für Primär-, Sekundär- und Maximalallokierung unverändert auf die current-Werte übertragen werden, gilt dies für die Voreinstellung des Dateiformats nur dann, wenn es in dem SM-Pubset erfüllbar ist. Erfüllbar bedeutet, dass in dem SM-Pubset ein Volume-Set mit der Nutzungsart STD vorhanden ist, der mit keiner Restriktion gegen Neuallokierung (NOT-ALLOWED, PHYSICAL-ONLY) versehen ist, und dessen Volume-Set-Format mit der Voreinstellung des Dateiformats verträglich ist. Ist der defined-Wert des voreingestellten Dateiformats in dem SM-Pubset nicht erfüllbar, weicht das System bei der Bestimmung des current-Wertes auf ein im Pubset erfüllbares Format aus (bei K bevorzugt auf NK2, ggf. auch auf NK4, bei NK2 auf NK4).

Die defined- und auch die current-Werte können durch den Systembetreuer modifiziert werden, während der Pubset in Betrieb ist. Die defined-Werte der Pubset-Space-Voreinstellungen von exportierten SM-Pubsets können jedoch im Unterschied zu SF-Pubsets nicht geändert werden.

#### 4.4.4 Large-Objects-Eigenschaft

SM-Pubsets können wie SF-Pubsets die Eigenschaft LARGE-OBJECTS haben. LARGE-OBJECTS ist dabei nicht eine Eigenschaft der einzelnen Volume-Sets, sondern des gesamten SM-Pubsets.

Besitzt der SM-Pubset die Eigenschaft LARGE-OBJECTS, dürfen zugehörige Volume-Sets Volumes mit einer Kapazität  $\geq 32$  GB enthalten (LARGE-DISKS-ALLOWED=\*YES). Zusätzlich kann auf einem Pubset mit großen Volumes auch das Anlegen großen Dateien ( $\geq 32$  GB) zugelassen werden (LARGE-FILES-ALLOWED=\*YES).

Die Eigenschaft LARGE-OBJECTS wird entweder beim Einrichten oder Modifizieren des Pubsets mit dem Dienstprogramm SIR bzw. über das Kommando SET-PUBSET-ATTRIBUTES vergeben. Siehe auch Handbuch „Dateien und Volumes größer 32 GB“ [7].

## 4.5 Systemdateien auf dem SM-Pubset

### Pubset-Konfigurationsdatei

Die zu dem SM-Pubsets gehörenden Volume-Sets und ihre Merkmale sind in der Pubset-Konfigurationsdatei :<catid>: \$TSOS.SYS.PUBSET.CONFIG vermerkt. Sie wird implizit bei der Pubset-Generierung erzeugt und liegt immer auf dem Control-Volume-Set.

### Pubset-Benutzerkatalog

Der Benutzerkatalog eines SM-Pubsets ist in der Datei :<catid>:\$TSOS.SYSSRPM abgelegt. Diese befindet sich immer auf dem Control-Volume-Set. Da SM-Pubsets nur als Daten-Pubsets und nicht als Home-Pubsets eingesetzt werden können, beziehen sich die im Benutzerkatalog hinterlegten Benutzerprofile ausschließlich auf den Pubset selbst. Systembezogene Benutzerfestlegungen sind darin zwar möglich, haben aber keine weiteren Auswirkungen. Als Instrument der Ressourcenkontrolle kann der Systembetreuer den einzelnen Benutzern differenzierte Space-Kontingente zuordnen. In den Benutzereinträgen ist auch das Benutzerrecht für physikalische Allokierung vermerkt. Es bezieht sich auf alle Volume-Sets des SM-Pubsets. Schließlich dienen die Benutzereinträge auch als Ablageort für die benutzerspezifischen Zuordnungen von voreingestellten Storage-Klassen. Diese bilden ein wichtiges Werkzeug für den Systembetreuer, um auf SM-Pubsets selbst definierte Strategien für die Platznutzung zum Einsatz zu bringen zu können.

Auf SM-Pubsets besteht wie auf SF-Pubsets die Möglichkeit, Benutzergruppen einzurichten. Für SM-Pubsets, die nicht als Home-Pubset nutzbar sind, sind vor allem die Space-Limit-Festlegungen in den Gruppeneinstellungen relevant. Die Benutzergruppeneinstellungen sind wie der Benutzerkatalog in der Datei :<catid>:\$TSOS.SYSSRPM abgespeichert.

## Dateikatalog des SM-Pubsets

Der Dateikatalog eines SF- oder eines SM-Pubsets dient als Behälter für die Katalogeinträge der dem Pubset zugehörigen Dateien und Jobvariablen, sowie auch für die Katalogeinträge von Privatplattendateien und Banddateien. Um den Volume-Sets als unabhängigen Ausfalleinheiten Rechnung zu tragen, ist der Dateikatalog bei SM-Pubsets im Unterschied zu SF-Pubsets mithilfe mehrerer Katalogdateien realisiert. Der Aufbau wird in [Bild 7](#) illustriert.

- Jeder Volume-Set eines SM-Pubsets enthält eine Volume-Set-spezifische Katalogdatei mit den Katalogeinträgen für die auf ihm liegenden Dateien. Der Pfadname der Volume-Set-spezifischen Katalogdatei ergibt sich aus Pubsetkennung und Volume-Set-Kennung wie folgt: `<catid>:$TSOS.TSOSCAT.<volset-id>`.
- Zusätzlich benötigt das Katalogmanagement weitere Katalogdateien, welche alle auf dem Control-Volume-Set des SM-Pubsets liegen:
  - Die Katalogeinträge für auf eine Hintergrundebene migrierte Dateien sowie katalogisierte Dateien, die keinen Platz belegen, werden in der Katalogdatei `<catid>:$TSOS.TSOSCAT.#MIN` aufbewahrt.
  - Als Behälter für Katalogeinträge von Banddateien und Privatplattendateien dient die Katalogdatei `<catid>:$TSOS.TSOSCAT.#PVT`.
  - Als Behälter für die Jobvariablen des SM-Pubsets dient die Katalogdatei `<catid>:$TSOS.TSOSCAT.#JVC`.
  - Die Katalogdateien `<catid>:$TSOS.TSOSCAT.$NLO`, `<catid>:$TSOS.TSOSCAT.$NLC` und `<catid>:$TSOS.TSOSCAT.$PFI` dienen dem Katalogmanagement zum schnellen Auffinden von Katalogeinträgen bzw. bei dem Entfernen eines defekten Volume-Sets zum Erstellen der Liste der betroffenen Dateien.

### *Hinweis*

Ab BS2000/OSD V6.0B können die Spezialkataloge mit dem Suffix #MIN, #PVT und #JVC in das Katalogformat „extra large“ konvertiert werden. Sie werden dabei umbenannt und erhalten den neuen Suffix M00, P00 und J00. Diesen Katalogen können bis zu 99 weitere Teilkataloge hinzugefügt werden (der Zähler im Suffix wird dabei fortlaufend hochgezählt). Näheres siehe Handbuch „Einführung in die Systembetreuung“ [\[5\]](#).



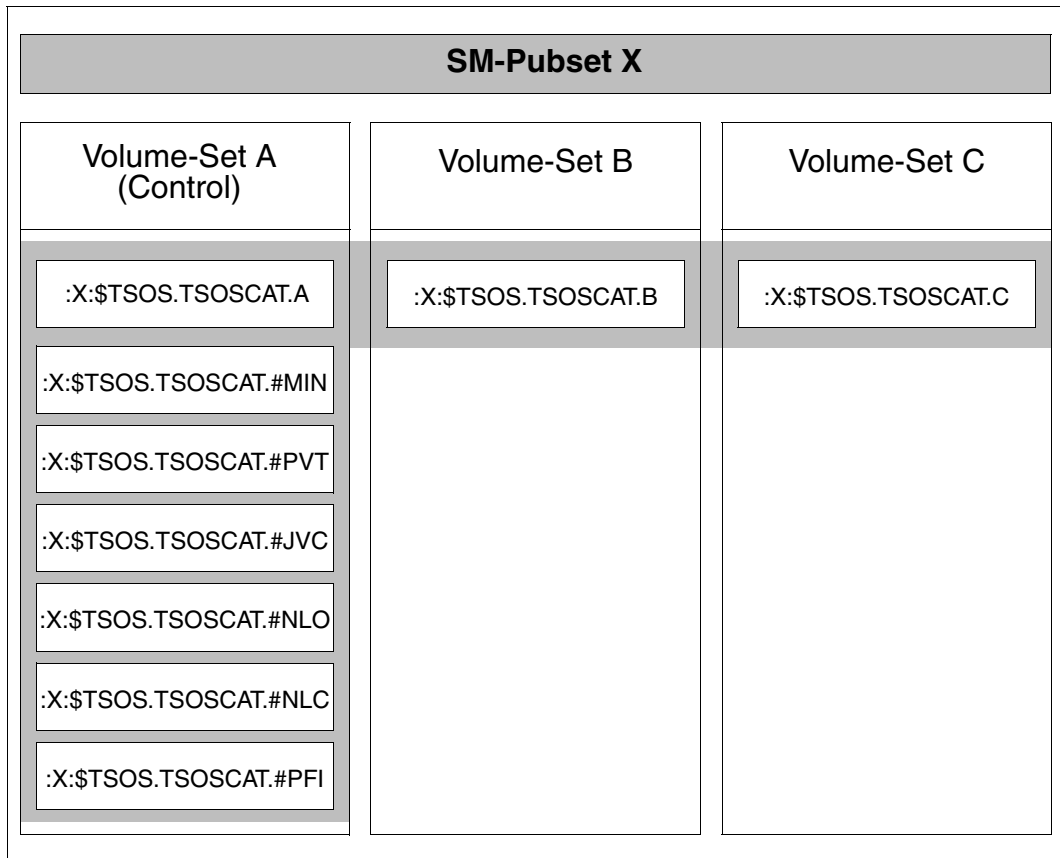


Bild 7: Aufbau des Dateikatalogs eines SM-Pubsets

Das Anlegen der Katalogdateien erfolgt implizit beim Generieren des Pubsets. Die Volume-Sets, auf denen die einzelnen Katalogdateien liegen, sind entsprechend der oben gegebenen Beschreibung festgelegt. Die maximale Größe der einzelnen Katalogdateien beträgt 16384 PAM-Seiten. Bei der Generierung des SM-Pubsets kann ihre Größe und ihre Verteilung auf bestimmte Volumes durch den Systembetreuer beeinflusst werden.

Später können die Katalogdateien durch den Systembetreuer mithilfe von SIR (für einen außer Betrieb befindlichen Pubset) oder durch das Kommando MODIFY-FILE-ATTRIBUTES (während einer Pubset-Session) vergrößert werden. Platz, der während einer Pubset-Session zusätzlich bereitgestellt wird, kann vom Katalog-Management direkt im Anschluss genutzt werden.

## GUARDS-Katalog

Zum Schutz der in einem SM-Pubset liegenden Dateien, Storage-Klassen, HSMS-Management-Klassen, Jobvariablen, etc. können auf einem Pubset GUARDS-Schutzprofile eingerichtet werden. Als Ablageort dient der GUARDS-Katalog des Pubsets. Er wird unter dem Pfadnamen :<catid>:\$TSOS.SYSCAT.GUARDS. implizit auf dem Control-Volume-Set angelegt, wenn für den Pubset erstmalig ein GUARDS-Schutzprofil eingerichtet wird. Die Bedeutung der GUARDS-Schutzprofile ist für SM-Pubsets und SF-Pubsets gleich.

## Kataloge für Storage-Klassen und Volume-Set-Listen

Der Storage-Klassen-Katalog (:<catid>:\$TSOS.SYSCAT.STORCLS) sowie der Volume-Set-Listen-Katalog (:<catid>:\$TSOS.SYSCAT.VSETLIST) werden implizit durch das Storage-Klassen-Management auf dem Control-Volume-Set angelegt. In ihnen werden die Definitionen der für den Pubset eingerichteten Storage-Klassen bzw. der mit den Storage-Klassen verknüpfbaren Volume-Set-Listen vermerkt.

## Watch-Dog-Datei

Der Watch-Dog-Mechanismus zur Rechnerüberwachung in einem Shared-Pubset-Verbund benötigt die Watch-Dog-Datei :<catid>:\$TSOS.SYS.PVS.SHARER.CONTROL. Sie wird auf einem Pubset automatisch eingerichtet, wenn er erstmalig im Modus „exclusive“ oder „shared-master“ importiert wird. Ihre Funktion ist bei SM- und SF-Pubsets gleich. Bei SM-Pubsets liegt die Watch-Dog-Datei auf dem Control-Volume-Set.

## Defect-Garbage-Datei

Defect-Garbage-Dateien dienen zum Erfassen von I/O-Fehlern von Volumes. Eine Defect-Garbage-Datei wird auf einem Volume-Set automatisch durch das System angelegt, wenn erstmalig für eine seiner Platten ein Defekt registriert wird. Sie erhält den Pfadnamen :<catid>:\$TSOS.SYSDAT.DEFECT.GARBAGE.<Volume-Set-id#>, wobei <Volume-Set-id#> die Kennung des Volume-Sets bezeichnet. Ist dessen Kennung kleiner als 4 Zeichen, wird sie mit # auf 4 Stellen aufgefüllt.

## SYSEAM-Datei

Die Datei :<catid>:\$TSOS.SYSEAM dient als Behälter für EAM-Dateien. Ihre Bedeutung ist für SM-Pubsets gleich wie bei SF-Pubsets. Sie kann auf einem beliebigen Volume-Set des SM-Pubsets liegen, insbesondere ist sie nicht an den Control-Volume-Set gebunden. Wenn sie der Systembetreuer vor dem erstmaligen Zugriff auf eine EAM-Datei durch das Kommando CREATE-FILE anlegt, kann er ihr bestimmte Dateiattribute zuordnen (z.B. Arbeitsdatei) oder ihre Lage innerhalb des SM-Pubsets durch physikalische Allokierung vorgeben. Andernfalls wird sie automatisch durch das System erzeugt, wobei ihre Dateiattribute Standardwerte erhalten und ihr Ablageort entsprechend durch das System bestimmt wird.

## 4.6 HSMS-Konfiguration eines SM-Pubsets

Unter der HSMS-Konfiguration eines SM-Pubsets im engeren Sinne verstehen wir die dem Pubset zugeordneten Hintergrundebenen und Backup-Archive. Unter den Backup-Archiven nimmt das System-Backup-Archiv des SM-Pubsets eine Sonderrolle ein. Im weiteren Sinne gehören zur HSMS-Konfiguration eines SM-Pubsets auch Langzeitarchive sowie Backup-Archive, die dem SM-Pubset nicht zugeordnet sind, aus welchen aber ein RESTORE von Dateien in den SM-Pubset möglich ist. Letztere können beispielsweise bei der Konvertierung von SF-Pubsets in einen SM-Pubset entstehen. Die HSMS-Konfiguration im engeren Sinne ist bei SM-Pubsets pubset-spezifisch konzipiert, d.h. sie ist genau einem SM-Pubset zugeordnet, auf dem die für ihre Benutzung erforderlichen Metadaten liegen. Sie trägt damit dem SM-Pubset als Umschalteneinheit Rechnung. Eine Illustration der HSMS-Konfiguration eines SM-Pubsets findet sich in [Bild 8](#).

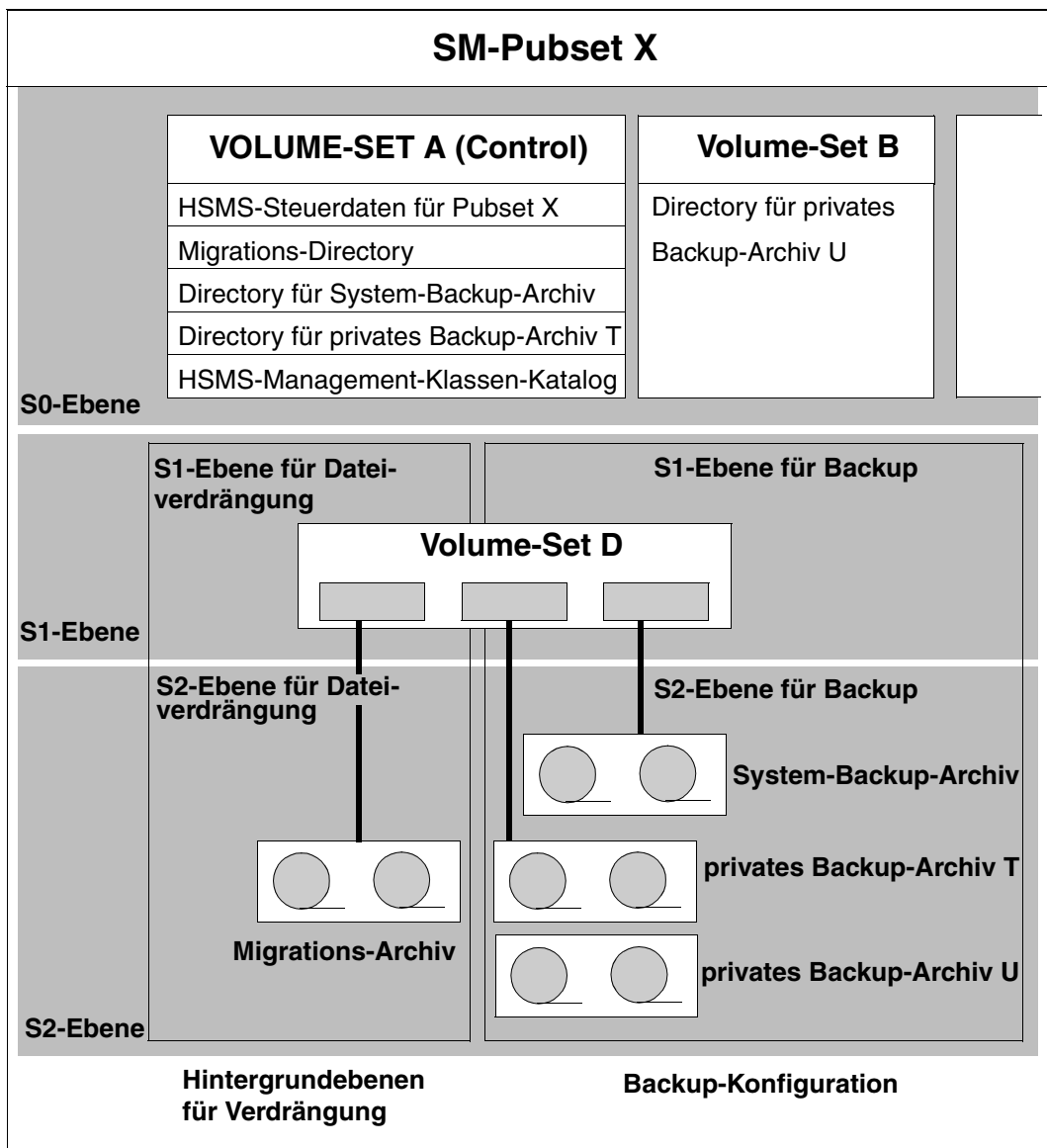


Bild 8: HSMS-Konfiguration eines SM-Pubsets

Wenn der Systembetreuer für einen SM-Pubset eine HSMS-Konfiguration einrichtet (HSMS-Anweisung CREATE-SM-PUBSET-PARAMETERS), gelangt der Pubset in den Zustand HSMS-SUPPORTED. Dies bedeutet, dass HSMS verfügbar sein muss, um den Pubset ohne Einschränkungen nutzen zu können.

## 4.6.1 Konfiguration der Hintergrundebenen

Ein SM-Pubset kann mithilfe von HSMS-Funktionen mit Hintergrundebenen für Dateiverdrängung ausgestattet werden. Sie sind dem SM-Pubset (im Unterschied zu SF-Pubsets) exklusiv zugeordnet und können nicht für andere SF-Pubsets oder SM-Pubsets benutzt werden. Im Unterschied zu der Verarbeitungsebene sind sie nicht obligatorische Pubset-Bestandteile.

Wesentliche Merkmale der Konfiguration der Hintergrundebenen sind:

- Der Name des Migrationsarchivs.  
Das Migrationsarchiv erhält durch HSMS einen Standardnamen, wenn der Systembetreiber den Namen nicht explizit vorgibt.
- Das Migrations-Directory.  
Das Migrations-Directory erhält durch HSMS einen Standardnamen, wenn der Systembetreiber den Namen nicht explizit vorgibt. Es muss auf dem SM-Pubset liegen, dem es zugeordnet ist (Unterschied zu SF-Pubsets!). Innerhalb des SM-Pubsets wird es durch HSMS standardmäßig auf dem Control-Volume-Set angelegt, falls von dem Systembetreiber nicht explizit ein anderer Volume-Set als Ablageort vorgegeben wird. Letzteres sollte nur in speziellen Anwendungsfällen erfolgen, da es einer Konzentration der Pubset-Metadaten auf dem Control-Volume-Set entgegenwirkt (Auswirkungen auf die Pubset-Verfügbarkeit und Pubset-Rekonfigurierbarkeit).
- Der Name des als S1-Ebene dienenden Volume-Sets, falls eine S1-Ebene vorgesehen wird.
- Der voreingestellte Device-Typ der S2-Datenträger und Name des als S2-Ebene dienenden Tape-Pools, falls eine S2 Ebene vorgesehen wird.
- Die Steuerparameter für die Migration.

## 4.6.2 Backup-Konfiguration eines SM-Pubsets

Auf einem SM-Pubset können mithilfe von HSMS Backup-Archive eingerichtet werden. Die Backup-Archive eines SM-Pubsets sind pubset-spezifisch (Unterschied zu SF-Pubsets). Dies bedeutet:

- In dem SM-Pubset liegende Dateien können nur in einem Backup-Archiv des SM-Pubsets gesichert werden.
- Das Backup-Archiv eines SM-Pubsets kann nur Sicherungen von Dateien des ihm zugeordneten SM-Pubsets enthalten.

Für das Wiedereinspielen von Sicherungen bestehen keine analogen Restriktionen wie für das Sichern. In SM- wie auch SF-Pubsets können Sicherungen aus beliebigen Backup-Archiven wiedereingespilt werden, sofern diese verfügbar sind.

Das Backup-Archiv eines SM-Pubsets ist charakterisiert durch:

- den Namen des Backup-Archivs,
- den Namen des zugeordneten Directory; es muss in dem SM-Pubset liegen, dem das Backup-Archiv zugeordnet ist,
- den voreingestellten Device-Typ der Backup-Datenträger und den Namen des Tape-Pools, in dem die Datenträger zusammengefasst sind,
- den Volume-Set, der für Sicherungen auf Platte genutzt wird. Hierfür wird der gleiche Volume-Set herangezogen, der auch als S1-Ebene für die Hintergrundverdrängung dient. Da Originaldateien und Sicherungskopien auf unterschiedlichen Datenträgern liegen sollen, ist er nicht für die Sicherung von Dateien geeignet, die auf die S1-Hintergrundebene verdrängt sind,
- Backup-Steuerparameter.

Unter den Backup-Archiven eines SM-Pubsets nimmt das System-Backup-Archiv eine Sonderrolle ein. Es wird als Standard-Archiv herangezogen, wenn der Benutzer bei Backup- und Restore-Aktivitäten das Backup-Archiv nicht explizit spezifiziert. Die Namen des System-Backup-Archivs und seines Directory werden von HSMS standardmäßig festgelegt, wenn sie nicht durch den Systembetreuer explizit vorgegeben werden. Die Directories der Backup-Archive eines SM-Pubsets müssen in dem Pubset selbst liegen. Innerhalb des Pubsets werden sie standardmäßig durch HSMS auf dem Control-Volume-Set angelegt. Insbesondere für das Directory des System-Backup-Archivs sollte die explizite Angabe eines anderen Volume-Sets als Ablageort nur in Sonderfällen erfolgen, da dies der Konzentration der Pubset-Metadaten auf dem Control-Volume-Set entgegenwirkt (Auswirkungen auf die Pubset-Verfügbarkeit und Pubset-Rekonfigurierbarkeit).

### 4.6.3 Langzeitarchive

Langzeitarchive können eine größere Lebensdauer haben als die Pubsets, deren archivierte Dateien sie enthalten. Die Langzeitarchive können daher pubset-übergreifend eingerichtet werden. Sie sind dann keinem bestimmten SF- oder SM-Pubset zugeordnet und können sowohl zum Erstellen als auch zum Wiedereinspielen von Datei-Archivierungen verschiedener Pubsets benutzt werden. Es besteht aber auch die Möglichkeit, pubset-spezifische Langzeitarchive zu definieren.

## 4.6.4 HSMS-Management-Klassen-Katalog

Zur Steuerung der Migrations- und Backup-Aktivitäten auf einem SM-Pubset durch den Benutzer können HSMS-Management-Klassen verwendet werden. Als Ablageort für die für einen SM-Pubset definierten HSMS-Management-Klassen dient der HSMS-Management-Klassen-Katalog. Er wird durch HSMS implizit auf dem Control-Volume-Set des SM-Pubsets angelegt.

## 4.6.5 HSMS-Dateien

Neben den Directories für das Migrationsarchiv und die Backup-Archive sowie dem HSMS-Management-Klassen-Katalog gibt es weitere HSMS-Dateien für SM-Pubsets, von denen die wichtigsten kurz beschrieben werden:

- Zusätzlich zu den rechner-spezifischen (zentralen) HSMS-Steuerdateien (`$SYSHSMS.HSMS.1.CONTROL.FILE`) ist für jeden SM-Pubset eine pubset-spezifische (lokale) Steuerdatei (`$SYSHSMS.HSM.1.SM.CONTROL.FILE`) vorgesehen. In ihr werden für den jeweiligen SM-Pubset die Archiv-Definitionen für das Migrationsarchiv und die Backup-Archive vermerkt.
- Der Systembetreuer kann für einen SM-Pubset eine Ausnahmedatei (Except-File) vorgeben, in der die Namen von Dateien eingetragen sind, die nicht auf Hintergrundebenen migriert werden dürfen. Sie muss auf dem SM-Pubset liegen.
- Die HSMS-Auftragsdatei eines SM-Pubsets (`$SYSHSMS.HSM.2.SM.REQUEST.FILE`) dient zur Verwaltung der für den Pubset erzeugten Migrations- und Backup-Aufträge.
- In der ARCHIVE-Checkpoint-Datei eines SM-Pubsets werden für die auf dem Pubset laufenden HSMS-Aufträge Wiederaufsetzpunkte gesichert, um Aufträge nach einer Unterbrechung, z.B. infolge eines Systemabsturzes, wieder anstarten zu können.
- Das Ergebnis eines HSMS-Auftrags wird in den Ergebnisdateien (Result-File und Report-File) hinterlegt.
- Der Benutzer kann vorgeben, ob das Ergebnis eines HSMS-Auftrags auf Drucker ausgegeben wird oder in druckaufbereiteter Form in einer von ihm angegebenen Berichtdatei hinterlegt wird. Die Catid im Dateinamen legt den Pubset fest, auf den die Berichtdatei gelangt.
- Für das Zurückholen einer Datei von einer Hintergrundebene in die Verarbeitungsebene wird kurzzeitig eine Hilfsdatei benötigt. Sie wird auf dem Pubset angelegt, in dem der Recall erfolgt.

Die HSMS-Dateien eines SM-Pubsets tragen der Umschaltbarkeit eines Pubsets besonders Rechnung, indem sie im Unterschied zu den entsprechenden HSMS-Dateien für SF-Pubsets nicht rechnerspezifisch, sondern pubset-bezogen gebildet werden. Sie liegen mit Ausnahme der Berichtdatei, deren Lage die Benutzer vorgeben, jeweils auf dem SM-Pubset, dem sie zugeordnet sind.

Innerhalb eines SM-Pubsets werden die automatisch durch HSMS erzeugten HSMS-Dateien, die für die Durchführung bzw. Wiederaufsetzbarkeit der HSMS-Funktionen wesentlich sind (Steuerdatei, Auftragsdatei, Checkpoint-Datei, Ergebnisdateien), auf dem Control-Volume-Set angelegt.

Für die Except-File, die der Systembetreuer anlegt, wird der Control-Volume-Set als Ablageort empfohlen. Die HSMS-Funktionalität für den SM-Pubset wird damit durch die Rekonfiguration des Pubsets bzw. durch den Ausfall einzelner Volume-Sets nicht tangiert. Der Control-Volume-Set ist nämlich obligatorischer Bestandteil des SM-Pubsets und kann während der Lebensdauer des Pubsets durch Pubset-Rekonfiguration nicht entfernt werden.

## 4.7 Nutzung von ARCHIVE

ARCHIVE unterstützt SM-Pubsets in gleicher Weise wie SF-Pubsets. Im Gegensatz zu HSMS bietet ARCHIVE jedoch keine Dateiverdrängung (Migration) und keine besonderen Funktionen für die zusätzlichen Strukturen von SM-Pubsets an. ARCHIVE bietet insbesondere keine Dateiselektion nach Management- oder Storage-Klassen, kein ausgezeichnetes S1-Volumeset und kein Restaurieren auf einen bestimmten oder ursprünglichen Volumeset an.

Kunden, die am SM-Pubset in seiner Eigenschaft als Umschalteinheit interessiert sind, sollten schon im ARCHIVE-Betrieb die zum Umschalten günstigere „dezentrale Verwaltung“ wählen (Directories pro SM-Pubset und im SM-Pubset) und können damit einen einfacheren Übergang auf HSMS planen. Die einfache Übernahme der Directories von SM-Pubsets beim Übergang von ARCHIVE- auf HSMS-Betrieb wird zudem erleichtert, wenn schon im ARCHIVE-Betrieb die Directories mit Catid geführt werden (PARAM CATID=\*YES).



## 4.8 SF-Pubset-Merkmale ohne Entsprechung bei SM-Pubsets

### Nutzung als Home-Pubset

SM-Pubsets sind nicht als Home-Pubsets nutzbar. Daraus ergeben sich folgende Konsequenzen:

1. Auf SM-Pubsets dürfen keine Dateien liegen, welche für die Systemeinleitung benötigt werden (IPL-Dateien, REP-Dateien, SLED-Dateien, IOCONF-Dateien, etc.). Die Volumes eines SM-Pubsets können auch nicht Träger von Boot-Blöcken sein.
2. Auf SM-Pubsets dürfen keine SNAPSHOT-Dateien angelegt werden

### SPEEDCAT-Nutzung

SPEEDCAT-Einstellungen sind für SM-Pubsets nicht vorgesehen, da diese standardmäßig über eine SPEEDCAT entsprechende Funktionalität verfügen.

### Generelle Erlaubnis für physikalische Allokierung

Die Möglichkeit, auf einem Pubset generell allen Benutzern die physikalische Allokierung zu erlauben ist für SM-Pubsets nicht vorgesehen, da sie durch eine verbesserte Funktionalität ersetzt ist (Recht für physikalische Allokierung für einzelne Benutzer).

### Rückkonvertieren des Dateikatalogs

Das Rückkonvertieren des Dateikatalogs in ein Format, das in Betriebssystemversionen < BS2000/OSD V3.0 benötigt wird, ist irrelevant, da diese Versionen nicht mehr unterstützt werden.

## 4.9 Nutzung eines SM-Pubsets an einem Rechner

Um einen Pubset (SF wie SM) an einem Rechner in Betrieb nehmen zu können, muss für den Pubset an diesem Rechner ein MRSCAT-Eintrag angelegt sein. Er dient u.a. zur Ablage von Information, die das System für die Durchführung der Pubset-Inbetriebnahme von dem Systembetreuer benötigt, sowie zur Festlegung von rechnerspezifischen Betriebsparametern und Nutzungseinschränkungen, die für eine Pubset-Session vorgegeben werden können. Als Mindestinformation für die Inbetriebnahme muss bei SM-Pubsets in dem MRSCAT-Eintrag der Name des Control-Volume-Sets eingetragen sein. Zum Einrichten eines MRSCAT-Eintrags und zum Ändern seines Inhalts stehen dem Systembetreuer die Kommandos ADD/MODIFY-MASTER-CATALOG-ENTRY und MODIFY-PUBSET-CACHE-ATTRIBUTES zur Verfügung.

Zusätzlich zum MRSCAT-Eintrag des SM-Pubsets benötigt das System als systeminterne Betriebsmittel auch für jeden seiner Volume-Sets einen MRSCAT-Eintrag. Dies gilt unabhängig von dem Modus der Inbetriebnahme (exclusive, shared-master oder shared-slave). Die MRSCAT-Einträge für die Volume-Sets werden bei der Inbetriebnahme des Pubsets automatisch durch das System angelegt, sofern sie nicht bereits vorhanden sind. Dies ist nur möglich, wenn nicht bereits ein anderweitig benutzter MRSCAT-Eintrag existiert, dessen Kennung mit der eines der Volume-Sets kollidiert. Die Inbetriebnahme wird in diesem Falle abgebrochen. Als anderweitig benutzt gilt ein MRSCAT-Eintrag dann, wenn er nicht vom passenden Typ ist (d.h. vom Typ SF- oder SM-Pubset), oder wenn er einem Volume-Set eines anderen SM-Pubsets zugeordnet ist. MRSCAT-Einträge, die zu Kollisionen führen, müssen vor der Inbetriebnahme des SM-Pubset durch den Systembetreuer gelöscht werden.

### 4.9.1 MRSCAT-Einträge für SM-Pubsets

Als Behälter für die an einem Rechner eingerichteten MRSCAT-Einträge dient die MRSCAT-Datei \$TSOS.SYSTEM.MRSCAT auf dem Home-Pubset des Rechners. Sie kann folgende unterschiedliche Typen von MRSCAT-Einträgen enthalten: Einträge für SF-Pubsets, für SM-Pubsets und für Volume-Sets. In [Bild 9](#) ist ein Beispiel für eine MRSCAT-Datei dargestellt. Zur Identifikation eines MRSCAT-Eintrags dient die Kennung des SM-Pubsets bzw. des SF-Pubsets bzw. des Volume-Sets. Für die eindeutige Adressierbarkeit der MRSCAT-Einträge ist es daher notwendig, dass sich die Kennungen aller in die MRSCAT-Datei eingetragenen SF-Pubsets, SM-Pubsets und Volume-Sets voneinander unterscheiden.

Solange der SM-Pubset an einem Rechner in Betrieb ist, werden dort die MRSCAT-Einträge der dem Pubset angehörenden Volume-Sets als Betriebsmittel benötigt und können durch den Systembetreuer nicht gelöscht werden. Wird der SM-Pubset an dem Rechner außer Betrieb genommen, bleiben die MRSCAT-Einträge der Volume-Sets erhalten, können aber entfernt werden. Der rechnerspezifische Status eines (Volume-Set-) MRSCAT-Eintrags beschreibt, welche Situation vorliegt. Er wird in dem MRSCAT-Eintrag des Volume-Sets hinterlegt und kann die Werte *connected* (Pubset, dem der Volume-Set angehört, ist in Betrieb) und *not connected* (Pubset, dem der Volume-Set angehört, ist außer Betrieb) annehmen.

Das System nutzt die MRSCAT-Einträge des SM-Pubsets und seiner Volume-Sets intern auch dazu, bestimmte in dem SM-Pubset selbst verankerte Pubset-Eigenschaften zusätzlich redundant zu hinterlegen, um bei Lesezugriffen performant darauf zugreifen zu können. Beispielsweise ermitteln bestimmte Pubset-Auskunftsfunktionen wie SHOW-MASTER-CATALOG-ENTRY oder SHOW-PUBSET-PARAMETERS die Pubset- und Volume-Set-Eigenschaften an Hand der MRSCAT-Einträge. Während einer Pubset-Session stellt das System die Konsistenz der Daten in den MRSCAT-Einträgen sicher. Nach der Außerbetriebnahme eines Pubsets werden die ihm zugeordneten MRSCAT-Einträge an diesem Rechner unverändert beibehalten. Wird der Pubset anschließend von einem anderen Rechner aus geändert, können sich Diskrepanzen mit den tatsächlichen Pubset-Eigenschaften ergeben. Bei der Inbetriebnahme eines Pubsets an einem Rechner werden dort die MRSCAT-Einträge automatisch durch das System in einen konsistenten Zustand gebracht.

MRSCAT-Datei für Home-Pubset F			
Kennung des Eintrags	Typ	bei Volume-Sets-Einträgen: zugeordneter SM-Pubset	sonstige Daten
C	SF-Pubset		
E	Volume-Set	S	
F	SF-Pubset		
H	SM-Pubset		
K	Volume-Set	S	
M	SF-Pubset		
O	Volume-Set		
P	SF-Pubset		
R	Volume-Set	H	
S	SM-Pubset		
U	Volume-Set	S	

MRSCAT-Einträge zu Pubset S

Bild 9: Inhalt einer MRSCAT-Datei

## 4.9.2 Festlegungen für die Pubset-Inbetriebnahme

Wir beschränken uns auf eine kurze Darstellung der im MRSCAT-Eintrag hinterlegten Festlegungen für die Pubset-Inbetriebnahme, bei welchen Besonderheiten für SM-Pubsets bestehen. Eine genaue Beschreibung findet sich in dem Handbuch „Einführung in die Systembetreuung“ [5].

### 1. Kennung des Control-Volume-Sets (Unterschied zu SF-Pubsets)

Für die Inbetriebnahme eines SM-Pubsets benötigt das System bestimmte Daten, die auf der Volres des Control-Volume-Sets hinterlegt sind. Die Volres hat dabei für einen SM-Pubset eine analoge Rolle wie die Pubres für einen SF-Pubset. Ihren Namen ermittelt das System aus der Kennung des Control-Volume-Sets, die der Systembetreuer über den MRSCAT-Eintrag des SM-Pubsets dem System bekannt machen muss.

### 2. Vorgaben für das System, wenn Cache-Bereiche nicht rekonnektiert werden können

Bei der Pubset-Inbetriebnahme wird überprüft, ob als Folge einer nicht ordnungsgemäß durchgeführten Außerbetriebnahme einzelne Volume-Sets noch mit Cache-Bereichen verknüpft sind. Das System versucht diese zu rekonnektieren, um Datenverluste zu vermeiden. Ist dies nicht möglich, so wird bei flüchtigen Cache-Medien (Hauptspeicher, Globalspeicher mit VOLATILITY=\*YES) die Inbetriebnahme ohne Rekonnektierung fortgeführt.

Bei nicht flüchtigen Cache-Medien (Globalspeicher mit VOLATILITY=\*NO) sind folgende Alternativen möglich, wenn das Rekonnektieren nicht gelingt:

- a) Die Inbetriebnahme wird abgebrochen. Vor der nächsten Inbetriebnahme behebt der Systembetreuer die Ursachen, die das Rekonnektieren der Cache-Bereiche verhinderten.
- b) Das System fragt den Operator, ob die Inbetriebnahme abzubrechen ist, oder ob sie ohne die Rekonnektierung der problematischen Cache-Bereiche erfolgen soll.

Das System entnimmt der FORCE-IMPORT-Einstellung im MRSCAT-Eintrag des Pubsets, welche Alternative der Systembetreuer wünscht. Werden Cache-Bereiche nicht rekonnektiert, sind Dateien mit darin zwischengespeicherten Ausgabedaten nicht mehr normal nutzbar. Es ist zu beachten, dass sich die FORCE-IMPORT-Einstellung immer auf den gesamten SM-Pubset, d.h. auf alle zugehörigen Volume-Sets bezieht. Eine differenzierte Behandlung einzelner Volume-Sets ist nicht möglich.

### 3. Vorgaben für das System bei Problemen beim Neukonnektieren von Cache-Bereichen

Die Rekonnectierung von Cache-Bereichen ist nur in dem Ausnahmefall erforderlich, wenn die vorausgegangene Pubset-Session nicht durch reguläres Exportieren des Pubsets beendet wurde. Im Normalfall versucht das System bei der Pubset-Inbetriebnahme für die einzelnen Volume-Sets des SM-Pubsets neue Cache-Bereiche bereitzustellen, wenn vom Systembetreuer durch das Kommando `MODIFY-PUBSET-CACHE-ATTRIBUTES` entsprechende Festlegungen getroffen worden sind. Sind die gewünschten Cache-Medien nicht bzw. nicht in dem gewünschten Umfang verfügbar oder ergeben sich sonstige Probleme bei ihrer Bereitstellung, bestehen folgende Alternativen:

- a) Die Inbetriebnahme wird abgebrochen. Vor der nächsten Inbetriebnahme sollte der Systembetreuer die Ursachen beseitigen, die das Bereitstellen der gewünschten Cache-Konfiguration verhinderten.
- b) Die Inbetriebnahme wird durchgeführt, wobei einzelne Volume-Sets keine bzw. kleinere Cache-Bereiche erhalten, als in der Konfigurationsdatei festgelegt ist.

Durch den Eintrag `SIZE-TOLERANCE` im `MRSCAT`-Eintrag des Pubsets gibt der Systembetreuer dem System bekannt, wie es sich in diesem Fall verhalten soll.

Auch bei der Ausführung des Kommandos `START-PUBSET-CACHING` für einen Volume-Set oder bei der Aufnahme eines freien Volume-Set in einen SM-Pubset kann es sich ergeben, dass für den Volume-Set der gewünschte Cache nicht oder nicht in ausreichendem Umfang verfügbar gemacht werden kann. Auch in diesem Fall kann durch `SIZE-TOLERANCE` das Systemverhalten gesteuert werden.

Auch die `SIZE-TOLERANCE`-Einstellung bezieht sich immer auf den gesamten SM-Pubset und erlaubt keine differenzierte Behandlung einzelner Volume-Sets.

## 4.9.3 Sonstige Festlegungen für die Pubset-Nutzung

Für die Festlegungen, ob bzw. wie ein Pubset im Verbund genutzt werden soll, bestehen zwischen SF- und SM-Pubsets keine Unterschiede (siehe Kommandos `ADD/MODIFY-MASTER-CATALOG-ENTRY` - Operanden `SHARED-PUBSET`, `BATCH-WAIT-TIME`, `DIALOG-WAIT-TIME`, `PARTNER-NAME` und Kommando `MODIFY-PUBSET-ATTRIBUTES`). Insbesondere beziehen sich die Festlegungen auch bei SM-Pubsets generell auf den gesamten SM-Pubset, nicht auf einzelne Volume-Sets.

Dies gilt auch für die Festlegungen der rechnerspezifischen Betriebsparameter, für die Festlegung der Pufferbereiche für das Katalog-Management, für die Steuerung von EAM (Kommandos `ADD/MODIFY-MASTER-CATALOG-ENTRY`-Operanden `RESIDENT-BUFFERS`, `NUMBER-OF-BUFFERS`, `EAM`).

Für detailliertere Informationen wird auf das Handbuch „Einführung in die Systembetreuung“ [5] verwiesen.



---

## 5 Neueinrichten eines SM-Pubsets

Das Erzeugen eines voll ausgestatteten SM-Pubsets (mit Storage-Klassen, Hintergrundebenen, Backup-Archiven, HSMS-Management-Klassen, GUARDS-Schutzprofilen, etc.) erfordert mehrere Schritte, deren Umfang, Reihenfolge und zeitliche Durchführung variiert werden können. Den ersten Schritt bildet jeweils die Generierung eines Basis-Pubsets mithilfe von SIR, der durch weitere Maßnahmen für die spezifischen Anforderungen der jeweiligen Benutzer erweitert und modifiziert wird. Die Vorgehensweise mit SIR wird in den Abschnitten 5.1 bis 5.3 beschrieben.

Eine spezielle Möglichkeit zum Erzeugen einer umbenannten Kopie eines SM-Pubsets bietet PVSREN mit der Funktion „Klonen eines SM-Pubsets“ für gespiegelte Pubsets (BCV-Spiegel, siehe Handbuch „SHC-OSD“ [8]. Diese Möglichkeit ist im [Abschnitt „Klonen eines SM-Pubsets mit PVSREN“ auf Seite 63](#) beschrieben.

### 5.1 Überblick über die SIR-Funktionalität für SM-Pubsets

SIR bietet sowohl für SF-Pubsets als auch SM-Pubsets Funktionen für die Pubset-Generierung und die Pflege bestehender Pubsets. Für SM-Pubsets bestehen folgende Besonderheiten:

- Aufgrund der komplexeren Struktur erfordern SM-Pubsets in SIR eine umfangreichere Funktionalität für die Pubset-Generierung.
- Der Teil der Funktionalität, der nur für das Generieren und die Pflege von Home-Pubsets von Bedeutung ist, wird für SM-Pubsets nicht bereitgestellt, da SM-Pubsets nicht als Home-Pubset eingesetzt werden können. Möglichkeiten für das Einrichten und die Pflege von SNAPSHOT-Dateien sowie das Einrichten der Upladefähigkeit auf Platten des Pubsets werden für SM-Pubsets nicht unterstützt.
- Die Funktionen zur Pflege bereits existierender Pubsets bestehen bei SM-Pubsets in der Hinzunahme neuer Volumes zu Volume-Sets (ggf. mit impliziter Volume-Initialisierung), der Hinzunahme neuer Volume-Sets zu einem SM-Pubset und der Pflege der Katalogdateien. Sie sind eine Alternative zu den Funktionen der dynamischen Pubset-Rekonfiguration.

- Für SM-Pubsets wird das Erzeugen freier leerer Volume-Sets ermöglicht. Freie Volume-Sets sind dadurch charakterisiert, dass sie keinem SM-Pubset angehören und abgesehen von Sondersituationen leer sind. Sie enthalten außer der Katalogdatei des Volume-Sets keine Dateien. Die freien Volume-Sets sind ein Hilfsmittel für die Pubset-Rekonfiguration. SM-Pubsets können durch die Hinzunahme freier leerer Volume-Sets vergrößert werden.

## 5.2 Generierung eines neuen SM-Pubsets durch SIR

### 5.2.1 Ausgangsmaterial

Als Ausgangsmaterial für die Bildung eines SM-Pubsets können einzelne Volumes oder bereits vorgenerierte freie Volume-Sets dienen:

a) einzelne Volumes

SIR bildet bei der Pubset-Generierung in einem internen Zwischenschritt entsprechend den Vorgaben des Systembetreuers aus einzelnen Volumes Volume-Sets. Dabei werden die Volumes neu formatiert, falls sie nicht schon formatiert sind oder ihre bisherige Formatierung nicht zu dem gewünschten Format des Volume-Sets passt. Die Volume-Sets erhalten auch die von dem Systembetreuer geforderten DRV-, RAID oder REMOTE-COPY-Eigenschaften.

b) vorgenerierte freie Volume-Sets

Mit Ausnahme des Control-Volume-Sets, der bei der Pubset-Generierung immer neu erzeugt wird, können freie Volume-Sets komplett in den SM-Pubset übernommen werden. Die ihnen fest anhaftenden Merkmale (z.B. Volume-Konfiguration, REMOTE-COPY, DRV, Format, Allocation-Unit) können nicht geändert werden.

### 5.2.2 Vorbereitung der Pubset-Generierung

Wie bei SF-Pubsets, muss ein MRSCAT-Eintrag für den Pubset nicht vorhanden sein. Er wird in diesem Fall durch SIR automatisch erzeugt. Ist aber bereits ein MRSCAT-Eintrag angelegt, dessen Kennung mit der Pubset-Kennung übereinstimmt, so werden die darin abgelegten Informationen (Pubset-Typ, Kennung des Control-Volume-Sets, Device-Typ der Volres) mit den Angaben des Systembetreuers gegengeprüft.



Im Unterschied zu SF-Pubsets muss aber darauf geachtet werden, dass an jedem Rechner, an dem der SM-Pubset später in Betrieb genommen werden soll, nicht nur für den Pubset selbst, sondern auch für jeden seiner Volume-Sets MRSCAT-Einträge angelegt werden können. Dies ist nur möglich, wenn nicht bereits namensgleiche MRSCAT-Einträge vorhanden sind, welche für andere SF- oder SM-Pubsets oder die Volume-Sets anderer SM-Pubsets benötigt werden. Vor der Bildung des SM-Pubsets sollte der Systembetreuer daher dafür sorgen, dass sich an keinem Rechner, an dem der SM-Pubset in Betrieb genommen werden soll, Konflikte mit der übrigen Konfiguration ergeben. An dem Rechner, an dem der SM-Pubset erzeugt wird, wird dies durch SIR überprüft. Ist dort bereits ein MRSCAT-Eintrag vorhanden, der mit der Kennung eines der Volume-Sets des SM-Pubsets übereinstimmt, wird die Generierung zurückgewiesen.

An dem Rechner, an dem ein SM-Pubset generiert wird, wird für jeden seiner Volume-Sets ein MRSCAT-Eintrag angelegt. Voraussetzung ist, dass dabei die an dem Rechner maximal mögliche Anzahl von MRSCAT-Einträgen nicht überschritten wird (Systemparameter DMCMAXP). Ist diese Voraussetzung nicht erfüllt, wird die Generierung des SM-Pubsets zurückgewiesen.

### 5.2.3 Festlegung der Pubset-Merkmale

Bezüglich der Festlegung der einzelnen Pubset-Merkmale durch SIR sind folgende Fälle zu unterscheiden:

- Vorgabe durch den Systembetreuer
- Ermittlung nach einem festen Verfahren
- keine Berücksichtigung bei der Generierung durch SIR

Nur ein kleiner Teil der Pubset-Merkmale gehört der ersten Kategorie an, welche für den Systembetreuer explizite Steuerungsmöglichkeiten vorsieht. Dem liegt der Gedanke zu Grunde, dass SIR einen Basis-Pubset bereitstellt, der in darauf aufbauenden Installationsschritten mit den üblichen Funktionen der Pubset-Pflege nachbehandelt wird. Steuerungsmöglichkeiten werden vor allem für die Merkmale des Basis-Pubsets angeboten, die sich später nicht oder nur schwer ändern lassen. Im Folgenden werden die einzelnen Merkmale und ihre Festlegung kurz besprochen.

#### **Vom Systembetreuer beeinflussbare Pubset-Merkmale**

##### *Kennung des Pubsets*

Bei der Festlegung der Pubset-Kennung sind die oben beschriebenen Randbedingungen bezüglich der Namenskollisionen zu beachten. Insbesondere muss die Kennung des Pubsets von der Kennung aller dem Pubset angehörenden Volume-Sets verschieden sein.

### *Physikalische Struktur*

Für das Festlegen der physikalischen Struktur des SM-Pubsets sind folgende Angaben möglich:

- Control-Volume-Set des SM-Pubsets
- Menge der sonstigen dem Pubset angehörenden Volume-Sets
- Menge der zu den einzelnen Volume-Sets gehörenden Volumes (für nicht bereits vorgenerierte Volume-Sets)
- Format und Größe der Allocation-Unit der einzelnen Volume-Sets (für nicht bereits vorgenerierte Volume-Sets)
- DRV-, RAID und REMOTE-COPY-Unterstützung für die einzelnen Volume-Sets (für nicht bereits vorgenerierte Volume-Sets)

### *Lage und Größe der Katalogdateien*

SIR gestattet dem Systembetreuer für die Katalogdateien der einzelnen Volume-Sets (`$TSOS.TSOSCAT.<volset-id>`) die Größe und Lage innerhalb der Volume-Sets zu spezifizieren. Die Größe und Lage der pubset-globalen Katalogdateien (`$TSOS.TSOSCAT.#JVC/#MIN / #PVT / $NLO / $NLC / $PFI`) wird in Abhängigkeit von der Größe der Katalogdateien der einzelnen Volume-Sets automatisch bestimmt. Falls dem Systembetreuer die ihm zur Verfügung stehenden Einflussmöglichkeiten für die pubset-globalen Katalogdateien nicht ausreichen, sollten bei der Generierung für die einzelnen Volume-Sets kleine Katalogdateien angelegt werden, was auch zu kleinen pubset-globalen Katalogdateien führt. Alle Katalogdateien können später im laufenden Pubset-Betrieb vergrößert werden, wobei ihre Lage auf den einzelnen Volumes mit den Möglichkeiten der physikalischen Allokierung festgelegt werden kann.

### *Eigenschaftsprofile der Volume-Sets*

Die Eigenschaftsprofile der Volume-Sets umfassen neben dem Format und der Größe der Allocation-Unit, welche den einzelnen Volume-Sets direkt durch die Initialisierung/Formattierung aufgeprägt werden, die Festlegung des Performance- und Verfügbarkeitsprofils. Letztere sind vom Systembetreuer so zu spezifizieren, dass sie die physikalischen Eigenschaften der Volume-Sets korrekt beschreiben. Wenn diese durch Maßnahmen nach der Pubset-Generierung beeinflusst werden - beispielsweise durch die Zuordnung von Cache-Bereichen zur Erhöhung der Performance, sind die später vorgesehenen physikalischen Eigenschaften zu berücksichtigen.

### *Nutzungsarten für Volume-Sets*

Die Nutzungsart der einzelnen Volume-Sets (STD, WORK, HSMS-CONTROLLED) kann vorgegeben werden.

### *Initialausstattung mit Anwenderdateien*

Der Systembetreuer kann veranlassen, dass bei der Pubset-Generierung Dateien von anderen Pubsets, Privatplatten oder Bändern auf den SM-Pubset kopiert werden. Dabei ist es im Unterschied zu SF-Pubsets nicht möglich, die pubset-interne Lage der Dateien zu beeinflussen.

### *Large-Objects Eigenschaft*

Der Systembetreuer hat die Large-Objects-Eigenschaft für den SM-Pubset explizit oder implizit festzulegen (siehe auch [Abschnitt „Large-Objects-Eigenschaft“ auf Seite 38](#)). SIR validiert die Angaben gegenüber der physikalischen Konfiguration. Ein SM-Pubset, der Volume-Sets mit großen Volumes aufnehmen kann, wird nur dann eingerichtet, wenn vom Systembetreuer LARGE-DISKS-ALLOWED=\*YES angegeben wurde. Soll der SM-Pubset auch für große Dateien eingerichtet werden, muss der Systembetreuer LARGE-DISKS-ALLOWED=\*YES(LARGE-FILES-ALLOWED=\*YES) angeben.

## **Von SIR fest voreingestellte Pubset-Merkmale**

Folgende Pubset-Merkmale werden nach einem nicht beeinflussbaren Verfahren bestimmt:

### *Konfigurationszustand der Volume-Sets*

Nach der Bildung eines SM-Pubsets sind alle seine Volume-Sets im Konfigurationszustand „normal use“.

### *Nutzungsrestriktionen für Volume-Sets und Volumes*

Für die Volume-Sets bestehen keine Nutzungsrestriktionen. Gleiches gilt für Volumes von Volume-Sets, die im Rahmen der Pubset-Generierung neu gebildet werden. Bei vorgenerierten freien Volume-Sets ist es von der Vorgeschichte des Volume-Sets abhängig, ob die zugehörigen Volumes mit Allokierungsrestriktionen belegt sind.

### *Cache-Zuordnung für Volume-Sets*

Bei der Pubset-Generierung erfolgt für jeden Volume-Set eine Voreinstellung der für ihn gewünschten Cache-Konfiguration und Cache-Betriebsparameter mit Standardwerten (CACHE-MEDIUM=\*NO, CACHE-SIZE=...). Diese werden bei SM-Pubsets im Gegensatz zu SF-Pubsets nicht im MRSCAT-Eintrag des Pubsets, sondern auf dem Pubset selbst abgelegt. Wenn bei der Pubset-Generierung nicht bereits ein MRSCAT-Eintrag für den Pubset vorhanden ist, wird dieser automatisch erzeugt. Dabei werden die Einstellungen für FORCE-IMPORT und SIZE-TOLERANCE, durch die der Systembetreuer die Behandlung von Cache-Problemen bei der Pubset-Inbetriebnahme steuern kann, mit Standardwerten initialisiert (FORCE-IMPORT=\*NO, SIZE-TOLERANCE=\*YES).

### *Sättigungsschwellwerte der Volume-Sets*

Die Sättigungsschwellwerte (defined-Werte) der einzelnen Volume-Sets erhalten den Wert BY-SYSTEM zugeordnet. Dieser bewirkt, dass bei der Pubset-Inbetriebnahme die current-Werte für die einzelnen Volume-Sets durch das System nach einem bestimmten Berechnungsverfahren ermittelt werden, in das die Volume-Set-Kapazität und der Systemparameter L4SPDEF eingehen.

### *Voreinstellungen für Pubset-Space (einschließlich der Voreinstellung für File-Format)*

Die Voreinstellungen für Primär-, Sekundär- und Maximalallokierung werden entsprechend den Systemparametern DMMAXSC, DMPRALL, DMSCALL des Rechners gewählt, an dem die Generierung erfolgt. Die Voreinstellung für das Dateiformat ist identisch mit dem Format des Control-Volume-Sets.

### *Benutzerkatalog*

Bei der Pubset-Generierung erfolgt implizit eine Pubset-Inbetriebnahme mit ACTUAL-JOIN=\*FIRST. Dabei wird der Benutzerkatalog mit Standard-Benutzerkennungen angelegt.

### *Lage und Größe von Systemdateien*

Mit Ausnahme der Katalogdateien kann die Lage und die Größe der von SIR angelegten Systemdateien (\$TSOS.SYS.PUBSET.CONFIG, \$TSOS.SYSSRPM,...) durch den Systembetreuer nicht beeinflusst werden.

### *Nutzung des Pubsets im Verbund*

Die Festlegung des gewünschten Pubset-Masters und Backup-Masters, der Shareability etc. erfolgt mit den Standardwerten (wie bei SF-Pubsets).

## **Von SIR nicht tangierte Pubset-Merkmale**

Nach der Pubset-Generierung sind noch nicht eingerichtet:

- Storage-Klassen
- GUARDS-Profile
- Benutzergruppen
- SYSEAM-Datei
- HSMS-Umgebung mit Hintergrundebenen, Backup-Archiven und HSMS-Management-Klassen (Pubset-Zustand: HSMS-SUPPORTED=\*NO)

## 5.3 Weitere Installationsmaßnahmen

Der mit SIR erzeugte Pubset bietet die Basis für weitere Installationsmaßnahmen. Im Allgemeinen ist es sinnvoll, den neu generierten Pubset speziell für die Nachbehandlung in Betrieb zu nehmen und dabei die von SIR fest vorgegebenen Einstellungen entsprechend dem jeweiligen Bedarf zu modifizieren. Dabei kann auch die Pubset-Umgebung eingerichtet werden, die durch SIR unberücksichtigt bleibt. Im Folgenden werden die wichtigsten Schnittstellen und Produkte aufgezählt, die dies ermöglichen. Für ihre Nutzung muss der Pubset in Betrieb sein.

### 5.3.1 Anpassung der von SIR getroffenen Voreinstellungen

#### Erweiterung des Dateikatalogs

Die von SIR angelegten Katalogdateien können vergrößert werden. Durch Verwendung von physikalischer Allokierung ist es möglich, ihre Lage auf den Volumes zu beeinflussen.

#### Nutzungsrestriktionen für Volume-Sets

Die für einzelne Volume-Sets bzw. Volumes gewünschten Nutzungsrestriktionen können durch das Kommando `MODIFY-PUBSET-RESTRICTIONS` eingestellt werden.

#### Sättigungsschwellwerte für Volume-Sets

Die für die einzelnen Volume-Sets gewünschten Sättigungsschwellwerte lassen sich durch das Kommando `MODIFY-SPACE-SATURATION-LEVELS` anpassen.

#### Voreinstellungen für Pubset-Space (einschließlich Voreinstellung für File-Format)

Die Voreinstellungen für Pubset-Space können durch das Kommando `MODIFY-PUBSET-SPACE-DEFAULTS` angepasst werden.

#### Storage-Klassen

Zum Einrichten von Storage-Klassen stehen die Funktionen des Storage-Klassen-Managements zur Verfügung. Der Storage-Klassen-Katalog kann auch aus Sicherungen eingelesen werden.

## Benutzereinträge

Der Systembetreuer kann durch die Kommandos der Benutzerverwaltung (ADD-USER, MODIFY-USER-ATTRIBUTES) die von ihm gewünschten Benutzereinträge erzeugen. Es können auch Sicherungen von Benutzerkatalogen eingespielt werden, die auch an anderen Pubsets, insbesondere auch SF-Pubsets erstellt worden sind.

## HSMS-Umgebung

Mit HSMS-Funktionen (CREATE-ARCHIVE, CREATE-SM-PUBSET-PARAMETERS, CREATE-MANAGEMENT-CLASS...) können die HSMS-Umgebung, die HSMS-Konfiguration für Hintergrundebenen und Backup sowie die HSMS-Management-Klassen eingerichtet werden.

## Cache-Zuordnung für Volume-Sets

Bei PFA lässt sich durch das Kommando MODIFY-PUBSET-CACHE-PROCESSING für einen Volume-Set die Voreinstellung bzgl. gewünschter Cache-Konfiguration und Cache-Betriebsparameter ändern. Zusätzlich müssen auf physikalischer Ebene dazu passende Cache-Konfigurationen eingerichtet werden. Die Cache-Konnektierung erfolgt dann bei der nächsten Pubset-Inbetriebnahme oder durch Aufruf des Kommandos START-PUBSET-CACHING für die einzelnen Volume-Sets.

## 5.4 Klonen eines SM-Pubsets mit PVSREN

In Symmetrix-Plattensubsystemen besteht mit Symmetrix Multi Mirror Facility die Möglichkeit, Platten zusätzliche Spiegel als BCVs (Business Continuance Volumes) zuzuordnen, die bei Bedarf abgetrennt und eigenständig verwendet werden können (siehe Handbuch „SHC-OSD“ [8]). Aus den BCV-Spiegeln der Platten eines SM- (und auch SF-) Pubsets kann mit PVSREN ein eigenständiger SM-Pubset mit neuen Katalogkennungen für Pubset und Volume-Sets erzeugt werden. Dieser Vorgang wird als „Klonen“ eines Pubsets bezeichnet. Bezüglich der Daten handelt es sich dabei um eine Kopie des Ausgangs-Pubsets. Einsatzfälle für geklonte Pubsets wären z.B. Auswertungen auf Basis eines eingefrorenen Datenbestands oder die Nutzung für Testzwecke.

Weitere Einzelheiten zum Klonen eines Pubsets sind dem Kapitel „PVSREN“ des Handbuchs „Dienstprogramme“ [1] zu entnehmen.





---

# 6 Erzeugen eines SM-Pubsets aus bestehenden SF-Pubsets

## 6.1 Grundsätzliche Aspekte

Damit Benutzer SM-Pubsets für einen bereits vorhandenen Dateibestand nutzen können, müssen bestehende SF-Pubsets in SM-Pubsets umgewandelt werden. Der Fall, dass ein SF-Pubset in einen namensgleichen SM-Pubset übergeführt wird, kann besonders einfach abgewickelt werden. Oft wird es auch erwünscht sein, mehrere SF-Pubsets zu einem SM-Pubset zusammenzufassen. Für die Umwandlung sind verschiedene Vorgehensweisen möglich, die wir anschließend näher betrachten. Unabhängig von dem gewählten Vorgehen sind folgende grundsätzlichen Aspekte zu beachten:

### **Änderung der Adressierung von Dateien, Jobvariablen, GUARDS-Profilen**

Die Kennung des Pubsets ist als Catid Bestandteil des Pfadnamens der in dem Pubset katalogisierten Dateien. Geht ein SF-Pubset in einen SM-Pubset ein, dessen Kennung sich von der des SF-Pubsets unterscheidet, ändert sich die Adressierung der Dateien (siehe [Bild 10](#)). Analoges gilt auch für Jobvariablen und GUARDS-Profile. Wir betrachten in diesem Abschnitt die Adressierungsproblematik anhand der Dateien.

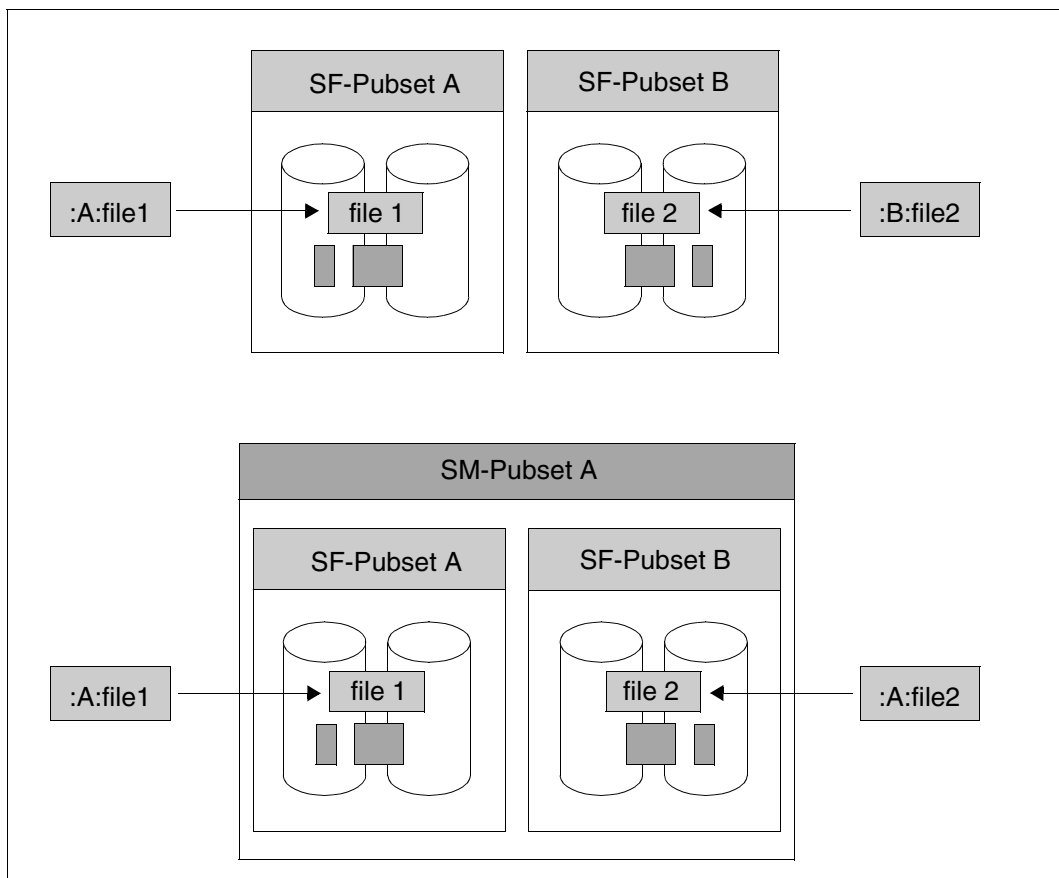


Bild 10: Änderung der Pfadnamen bei der Bildung des SM-Pubsets aus SF-Pubsets

Der veränderten Adressierung der Dateien kann auf folgende Weise Rechnung getragen werden:

- Soweit die Benutzer zur Adressierung von Dateien die Default-Catid (d.h. die Catid des Default-Pubsets) benutzt haben, ist es ausreichend, dass der Systembetreuer die Default-Catids in den Benutzereinträgen (der potenziellen Home-Pubsets) anpasst.
- Wenn Benutzer Dateien mit vollständigem Pfadnamen adressiert haben, d.h. mit expliziter Angabe der Catid (z.B. in Kommandoprozeduren, Benutzerprogrammen oder sonstigen Benutzerdatenstrukturen), müssen sie die Pfadnamen anpassen. Eine Beispiel für die Verwendung vollständiger Pfadnamen sind in GUARDS-Profilen referenzierte Programmnamen.

- Ein Grund für die Verwendung vollständiger Pfadnamen besteht darin, dass ein Benutzer seine Dateien über mehrere SF-Pubsets verteilt, um die unterschiedlichen Eigenschaften der einzelnen SF-Pubsets (Performance, Verfügbarkeit, Format) gezielt zu nutzen. Enthalten Kommandoprozeduren oder Benutzerprogramme Anweisungen, durch welche Dateien mit Angabe des vollständigen Pfadnamens angelegt werden, müssen ggf. Anpassungen erfolgen, damit die Dateien auch innerhalb des SM-Pubsets auf einen geeigneten Volume-Set gelangen. Den Dateien müssen die gewünschten Eigenschaften (Performance, Verfügbarkeit, Format) explizit durch Dateiattribute zugewiesen werden.
- Bei SF-Pubsets ergeben sich für Dateien, welche mit der Default-Catid angelegt werden, bestimmte Eigenschaften implizit aus den Eigenschaften des jeweiligen Default-Pubsets. Damit Dateien, die über eine Default-Catid angelegt werden, auch nach dem Übergang auf einen SM-Pubset die gleichen Eigenschaften erhalten wie bisher, kann der Systembetreuer den einzelnen Benutzern Default-Storage-Klassen zuweist, welche die gleichen Eigenschaften repräsentieren, wie die bisher zugeordneten Default-Pubsets. Die Kommandoprozeduren und Programme brauchen nicht geändert zu werden. Wenn beim Anlegen einer Datei keine Dateiattribute spezifiziert werden, werden der Datei die Werte zugewiesen, welche in der als Voreinstellung angegebenen Storage-Klasse des Benutzers hinterlegt sind.

### **Kollisionen und Überlängen von Pfadnamen**

Befinden sich auf verschiedenen SF-Pubsets, die in den SM-Pubset eingehen, Dateien, deren Pfadnamen sich lediglich durch die Catid unterscheidet, so würden sich bei der Bildung des SM-Pubsets Namenskollisionen ergeben, d.h. die Pfadnamen wären anschließend nicht mehr eindeutig. Davon können nur Benutzer betroffen sein, deren Dateien über mehrere SF-Pubsets verteilt sind. Geht ein SF-Pubset in einen SM-Pubsets ein, dessen Kennung mehr Zeichen enthält als die des SF-Pubsets, verlängern sich die Pfadnamen der auf dem SF-Pubset liegenden Dateien. Dadurch können sich Überschreitungen der maximalen Pfadnamenlänge ergeben. Namenskollisionen und Überlängen von Pfadnamen sind wie folgt zu berücksichtigen:

- Die betroffenen Dateien werden umbenannt oder gelöscht.
- In Kommandoprozeduren oder Programmen werden Anweisungen mit Referenzen auf Dateien, die umbenannt werden, angepasst.

### **Inkompatible Kommandos, Anweisungen, Programmschnittstellen**

Einige Systemkommandos, Programmschnittstellen und Anweisungen einzelner Utilities sind für SM-Pubsets nicht in jeder für SF-Pubsets erlaubten Ausprägung verwendbar. So müssen HSMS-Aufträge für Backup, Restore, Migrate und Recall bei SM-Pubsets pubset-spezifisch gebildet werden. Pubset-übergreifende Aufträge sind im Unterschied zu SF-Pubsets nicht möglich. Dies tangiert z.B. auch die Privatplattendateien. Für ihre komplette Sicherung ist für jeden SM-Pubset, in dem eine Privatplattendatei katalogisiert ist, ein eigener Backup-Auftrag erforderlich. Kommandoprozeduren und Programme, die von solchen Inkompatibilitäten betroffen sind, müssen angepasst werden.

### **Anpassung der MRSCAT-Einträge**

An jedem Rechner, an dem der SM-Pubset in Betrieb genommen wird, müssen die MRSCAT-Einträge angepasst werden:

- Es muss ein MRSCAT-Eintrag vom Typ SM-Pubset angelegt werden.
- Die MRSCAT-Einträge der SF-Pubsets, aus denen der SM-Pubset gebildet wird, werden nicht mehr benötigt und sollten gelöscht werden.
- Für jeden Volume-Set des SM-Pubsets ist sicherzustellen, dass nicht bereits ein MRSCAT-Eintrag für einen SF-Pubset-, SM-Pubset oder Volume-Set eines anderen SM-Pubsets vorhanden ist, dessen Kennung mit der des Volume-Sets übereinstimmt.

### **Sonstige Referenzen auf die Kennungen von SF-Pubsets**

Alle sonstigen Referenzen auf Kennungen von SF-Pubsets, die nach der Bildung des SM-Pubsets nicht mehr gültig sind, müssen angepasst werden, wie die Default-Catids in den Benutzereinträgen potenzieller Home-Pubsets, Pubset-Kennungen in Kommandoprozeduren und Benutzerprogrammen.

## 6.2 Übersicht über die Vorgehensweisen

In die Umwandlung von SF-Pubsets in einen SM-Pubset sind Verarbeitungsebene, Hintergrundebenen und Backup-Archive einzubeziehen. Wir skizzieren im Folgenden mögliche Vorgehensweisen. Sie werden in den folgenden Abschnitten weiter präzisiert.

### 6.2.1 Anpassung der Verarbeitungsebene

Für die Anpassung der Verarbeitungsebene bestehen folgende Möglichkeiten:

#### a) In-place-Konvertierung

Die in-place-Konvertierung der Verarbeitungsebene wird durch die Utility SMPGEN ermöglicht. Sie gestattet aus einem oder mehreren SF-Pubsets einen neuen SM-Pubset zu bilden bzw. einen bereits bestehenden SM-Pubset zu erweitern (siehe dazu auch [Abschnitt „Erweitern eines SM-Pubsets mit SF-Pubsets“ auf Seite 162](#)), wobei die SF-Pubsets jeweils in einen Volume-Set übergeführt werden. Die auf den SF-Pubsets liegenden Benutzerdateien bleiben bei der Zusammenführung unverändert erhalten, die Metadatendateien der SF-Pubsets wie Dateikataloge, Benutzerkataloge, GUARDS-Kataloge etc. werden in entsprechende Metadatendateien des SM-Pubsets umgewandelt. Bestimmte Pubset- und Volume-Set-Merkmale, Benutzerkontingente, etc. werden für den SM-Pubset durch SMPGEN automatisch ermittelt. Falls der Systembetreuer davon abweichende Einstellungen wünscht, muss er den SM-Pubset nach der Generierung entsprechend nachbehandeln.

Bei SF-Pubsets lässt sich für Dateien eine bestimmte Verfügbarkeit oder Performance dadurch erzielen, dass sie auf spezielle SF-Pubsets gelegt werden (wie durch DRV oder REMOTE-COPY gespiegelte Pubsets, etc.), ohne dass diese Eigenschaften explizit durch Dateiattribute spezifiziert werden. Dateien, die auf SF-Pubsets mit speziellen Eigenschaften liegen, müssen bei der Bildung des SM-Pubsets entsprechende Dateiattribute zugewiesen werden. Bei SF-Pubsets, die in Volume-Sets mit dem Attribut AVAILABILITY=\*HIGH übergeführt werden, geschieht dies automatisch durch SMPGEN. Bei SF-Pubsets mit Caches, welche über PFA genutzt werden, müssen den Dateien bereits vor der Bildung des SM-Pubsets passende Performance-Attribute zugewiesen sein. In allen übrigen Fällen, z.B. bei SF-Pubsets, die aus im Globalspeicher emulierten Volumes bestehen, muss der Systembetreuer die Attributierung explizit vornehmen. Werden den Dateien keine passenden Attribute zugeordnet, bleibt ihre Lage bei der Pubset-Bildung durch SMPGEN zwar zunächst gleich, bei einem späteren Sichern und Wiedereinspielen bzw. Verdrängen und Zurückholen in die Verarbeitungsebene wird aber der Ablageort neu bestimmt, wobei die nicht angepassten Dateiattribute zum Tragen kommen.

## b) Sichern und Wiedereinspielen von Dateien

Für alle SF-Pubsets, die in den SM-Pubset eingehen, erfolgt eine Sicherung der in der Verarbeitungsebene liegenden Dateien, ggf. im Rahmen einer Vollsicherung unter Einbeziehung der Hintergrundebenen. Anschließend wird mit SIR der SM-Pubset neu erzeugt. Dann legt der Systembetreuer auf dem SM-Pubset für die bisherigen Benutzer der SF-Pubsets Benutzereinträge an. Er wird dabei durch HSMS unterstützt, mit dem er die Sicherung des Benutzerkatalogs eines SF-Pubsets wiedereinspielen kann. Dabei erhalten die für SM-Pubsets neu eingeführten Benutzerattribute (z.B. Benutzerkontingente für hochverfügbare Dateien) Voreinstellungen zugewiesen, die eine möglichst große Kompatibilität zu den SF-Pubsets gewährleisten. Wenn für die übrigen SF-Pubsets, die in den SM-Pubset eingehen, noch weitere Benutzer vorhanden waren, muss der Systembetreuer für diese mit dem Kommando ADD-USER explizit neue Benutzereinträge erstellen. Anschließend werden GUARDS-Kataloge eingerichtet, wobei wiederum auf SF-Pubsets erstellte Sicherungen benutzt werden können. Schließlich werden die Benutzerdateien wiedereingespielt.

Den Dateien, welche bisher auf SF-Pubsets mit spezieller Performance oder Verfügbarkeit liegen, müssen vor dem Sichern passende Dateiattribute zugeordnet werden, damit sie beim Wiedereinspielen auf den SM-Pubset auf geeignete Volume-Sets gelangen.

Soll beim Wiedereinspielen die Verteilung der Dateien auf dem SM-Pubset durch Default-Storage-Klassen mit zugeordneten Volume-Set-Listen beeinflusst werden, müssen zuvor passende Storage-Klassen eingerichtet und den einzelnen Benutzern als Default-Storage-Klassen zugewiesen werden. Damit sie bei dem Restore durch HSMS zum Tragen kommen, muss in der HSMS-Anweisung RESTORE-FILES die Angabe STORAGE-CLASS=\*STD erfolgen.

Die in-place-Konvertierung durch SMPGEN ermöglicht einen sehr komfortablen Umstieg. Dies gilt insbesondere auch für das Erstellen des Benutzerkatalogs sowie der GUARDS-Kataloge wenn mehrere SF-Pubsets in den SM-Pubset eingehen. Wenn aber für den SM-Pubset ein anderer physikalischer Aufbau gewünscht wird, als er sich durch die in-place-Konvertierung der SF-Pubset ergäbe, muss der Umstieg durch Sichern und Wiedereinspielen von Dateien erfolgen. Ein Beispiel dafür ist der Fall, dass ein aus vielen Volumes bestehender SF-Pubset in einen SM-Pubset aus mehreren kleinen Volume-Sets übergeführt werden soll.

## 6.2.2 Anpassung der Hintergrundebenen für Verdrängung

Die Anpassung der Hintergrundebenen für Verdrängung kann auf folgende Weise erfolgen:

a) In-place-Konvertierung der S2-Ebene

Die HSMS-Utility DIRCONV erlaubt aus den Migrations-Directories der SF-Pubsets ein neues Directory zu erzeugen, das später als Migrations-Directory des SM-Pubsets benutzt werden kann. Zugriffe auf die Datenträger mit den migrierten Dateien sind dabei nicht erforderlich. Allerdings müssen für die in-place-Konvertierbarkeit bestimmte Bedingungen erfüllt sein, die ggf. zuvor mithilfe von HSMS-Funktionen hergestellt werden müssen. Beispielsweise dürfen auf den betroffenen SF-Pubsets keine Dateien auf die S1-Ebene verdrängt sein. Um die Voraussetzungen für die in-place-Konvertierung der Migrations-Directories herzustellen, können sich auch Zugriffe auf die Datenträger mit den migrierten Dateien als notwendig erweisen. Der Umfang dieser Bereinigungsmaßnahmen hängt stark von dem jeweiligen Anwendungsumfeld ab. Das eigentliche Übertragen der S2-Ebenen der SF-Pubsets auf den SM-Pubsets erfolgt dadurch, dass der Systembetreuer nach der Anpassung der Verarbeitungsebene eine HSMS-Umgebung für den SM-Pubset einrichtet und dabei das vorher durch Konvertierung erzeugte Directory als Migrations-Directory zuweist.

b) Anpassung der Hintergrundebenen durch Sichern und Wiedereinspielen von Dateien

Die auf den Hintergrundebenen befindlichen Dateien der SF-Pubsets werden gesichert (ggf. im Rahmen einer Vollsicherung der in den SM-Pubset eingehenden SF-Pubsets mit Einbeziehung der Hintergrundebenen). Nachdem die Anpassung der Verarbeitungsebene erfolgt ist, wird auf dem SM-Pubset eine HSMS-Umgebung mit (zunächst leeren) Hintergrundebenen eingerichtet. Auf diese werden die Sicherungen der früher auf den Hintergrundebenen der SF-Pubsets liegenden Dateien wiedereingespilt.

Die in-place-Konvertierung der S2-Ebene ist gegenüber der Anpassung durch Sichern und Wiedereinspielen dann zu empfehlen, wenn die Voraussetzungen für die Konvertierung der Migrations-Directories von vornherein gegeben sind oder einfach hergestellt werden können.

## 6.2.3 Anpassung der Backup-Archive

Die Backup-Archive der SF-Pubsets, welche in den SM-Pubset eingehen, können ähnlich wie die Migrationsarchive angepasst werden, indem zunächst mithilfe von DIRCONV aus den bestehenden Directories für den SM-Pubset geeignete Directories erzeugt werden. Nach dem Einrichten der HSMS-Umgebung auf dem SM-Pubset können darauf pubset-spezifische Backup-Archive eingerichtet werden, denen die zuvor passend vorbereiteten Backup-Directories zugewiesen werden. Dies gilt insbesondere auch für das System-Backup-Archiv des SM-Pubsets.

Das Anpassen der Backup-Directories durch DIRCONV erfolgt in-place, d.h. ohne Zugriff auf die Datenträger, auf denen die Dateisicherungen abgelegt sind. Wie bei den Migrations-Directories müssen für die Konvertierbarkeit bestimmte Voraussetzungen erfüllt sein, wobei diese für bestehende Backup-Archive meist wesentlich schwieriger herzustellen sind und im Allgemeinen auch Zugriffe auf die Datenträger mit den Dateisicherungen erfordern. Der Vorbereitungsaufwand hängt sehr stark von dem jeweiligen Anwendungsumfeld ab.

In Fällen, in denen die Anpassung der bestehende Backup-Archivs zu aufwändig oder schwierig ist, ist es ratsam, unmittelbar vor der Bildung des SM-Pubsets (d.h. nachdem die für die Anpassung der Verarbeitungsebene notwendigen Bereinigungen erfolgt sind und die SMPGEN-Checks keine Verstöße mehr aufzeigen) ein neues Backup-Archiv einzurichten und in dieses aktuelle Dateisicherungen einzubringen. Für das Backup-Archiv, das später in das System-Backup-Archiv des SM-Pubsets übergeführt wird, empfiehlt sich eine aktuelle Vollsicherung aller SF-Pubsets, die in den SM-Pubset eingehen. Bei diesem Vorgehen entfallen alle schwierig durchzuführende Bereinigungsmaßnahmen für die Anpassung der Archive. Beispielsweise können keine Kollisionen von Dateinamen auftreten, da wegen der zuvor durchgeführten SMPGEN-Checks nur Dateien in die Sicherung eingehen, deren Namen nicht kollidieren. Die bisherigen System-Backup-Archive der SF-Pubsets, die in den SM-Pubset eingehen, werden unangepasst beibehalten. Sie können herangezogen werden, wenn nach Bildung des SM-Pubsets Zugriffe auf Dateisicherungen notwendig sind, die vor der Bildung des SM-Pubsets erstellt wurden und nicht in das System-Backup-Archiv des SM-Pubset eingegangen sind. HSMS bietet nämlich die Möglichkeit, auf SM-Pubsets (wie auch auf SF-Pubsets) Dateisicherungen aus beliebigen Backup-Archiven für SF-Pubsets oder SM-Pubsets einzuspielen. Backup-Archive für SF-Pubsets können aber nicht benutzt werden, um Sicherungen von SM-Pubsets aufzunehmen.

#### 6.2.4 Maßnahmen, die ein Rücksetzen ermöglichen

Wenn bei Anpassungsmaßnahmen Probleme auftreten, ist es oft wichtig, auf einen bestimmten Ausgangsstand rücksetzen zu können. Dies erfordert im Allgemeinen zusätzliche Vorkehrungen, wie das Erstellen von Sicherungskopien (z.B. der Directories von Migrations- und Backup-Archiven). Um die Darstellung möglichst einfach zu halten, wird im [Abschnitt „Konvertierung der S0-Ebene durch SMPGEN“ auf Seite 75](#) und im [Abschnitt „Anpassung der Migrations- und Backup-Archive“ auf Seite 86](#), welche die Anpassung der Verarbeitungsebene, Hintergrundebene und der Backup-Archive beschreiben, das Rücksetzen weitgehend ausgeklammert. Es wird an Stelle dessen übergreifend in [Abschnitt „Rückstieg von einem SM-Pubset auf SF-Pubsets“ auf Seite 117](#) behandelt. Dem Systembetreuer wird dringend empfohlen, die für das Rücksetzen erforderlichen Schritte bereits als Teil des Vorgehens für die Konvertierung einzuplanen.



## 6.2.5 Vorbereitungen

Die Bildung des SM-Pubsets aus SF-Pubsets bedarf einer sorgfältigen Vorbereitung, wobei der Systembetreuer und die Benutzer betroffen sind. Die Utilities SMPGEN und DIRCONV stellen hierfür Funktionen bereit.

- Erfolgt die Anpassung der SF-Pubsets mithilfe von SMPGEN, müssen die betroffenen SF-Pubsets so vorbereitet werden, dass SMPGEN sie konsistent zu einem SM-Pubset zusammenführen kann. Beispielsweise dürfen sich keine Konflikte oder Überlängen bei Dateinamen ergeben. Für die Durchführung der erforderlichen Bereinigungen unterstützt SMPGEN die Benutzer und den Systembetreuer durch eine CHECK-Funktion. Erst, wenn dabei keine Probleme erkannt werden, ist das Generieren des SM-Pubsets möglich.
- Ein großer Teil der SMPGEN-Checks (z.B. Überprüfungen auf Kollisionen und Überlängen von Dateinamen) sollte unbedingt auch dann durchgeführt werden, wenn die Anpassung der Verarbeitungsebene über Sichern und Wiedereinspielen von Dateien erfolgt (z.B. Überprüfen von Namenskollisionen). Damit können Probleme, die später beim Wiedereinspielen der Dateien oder beim Anpassen der Migrations- und Backup-Archive sehr schwierig zu beheben wären, bereits im Vorfeld entschärft werden.
- Vor der Bildung des SM-Pubsets müssen Vorkehrungen dafür erfolgen, dass später auf ihm die Hintergrundebenen (mit dem Dateibestand der Hintergrundebenen der SF-Pubsets) eingerichtet werden können. Sie sind davon abhängig, ob die Hintergrundebenen durch Sichern und Wiedereinspielen der darauf befindlichen Dateien oder durch in-place-Konvertierung angepasst werden.
- Die Backup-Archive für SF-Pubsets, die in Backup-Archive des SM-Pubsets übergeführt werden sollen, müssen vor der Bildung des SM-Pubsets mithilfe von DIRCONV angepasst werden.
- Von den einzelnen Benutzern muss die Anpassung der Dateiadressierung und der Austausch nicht mehr gültiger Catids in Kommandoprozeduren und Programmen durchgeführt werden.
- Von den einzelnen Benutzern müssen Kommandoprozeduren und Programme, welche für SM-Pubsets nicht zulässige Kommandos/Anweisungen enthalten, angepasst werden.
- In GUARDS-Profilen referenzierte Programmnamen sind anzupassen. Erfolgt die Bildung des SM-Pubsets durch SMPGEN, erhält der Systembetreuer für die Anpassung der GUARDS-Kataloge Unterstützung, allerdings nur für die SF-Pubsets, die in den SM-Pubset eingehen.
- An allen Rechnern, an denen der SM-Pubset in Betrieb genommen werden soll, muss der Systembetreuer die MRSCAT-Einträge anpassen. Bei der Bildung des SM-Pubsets mit SMPGEN erfolgt die Anpassung an dem Rechner, an dem SMPGEN abläuft, automatisch.

- Dateien, für welche spezielle Anforderungen hinsichtlich Performance und Verfügbarkeit bestehen, müssen passende Dateiattribute bekommen. Bei SF-Pubsets, die in Volume-Sets mit AVAILABILITY=\*HIGH übergehen, passt SMPGEN das Dateiaattribut Verfügbarkeit automatisch an.

Der Vorbereitungsaufwand hängt von den einzelnen Umstiegsszenarien ab. In bestimmten Fällen ist er sehr gering (z.B. Umwandlung eines SF-Pubsets ohne S1-Ebene mit eigenem Migrations- und Backup-Archiv). In komplexeren Fällen kann er erheblichen Umfang annehmen (z.B. Bildung eines SM-Pubsets aus mehreren SF-Pubsets mit Migrations- und Backup-Archiven, welche durch weitere Pubsets genutzt werden). Konkrete Umstiegsszenarien werden im [Abschnitt „Exemplarische Umstiegsszenarien und Vorgehensweisen“ auf Seite 122](#) besprochen.

## 6.3 Konvertierung der S0-Ebene durch SMPGEN

### 6.3.1 Voraussetzungen

Wir betrachten die Voraussetzungen, die SF-Pubsets erfüllen müssen, damit aus ihnen durch SMPGEN ein SM-Pubset gebildet werden kann.

#### Kennung des SM-Pubsets

Die Kennung des SM-Pubsets darf nicht mit dem Namen eines anderen SF- oder SM-Pubsets oder eines Volume-Sets kollidieren. Insbesondere muss sie sich von den Kennungen aller SF-Pubsets unterscheiden, die in den SM-Pubset eingehen. Soll der SM-Pubset die Kennung eines früheren SF-Pubsets übernehmen, muss dieser SF-Pubset vor der Bildung des SM-Pubsets mithilfe der Utility PVSREN umbenannt werden. Dies ist z.B. dann erforderlich, wenn ein SF-Pubset in einen namensgleichen SM-Pubset mit genau einem Volume-Set umgewandelt wird.

#### Namenskollisionen von Dateien, Jobvariablen, GUARDS-Profilen

Namenskollisionen liegen vor, wenn sich auf mehreren in den SM-Pubset eingehenden SF-Pubsets namensgleiche Dateien, Jobvariablen oder GUARDS-Profile befinden. Namensgleichheit bedeutet dabei, dass sich die Pfadnamen nur durch die Catid unterscheiden. Abgesehen von bestimmten Systemdateien, die durch SMPGEN automatisch umbenannt oder gelöscht werden (z.B. \$TSOS.TSOSCAT, \$TSOS.SYSSRPM), müssen Namenskollisionen vor der Bildung des SM-Pubsets durch die Benutzer bereinigt werden. SMPGEN weist sonst den Generierungsauftrag zurück. Die Bereinigungen können durch Löschen oder Umbenennung erfolgen. Migrierte Dateien müssen vor der Umbenennung in die Verarbeitungsebene zurückgeholt werden.

#### Pfadnamen-Überlängen

Hat einer der SF-Pubsets, aus denen der SM-Pubset gebildet wird, eine kürzere Kennung als der SM-Pubset, verlängern sich durch die SM-Pubset-Generierung die Pfadnamen der auf dem SF-Pubset katalogisierten Dateien, Jobvariablen und GUARDS-Profile. SMPGEN überprüft, ob dadurch die maximal zulässige Pfadnamenlänge von 54 Zeichen überschritten wird. Namensüberlängen können wie Namenskollisionen durch Umbenennen oder Löschen bereinigt werden.

Auch für Programmnamen, die in GUARDS-Bedingungen referenziert werden, kann das Ersetzen der Catid zu unzulässigen Pfadnamenlängen führen. Dabei können auch GUARDS-Profile betroffen sein, die auf SF-Pubsets liegen, welche nicht in den SM-Pubset eingehen. SMPGEN überprüft die Pfadlängen der Programmnamen in GUARDS-Profilen, allerdings nur für die SF-Pubsets, aus denen der SM-Pubset gebildet wird. Die Bildung eines SM-Pubsets wird auch dann zugelassen, wenn SMPGEN nicht bereinigte Programmnamen in GUARDS-Profilen findet (mit Ausgabe von Warnungen). Ihre Bereinigung kann damit auch erst nach Bildung des SM-Pubsets erfolgen.

### **Dateinamen, die für Systemdateien reserviert sind**

Bestimmte Dateinamen, welche auf SF-Pubsets für Benutzerdateien erlaubt sind (wie \$TSOS.TSOSCAT.<volsetid> oder \$TSOS.SYS.PUBSET.CONFIG), sind auf SM-Pubsets für Systemdateien reserviert. Findet SMPGEN auf einem der in den SM-Pubset eingehenden SF-Pubsets eine Datei mit einem solchen Namen vor, wird die Generierung des SM-Pubsets zurückgewiesen.

### **Paging-Dateien**

Auf den in den SM-Pubset eingehenden SF-Pubsets dürfen keine Dateien liegen, deren Pfadnamen mit \$TSOS.SYS.PAGING. beginnt.

### **Dateien für die Systemeinleitung**

Auf den SF-Pubsets liegende Dateien für die Systemeinleitung müssen für die Pubset-Generierung nicht entfernt werden. Sie sind aber nach der Bildung des SM-Pubsets nicht mehr nutzbar.

### **Dateien auf der S1-Ebene für Dateiverdrängung**

Wenn die Migrations-Directories der SF-Pubsets in-place in das Migrations-Directory des SM-Pubsets konvertiert werden sollen, dürfen von keinem der betroffenen Pubsets Dateien in der S1-Ebene für Dateiverdrängung liegen. Der Systembetreuer kann dies mithilfe von SMPGEN überprüfen (Operand: S1-MIGRATED-FILES=\*NOT-ALLOWED).

### **Konnectierte Cache-Bereiche**

Keiner der in den SM-Pubset eingehenden SF-Pubsets darf mit einem Cache-Bereich konnectiert sein. Die Konnectierung mit einem Cache-Bereich kann aufgehoben werden, indem der Pubset importiert wird (ggf. in dem Modus FORCE-IMPORT=\*YES) und anschließend wieder außer Betrieb genommen wird.

## Platz für Systemdateien

Bei der Bildung des SM-Pubsets werden auf dem SF-Pubset, der die Rolle des Control-Volume-Sets übernimmt, Systemdateien angelegt. Für diese muss genügend Platz vorhanden sein.

## Überlaufprobleme

Die Generierung des SM-Pubsets kann an Überlaufproblemen scheitern. Die meisten Überlaufbedingungen (z.B. maximale Anzahl von SF-Pubsets, maximale Anzahl von Benutzern) werden durch SMPGEN vorab geprüft, einige (z.B. Überlauf der Größe der Katalogdateien) werden jedoch erst bei der Durchführung der Generierung erkannt.

### 6.3.2 Vorbereitung und Durchführung

Um die für die Bildung des SM-Pubsets geforderten Voraussetzungen herstellen zu können, wird dem Systembetreuer und den Benutzern die SMPGEN-Check-Funktion mit der Anweisung CREATE-SYSTEM-MANAGED-PUBSET zur Verfügung gestellt. Sie bietet die zwei Ausführungsvarianten „Checks während einer regulären Pubset-Session“ und „Checks in einem dem Systembetreuer vorbehaltenen exklusiven Modus“. Dadurch wird den unterschiedlichen Anforderungen in verschiedenen Vorbereitungsphasen Rechnung getragen. Nicht-privilegierten Benutzern steht die Anweisung CREATE-SYSTEM-MANAGED-PUBSET ausschließlich im Prüfmodus zur Verfügung. Der Systembetreuer muss den Prüfmodus explizit über den Operanden OPERATIONAL-MODE=\*CHECK-NAME-CONSISTENCY(...), der weitere Auswahlmöglichkeiten zu Benutzerkennungen und Pubset-Status anbietet, anfordern.

#### Von den Benutzern durchzuführende Überprüfungen

Zunächst sollten alle Benutzer für ihre Kennungen Namenskonflikte und Namensüberlängen bereinigen. Hierfür eignen sich SMPGEN-Checks im Rahmen von laufenden Pubset-Sessions (siehe CREATE-SYSTEM-MANAGED-PUBSET, Operand PUBSET-STATE=\*IMPORTED). Für die Überprüfungen müssen alle SF-Pubsets, die in den SM-Pubset eingehen, importiert sein.

Für Benutzer ohne TSOS-Privileg beziehen sich die von SMPGEN durchgeführten Checks auf ihre jeweilige Kennung. Der Systembetreuer hat zusätzlich die Möglichkeit zu benutzerübergreifenden Überprüfungen. Er kann damit feststellen, in welchem Umfang die einzelnen Benutzer die Bereinigungen bereits durchgeführt haben. Wenn ihm bei Überprüfungen im Modus PUBSET-STATE=\*IMPORTED keine Verstöße mehr angezeigt worden sind, kann er aber nicht sicher sein, dass sich in der Zwischenzeit durch Aktivitäten einzelner Benutzer nicht neue Verstöße ergeben haben.

### **Durch den Systembetreuer durchzuführende Überprüfungen**

Nachdem die einzelnen Benutzer die Bereinigungen durchgeführt haben, muss der Systembetreuer unmittelbar vor der Konvertierung der SF-Pubsets bestimmte Überprüfungen vornehmen, bei denen er nicht durch Aktivitäten anderer Benutzer gestört werden sollte (z.B. die zuverlässige Überprüfung, ob alle erforderlichen Bereinigungen erfolgt sind). Dies wird durch SMPGEN-Checks im Modus PUBSET-STATE=\*NOT-IMPORTED ermöglicht. Bei dem Aufruf der Check-Funktion in diesem Modus müssen alle in den SM-Pubset eingehenden SF-Pubsets exportiert sein. Für die Ausführung erfolgt ein impliziter Import mit exklusiver Belegung für den Systembetreuer. Wenn bei den unter exklusiver Belegung durchgeführten Checks keine Probleme erkannt werden, ist sichergestellt, dass die anschließende Generierung des SM-Pubsets durch SMPGEN nicht zurückgewiesen wird. Eine erfolgreiche Ausführung ist damit nicht garantiert, da bestimmte Probleme erst bei der Konvertierung erkannt werden (z.B. nicht ausreichender Platz auf dem Control-Volumen-Set).

Die Bildung des SM-Pubsets erfolgt durch Aufruf von CREATE-SYSTEM-MANAGED-PUBSET im Ausführungsmodus (OPERATIONAL-MODE=\*OPERATION ist implizit voreingestellt). Neben passend vorbereiteten SF-Pubsets muss der Systembetreuer auch für eine bestimmte Ablaufumgebung sorgen. Diese ist im Handbuch „Dienstprogramme“ [1] im Detail beschrieben.

Treten während der Generierung des SM-Pubsets Probleme auf, die eine erfolgreiche Beendigung verhindern, so werden im Allgemeinen die SF-Pubsets in ihrem ursprünglichen Zustand wiederhergestellt. Trotzdem kann nicht ausgeschlossen werden, dass in bestimmten Abbruchsituationen (z.B. bei einem System-Crash) ein nicht reparabler Zwischenzustand zurückbleibt. Der Systembetreuer sollte daher vor der Bildung des SM-Pubsets die darin involvierten SF-Pubsets sichern (z.B. mit FDDRL), um sie - falls notwendig - wieder herstellen zu können.

### 6.3.3 Merkmale eines durch SMPGEN erzeugten SM-Pubsets

Bezüglich der Festlegung der einzelnen Pubset-Merkmale durch SMPGEN sind folgende Fälle zu unterscheiden:

- Festlegung entsprechend den Eigenschaften der SF-Pubsets, die in den SM-Pubset eingehen
- Vorgabe durch den Systembetreuer
- automatische Bestimmung durch SMPGEN

Im Folgenden wird kurz dargestellt, wie die einzelnen Pubset-Merkmale durch SMPGEN festgelegt werden.

#### **Kennung des Pubsets**

Die Kennung des SM-Pubsets wird durch den Systembetreuer vorgegeben.

#### **Physikalische Struktur des SM-Pubsets**

Der Systembetreuer bestimmt, welche SF-Pubsets in den SM-Pubset aufgenommen werden, und welcher von ihnen die Rolle des Control-Volume-Sets übernimmt. Auf Grund dieser Angaben ist die physikalische Struktur des SM-Pubsets eindeutig festgelegt, da die Volume-Konfiguration, die Größe der Allocation-Unit, das Format, sowie DRV- und REMOTE-COPY-Eigenschaften bei der Umwandlung der SF-Pubsets in Volume-Sets beibehalten werden. Die Kennungen der Volume-Sets ergeben sich aus den Kennungen der SF-Pubsets.

#### **Large-Objects-Eigenschaft**

Befindet sich unter den SF-Pubsets, die in den SM-Pubset eingehen, ein Pubset mit der Eigenschaft LARGE-OBJECTS, so erhält auch der resultierende SM-Pubset diese Eigenschaft (d.h. er wird ein LARGE-OBJECTS-Pubset).

Befindet sich unter den zusammengeführten SF-Pubsets mit der Eigenschaft LARGE-OBJECTS mindestens einer, der neben großen Volumes (LARGE-DISKS-ALLOWED=\*YES) auch große Dateien unterstützt (LARGE-FILES-ALLOWED=\*YES), so besitzt der SM-Pubset ebenfalls diese Eigenschaft. Ist dies nicht der Fall, können auf dem SM-Pubset keine große Dateien angelegt werden (LARGE-FILES-ALLOWED=\*YES).

#### **Cache-Konfiguration der Volume-Sets**

Die Cache-Konfiguration eines Volume-Sets (Cache-Medium, -Größe, -Betriebsparameter) ergibt sich aus den Festlegungen im MRSCAT-Eintrag des jeweiligen SF-Pubsets.

### **Konfigurationszustand der Volume-Sets**

Nach der Bildung eines SM-Pubsets befinden sich alle seine Volume-Sets in dem Konfigurationszustand „normal use“.

### **Eigenschaftsprofile für Volume-Set-Selektion**

Die Größe der Allocation-Unit sowie das Format der Volumes-Sets sind durch die Eigenschaften der SF-Pubsets bestimmt. Das Performance- und Verfügbarkeitsprofil werden von dem Systembetreuer vorgegeben.

### **Nutzungsarten der Volume-Sets**

Standardmäßig wird allen Volume-Sets die Nutzungsart STD zugewiesen. Daneben kann der Systembetreuer auch für einen Volume-Set die Nutzungsart HSMS-CONTROLLED anfordern. Diese Funktion ist nur in dem speziellen Fall der Wiederherstellung eines SM-Pubsets nach dem Ausfall des Control-Volume-Sets sinnvoll. Damit können Dateisicherungen und Inhalte von nach S1 verdrängten Dateien, die sich auf dem SF-Pubset befinden, der aus dem früheren S1-Volume-Set hervorgegangen ist, wieder in den SM-Pubset eingebracht werden.

### **Nutzungsrestriktionen für Volume-Sets**

Für die Volume-Sets bestehen keine Nutzungsrestriktionen.

### **Nutzungsrestriktionen für Volumes**

Sind einzelne Volumes eines in den SM-Pubsets eingehenden SF-Pubsets mit Allokierungsrestriktionen belegt, werden diese beibehalten.

### **Sättigungsschwellwerte für Volume-Sets**

Die Sättigungsschwellwerte der einzelnen Volume-Sets bleiben gemäß den MRSCAT-Einträgen der SF-Pubsets bestehen.

### **Voreinstellungen für Pubset-Space (einschließlich Voreinstellung des File-Formats)**

Die Voreinstellungen für Primär-, Sekundär- und Maximalallokierung werden aus dem MRSCAT-Eintrag des SF-Pubsets entnommen, welcher die Funktion des Control-Volume-Sets übernimmt.

Die Voreinstellung für das Dateiformat wird gleich dem Format des Control-Volume-Sets gewählt.



## Nutzung des Pubsets im Verbund

Die Festlegungen hinsichtlich der Nutzung im Rechnerverbundverbund, die im SVL der Volres hinterlegt werden, ergeben sich für den SM-Pubset aus den entsprechenden Festlegungen für den SF-Pubset, welcher die Funktion des Control-Volume-Sets übernimmt.

## MRSCAT-Einträge

Für den SM-Pubset wird ein MRSCAT-Eintrag angelegt. Sein Inhalt entspricht einem durch das Kommando ADD-MASTER-CATALOG-ENTRY initialisierten MRSCAT-Eintrag für einen SM-Pubset. Mit Ausnahme der Kennung des Pubsets und des Control-Volume-Sets sowie des Device-Typs der Volres des Control-Volume-Sets sind alle Werte durch Zuweisung von Voreinstellungen bestimmt worden. Dies führt z.B zu den Einstellungen SHARE=\*NO, FORCE-IMPORT=\*YES, SIZE-TOLERANCE=\*NO. Die MRSCAT-Einträge der in den SM-Pubset eingehenden SF-Pubsets werden gelöscht.

## Lage der Katalogdateien und der Pubset-Konfigurationsdateii

Die Katalogdateien für die einzelnen Volume-Sets ergeben sich durch Umwandlung der Katalogdateien der SF-Pubsets. Die pubset-globalen Katalogdateien und die Pubset-Konfigurationsdatei werden von SMPGEN auf dem Control-Volume-Set angelegt. Ihre Lage und Größe ist vom Systembetreuer nicht beeinflussbar.

## GUARDS-Profile

Die auf den SF-Pubsets eingerichteten GUARDS-Profile bleiben erhalten. Intern werden die GUARDS-Kataloge der in den SM-Pubset eingehenden SF-Pubsets zu dem GUARDS-Katalog des SM-Pubsets zusammengefasst.

Bei der Bildung des GUARDS-Katalogs für den SM-Pubset werden in den GUARDS-Profilen enthaltene Referenzen im Allgemeinen auf Programmnamen angepasst. Die Catids der SF-Pubsets werden durch die Catid des SM-Pubsets ersetzt. Für die Behandlung von Sonderfällen (Namenskollisionen, Behandlung von Wildcard-Konstrukten in GUARDS-Profilen) wird auf das Handbuch „SECOS“ [\[2\]](#) verwiesen.

## Benutzerkatalog

Der Benutzerkatalog des SM-Pubsets entsteht durch Zusammenführung der Benutzerkataloge der einzelnen SF-Pubsets. Die Benutzermenge ergibt sich durch die Vereinigung der Benutzermengen der SF-Pubsets. Für jeden Benutzer wird durch SMPGEN ein neuer Benutzereintrag erstellt. Wenn für einen Benutzer mehrere Einträge auf verschiedenen SF-Pubsets vorhanden, wird ihr Inhalt abgeglichen und zusammengeführt. Die Werte für die Benutzerkontingente eines SM-Pubsets, die auf SF-Pubsets keine Entsprechung haben, werden passend ergänzt.

Bei der Ermittlung des Inhalts der Benutzereinträge kann zwischen folgenden Datenkategorien unterschieden werden:

- Space-Limit-Werte, Space-Used-Werte

Die Werte der Space-Limits, die bereits auf SF-Pubsets in äquivalenter Bedeutung vorhanden sind, werden durch Summation der Space-Limit-Werte gebildet, die für den jeweiligen Benutzer auf den betroffenen SF-Pubsets eingetragen sind (Beispiel: S0-LEVEL-SPACE-LIMIT korrespondiert mit PUBLIC-SPACE-LIMIT).

Die Voreinstellungen der Werte von Space-Limits, die keine Entsprechung auf SF-Pubsets haben (z.B. HIGH-PERF-SPACE-LIMIT), werden so gewählt, dass sie sich in die hierarchische Kontingenzstruktur der SM-Pubsets einpassen.

Die Space-Used-Werte werden aus dem Platz, den die Dateien eines Benutzers belegen, und den Attributen, die den einzelnen Dateien zugeordnet sind, ermittelt.

- aktuelle Anzahl der katalogisierten Dateien/Jobvariablen und Limit-Wert

Die aktuelle Anzahl der für die einzelnen Benutzer katalogisierten Dateien/Jobvariablen sowie die erlaubten Maximalwerte werden durch Summation der entsprechenden Werte auf den betroffenen SF-Pubsets berechnet.

- für Daten-Pubsets relevante Benutzerberechtigungen

Die für Daten-Pubsets relevanten Benutzerberechtigungen PUBLIC-SPACE-EXCESS (Überschreiten von Space-Limits), DMS-TUNING-RESOURCES (Anforderung erhöhter Performance) und PHYSICAL-ALLOCATION (Recht für physikalische Allokierung) werden so festgelegt, dass von allen SF-Pubsets die am wenigsten restriktive Festlegung übernommen wird.

- Privilegien und sonstige Benutzermerkmale

Benutzerprivilegien und sonstige systembezogene Benutzermerkmale (z.B. Privilegien), die auf dem Home-Pubset abgelegt werden, sind für SM-Pubsets von untergeordneter Bedeutung, solange diese nicht als Home-Pubset nutzbar sind. Für die Festlegung dieser Benutzermerkmale - insbesondere des LOGON-Passworts - können die Voreinstellungen von SMPGEN bestimmt werden. Sie können aber auch von einem der SF-Pubsets, die in den SM-Pubset eingehen, übernommen werden (Operand KEEP-USER-ATTRIBUTES). Letzteres ist nur für Benutzer möglich, für welche auf dem SF-Pubset ein Benutzereintrag existiert.

– Gruppenstrukturen

Der Operand KEEP-USER-ATTRIBUTES steuert beim Generieren des SM-Pubsets auch, ob das Einrichten von Benutzergruppen unterbleibt oder ob die Gruppenstrukturen von einem der SF-Pubsets, die in den SM-Pubset eingehen, übernommen werden. Die Gruppeneinträge für SM-Pubsets enthalten Space-Limits für Benutzerkontingente, die in den Gruppeneinträgen für SF-Pubsets keine Entsprechung haben. Diesen Space-Limits werden bei der Übernahme Maximalwerte zugewiesen. Benutzer des SM-Pubsets, die auf dem SF-Pubset, von dem die Gruppeneinträge übernommen werden, nicht eingetragen waren oder dort keiner Gruppe angehörten, werden der Gruppe \*UNIVERSAL zugeordnet.

### Dateien und Dateiattribute

Die Lage der Benutzerdateien auf den Volumes wird durch die Bildung des SM-Pubsets nicht geändert.

Die auf SF-Pubsets katalogisierten Dateien, die auf dem SF-Pubset selbst keinen Platz belegen (auf Hintergrundebenen liegende Dateien, Banddateien, Privatplattendateien, katalogisierte Dateien ohne zugewiesenem Platz) bleiben weiterhin katalogisiert. Intern werden ihre Katalogeinträge in die jeweiligen Spezialkatalogdateien transferiert.

Bei der Festlegung der Dateiattribute ist zu unterscheiden, ob diese auf SF-Pubsets bereits eine Entsprechung haben (zur Bedeutung der Dateiattribute siehe [Abschnitt „Ablageortrelevante Dateiattribute und zugewiesene Voreinstellungen“ auf Seite 210](#)).

a) Dateiattribute mit Entsprechung auf SF-Pubsets

Abgesehen von dem Dateiautribut AVAILABILITY bleiben die Dateiattribute, die auf SF-Pubsets eine Entsprechung haben, unverändert. Das Dateiautribut AVAILABILITY wird durch das Verfügbarkeitsprofil des Volume-Sets beeinflusst, auf dem die Datei liegt: Alle nicht-temporären Dateien auf einem Volume-Set mit AVAILABILITY=\*HIGH bekommen automatisch das Dateiautribut AVAILABILITY=\*HIGH zugewiesen, alle übrigen Dateien behalten ihren bisherigen Wert für AVAILABILITY bei.

Der Systembetreuer sollte dafür sorgen, dass die SF-Pubsets, auf denen Dateien mit Attribut AVAILABILITY=\*HIGH liegen, als Volume-Sets das Verfügbarkeitsprofil AVAILABILITY=\*HIGH zugewiesen bekommen. Andernfalls würden nach der Bildung des SM-Pubsets Dateien mit dem Dateiautribut AVAILABILITY=\*HIGH auf Volume-Sets mit dem Verfügbarkeitsprofil AVAILABILITY=\*STD liegen (was in Widerspruch mit der Verbindlichkeit des Dateiautributs AVAILABILITY steht).

## b) Dateiattribute ohne Entsprechung auf SF-Pubsets

Bei den Dateiattributen, die auf SF-Pubsets keine Entsprechung haben, handelt es sich um das Preformat und die Sperre für S0-Migration.

- Alle Dateien erhalten automatisch das Attribut S0-MIGRATION=\*ALLOWED. Sie sind damit anschließend nicht mehr an einen bestimmten Volume-Set gebunden. Wünscht ein zur physikalischen Allokierung berechtigter Benutzer, dass bestimmte Dateien auf dem Volume-Set bleiben, auf dem sie sich unmittelbar nach der Bildung des SM-Pubsets befinden, muss er ihnen durch das Kommando MODIFY-FILE-ATTRIBUTES explizit das Attribut S0-MIGRATION=\*FORBIDDEN zuweisen.
- Das Preformat ist für von den SF-Pubsets übernommene Dateien nur dann relevant, wenn ihnen auf den SF-Pubsets bereits Platz zugewiesen wurde, sie aber dort noch nicht eröffnet worden sind. Sie werden bei der erstmaligen Dateieröffnung so behandelt, als ob der Datei das Format des Volume-Sets, auf dem sie zu diesem Zeitpunkt liegt, als Preformat zugewiesen wäre.

### 6.3.4 Nachbearbeitung der durch SMPGEN getroffenen Voreinstellungen

Nach der Konvertierung sollte der Systembetreuer überprüfen, ob die von SMPGEN automatisch vorgenommenen Voreinstellungen z.B. bezüglich der Benutzerkontingente, Allokierung oder Nutzungsrestriktionen bereits passend sind oder noch modifiziert werden sollen. Anpassungen können mit den Funktionen für die Pubset-Pflege (z.B. den Kommandos MODIFY-USER-ATTRIBUTES, MODIFY-PUBSET-DEFINITION-FILE, MODIFY-PUBSET-RESTRICTIONS, MODIFY-MASTER-CATALOG-ENTRY) vorgenommen werden. Insbesondere sollte dabei analysiert werden, ob sich hinsichtlich der zugewiesenen Voreinstellungen der Dateiattribute für die Benutzer unerwünschte Inkompatibilitäten ergeben. Durch passende Einstellungen des voreingestellten File-Formats für den SM-Pubset sowie durch die Zuordnung passender voreingestellter Storage-Klassen für die einzelnen Benutzer lassen sich diese weitgehend vermeiden.

#### *Beispiel*

Für einen Benutzer, dem bisher ein SF-Pubset mit dem Format K als voreingestelltes Pubset zugewiesen war, bleibt die Voreinstellung des Dateiformats gleich, wenn dem SM-Pubset als Voreinstellung das File-Format K zugeordnet wird. Erhält der SM-Pubset aber NK4 als Voreinstellung, kann für diesen Benutzer ein kompatibles Verhalten dadurch gewährleistet werden, dass ihm der Systembetreuer als Voreinstellung eine Storage-Klasse mit dem Preformat K zuordnet.

### 6.3.5 Funktionale Reduzierungen als Folge der Pubset-Konvertierung

Für die in den SM-Pubset eingehenden SF-Pubsets ergeben sich folgende funktionale Reduzierungen:

#### **EAM-Dateien**

Auf den SF-Pubsets liegende SYSEAM-Dateien werden gelöscht.

#### **Eignung als Home-Pubset**

Bei der Konvertierung der SF-Pubsets in einen SM-Pubset geht die Nutzungsmöglichkeit als Home-Pubset verloren.

#### **Spoolaufträge**

Spoolaufträge, die sich auf Dateien beziehen, die durch die Bildung des SM-Pubsets ihren Pfadnamen ändern, sollten vor der Konvertierung abgeschlossen werden. Die Spoolaufträge würden sonst ungültig werden und könnten nicht mehr ausgeführt werden. Temporäre Dateien, die zu ungültig gewordenen Spoolaufträgen gehören, muss der Systembetreuer nach der Bildung des SM-Pubsets explizit löschen.

#### **Sicherung der Benutzerdatei**

Die Datei \$TSOS.SYSSRPM.BACKUP geht verloren.

## 6.4 Anpassung der Migrations- und Backup-Archive

Dieser Abschnitt gibt einen Überblick über die HSMS- und DIRCONV-Funktionen zur Anpassung der Backup- und Migrationsarchive. Die HSMS-Anweisungen (BACKUP-FILES, COPY-SAVE-FILE, etc.) und HSMS-Begriffe (wie Archiv, Save-Files, Save-Versionen) werden als bekannt vorausgesetzt (siehe Handbuch „HSMS“ [4]). Gleiches gilt für die Arbeitsweise von DIRCONV. Bei dessen Benutzung ist zu beachten, dass (z.B. bei RENAME-CATID) die vom Aufrufer bereitgestellten Directories unverändert bleiben und zusätzlich ein neues Directory mit den gewünschten Änderungen erzeugt wird. Es wird empfohlen, vor dem DIRMERGE-Aufruf die bisherigen Directories so umzubenennen, dass aus ihrem Pfadnamen die künftige Rolle als Sicherungskopie hervorgeht. Der Pfadname für das neue Directory sollte so gewählt werden, dass der Anpassungsaufwand für bestehende Archivdefinitionen möglichst gering ist, z.B. indem es den Pfadnamen des bisherigen Directory übernimmt.

Im Vergleich zur Verarbeitungsebene sind die Voraussetzungen für die in-place-Konvertierung des Migrationsarchivs und des System-Backup-Archivs oft schwer herzustellen. Gleiches gilt für die privaten Backup-Archive. Vor der Umstellung sollten der Systembetreuer bzw. die Eigentümer der privaten Backup-Archive untersuchen, ob die einzelnen Archive von den erforderlichen Voraussetzungen für die in-place-Konvertierung abweichen, und welcher Aufwand notwendig ist, sie herzustellen. Dieser sollte mit dem Aufwand für alternative Möglichkeiten der Anpassung verglichen werden. Alternative Möglichkeiten bestehen darin, für den SM-Pubset neue Archive (genauer: Archive mit neuen Directories und neuen Datenträgern) einzurichten, und sie durch den Transfer von Dateien bzw. Jobvariablen (Save/Restore, COPY-SAVE-FILE) aus anderen Archiven mit einem geeigneten Inhalt zu versorgen. Im Unterschied zur in-place-Konvertierung sind dabei Zugriffe auf die Datenträger erforderlich, auf welchen migrierte Dateien bzw. Datei-/JV-Sicherungen abgelegt sind.

Für die Durchführung von HSMS-Funktionen sind Zugriffe auf den MAREN-Katalog erforderlich. Liegt dieser auf einem der SF-Pubsets, die in den SM-Pubset eingehen, ist er während der Bildung des SM-Pubsets nicht nutzbar. Spätestens, wenn auf dem SM-Pubset die HSMS-Umgebung eingerichtet wird, sollten wieder Zugriffe möglich sein. Im Falle, dass die Anpassung der Verarbeitungsebene nicht in-place, sondern durch Sichern und Wiedereinspielen von Dateien erfolgt, sind dafür besondere Vorkehrungen notwendig. In diesem Fall wird empfohlen, den MAREN-Katalog vor der Bildung des SM-Pubsets auf einen davon nicht betroffenen Pubset (bzw. Privatplatten) zu verlagern.

## 6.4.1 Voraussetzungen für die in-place-Konvertierung

Für die in-place-Konvertierung sind folgende Voraussetzungen erforderlich:

- Die Migrations- und Backup-Archive von SF-Pubsets sind gleichzeitig durch mehrere SF-Pubsets nutzbar, wobei nicht davon ausgegangen werden kann, dass diese alle in den SM-Pubset eingehen sollen. Die Migrations- und Backup-Archive von SM-Pubsets sind hingegen pubset-spezifisch und dürfen keine Dateien bzw. Dateisicherungen anderer Pubsets enthalten. Die in-place-Konvertierung der SF-Pubset-Archive ist nur möglich, wenn sie von „fremden“ Pubsets entkoppelt sind, d.h. nicht für SF-Pubsets genutzt werden, die nicht in den SM-Pubset eingehen
- Es können nur Backup- bzw. Migrationsarchive in-place konvertiert werden, deren gesamter Dateibestand auf Bändern liegt. Bei den Migrationsarchiven ist dies darin begründet, dass für die in-place-Konvertierung eines S1-Pubsets in einen S1-Volume-Set keine Unterstützung gegeben wird. Bei den Backup-Archiven ist dies die Folge einer grundsätzlichen Einschränkung, die für SM-Pubsets im Unterschied zu SF-Pubsets nur Bänder als Sicherungsmedium erlaubt.
- In den Pfadnamen der in den Archiven abgelegten Dateien und Jobvariablen müssen die Catids der bisherigen SF-Pubsets durch die des SM-Pubsets ersetzt werden. Zuvor ist sicherzustellen, dass die Umbenennung nicht zu Überlängen oder Kollisionen von Dateinamen führt.

Erfüllt ein Archiv diese Voraussetzungen nicht, muss es vor der in-place-Konvertierung bereinigt werden. Dies erfordert im Allgemeinen Zugriffe auf die Datenträger mit den migrierten Dateien bzw. den Sicherungen, was den Vorteil der in-place-Konvertierung erheblich einschränken kann.

## 6.4.2 Anpassung der Migrationsarchive

Zunächst wird das Vorgehen für die in-place-Konvertierung der Migrationsarchive dargestellt. Anschließend wird die Anpassung der Migrationsarchive auf Basis von Sicherungen beschrieben. Die erforderlichen Schritte müssen zum Teil vor, zum Teil nach der Anpassung der Verarbeitungsebene erfolgen. Die Migrationsarchive können auch dann in-place angepasst werden, wenn die Anpassung der Verarbeitungsebene durch Sichern und Wiederherstellen von Dateien erfolgt.

### 6.4.2.1 In-place-Konvertierung von Migrationsarchiven

Es erfolgt zunächst eine Übersicht der erforderlichen Maßnahmen und anschließend eine detaillierte Beschreibung der einzelnen Schritte.

#### a) Maßnahmen vor der Anpassung der Verarbeitungsebene

A1	Bereinigung von Namenskonflikten von auf Hintergrundebenen migrierten Dateien
A2	Ermittlung der betroffenen Migrationsarchive und Migrations-Directories
A3	Entkopplung der Migrationsarchive von SF-Pubsets, die nicht in den SM-Pubset eingehen
A4	Verlagern der in der S1-Ebene befindlichen Dateien
A5	Entfernen nicht mehr aktueller Einträge aus den betroffenen Migrations-Directories
A6	Zusammenfassen der betroffenen Directories zu einem einzigen Directory
A7	optional: Maßnahmen, um das Migrationsarchiv noch für eine bestimmte Zeit für SF-Pubsets nutzen zu können
A8	Anpassung des Directory für den künftigen SM-Pubset durch Catid-Ersetzung
A9	Anpassung des dem Archiv zugeordneten MAREN-Katalogs
A10	Sichern des Migrations-Directory (nur erforderlich, wenn die Anpassung der Verarbeitungsebene nicht in-place erfolgt)
A11	Bereinigung der HSMS-Definitionen für die in den SM-Pubset eingehenden SF-Pubsets

#### b) Maßnahmen nach der Anpassung der Verarbeitungsebene

A12	Einrichten der HSMS-Umgebung auf dem SM-Pubset
A13	Wiedereinspielen des Migrations-Directory (nur erforderlich, wenn die Anpassung der Verarbeitungsebene nicht in-place erfolgt, siehe A10)
A14	Einrichten des Migrationsarchivs mit S2-Ebene und Zuordnung des früher angepassten Migrations-Directory
A15	Anpassung der Migration-Control-Parameter (z.B. Except-Files)
A16	Einrichten der S1-Ebene und Verlagern von Dateien auf die S1-Ebene



## a) Maßnahmen vor der Anpassung der Verarbeitungsebene

A1	<p><b>Bereinigung von Namenskonflikten für auf Hintergrundebenen migrierte Dateien</b></p> <p>Durch SMPGEN-Checks, die für die Anpassung der Verarbeitungsebene erfolgen, werden auch Namenskonflikte der auf Hintergrundebenen befindlichen Dateien erkannt. Sie lassen sich durch Umbenennen oder Löschen von Dateien beheben. Migrierte Dateien müssen zum Umbenennen in die Verarbeitungsebene gebracht werden.</p>				
A2	<p><b>Ermitteln der Migrationsarchive, die den in den SM-Pubset eingehenden SF-Pubsets zugeordnet sind, und ihrer Directories</b></p> <p>Das einem Pubset zugewiesene Migrationsarchiv kann durch die HSMS-Anweisung SHOW-PUBSET-PARAMETERS ermittelt werden. Das Directory des Archivs lässt sich mithilfe der HSMS-Anweisung SHOW-ARCHIVE-ATTRIBUTES bestimmen.</p>				
A3	<p><b>Entkopplung der Migrationsarchive von SF-Pubsets, die nicht in den SM-Pubset eingehen</b></p> <p>Um Kopplungen eines Migrationsarchivs mit fremden SF-Pubsets (d.h. SF-Pubsets, die nicht in den SM-Pubset eingehen) zu erkennen, kann sich der Systembetreuer durch die DIRCONV-Anweisung</p> <table border="1" data-bbox="316 860 1228 901"> <tr> <td data-bbox="316 860 589 901"><b>//SHOW-CATID</b></td> <td data-bbox="595 860 1228 901"><b>DIRECTORY-NAME = ... , ...</b></td> </tr> </table> <p>alle Pubsets anzeigen lassen, für welche das Migrations-Directory des Archivs genutzt wird. Kopplungen zu einem fremden Pubset können auf folgende Art aufgelöst werden:</p> <p>Dem fremden Pubset wird mit der HSMS-Anweisung MODIFY-PUBSET-PARAMETERS ein neues Migrationsarchiv zugewiesen:</p> <table border="1" data-bbox="316 1087 1228 1194"> <tr> <td data-bbox="316 1087 589 1194"><b>//MODIFY-PUBSET-PARAMETERS</b></td> <td data-bbox="595 1087 1228 1194"><b>PUBSET-ID = &lt;Kennung des fremden Pubsets&gt; , STORAGE-LEVEL = S0 (SYSMIGRATE = &lt;Name des neuen Migrationsarchivs&gt; ) , ...</b></td> </tr> </table> <p>Gegebenfalls muss das neue Migrationsarchiv zuvor neu eingerichtet werden. Um die migrierten Dateien des fremden Pubsets in das neue Migrationsarchiv zu übernehmen, sind zwei alternative Vorgehensweisen möglich:</p>	<b>//SHOW-CATID</b>	<b>DIRECTORY-NAME = ... , ...</b>	<b>//MODIFY-PUBSET-PARAMETERS</b>	<b>PUBSET-ID = &lt;Kennung des fremden Pubsets&gt; , STORAGE-LEVEL = S0 (SYSMIGRATE = &lt;Name des neuen Migrationsarchivs&gt; ) , ...</b>
<b>//SHOW-CATID</b>	<b>DIRECTORY-NAME = ... , ...</b>				
<b>//MODIFY-PUBSET-PARAMETERS</b>	<b>PUBSET-ID = &lt;Kennung des fremden Pubsets&gt; , STORAGE-LEVEL = S0 (SYSMIGRATE = &lt;Name des neuen Migrationsarchivs&gt; ) , ...</b>				

<b>A3</b>	<i>Alternative a)</i>		
Forts.	<p>Jedes Save-File, das eine migrierte Datei des fremden Pubsets enthält, wird durch die HSMS-Anweisung</p>		
<table border="1"> <tr> <td data-bbox="312 272 589 350"><b>//COPY-SAVE-FILE</b></td> <td data-bbox="589 272 1236 350"><b>SAVE-FILE-ID = ... , SELECT-FILES =</b> <b>:&lt; Kennung des fremden Pubsets&gt;:\$*.* , ...</b></td> </tr> </table>		<b>//COPY-SAVE-FILE</b>	<b>SAVE-FILE-ID = ... , SELECT-FILES =</b> <b>:&lt; Kennung des fremden Pubsets&gt;:\$*.* , ...</b>
<b>//COPY-SAVE-FILE</b>	<b>SAVE-FILE-ID = ... , SELECT-FILES =</b> <b>:&lt; Kennung des fremden Pubsets&gt;:\$*.* , ...</b>		
<p>selektiv (d.h. unter Mitnahme nur der Dateien des fremden Pubsets) in das Migrationsarchiv des fremden Pubsets übertragen. Anschließend muss für die migrierten Dateien des fremden Pubsets die Lokalisierungsinformation in den Katalogeinträgen aktualisiert werden. Dies kann durch die folgende HSMS-Anweisung erfolgen:</p>			
<table border="1"> <tr> <td data-bbox="312 520 589 627"><b>//REPAIR-CATALOG-BY-EXCHANGE</b></td> <td data-bbox="589 520 1236 627"><b>S0-PUBSET-ID =</b> <b>&lt;Kennung des fremden Pubsets&gt; , ...</b></td> </tr> </table>		<b>//REPAIR-CATALOG-BY-EXCHANGE</b>	<b>S0-PUBSET-ID =</b> <b>&lt;Kennung des fremden Pubsets&gt; , ...</b>
<b>//REPAIR-CATALOG-BY-EXCHANGE</b>	<b>S0-PUBSET-ID =</b> <b>&lt;Kennung des fremden Pubsets&gt; , ...</b>		
<p>Eine Schwierigkeit kann bei dieser Vorgehensweise in der Ermittlung der betroffenen Save-Files bestehen. Sie ist auf folgende Weise möglich:</p>			
<p>Die HSMS-Anweisung</p>			
<table border="1"> <tr> <td data-bbox="312 751 589 829"><b>//SHOW-ARCHIVE</b></td> <td data-bbox="589 751 1236 829"><b>SELECT = FILES(FILE-NAMES =</b> <b>:&lt;Kennung des fremden Pubsets&gt;:\$*.* ) , ...</b></td> </tr> </table>		<b>//SHOW-ARCHIVE</b>	<b>SELECT = FILES(FILE-NAMES =</b> <b>:&lt;Kennung des fremden Pubsets&gt;:\$*.* ) , ...</b>
<b>//SHOW-ARCHIVE</b>	<b>SELECT = FILES(FILE-NAMES =</b> <b>:&lt;Kennung des fremden Pubsets&gt;:\$*.* ) , ...</b>		
<p>zeigt die Save-Versionen an, die Save-Files mit migrierten Dateien des fremden Pubsets enthalten. Die betroffenen Save-Files können daraus mithilfe der HSMS-Anweisung</p>			
<table border="1"> <tr> <td data-bbox="312 933 589 982"><b>//SHOW-ARCHIVE</b></td> <td data-bbox="589 933 1236 982"><b>SELECT = SAVE-VERSIONS(... ) , ...</b></td> </tr> </table>		<b>//SHOW-ARCHIVE</b>	<b>SELECT = SAVE-VERSIONS(... ) , ...</b>
<b>//SHOW-ARCHIVE</b>	<b>SELECT = SAVE-VERSIONS(... ) , ...</b>		
<p>bestimmt werden.</p>			
<p>Alternative a) sollte nur dann in Betracht gezogen werden, wenn die Kopplungen zu anderen SF-Pubsets gering sind, da ansonsten erhebliche Aufwände entstehen.</p>			
<p><i>Alternative b)</i></p>			
<p>Zunächst werden die migrierten Dateien des fremden Pubsets gesichert. Hierfür ist es empfehlenswert, zuvor die migrierten Dateien des fremden Pubsets mit SHOW-FILE-ATTRIBUTES zu ermitteln und dabei ihre Dateinamen in einer Liste zu hinterlegen. Diese kann bei dem BACKUP-FILES-Aufruf mitgegeben werden.</p>			
<table border="1"> <tr> <td data-bbox="312 1346 589 1519"><b>//BACKUP-FILES</b></td> <td data-bbox="589 1346 1236 1519"><b>FILE-NAMES = *FROM-FILE</b> <b>(&lt;Liste der fremden Dateien&gt;),</b> <b>SAVE-OPTIONS = PAR(SAVE-DATA = S2-S1-S0) ,</b> <b>ARCHIVE-NAME = &lt;Name des Backup-Archivs&gt; ,</b> <b>...</b></td> </tr> </table>		<b>//BACKUP-FILES</b>	<b>FILE-NAMES = *FROM-FILE</b> <b>(&lt;Liste der fremden Dateien&gt;),</b> <b>SAVE-OPTIONS = PAR(SAVE-DATA = S2-S1-S0) ,</b> <b>ARCHIVE-NAME = &lt;Name des Backup-Archivs&gt; ,</b> <b>...</b>
<b>//BACKUP-FILES</b>	<b>FILE-NAMES = *FROM-FILE</b> <b>(&lt;Liste der fremden Dateien&gt;),</b> <b>SAVE-OPTIONS = PAR(SAVE-DATA = S2-S1-S0) ,</b> <b>ARCHIVE-NAME = &lt;Name des Backup-Archivs&gt; ,</b> <b>...</b>		

<p><b>A3</b></p> <p>Forts.</p>	<p>Anschließend wird durch die HSMS-Anweisung</p> <table border="1" data-bbox="316 199 1233 340"> <tr> <td data-bbox="316 199 589 340"><b>//REPAIR-CATALOG-BY-RESTORE</b></td> <td data-bbox="593 199 1233 340"><b>S0-PUBSET-ID = &lt;Kennung des fremden Pubsets&gt;, ARCHIVE-NAME = &lt;Backup-Archiv mit den Sicherungen der migrierten Dateien des fremden Pubsets&gt; , ...</b></td> </tr> </table> <p>das Migrationsarchiv des fremden Pubsets mithilfe der zuvor erstellten Sicherungen wieder rekonstruiert und die Lokalisierungsinformation in den Katalogeinträgen der migrierten Dateien aktualisiert.</p> <p>Nachdem die migrierten Dateien der fremden SF-Pubsets in deren neue Migrationsarchive übertragen sind, müssen in dem ursprünglichen Directory die Bezüge zu dem fremden Pubset gelöscht werden. Dies wird durch die DIRCONV-Anweisung</p> <table border="1" data-bbox="316 584 1233 629"> <tr> <td data-bbox="316 584 589 629"><b>//REMOVE-CATID</b></td> <td data-bbox="593 584 1233 629"><b>DIRECTORY-NAME = ... , CATID = ... , ...</b></td> </tr> </table> <p>ermöglicht. Der Inhalt der Datenträger bleibt dadurch unverändert, d.h. die migrierten Dateien für den fremden Pubset belegen weiterhin (in den Save-Files) Platz, auch wenn sie nicht mehr genutzt werden können. Sollen außer dem Directory auch die S2-Datenträger bereinigt werden (was nicht unbedingt notwendig ist), muss das Migrationsarchiv reorganisiert werden (siehe A5, Alternative a).</p>	<b>//REPAIR-CATALOG-BY-RESTORE</b>	<b>S0-PUBSET-ID = &lt;Kennung des fremden Pubsets&gt;, ARCHIVE-NAME = &lt;Backup-Archiv mit den Sicherungen der migrierten Dateien des fremden Pubsets&gt; , ...</b>	<b>//REMOVE-CATID</b>	<b>DIRECTORY-NAME = ... , CATID = ... , ...</b>
<b>//REPAIR-CATALOG-BY-RESTORE</b>	<b>S0-PUBSET-ID = &lt;Kennung des fremden Pubsets&gt;, ARCHIVE-NAME = &lt;Backup-Archiv mit den Sicherungen der migrierten Dateien des fremden Pubsets&gt; , ...</b>				
<b>//REMOVE-CATID</b>	<b>DIRECTORY-NAME = ... , CATID = ... , ...</b>				
<p><b>A4</b></p>	<p><b>Verlagern der in der S1-Ebene liegenden Dateien</b></p> <p>Keiner der in die Bildung des SM-Pubsets involvierten SF-Pubsets darf eine auf S1 migrierte Datei enthalten. Diese Bedingung kann durch SMPGEN-Check überprüft werden (SMPGEN-Operand: S1-MIGRATED-FILES=*NOT-ALLOWED). Ist sie nicht erfüllt, kann sie der Systembetreuer durch eine der folgende Maßnahmen herstellen:</p> <p><i>Alternative a)</i></p> <p>Für jedes Migrationsarchiv, das einem in den SM-Pubset eingehenden SF-Pubset zugeordnet ist, werden alle auf S1 migrierten Dateien durch einen Aufruf von MIGRATE-FILES von der S1-Ebene in die S2-Ebene verlagert:</p> <table border="1" data-bbox="316 1179 1233 1285"> <tr> <td data-bbox="316 1179 589 1285"><b>//MIGRATE-FILES</b></td> <td data-bbox="593 1179 1233 1285"><b>FROM-STORAGE-LEVEL = S1-STORAGE-LEVEL , TO-STORAGE-LEVEL = S2-STORAGE-LEVEL , ARCHIVE-NAME = ... , ...</b></td> </tr> </table> <p>Da aus den Migrations-Directories zuvor die Einträge von fremden SF-Pubsets entfernt wurden, werden durch diese Anweisung ausschließlich die auf S1 migrierten Dateien der in die Bildung des SM-Pubsets involvierten SF-Pubsets erfasst.</p>	<b>//MIGRATE-FILES</b>	<b>FROM-STORAGE-LEVEL = S1-STORAGE-LEVEL , TO-STORAGE-LEVEL = S2-STORAGE-LEVEL , ARCHIVE-NAME = ... , ...</b>		
<b>//MIGRATE-FILES</b>	<b>FROM-STORAGE-LEVEL = S1-STORAGE-LEVEL , TO-STORAGE-LEVEL = S2-STORAGE-LEVEL , ARCHIVE-NAME = ... , ...</b>				

<p><b>A4</b></p> <p>Forts.</p>	<p><i>Alternative b)</i></p> <p>Die auf S1-migrierten Dateien werden durch einen Recall-Aufruf in die Verarbeitungsebene gebracht:</p> <table border="1" data-bbox="316 294 1229 434"> <tr> <td data-bbox="316 294 589 434"><b>//RECALL-MIGRATED-FILES</b></td> <td data-bbox="595 294 1229 434"><b>FILE-NAMES = *FROM-FILE (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES) , FROM-STORAGE-LEVEL = S1-STORAGE-LEVEL , ...</b></td> </tr> </table> <p><i>Alternative c)</i></p> <p>Die auf S1-migrierten Dateien werden gesichert und gelöscht:</p> <table border="1" data-bbox="316 525 1229 632"> <tr> <td data-bbox="316 525 589 632"><b>//BACKUP-FILES</b></td> <td data-bbox="595 525 1229 632"><b>FILE-NAMES = *FROM-FILE (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES) , DELETE-FILES-AND-JV = YES , ...</b></td> </tr> </table> <p>Sollen die bisher auf der S1-Ebene befindlichen Dateien später auf die S1-Ebene des SM-Pubsets gelangen, ist es empfehlenswert, ihre Namen in einer Datei zu vermerken (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES). Jede der Alternativen setzt voraus, dass in der Verarbeitungsebene ausreichend Speicher zur Verfügung steht (oder gestellt wird), um die Dateien der S1-Ebene temporär aufzunehmen. Bei Alternative b) betrifft dies bereits die SF-Pubsets, bei den Alternativen a) und c) erst den späteren SM-Pubset.</p>	<b>//RECALL-MIGRATED-FILES</b>	<b>FILE-NAMES = *FROM-FILE (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES) , FROM-STORAGE-LEVEL = S1-STORAGE-LEVEL , ...</b>	<b>//BACKUP-FILES</b>	<b>FILE-NAMES = *FROM-FILE (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES) , DELETE-FILES-AND-JV = YES , ...</b>
<b>//RECALL-MIGRATED-FILES</b>	<b>FILE-NAMES = *FROM-FILE (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES) , FROM-STORAGE-LEVEL = S1-STORAGE-LEVEL , ...</b>				
<b>//BACKUP-FILES</b>	<b>FILE-NAMES = *FROM-FILE (z.B. Ausgabeliste von SHOW-FILE-ATTRIBUTES) , DELETE-FILES-AND-JV = YES , ...</b>				
<p><b>A5</b></p>	<p><b>Entfernen nicht mehr aktueller Einträge in den Migrations-Directories</b></p> <p>Um Namensprobleme beim Anpassen der Migrationsarchive zu verhindern, müssen von Namensproblemen betroffene Dateien umbenannt oder gelöscht werden (Schritt A1) und außerdem nicht mehr aktuelle Einträge aus den Migrations-Directories entfernt werden. Für das Löschen veralteter Einträge sind folgende Vorgehensweisen möglich:</p>				

<b>A5</b>	<i>Alternative a)</i>
Forts.	Das gesamte Migrationsarchiv wird reorganisiert. Dazu wird jedes Save-File durch die Anweisung
<b>//COPY-SAVE-FILES</b>	<b>SAVE-FILE = ... , MIGRATION-STATE = MIGRATED-FILES , ...</b>
in ein neues Save-File kopiert, wobei nur die Inhalte aktuell migrierter Dateien übernommen werden. Für die Save-Files, welche keine aktuellen Dateien enthalten, kann das Kopieren entfallen. Eine Aufstellung der betroffenen Save-Files kann der Systembetreuer mithilfe folgender HSMS-Anweisung erstellen:	
<b>//SHOW-ARCHIVE</b>	<b>ARCHIVE-NAME = ... , SELECT = SAVE-FILES(SAVE-FILE-ID = *BY- ATTRIBUTES (SAVE-FILE-STORAGE = TAPE)) , ...</b>
Die bisherigen Save-Files werden mithilfe der HSMS-Anweisung	
<b>//MODIFY-ARCHIVE</b>	<b>ARCHIVE-NAME = ... , SAVE-FILES = DELETE(SAVE-FILE-ID = ... , FORCED-DELETE=YES) , ...</b>
gelöscht. Wenn sie alle vor dem Tag erzeugt worden sind, an dem die Reorganisation erfolgt, lassen sie sich durch die Angabe SAVE-FILES=DELETE(SAVE-FILE-ID = *BY-ATTRIBUTES (CREATED-BEFORE = <Datum des Tages vor der Reorganisation>) besonders einfach erfassen. Bei der Reorganisation werden sowohl im Migrations-Directory als auch auf den Bändern veraltete Einträge entfernt.	
<i>Alternative b)</i>	
Für die Reorganisation des Migrationsarchivs müssen Zugriffe auf die S2-Datenträger erfolgen. Dies kann im Rahmen einer in-place-Konvertierung unerwünscht sein. Deswegen wird durch die DIRCONV-Anweisung	
<b>//REMOVE- UNCATALOGED- FILES</b>	<b>DIRECTORY-NAME = ... , ...</b>
auch die Möglichkeit geboten, ausschließlich das Migrations-Directory zu bereinigen. Der Inhalt der S2-Datenträgern ändert sich dabei nicht, auf die darauf befindlichen nicht mehr aktuellen Einträge kann aber anschließend nicht mehr zugegriffen werden. Sie können im Rahmen einer Reorganisation des Archivs entfernt werden.	

<b>A6</b>	<p><b>Zusammenfassen der Directories zu einem einzigen Directory</b></p> <p>Sind den SF-Pubsets, die in den SM-Pubset eingehen, verschiedene Migrations-Directories zugewiesen, müssen sie zu einem einzigen Directory zusammengefasst werden. Dies wird durch folgende DIRCONV-Anweisung ermöglicht:</p> <table border="1" data-bbox="316 310 1229 384"> <tr> <td data-bbox="316 310 589 384"><b>//MERGE-DIRECTORIES</b></td> <td data-bbox="595 310 1229 384"><b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b></td> </tr> </table>	<b>//MERGE-DIRECTORIES</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>
<b>//MERGE-DIRECTORIES</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>		
<b>A7</b>	<p><b>optional: Anpassungen des Archivs für die Nutzung für SF-Pubsets</b></p> <p>Das Zusammenfassen der Migrations-Directories kann als Vorleistung für einen späteren Umstieg auf den SM-Pubset betrachtet werden, der aber nicht unmittelbar im Anschluss erfolgen muss. Um in diesem Fall das durch MERGE gebildete Migrations-Directory zwischenzeitlich nutzen zu können, sind folgende Vorkehrungen notwendig:</p> <ol style="list-style-type: none"> <li>1. Ein Migrationsarchiv muss definiert sein, dem das neu erzeugte Directory zugeordnet ist.</li> <li>2. Dieses muss allen in die Bildung des SM-Pubsets involvierten SF-Pubsets als Migrationsarchiv zugewiesen werden.</li> <li>3. Der Übergang zum neuen Directory muss konsistent in den Datenträgerzuordnungen des MAREN-Katalogs berücksichtigt werden. Hierzu bietet DIRCONV die Anweisung:</li> </ol> <table border="1" data-bbox="316 880 1229 954"> <tr> <td data-bbox="316 880 589 954"><b>//UPDATE-VOLUME-CATALOG</b></td> <td data-bbox="595 880 1229 954"><b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b></td> </tr> </table> <p>Dabei wird in dem MAREN-Katalog für alle Datenträger, die in dem mit dem Operanden NEW-DIRECTORY-NAME spezifizierten Directory eingetragen sind, die Zuordnung zu diesem Directory hergestellt. Mit dem Operanden DIRECTORY-NAMES bietet die Anweisung UPDATE-VOLUME-CATALOG auch die Möglichkeit, dass freie Bänder, die für die bisherigen Directories vorreserviert waren, in den Pool der freien Datenträger des neuen Directory eingehen.</p> <p>Wenn der Umstieg auf den SM-Pubset direkt im Anschluss an das Zusammenfassen der Migrations-Directories erfolgen soll, entfallen die oben aufgezählten Maßnahmen.</p>	<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>
<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>		

<b>A8</b>	<b>Catid-Austausch in dem Migrations-Directory</b>		
Mithilfe der DIRCONV-Anweisung			
<table border="1"> <tr> <td data-bbox="316 244 589 353"><b>//RENAME-CATID</b></td> <td data-bbox="589 244 1233 353"> <b>DIRECTORY-NAME = ... ,  OLD-CATID = *ALL ,  NEW-CATID = PUBSET-ID , ...</b> </td> </tr> </table>		<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... ,  OLD-CATID = *ALL ,  NEW-CATID = PUBSET-ID , ...</b>
<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... ,  OLD-CATID = *ALL ,  NEW-CATID = PUBSET-ID , ...</b>		
<p>wird ein neues Migrations-Directory erzeugt, in dem Referenzen auf Catids der bisherigen SF-Pubsets durch die Referenz auf die Catid des künftigen SM-Pubset ersetzt sind. Im Fall, dass der SM-Pubset aus genau einem SF-Pubset gebildet wird, und dabei dessen Kennung übernimmt, entfällt dieser Schritt.</p> <p>Wenn die Verarbeitungsebene durch SMPGEN angepasst wird, muss das künftige Directory des SM-Pubsets auf den SF-Pubset gelegt werden, der später zum Control-Volume-Set wird (z.B. durch Angabe eines entsprechenden Pfadnamens für das neue Directory in der DIRCONV-Anweisung).</p> <p>Die bisher für die Abwicklung der Migration benutzten Directories können beibehalten werden, um die Möglichkeit für ein späteres Rücksetzen offen zu halten. Sie dürfen aber ab sofort nicht mehr zur Durchführung von Migrations- und Recall-Aufträgen verwendet werden, da sonst das neu erzeugte Directory inkonsistent würde.</p>			
<b>A9</b>	<b>Anpassungen in dem MAREN-Katalog</b>		
<p>Unterscheidet sich der künftige Pfadname des Migrations-Directory auf dem SM-Pubset von dem Pfadnamen, muss der neue Pfadname konsistent in dem MAREN-Katalog berücksichtigt werden (siehe auch A7). Dies erfolgt mithilfe der DIRCONV-Anweisung</p>			
<table border="1"> <tr> <td data-bbox="316 997 589 1075"><b>//UPDATE-VOLUME-CATALOG</b></td> <td data-bbox="589 997 1233 1075"> <b>DIRECTORY-NAMES =... ,  NEW-DIRECTORY-NAME = ... , ...</b> </td> </tr> </table>		<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES =... ,  NEW-DIRECTORY-NAME = ... , ...</b>
<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES =... ,  NEW-DIRECTORY-NAME = ... , ...</b>		
<b>A10</b>	<b>Sichern des Migrations-Directory</b>		
<b>(entfällt, wenn die Verarbeitungsebene in-place angepasst wird)</b>			
<p>Wenn die Verarbeitungsebene nicht in-place sondern durch Sichern und Wiedereinspielen angepasst wird, muss auch das Migrations-Directory gesichert werden. Dies kann z.B auch dadurch erfolgen, dass es auf einen von der Bildung des SM-Pubsets nicht betroffenen Pubset kopiert wird.</p>			
<b>A11</b>	<b>Bereinigungen der HSMS-Definitionen für die bisherigen SF-Pubsets</b>		
<p>Falls der SM-Pubset die Kennung eines SF-Pubsets übernimmt, muss die HSMS-Definition dieses SF-Pubsets entfernt werden. Andernfalls können sich später beim Einrichten der HSMS-Umgebung auf dem SM-Pubset Probleme ergeben. Für die übrigen SF-Pubsets, die in den SM-Pubset eingehen, und für künftig nicht mehr benötigte Migrationsarchive können die Definitionen auch erst nach dem Umstieg bereinigt werden.</p>			

## b) Maßnahmen nach der Anpassung der Verarbeitungsebene

A12	<p><b>Einrichten der HSMS-Umgebung auf dem SM-Pubset</b></p> <p>Damit nach der Anpassung der Verarbeitungsebene HSMS-Funktionen genutzt werden können, muss durch die HSMS-Anweisung CREATE-SM-PUBSET-PARAMETERS auf dem Pubset die HSMS-Umgebung installiert werden. Der Pubset kommt dadurch in den Zustand HSMS-SUPPORTED. Um anschließend HSMS-Funktionen ohne Probleme nutzen zu können, müssen auch Zugriffe auf den MAREN-Katalog möglich sein.</p>
A13	<p><b>Wiedereinspielen des Migrations-Directory (entfällt wenn die Verarbeitungsebene in-place angepasst wurde, siehe A10)</b></p> <p>Wenn die Verarbeitungsebene nicht in-place, sondern durch Sichern und Wiedereinspielen von Dateien angepasst wurde, muss das Migrations-Directory wieder eingespielt werden oder falls es auf einen anderen Pubset verlagert wurde, von dort auf den SM-Pubset kopiert werden (siehe A10). Dabei ist darauf zu achten, dass es auf den Control-Volume-Set gelangt.</p>
A14	<p><b>Einrichten des Migrationsarchivs und Zuweisung des Migrations-Directory</b></p> <p>Um die S2-Ebene wieder funktionsfähig zu machen, ist ein Migrationsarchiv einzurichten, dem das vor der Bildung des SM-Pubsets passend vorbereitete und ggf. in Schritt A13 wiedereingespielte Directory zugewiesen wird. Dies kann entweder bereits bei dem CREATE-SM-PUBSET-PARAMETERS-Aufruf oder später durch einen Aufruf von MODIFY-SM-PUBSET-PARAMETERS erfolgen.</p>
A15	<p><b>Einstellung der MIGRATION-CONTROL-Parameter</b></p> <p>Beim Einrichten des Archivs muss der Systembetreuer die MIGRATION-CONTROL-Parameter passend versorgen. Besondere Beachtung verdienen dabei die Steuerungsmöglichkeiten zur Verhinderung von unerwünschter Verdrängung auf Hintergrundebenen. Es ist zu beachten, dass für die SF-Pubsets, die in den SM-Pubset eingehen, die bisherige Except-File ihre Wirkung verliert. Um den Schutz gegen unerwünschte Dateiverdrängung wiederherzustellen, kann der Systembetreuer auf dem SM-Pubset eine pubset-spezifische Except-File einrichten, in welche die für Migration gesperrten Dateien des SM-Pubsets eingetragen werden, und diese dem SM-Pubset zuordnen. Die Zuordnung kann bei dem CREATE-SM-PUBSET-PARAMETERS-Aufruf oder später durch einen MODIFY-SM-PUBSET-PARAMETERS-Aufruf erfolgen. Eine andere Möglichkeit, unerwünschte Verdrängung zu verhindern, bietet die Benutzung der dateispezifischen Migrationssperre MIGRATE=FORBIDDEN, die durch das Kommando MODIFY-FILE-ATTRIBUTES gesetzt werden kann.</p>



<b>A16</b>	<p><b>Einrichten der S1-Ebene und Verlagern von Dateien auf die S1-Ebene</b></p> <p>Wünscht der Systembetreuer, dass nach der Erzeugung des SM-Pubsets die für die Konvertierung aufgelöste S1-Ebene wiederhergestellt wird, kann er dies mit folgenden Maßnahmen erreichen:</p> <ol style="list-style-type: none"><li>1. Der SM-Pubset muss zunächst mit einem S1-Volume-Set ausgestattet werden. Hierzu wird mit den Möglichkeiten der dynamischen Pubset-Rekonfiguration ein Volume-Set mit Nutzungsart HSMS-CONTROLLED in den Pubset aufgenommen. Dies erfolgt in den Schritten:<ul style="list-style-type: none"><li>– Eintragen des Volume-Sets in die Pubset-Konfigurationsdatei mit Nutzungsart HSMS-CONTROLLED (Kommando: MODIFY-PUBSET-DEFINITION-FILE)</li><li>– Hinzunahme des Volume-Sets zur physikalischen Volume-Set-Konfiguration (Kommando: MODIFY-PUBSET-PROCESSING)</li></ul>Dieser Volume-Set muss HSMS als S1-Volume-Set bekannt gegeben werden (HSMS-Anweisungen: CREATE-SM-PUBSET-PARAMETERS oder MODIFY-SM-PUBSET-PARAMETERS, Operanden S1-DEFINITION, S1-VOLUME-SET-ID).</li><li>2. Die vor der Generierung des SM-Pubsets auf S1 migrierten Dateien können in die S1-Ebene des SM-Pubsets gebracht werden.<p>Zunächst müssen die Dateien in die Verarbeitungsebene gebracht werden, falls in Schritt A4 die Alternative a) oder c) gewählt wurde: Wurden die auf S1 migrierten Dateien in die S2-Ebene verlagert (Alternative a), so müssen sie durch einen RECALL-Auftrag in die Verarbeitungsebene zurückgebracht werden. Wurden sie gesichert und anschließend gelöscht (Alternative c), muss die Sicherung wiedereingespielt werden. Dabei hat der Systembetreuer dafür sorgen, dass die Verarbeitungsebene ausreichend Platz bietet, ggf. indem er den Pubset temporär durch Hinzunahme von Volumes oder eines Volume-Sets vergrößert. Bei der Alternative b) befinden sich die Dateien zu diesem Zeitpunkt bereits in der Verarbeitungsebene.</p>Aus der Verarbeitungsebene können die Dateien durch einen Migrations-Auftrag in die S1-Ebene gebracht werden. Die Durchführung dieser Maßnahmen wird erleichtert, wenn in Schritt A4 die Namen der auf den SF-Pubsets auf S1 migrierten Dateien in einer Datei vermerkt worden sind.</li></ol>
------------	--

Die Hintergrundebenen sollten für den SM-Pubset verfügbar gemacht werden, bevor er zur allgemeinen Nutzung freigegeben wird. Im Anschluss können sowohl neue Migrationsaufträge initiiert werden, als auch die bereits vor der Bildung des SM-Pubsets auf den Hintergrundebenen liegenden Dateien wieder in die Verarbeitungsebene zurückgeholt werden.

Das dargestellte Vorgehen für die in-place-Konvertierung der Migrationsarchive zeigt deutlich, dass die dabei zu bewältigenden Probleme erheblich von dem jeweiligen Anwendungsumfeld der einzelnen Benutzer abhängen. Ideale Voraussetzungen liegen vor, wenn keines der betroffenen Migrationsarchive Bezüge zu Pubsets enthält, die nicht in den SM-Pubset eingehen. Die Entkopplung von fremden Pubsets ist dann von vornherein gegeben. Mit großem Vorbereitungsaufwand ist zu rechnen, wenn die anzupassenden Migrationsarchive auch durch fremde Pubsets benutzt werden. Hier dürfte oft der Weg vorzuziehen sein, die Migrationsarchive mithilfe von Dateisicherungen anzupassen.

#### 6.4.2.2 Anpassung der Migrationsarchive mithilfe von Dateisicherungen

Es erfolgt zunächst eine Übersicht der erforderlichen Maßnahmen und anschließend eine detaillierte Beschreibung der einzelnen Schritte

##### a) Maßnahmen vor der Anpassung der Verarbeitungsebene

B1	Durchführung der SMPGEN-Checks mit der Option „Dateien in der S1-Ebene erlaubt“
B2	Sicherung der auf den Hintergrundebenen liegenden Dateien
B3	Bereinigung der bestehenden Migrationsarchive für die SF-Pubsets
B4	Anpassung des Backup-Archivs, in dem die Sicherungen abgelegt sind

##### b) Maßnahmen nach der Anpassung der Verarbeitungsebene

B5	Einrichten der HSMS-Umgebung auf dem SM-Pubset
B6	Verfügbarmachen des Backup-Archivs mit den Sicherungen
B7	Einrichten des Migrationsarchivs mit S2-Ebene
B8	Festlegen der MIGRATION-CONTROL-Parameter für den SM-Pubset (Except-File)
B9	Einrichten der S1-Ebene
B10	Wiedereinspielen von Sicherungen auf die S1- und S2-Ebene

## a) Maßnahmen vor der Anpassung der Verarbeitungsebene

<b>B1</b>	<p><b>Durchführung der SMPGEN-Checks</b></p> <p>Durch SMPGEN-Checks, die für die Anpassung der Verarbeitungsebene erfolgen, werden auch Namenskonflikte der auf Hintergrundebenen befindlichen Dateien erkannt. Sie lassen sich durch Umbenennen oder Löschen von Dateien beheben. Die migrierte Dateien müssen zum Umbenennen in die Verarbeitungsebene gebracht werden.</p> <p>Im Gegensatz zur in-place-Konvertierung der Migrationsarchive sind für die auf S1 migrierten Dateien keine besonderen Vorkehrungen erforderlich (wie Verlagern auf die S2-Ebene, siehe A4). Damit sie bei den SMPGEN-Checks nicht als Inkonsistenzen bewertet werden, sind diese mit der Option S1-MIGRATED-FILES=*ALLOWED durchzuführen.</p>
<b>B2</b>	<p><b>Sicherung der auf den Hintergrundebenen liegenden Dateien</b></p> <p>Die auf den Hintergrundebenen liegenden Dateien werden gesichert. Die Sicherung kann explizit oder im Rahmen einer Gesamtsicherung aller in die Bildung des SM-Pubsets involvierten SF-Pubsets erfolgen. Sie sollte nach der Durchführung der SMPGEN-Checks und vor der Anpassung der Verarbeitungsebene erfolgen. Um zu verhindern, dass sich durch unkoordinierte Benutzeraktivitäten Inkonsistenzen zu dem Sicherungsstand ergeben (z.B. durch Bearbeitung der Datei während der Sicherung), empfiehlt es sich, dass die betroffenen SF-Pubsets für die Durchführung der Sicherung so importiert werden, dass nur dem Systembetreuer der Zugang ermöglicht wird (Kommando ADD-MASTER-CATALOG-ENTRY, Operand ACCESS-CONTROLLED).</p>

<b>B3</b>	<p><b>Bereinigung der HSMS-Definitionen und der Migrationsarchive</b></p> <p>Falls der SM-Pubset die Kennung eines SF-Pubsets übernimmt, muss die HSMS-Definition dieses SF-Pubsets entfernt werden. Andernfalls können sich später beim Einrichten der HSMS-Umgebung auf dem SM-Pubset Probleme ergeben. Die übrigen in Schritt B3 aufgezählten Bereinigungsmaßnahmen können auch erst später erfolgen, z.B. nachdem der Umstieg auf den SM-Pubset erfolgreich durchgeführt wurde. In diesem Fall wird das Rücksetzen auf den Ausgangszustand erleichtert, wenn während des Umstiegs Probleme auftreten.</p> <p>Die HSMS-Definitionen für die SF-Pubsets, die in den SM-Pubset eingehen, können entfernt werden. Auch die ihnen zugeordneten Migrationsarchive und Directories können gelöscht werden, falls sie nicht noch für andere SF-Pubsets benötigt werden. Wenn sie für andere SF-Pubsets beibehalten werden müssen, ist es sinnvoll, sie zu bereinigen:</p> <ol style="list-style-type: none"> <li>1. Durch die DIRCONV-Anweisung REMOVE-CATID werden in den betroffenen Directories die Bezüge zu den SF-Pubsets, die in den SM-Pubset eingehen, gelöscht.</li> <li>2. Auch die Reorganisation der Archive ist in Betracht zu ziehen, um die migrierten Dateien, für die in den SM-Pubset eingehenden SF-Pubsets aus den Save-Files und damit von den Datenträgern zu entfernen. Sie kann erfolgen, indem jedes Save-File durch <table border="1" data-bbox="307 839 1238 915"> <tr> <td><b>//COPY-SAVE-FILES</b></td> <td><b>SAVE-FILE = ... , MIGRATION-STATE = MIGRATED-FILES , ...</b></td> </tr> </table> in ein neues Save-File kopiert wird, wobei nur die Inhalte der zurzeit migrierten Dateien übernommen werden. Eine Aufstellung der betroffenen Save-Files kann sich der Systembetreuer mithilfe folgender HSMS-Anweisung erzeugen: <table border="1" data-bbox="307 1020 1238 1129"> <tr> <td><b>//SHOW-ARCHIVE</b></td> <td><b>ARCHIVE-NAME = ... , SELECT = SAVE-FILES(SAVE-FILE-ID = *BY-ATTR(SAVE-FILE-STORAGE = TAPE)) , ...</b></td> </tr> </table> Die bisherigen Save-Files werden durch folgende HSMS-Anweisung entfernt: <table border="1" data-bbox="307 1169 1238 1278"> <tr> <td><b>//MODIFY-ARCHIVE</b></td> <td><b>ARCHIVE-NAME = ... , SAVE-FILES = DELETE (SAVE-FILE-ID = *BY-ATTR (CREATED-BEFORE..)) , ...</b></td> </tr> </table> </li> </ol>	<b>//COPY-SAVE-FILES</b>	<b>SAVE-FILE = ... , MIGRATION-STATE = MIGRATED-FILES , ...</b>	<b>//SHOW-ARCHIVE</b>	<b>ARCHIVE-NAME = ... , SELECT = SAVE-FILES(SAVE-FILE-ID = *BY-ATTR(SAVE-FILE-STORAGE = TAPE)) , ...</b>	<b>//MODIFY-ARCHIVE</b>	<b>ARCHIVE-NAME = ... , SAVE-FILES = DELETE (SAVE-FILE-ID = *BY-ATTR (CREATED-BEFORE..)) , ...</b>
<b>//COPY-SAVE-FILES</b>	<b>SAVE-FILE = ... , MIGRATION-STATE = MIGRATED-FILES , ...</b>						
<b>//SHOW-ARCHIVE</b>	<b>ARCHIVE-NAME = ... , SELECT = SAVE-FILES(SAVE-FILE-ID = *BY-ATTR(SAVE-FILE-STORAGE = TAPE)) , ...</b>						
<b>//MODIFY-ARCHIVE</b>	<b>ARCHIVE-NAME = ... , SAVE-FILES = DELETE (SAVE-FILE-ID = *BY-ATTR (CREATED-BEFORE..)) , ...</b>						
<b>B4</b>	<p><b>Anpassungsmaßnahmen für das Backup-Archiv mit den Sicherungen der migrierten Dateien</b></p> <p>Damit das Backup-Archiv mit den Sicherungen der migrierten Dateien nach der Bildung des SM-Pubsets zum Wiedereinspielen der Sicherungen genutzt werden kann, müssen vor der Bildung des SM-Pubsets bestimmte Vorkehrungen erfolgen. Siehe hierzu <a href="#">Abschnitt „Anpassung der Backup-Archive“ auf Seite 103</a> .</p>						

**b) Maßnahmen nach der Anpassung der Verarbeitungsebene**

<b>B5</b>	<p><b>Einrichten der HSMS-Umgebung auf dem SM-Pubset</b></p> <p>Damit nach der Anpassung der Verarbeitungsebene HSMS-Funktionen genutzt werden können, muss durch die HSMS-Anweisung CREATE-SM-PUBSET-PARAMETERS auf dem Pubset die HSMS-Umgebung installiert werden. Der Pubset kommt dadurch in den Zustand HSMS-SUPPORTED. Um anschließend HSMS-Funktionen ohne Probleme nutzen zu können, müssen auch Zugriffe auf den MAREN-Katalog möglich sein.</p>
<b>B6</b>	<p><b>Verfügbarmachen des Backup-Archivs mit den Sicherungen auf dem SM-Pubset</b></p> <p>Das Backup-Archiv, in dem die Sicherungen der migrierten Dateien abgelegt sind, wird als Backup-Archiv des SM-Pubsets verfügbar gemacht. Die dazu erforderlichen Maßnahmen werden im <a href="#">Abschnitt „Anpassung der Backup-Archive“ auf Seite 103</a> behandelt.</p>
<b>B7</b>	<p><b>Einrichten des Migrationsarchivs</b></p> <p>Das Migrationsarchiv wird eingerichtet (entweder bereits durch CREATE-SM-PUBSET-PARAMETERS oder durch MODIFY-SM-PUBSET-PARAMETERS). Im Unterschied zu der in-place-Konvertierung wird dem Migrationsarchiv ein neues Directory zugeordnet, das zunächst leer ist.</p>
<b>B8</b>	<p><b>Festlegen der MIGRATION-CONTROL-Parameter</b></p> <p>Beim Einrichten des Migrationsarchivs muss der Systembetreuer die MIGRATION-CONTROL-Parameter passend versorgen. Besondere Beachtung verdienen dabei die Steuerungsmöglichkeiten zur Verhinderung von unerwünschter Verdrängung auf Hintergrundebenen. Dabei ist zu beachten, dass für die SF-Pubsets, die in den SM-Pubset eingehen, die bisherige Except-File ihre Wirkung verliert. Um den Schutz gegen unerwünschte Verdrängung wiederherzustellen, kann der Systembetreuer auf dem SM-Pubset eine pubset-spezifische Except-File einrichten, in welche die für Verdrängung gesperrten Dateien des SM-Pubsets eingetragen werden, und diese dem SM-Pubset zuordnen. Die Zuordnung kann bei dem CREATE-SM-PUBSET-PARAMETERS-Aufruf oder später durch einen MODIFY-SM-PUBSET-PARAMETERS-Aufruf erfolgen. Eine andere Möglichkeit, unerwünschte Verdrängung zu verhindern, bietet die Benutzung der dateispezifischen Migrationssperre MIGRATE=*FORBIDDEN, die durch das Kommando MODIFY-FILE-ATTRIBUTES gesetzt werden kann.</p>

<b>B9</b>	<p><b>Einrichten der S1-Ebene</b></p> <p>Sollen die früher in der S1-Ebene der SF-Pubsets liegenden Dateien auf die S1-Ebene des SM-Pubsets gebracht werden, muss der SM-Pubset zunächst mit einem S1-Volume-Set ausgestattet werden. Hierzu wird mit den Möglichkeiten der dynamischen Pubset-Rekonfiguration ein Volume-Set mit Nutzungsart HSMS-CONTROLLED in den Pubset aufgenommen. Dies erfolgt in den Schritten:</p> <ul style="list-style-type: none"> <li>– Eintragen des Volume-Sets in die Pubset-Rekonfigurationsdatei mit Angabe der Nutzungsart HSMS-CONTROLLED (Kommando: MODIFY-PUBSET-DEFINITION-FILE).</li> <li>– Hinzunahme des Volume-Sets zur physikalischen Volume-Set-Konfiguration (Kommando: MODIFY-PUBSET-PROCESSING).</li> </ul> <p>Dieser Volume-Set muss HSMS als S1-Volume-Set bekannt gegeben werden (HSMS-Anweisungen CREATE-SM-PUBSET-PARAMETERS, MODIFY-SM-PUBSET-PARAMETERS, Operanden S1-DEFINITION, S1-VOLUME-SET-ID).</p>		
<b>B10</b>	<p><b>Wiedereinspielen der Sicherungen auf die S1- und S2-Ebene</b></p> <p>Durch die HSMS-Anweisung</p> <table border="1" data-bbox="308 773 1229 910"> <tr> <td data-bbox="308 773 618 910"><b>//REPAIR-CATALOG-BY-RESTORE</b></td> <td data-bbox="624 773 1229 910"><b>S0-PUBSET-ID = &lt;Kennung des SM-Pubsets&gt; , TO-STORAGE-LEVEL = ... , ARCHIVE-NAME = &lt;Backup-Archiv mit den Sicherungen der migrierten Dateien&gt; , ...</b></td> </tr> </table> <p>wird das Migrationsarchiv mithilfe der früher erstellten Sicherungen wieder rekonstruiert und die Lokalisierungsinformation in den Katalogeinträgen der migrierten Dateien aktualisiert. Mit der Option TO-STORAGE=*SAME kann die ursprüngliche Verteilung auf die S1- und S2-Ebene wiederhergestellt werden. Sollen auch die früher in der S1-Ebene liegenden Dateien in die S2-Ebene übernommen werden, z.B. wenn in dem SM-Pubset keine S1-Ebene vorgesehen wird oder diese noch nicht eingerichtet ist, muss die Option TO-STORAGE=S2-STORAGE-LEVEL gewählt werden.</p>	<b>//REPAIR-CATALOG-BY-RESTORE</b>	<b>S0-PUBSET-ID = &lt;Kennung des SM-Pubsets&gt; , TO-STORAGE-LEVEL = ... , ARCHIVE-NAME = &lt;Backup-Archiv mit den Sicherungen der migrierten Dateien&gt; , ...</b>
<b>//REPAIR-CATALOG-BY-RESTORE</b>	<b>S0-PUBSET-ID = &lt;Kennung des SM-Pubsets&gt; , TO-STORAGE-LEVEL = ... , ARCHIVE-NAME = &lt;Backup-Archiv mit den Sicherungen der migrierten Dateien&gt; , ...</b>		

Anschließend sind die Hintergrund-Ebenen des SM-Pubsets funktionsfähig. Auf die bereits vor der Bildung des SM-Pubsets verdrängten Dateien kann wieder zugegriffen werden.

### 6.4.3 Anpassung der Backup-Archive

Die Anpassung der Backup-Archive unterscheidet sich in einigen Punkten wesentlich von der Anpassung der Migrationsarchive:

- Die Ermittlung der betroffenen Archive erfordert größeren Aufwand. Die System-Backup-Archive können analog den Migrationsarchiven durch die HSMS-Anweisung SHOW-PUBSET-PARAMETERS bestimmt werden. Zusätzlich sind aber auch alle privaten Backup-Archive zu berücksichtigen, in welchen für einen der involvierten SF-Pubsets Dateisicherungen liegen oder in welche später Sicherungen aus dem SM-Pubset gebracht werden sollen.
- Auch Sicherungen, die in SF-Pubset-Archiven abgelegt sind, können auf SM-Pubsets wiedereingespielt werden. Der Zwang, die bestehenden SF-Pubset-Archive in SM-Pubset-Archive umzuwandeln, ist dadurch erheblich gemindert. Der in ihnen abgelegte Sicherungsbestand ist weiterhin zugreifbar, auch wenn sie nicht angepasst werden. Allerdings können sie dann nicht mehr für neue Sicherungen aus dem SM-Pubset verwendet werden. Die Entscheidung, ob ein Backup-Archiv konvertiert werden soll, kann von dem jeweiligen Eigentümer des Archivs, unter Berücksichtigung von speziellen Randbedingungen, getroffen werden.
- Im Vergleich zur Anpassung des Migrationsarchivs ist für die Anpassung der Backup-Archive ein wesentlich größerer Spielraum zwischen verschiedenen Vorgehensweisen gegeben. Während beispielsweise der Inhalt des Migrationsarchivs nach der Konvertierung durch die zu diesem Zeitpunkt migrierten Dateien fest vorgegeben ist, kann der Sicherungsbestand, der in die Backup-Archive des SM-Pubsets übernommen wird, anwenderspezifisch bestimmt werden. Die Übernahme ist auch nicht an einen bestimmten Zeitpunkt gekoppelt, sie kann z.B. bedarfsorientiert in mehreren Schritten erfolgen.
- Bei der in-place-Konvertierung der bestehenden Backup-Archive ist der Umfang von Kollisionen und Überlängen von Dateinamen schwieriger zu überschauen, da alle in den betroffenen Archiven früher einmal gesicherten Dateien relevant sind. Dadurch hängen die für die in-place-Konvertierung erforderlichen Vorbereitungen wesentlich von der Vorgeschichte der Archive ab.

Im Folgenden werden zwei exemplarische Vorgehensweisen detaillierter dargestellt:

1. Der erste Fall behandelt die in-place-Konvertierung bereits bestehender Backup-Archive für SF-Pubsets, dargestellt am Fall der System-Backup-Archive. Dabei wird aus den bestehenden System-Backup-Archiven aller in den SM-Pubset aufgenommenen SF-Pubsets das System-Backup-Archiv für den SM-Pubset gebildet. Ziel ist es, dass für die betroffenen SF-Pubsets der gesamte in den System-Backup-Archiven enthaltene Sicherungsbestand in das System-Backup-Archiv des SM-Pubsets eingeht und dabei der Zugriff auf die Sicherungen, die Relation zwischen Sicherung, Save-File, Save-Version nicht bzw. nur minimal geändert wird. Hinsichtlich der Systemsicherungen bleibt den Benutzern der Übergang auf den SM-Pubset weitgehend verborgen.
2. Im zweiten Fall wird der Dateibestand der in den SM-Pubset eingehenden SF-Pubsets unmittelbar vor der Bildung des SM-Pubsets in ein neu eingerichtetes Backup-Archiv gesichert, aus dem durch in-place-Konvertierung das System-Backup-Archiv des SM-Pubsets gebildet wird. Die bisherigen System-Backup-Archive der SF-Pubsets werden mit ihrem Sicherungsbestand als private Backup-Archive beibehalten. Für den Benutzer bedeutet dies unterschiedliche Zugriffe auf Systemsicherungen, je nachdem zu welchem Zeitpunkt (vor oder nach Bildung des SM-Pubsets) sie erstellt worden sind.

Das im ersten Fall dargestellte Vorgehen ist besonders für einfach überschaubare Anwendungsszenarien geeignet. Dies sind Anwendungsszenarien, in denen Namenskollisionen zwischen Dateien und damit auch zwischen Dateisicherungen nicht auftreten können und bei denen die involvierten Archive nicht durch Pubsets genutzt werden, die nicht in den SM-Pubset eingehen.

Bei komplexeren Situationen dürfte der im zweiten Beispiel gezeigte Weg vorzuziehen sein. Wie bei der Anpassung der Migrationsarchive ist durch den Systembetreuer und die Eigentümer der privaten Backup-Archive zu analysieren, welcher Weg im Hinblick auf die speziellen Randbedingungen am besten geeignet ist.



### 6.4.3.1 Anpassung bestehender System-Backup-Archive

Es erfolgt zunächst eine Übersicht der erforderlichen Maßnahmen und anschließend eine detaillierte Beschreibung der einzelnen Schritte.

#### a) Maßnahmen vor der Anpassung der Verarbeitungsebene

C1	Ermittlung der betroffenen System-Backup-Archive und ihrer Directories
C2	Entkopplung der Backup-Archive von SF-Pubsets, die nicht in den SM-Pubset eingehen
C3	Verlagern von Save-Files, die auf Privatplatten oder Pubsets liegen
C4	Überprüfen von Kollisionen und Überlängen von Namen gesicherter Dateien
C5	Bereinigung von Kollisionen und Überlängen von Namen gesicherter Dateien
C6	Zusammenfassen der Directories
C7	Austausch der Catids in dem Directory
C8	Anpassungen in dem MAREN-Katalog
C9	Sichern des Backup-Directory und ggf. des MAREN-Katalogs (nur erforderlich, wenn die Anpassung der Verarbeitungsebene nicht in-place erfolgt)
C10	Bereinigung der Archivdefinitionen für die SF-Pubsets

#### b) Maßnahmen nach der Bildung des SM-Pubsets

C11	Einrichten der HSMS-Umgebung auf dem SM-Pubset
C12	Wiedereinspielen des Backup-Directory und ggf. des MAREN-Katalogs (nur erforderlich, wenn die Anpassung der Verarbeitungsebene nicht in-place erfolgt)
C13	Einrichten des System-Backup-Archivs mit Verwendung des für den SM-Pubset angepassten Directories

Die in place-Konvertierung der für die SF-Pubsets bereits bestehenden System-Backup-Archive ist nur dann zu empfehlen, wenn für sie Kopplungen zu fremden Pubsets nicht bestehen oder gering sind, und wenn keine oder nur wenige Namenskollisionen und Namensüberlängen bereinigt werden müssen. Andernfalls ist ihre Durchführung (Schritte C2,C5) sehr schwierig und zeitaufwändig. In vielen Anwendungsfällen dürften dafür günstige Voraussetzungen bestehen.

## a) Maßnahmen vor der Anpassung der Verarbeitungsebene

C1	<p><b>Ermittlung der betroffenen System-Backup-Archive und ihrer Directories</b></p> <p>Das einem Pubset zugewiesene System-Backup-Archiv wird durch die HSMS-Anweisung SHOW-PUBSET-PARAMETERS angezeigt, das Directory des Archivs lässt sich mithilfe der HSMS-Anweisung SHOW-ARCHIVE-ATTRIBUTES bestimmen.</p>								
C2	<p><b>Entkopplung der Backup-Archive von SF-Pubsets, die nicht in den SM-Pubset eingehen</b></p> <p>Die betroffenen Backup-Archive dürfen keine Dateisicherungen von Pubsets enthalten, die nicht in den SM-Pubset eingehen. Um Kopplungen mit fremden SF-Pubsets zu erkennen, kann sich der Systembetreuer durch die DIRCONV-Anweisung</p> <table border="1" data-bbox="316 604 1227 645"> <tr> <td><b>//SHOW-CATID</b></td> <td><b>DIRECTORY-NAME = ... , ...</b></td> </tr> </table> <p>alle Pubsets anzeigen lassen, für welche ein bestimmtes Backup-Directory genutzt wird. Bestehende Kopplungen zu einem fremden Pubset können wie folgt aufgelöst werden:</p> <ol style="list-style-type: none"> <li>1. Dem fremden Pubset wird ein anderes System-Backup-Archiv zugewiesen. Gegebenenfalls muss es dazu zuvor neu eingerichtet werden.</li> <li>2. In dieses werden die gesicherten Dateien des fremden Pubsets übernommen. Hierfür ist folgende Vorgehensweise möglich:</li> </ol> <p>Die Save-Files, die eine Dateisicherung eines fremden Pubsets enthalten, werden ermittelt. Dazu werden zunächst mithilfe der HSMS-Anweisung</p> <table border="1" data-bbox="316 997 1227 1070"> <tr> <td><b>//SHOW-ARCHIVE</b></td> <td><b>SELECT = FILES(FILE-NAMES = : &lt;Kennung des fremden Pubsets&gt;:\$*.* ) , ...</b></td> </tr> </table> <p>alle Save-Versionen bestimmt, welche Save-Files mit Dateisicherungen des fremden Pubsets enthalten. Mithilfe der HSMS-Anweisung</p> <table border="1" data-bbox="316 1149 1227 1191"> <tr> <td><b>//SHOW-ARCHIVE</b></td> <td><b>SELECT = SAVE-VERSIONS(...) , ...</b></td> </tr> </table> <p>werden daraus die betroffenen Save-Files bestimmt. Unter Berücksichtigung der aufsteigenden Reihenfolge der Save-Versionen wird jedes Save-File, das eine Dateisicherung des fremden Pubsets enthält, selektiv (d.h. nur unter Mitnahme der Dateien des fremden Pubsets) in dessen neues System-Backup-Archiv übertragen:</p> <table border="1" data-bbox="316 1364 1227 1432"> <tr> <td><b>//COPY-SAVE-FILE</b></td> <td><b>SAVE-FILE-ID = ... ,SELECT-FILES = :&lt;Kennung des fremden Pubsets&gt;:\$*.* , ...</b></td> </tr> </table>	<b>//SHOW-CATID</b>	<b>DIRECTORY-NAME = ... , ...</b>	<b>//SHOW-ARCHIVE</b>	<b>SELECT = FILES(FILE-NAMES = : &lt;Kennung des fremden Pubsets&gt;:\$*.* ) , ...</b>	<b>//SHOW-ARCHIVE</b>	<b>SELECT = SAVE-VERSIONS(...) , ...</b>	<b>//COPY-SAVE-FILE</b>	<b>SAVE-FILE-ID = ... ,SELECT-FILES = :&lt;Kennung des fremden Pubsets&gt;:\$*.* , ...</b>
<b>//SHOW-CATID</b>	<b>DIRECTORY-NAME = ... , ...</b>								
<b>//SHOW-ARCHIVE</b>	<b>SELECT = FILES(FILE-NAMES = : &lt;Kennung des fremden Pubsets&gt;:\$*.* ) , ...</b>								
<b>//SHOW-ARCHIVE</b>	<b>SELECT = SAVE-VERSIONS(...) , ...</b>								
<b>//COPY-SAVE-FILE</b>	<b>SAVE-FILE-ID = ... ,SELECT-FILES = :&lt;Kennung des fremden Pubsets&gt;:\$*.* , ...</b>								

<b>C2</b> Forts.	<p>3. Nachdem die Dateisicherungen der fremden SF-Pubsets in deren neue Backup-Archive übertragen sind, müssen mithilfe der folgenden DIRCONV-Anweisung in dem ursprünglichen Directory die Bezüge zu dem fremden Pubset gelöscht werden:</p> <table border="1" data-bbox="316 294 1232 338"> <tr> <td><b>//REMOVE-CATID</b></td> <td><b>DIRECTORY-NAME = ... ,CATID = ... , ...</b></td> </tr> </table> <p>Der Inhalt der Datenträger bleibt dadurch unverändert, d.h. die Sicherungen für den fremden Pubset belegen weiterhin in den Save-Files Platz, können aber nicht mehr genutzt werden können. Sollen außer dem Directory auch die Datenträger bereinigt werden (was nicht unbedingt notwendig ist), muss das Archiv reorganisiert werden.</p>	<b>//REMOVE-CATID</b>	<b>DIRECTORY-NAME = ... ,CATID = ... , ...</b>				
<b>//REMOVE-CATID</b>	<b>DIRECTORY-NAME = ... ,CATID = ... , ...</b>						
<b>C3</b>	<p><b>Verlagern von Save-Files, die auf Privatplatten oder Pubsets liegen</b></p> <p>Mithilfe der HSMS-Anweisung</p> <table border="1" data-bbox="316 602 1232 707"> <tr> <td><b>//SHOW-ARCHIVE</b></td> <td><b>SELECT = SAVE-FILES(SAVE-FILE-ID =*BY-ATTR (SAVE-FILE-STORAGE = PUBLIC-DISK / PRIVATE-DISK)) , ...</b></td> </tr> </table> <p>wird ermittelt, ob die betroffenen Backup-Archive Save-Files enthalten, die auf einem Pubset oder einer Privatplatte liegen. Enthalten diese Save-Files Sicherungen, die in das System-Backup-Archiv des SM-Pubsets eingehen sollen, müssen sie auf Bänder des jeweiligen Backup-Archivs gebracht werden. Dies wird durch die HSMS-Anweisung</p> <table border="1" data-bbox="316 880 1232 984"> <tr> <td><b>//COPY-SAVE-FILE</b></td> <td><b>SAVE-FILE-ID = ... , ARCHIVE-NAME = ... , TO-ARCHIVE-NAME = ..., TO-STORAGE = S2-STORAGE-LEVEL , ...</b></td> </tr> </table> <p>ermöglicht. Da ein COPY-SAVE-FILE-Aufruf innerhalb eines bestimmten Backup-Archivs (ARCHIVE-NAME = *SAME) nur die Angabe SAVE-FILE-ID = *LATEST erlaubt, ist im Allgemeinen ein Umweg über ein weiteres Backup-Archiv notwendig. Anschließend werden für die betroffenen Backup-Archive alle auf Pubsets oder Privatplatten liegenden Save-Files gelöscht:</p> <table border="1" data-bbox="316 1158 1232 1197"> <tr> <td><b>//MODIFY-ARCHIVE</b></td> <td><b>SAVE-FILES = DELETE , ...</b></td> </tr> </table>	<b>//SHOW-ARCHIVE</b>	<b>SELECT = SAVE-FILES(SAVE-FILE-ID =*BY-ATTR (SAVE-FILE-STORAGE = PUBLIC-DISK / PRIVATE-DISK)) , ...</b>	<b>//COPY-SAVE-FILE</b>	<b>SAVE-FILE-ID = ... , ARCHIVE-NAME = ... , TO-ARCHIVE-NAME = ..., TO-STORAGE = S2-STORAGE-LEVEL , ...</b>	<b>//MODIFY-ARCHIVE</b>	<b>SAVE-FILES = DELETE , ...</b>
<b>//SHOW-ARCHIVE</b>	<b>SELECT = SAVE-FILES(SAVE-FILE-ID =*BY-ATTR (SAVE-FILE-STORAGE = PUBLIC-DISK / PRIVATE-DISK)) , ...</b>						
<b>//COPY-SAVE-FILE</b>	<b>SAVE-FILE-ID = ... , ARCHIVE-NAME = ... , TO-ARCHIVE-NAME = ..., TO-STORAGE = S2-STORAGE-LEVEL , ...</b>						
<b>//MODIFY-ARCHIVE</b>	<b>SAVE-FILES = DELETE , ...</b>						

<b>C4</b>	<p><b>Überprüfen von Überlängen und Kollisionen von Dateinamen</b></p> <p>Zum Konvertieren der Archive dienen die DIRCONV-Funktionen MERGE-DIRECTORIES und RENAME-CATID. Sie können nur dann erfolgreich ausgeführt werden, wenn sich dabei keine Kollisionen und Überlängen von Dateinamen bzw. von Namen von Jobvariablen ergeben. Zum Herstellen dieser Voraussetzung wird dem Anwender dringend empfohlen, sich vor der eigentlichen Konvertierung mithilfe folgender DIRCONV-Check-Funktionen (SILENT-RUN) alle Verstöße anzeigen zu lassen:</p> <table border="1" data-bbox="316 439 1227 621"> <tr> <td data-bbox="316 439 589 513">//MERGE-DIRECTORIES</td> <td data-bbox="594 439 1227 513">DIRECTORY-NAMES = ... , SILENT-RUN = YES , ...</td> </tr> <tr> <td data-bbox="316 520 589 621">//RENAME-CATID</td> <td data-bbox="594 520 1227 621">OLD-CATID = ... , NEW-CATID = ... , SILENT-RUN = YES , ...</td> </tr> </table>	//MERGE-DIRECTORIES	DIRECTORY-NAMES = ... , SILENT-RUN = YES , ...	//RENAME-CATID	OLD-CATID = ... , NEW-CATID = ... , SILENT-RUN = YES , ...
//MERGE-DIRECTORIES	DIRECTORY-NAMES = ... , SILENT-RUN = YES , ...				
//RENAME-CATID	OLD-CATID = ... , NEW-CATID = ... , SILENT-RUN = YES , ...				
<b>C5</b>	<p><b>Bereinigung von Kollisionen und Überlängen von Dateinamen</b></p> <p>Ein einfacher Weg für die Bereinigung von Verstößen besteht darin, aus den Directories die Einträge, welche Kollisionen oder Namensüberlängen erzeugen, zu löschen. Ermöglicht wird dies durch die DIRCONV-Anweisung REMOVE-CATID. Der Inhalt der Save-Files auf den Sicherungsdatenträgern bleibt dabei unverändert:</p> <table border="1" data-bbox="316 844 1227 981"> <tr> <td data-bbox="316 844 589 981">//REMOVE-CATID</td> <td data-bbox="594 844 1227 981">DIRECTORY-NAME = ... , NEW-DIRECTORY-NAME = ... , CATID = ... , USER-ID = ... , FILE-NAME = ... , JV-NAME = ... , ...</td> </tr> </table> <p>Schwieriger ist es, wenn Sicherungen von Dateien oder Jobvariablen unter einem anderen Namen beibehalten werden sollen. Für eine bestimmte Sicherung kann dies auf folgende Weise erreicht werden:</p> <ol data-bbox="272 1100 1227 1400" style="list-style-type: none"> <li>1. Wiedereinspielen der Datei-/JV-Sicherung (existiert bereits eine namensgleiche Datei/Jobvariable, muss diese umbenannt und später zurückbenannt werden)</li> <li>2. Umbenennen der durch Wiedereinspielen erzeugten Datei/Jobvariablen</li> <li>3. Sichern der umbenannten Datei/Jobvariablen ggf. mit implizitem Löschen (BACKUP-FILES, Operand DELETE-FILES);</li> <li>4. Entfernen des Eintrags für die ursprünglichen Sicherung aus dem Directory durch REMOVE-CATID</li> </ol> <p>Es ist zu beachten, dass bei diesem Vorgehen für das Umbenennen die Relation zwischen Sicherung, Save-File, Save-Version und Sicherungsdatum geändert wird.</p>	//REMOVE-CATID	DIRECTORY-NAME = ... , NEW-DIRECTORY-NAME = ... , CATID = ... , USER-ID = ... , FILE-NAME = ... , JV-NAME = ... , ...		
//REMOVE-CATID	DIRECTORY-NAME = ... , NEW-DIRECTORY-NAME = ... , CATID = ... , USER-ID = ... , FILE-NAME = ... , JV-NAME = ... , ...				

<b>C6</b>	<p><b>Zusammenfassen der Directories</b></p> <p>Sind den SF-Pubsets, die in den SM-Pubset eingehen, unterschiedliche System-Backup-Directories zugewiesen, müssen diese durch folgende DIRCONV-Anweisung zu einem gemeinsamen Directory zusammengefasst werden:</p> <table border="1" data-bbox="316 310 1228 384"> <tr> <td data-bbox="316 310 589 384"><b>//MERGE-DIRECTORIES</b></td> <td data-bbox="595 310 1228 384"><b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b></td> </tr> </table>	<b>//MERGE-DIRECTORIES</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>
<b>//MERGE-DIRECTORIES</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>		
<b>C7</b>	<p><b>Austausch der Catids in dem Directory</b></p> <p>Durch die DIRCONV-Anweisung</p> <table border="1" data-bbox="316 475 1228 583"> <tr> <td data-bbox="316 475 589 583"><b>//RENAME-CATID</b></td> <td data-bbox="595 475 1228 583"><b>DIRECTORY-NAME = ... , OLD-CATID = *ALL = ... , NEW-CATID = &lt;Kennung des SM-Pubsets&gt; , ...</b></td> </tr> </table> <p>wird ein neues System-Backup-Directory erzeugt, in dem alle Referenzen auf die bisherigen SF-Pubsets durch die Referenz auf den künftigen SM-Pubset ersetzt sind. Im Fall, dass die Verarbeitungsebene durch SMPGEN angepasst wird, muss das neue Directory (durch Angabe der Catid in dem Pfadnamen) auf den SF-Pubset gelegt werden, der später zum Control-Volume-Set wird.</p> <p>Die bisherigen Directories können beibehalten werden, um bei Problemen während des Umstiegs wieder auf den Ausgangszustand rücksetzen zu können. Sie dürfen aber nicht mehr für neue Backup-Aufträge verwendet werden, da dies Inkonsistenzen zwischen dem neu erzeugten Directory und dem Inhalt der Datenträger zur Folge hätte.</p>	<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... , OLD-CATID = *ALL = ... , NEW-CATID = &lt;Kennung des SM-Pubsets&gt; , ...</b>
<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... , OLD-CATID = *ALL = ... , NEW-CATID = &lt;Kennung des SM-Pubsets&gt; , ...</b>		
<b>C8</b>	<p><b>Anpassungen in dem MAREN-Katalog</b></p> <p>Unterscheidet sich der künftige Pfadname des System-Backup-Directory auf dem SM-Pubset von dem aktuellen Pfadnamen, muss der neue Pfadname in den Datenträgerzuordnungen des MAREN-Katalogs berücksichtigt werden</p> <table border="1" data-bbox="316 1087 1228 1161"> <tr> <td data-bbox="316 1087 589 1161"><b>//UPDATE-VOLUME-CATALOG</b></td> <td data-bbox="595 1087 1228 1161"><b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b></td> </tr> </table> <p>Dabei wird in dem MAREN-Katalog für alle Datenträger, die in dem mit dem Operanden NEW-DIRECTORY-NAME spezifizierten Directory eingetragen sind, die Zuordnung zu diesem Directory hergestellt. Mit dem Operanden DIRECTORY-NAMES bietet die Anweisung UPDATE-VOLUME-CATALOG auch die Möglichkeit, dass freie Bänder, die für die bisherigen Directories vorreserviert waren, dem neuen Directory zugewiesen werden können, d.h. in den Pool der freien Datenträger des neuen Directory eingehen.</p>	<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>
<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = ... , ...</b>		

<b>C9</b>	<b>Sichern des Backup-Directory (entfällt, wenn Verarbeitungsebene in-place angepasst wird)</b>  Wenn die Verarbeitungsebene nicht in-place sondern durch Sichern und Wiedereinspielen angepasst wird, muss auch das System-Backup-Directory gesichert werden. Dies kann z.B dadurch erfolgen, dass es auf einen von der Bildung des SM-Pubsets nicht betroffenen Pubset kopiert wird.
<b>C10</b>	<b>Bereinigung der Archivdefintionen für die SF-Pubsets</b>  Wenn auf den SM-Pubset übergegangen wird, haben die Archivdefintionen für die alten System-Backup-Archive sowie die Zuordnungen dieser Archive zu den SF-Pubsets keine Bedeutung mehr und können gelöscht werden. Dieser Schritt kann auch erst später erfolgen, z.B. nachdem der Umstieg erfolgreich durchgeführt wurde. In diesem Fall wird bei Problemen während des Umstiegs auf den SM-Pubset das Rücksetzen auf den Ausgangsstand erleichtert.

**b) Maßnahmen nach der Anpassung der Verarbeitungsebene**

<b>C11</b>	<p><b>Einrichten der HSMS-Umgebung auf dem SM-Pubset</b></p> <p>Damit nach der Anpassung der Verarbeitungsebene HSMS-Funktionen genutzt werden können, muss auf dem SM-Pubset durch die HSMS-Anweisung CREATE-SM-PUBSET-PARAMETERS eine HSMS-Umgebung eingerichtet werden. Der Pubset kommt dadurch in den Zustand HSMS-SUPPORTED. Um anschließend HSMS-Funktionen ohne Probleme nutzen zu können, müssen auch Zugriffe auf den MAREN-Katalog möglich sein.</p>
<b>C12</b>	<p><b>Wiedereinspielen des Directory des System-Backup-Archivs (entfällt wenn die Verarbeitungsebene in-place angepasst wurde)</b></p> <p>Wenn die Verarbeitungsebene nicht in-place, sondern durch Sichern und Wiedereinspielen von Dateien angepasst wurde, muss das Backup-Directory auf den SM-Pubset wiedereingespielt werden oder falls es auf einen anderen Pubset verlagert wurde, von dort auf den SM-Pubset kopiert werden (siehe C9). Dabei ist darauf zu achten, dass es auf den Control-Volume-Set gelangt.</p>
<b>C13</b>	<p><b>Einrichten des System-Backup-Archivs</b></p> <p>Um den Sicherungsbestand der System-Backup-Archive der bisherigen SF-Pubsets für den SM-Pubset nutzbar zu machen, ist für den SM-Pubset ein System-Backup-Archiv einzurichten, dem das passend vorbereitete und ggf. in Schritt C12 wiedereingespielte Directory zugewiesen wird. Dies kann entweder bereits bei dem CREATE-SM-PARAMETERS-Aufruf oder später durch einen MODIFY-SM-PUBSET-PARAMETERS-Aufruf erfolgen.</p>

Das System-Backup-Archiv sollte verfügbar gemacht werden, bevor der SM-Pubset zur allgemeinen Nutzung freigegeben wird. Anschließend kann es für erneute Sicherungen wie auch für Zugriffe auf den vor der Bildung des SM-Pubsets erzeugten Sicherungsbestand benutzt werden. Soll das System-Backup-Archiv zur Wiederherstellung der Hintergrundebenen benutzt werden, muss es vorher (siehe B6) nutzbar gemacht werden.

### 6.4.3.2 Erzeugen eines neuen System-Backup-Archivs für den SM-Pubset

Das im allgemeinen empfehlenswerte Standardverfahren für die Anpassung der System-Backup-Archive besteht darin, unmittelbar vor der Bildung des SM-Pubsets (d.h. nachdem die SMPGEN-Checks keine Verstöße mehr aufzeigen) ein neues (privates) Backup-Archiv einzurichten. In diesem wird dann der aktuelle Dateibestand des SF-Pubsets, der in das SM-Pubset eingehen soll, gesichert. Das Archiv dient später als System-Backup-Archiv. Das Backup-Archiv ist automatisch von fremden Pubsets entkoppelt und enthält keine Sicherungen von Dateien, die bei der Anpassung Namenskollisionen oder Namensüberlängen verursachen können. Dieses Verfahren ist besonders dann in Betracht zu ziehen, wenn die in-place-Konvertierung der bestehenden Backup-Archive mit Schwierigkeiten (z.B. wegen der Entkopplung von fremden Pubsets oder der Bereinigung von Namenskollisionen) verbunden ist und die Übernahme des in ihnen abgelegten Sicherungsbestands in das System-Backup-Archiv des SM-Pubsets nur untergeordnete Bedeutung hat.

Es erfolgt zunächst eine Übersicht der für dieses Verfahren erforderlichen Maßnahmen und anschließend eine detaillierte Beschreibung der einzelnen Schritte.

#### a) Maßnahmen vor der Anpassung der Verarbeitungsebene

D1	Einrichten eines neuen Backup-Archivs, das später die Rolle des System-Backup-Archivs des SM-Pubsets übernimmt
D2	Erstellen einer aktuellen Vollsicherung für die in den SM-Pubset eingehenden SF-Pubsets
D3	Catid-Ersetzung in dem Directory
D4	Anpassungen in dem MAREN-Katalog
D5	Sichern des Backup-Directory (entfällt, wenn die Verarbeitungsebene in-place angepasst wird)
D6	Bereinigung der Archivdefinition
D7	Vermerken der Namen der System-Backup-Archive und ihrer Zuordnungen zu den bisherigen SF-Pubsets

#### b) Maßnahmen nach der Anpassung der Verarbeitungsebene

D8	Einrichten der HSMS-Umgebung auf dem SM-Pubset
D9	Wiedereinspielen des Directory des System-Backup-Archivs (entfällt wenn die Verarbeitungsebene in-place angepasst wurde)
D10	Einrichten des System-Backup-Archivs



## a) Maßnahmen vor der Anpassung der Verarbeitungsebene

<b>D1</b>	<p><b>Einrichten eines neuen Backup-Archivs</b></p> <p>Nachdem die SMPGEN-Checks erfolgreich durchgeführt worden sind, wird für alle in den SM-Pubset eingehenden SF-Pubsets ein neues gemeinsames (SF-) Backup-Archiv eingerichtet, dem ausschließlich Bänder als Datenträger zugeordnet sind. Dieses kann später für den SM-Pubset die Rolle des System-Backup-Archivs übernehmen.</p>		
<b>D2</b>	<p><b>Erstellen einer aktuellen Vollsicherung für die in den SM-Pubset eingehenden SF-Pubsets</b></p> <p>Für jeden in den SM-Pubset eingehenden SF-Pubset erfolgt eine Vollsicherung in dieses Backup-Archiv. Dieser Schritt sollte unmittelbar nach erfolgreich durchgeführten SMPGEN-Checks erfolgen.</p>		
<b>D3</b>	<p><b>Catid-Ersetzung in dem Directory</b></p> <p>Durch die DIRCONV-Anweisung</p> <table border="1" data-bbox="309 698 1227 835"> <tr> <td data-bbox="309 698 665 835"><b>//RENAME-CATID</b></td> <td data-bbox="671 698 1227 835"><b>DIRECTORY-NAME = ... , OLD-CATID = *ALL , NEW-CATID = &lt;Kennung des SM-Pubsets&gt; , ...</b></td> </tr> </table> <p>wird ein neues Directory erzeugt, in dem alle Referenzen auf die bisherigen SF-Pubsets durch die Referenz auf den künftigen SM-Pubset ersetzt sind. Probleme auf Grund von Namenskollisionen und Namensüberlängen können sich nicht ergeben, da nur Dateien/Jobvariablen gesichert sind, die im Rahmen der SMPGEN-Checks bereits auf Namenskollisionen und Namensüberlängen überprüft worden sind.</p> <p>Im Fall, dass die Anpassung der Verarbeitungsebene in-place erfolgt, muss das neue Directory auf den SF-Pubset gelegt werden, der Control-Volume-Set wird.</p>	<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... , OLD-CATID = *ALL , NEW-CATID = &lt;Kennung des SM-Pubsets&gt; , ...</b>
<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... , OLD-CATID = *ALL , NEW-CATID = &lt;Kennung des SM-Pubsets&gt; , ...</b>		
<b>D4</b>	<p><b>Anpassungen in dem MAREN-Katalog</b></p> <p>Unterscheidet sich der künftige Pfadname des System-Backup-Directory auf dem SM-Pubset von dem aktuellen Pfadnamen, muss der neue Pfadname in den Datenträgerzuordnungen des MAREN-Katalogs berücksichtigt werden:</p> <table border="1" data-bbox="309 1276 1227 1344"> <tr> <td data-bbox="309 1276 665 1344"><b>//UPDATE-VOLUME-CATALOG</b></td> <td data-bbox="671 1276 1227 1344"><b>DIRECTORY-NAMES = , NEW-DIRECTORY-NAME = ... , ...</b></td> </tr> </table>	<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = , NEW-DIRECTORY-NAME = ... , ...</b>
<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = , NEW-DIRECTORY-NAME = ... , ...</b>		

<b>D4</b> Fort.	Dabei wird in dem MAREN-Katalog für alle Datenträger, die in dem mit dem Operanden NEW-DIRECTORY-NAME spezifizierten Directory eingetragen sind, die Zuordnung zu diesem Directory hergestellt. Mit dem Operanden DIRECTORY-NAMES bietet die Anweisung UPDATE-VOLUME-CATALOG auch die Möglichkeit, dass freie Bänder, die für die bisherigen Directories vorreserviert waren, dem neuen Directory zugewiesen werden können, d.h. in den Pool der freien Datenträger des neuen Directory eingehen.
<b>D5</b>	<p><b>Sichern des Backup-Directory (entfällt, wenn die Verarbeitungsebene in-place angepasst wird)</b></p> <p>Wird die Verarbeitungsebene nicht in-place sondern durch Sichern und Wiedereinspielen angepasst, muss auch das System-Backup-Directory gesichert werden. Dies kann z.B dadurch erfolgen, dass es auf einen von der Bildung des SM-Pubsets nicht betroffenen Pubset kopiert wird.</p>
<b>D6</b>	<p><b>Bereinigung der Archivdefinition</b></p> <p>Wenn auf den SM-Pubset übergegangen wird, wird die bisherige (SF-) Archivdefinition für das neu erzeugte Archiv nicht mehr benötigt und sollte gelöscht werden.</p>
<b>D7</b>	<p><b>Vermerken der Namen der System-Backup-Archive und ihrer Zuordnungen zu den bisherigen SF-Pubsets</b></p> <p>Die bisherigen System-Backup-Archive werden als Benutzer-Backup-Archive beibehalten, d.h. die Zuordnungen als System-Backup-Archive werden aufgehoben. Der Systembetreuer sollte ein Verzeichnis erstellen, in dem für jeden in den SM-Pubset aufgenommenen SF-Pubset der Namen seines früheren System-Backup-Archivs festgehalten wird. Dieses sollte den Benutzern zur Verfügung gestellt werden.</p>

## b) Maßnahmen nach Anpassung der Verarbeitungsebene

<b>D8</b>	<p><b>Einrichten der HSMS-Umgebung auf dem SM-Pubset</b></p> <p>Damit nach der Anpassung der Verarbeitungsebene HSMS-Funktionen genutzt werden können, muss auf dem SM-Pubset durch die HSMS-Anweisung CREATE-SM-PUBSET-PARAMETERS eine HSMS-Umgebung eingerichtet werden. Der Pubset kommt dadurch in den Zustand HSMS-SUPPORTED. Um HSMS-Funktionen ohne Probleme nutzen zu können, müssen auch Zugriffe auf den MAREN-Katalog möglich sein.</p>
-----------	--

<b>D9</b>	<p><b>Wiedereinspielen des Directory des System-Backup-Archivs (entfällt wenn Verarbeitungsebene in-place angepasst wurde)</b></p> <p>Wenn die Verarbeitungsebene nicht in-place sondern durch Sichern und Wiedereinspielen von Dateien angepasst wurde, muss das Backup-Directory auf den SM-Pubset gebracht werden (siehe D5). Dabei ist darauf zu achten, dass es auf den Control-Volume-Set gelangt.</p>		
<b>D10</b>	<p><b>Einrichten des System-Backup-Archivs</b></p> <p>Für den SM-Pubset wird das System-Backup-Archiv eingerichtet, dem das vor der Bildung des SM-Pubsets passend vorbereitete Directory zugewiesen wird. Dies kann entweder bereits bei dem CREATE-SM-PARAMETERS-Aufruf oder später durch einen MODIFY-SM-PUBSET-PARAMETERS-Aufruf erfolgen. Es enthält aktuelle Sicherungen aller in dem SM-Pubset liegenden Dateien, nicht jedoch frühere Sicherungsstände.</p> <p>Sicherungen, die in den System-Backup-Archiven der früheren SF-Pubsets abgelegt sind, können durch die HSMS-Funktion</p> <table border="1" data-bbox="308 707 1227 814"> <tr> <td data-bbox="308 707 630 814"><b>//RESTORE-FILES</b></td> <td data-bbox="636 707 1227 814"><b>ARCHIVE-NAME = ... , NEW-FILE-NAMES = *BY-RULE(NEW-CATALOG-ID=&lt;Kennung des SM-Pubsets&gt; ) ,..</b></td> </tr> </table> <p>eingespielt werden. Dafür wird der explizite Archivname benötigt. Um ihn zu ermitteln, muss der Benutzer wissen, auf welchem SF-Pubset die Datei vor der Bildung des SM-Pubsets gelegen hat, und welches Archiv diesem als System-Backup-Archiv zugewiesen war.</p> <p>Damit bei Bedarf die erforderlichen Informationen bei Bedarf zur Verfügung stehen, sollten bei dem Übergang auf den SM-Pubset durch den Systembetreuer bzw. die Benutzer folgende Daten festgehalten werden:</p> <ul style="list-style-type: none"> <li>● Die Namen der System-Backup-Archive der früheren SF-Pubsets. Sie sollten vom Systembetreuer protokolliert und den Benutzern bekannt gegeben werden (siehe D7).</li> <li>● Die Kennung der SF-Pubsets, auf denen sich die Dateien vor der Bildung des SM-Pubsets befanden. Hierfür ist es empfehlenswert, vor der Bildung des SM-Pubsets alle in den SM-Pubset eingehenden Dateien mit ihrem vollständigen Pfadnamen zu protokollieren (z.B. in einer Ausgabeliste des Kommandos SHOW-FILE-ATTRIBUTES). Diese Maßnahme kann durch die einzelnen Benutzer für ihre jeweiligen Benutzerkennungen vorgenommen werden. Zusätzlich sollte auch der Systembetreuer ein benutzerübergreifendes Protokoll erstellen.</li> <li>● Wenn Dateien durch Benutzer für die Bildung des SM-Pubsets umbenannt werden, um Namenskonflikte zu bereinigen, sollten ihre ursprünglichen Dateinamen festgehalten werden.</li> </ul>	<b>//RESTORE-FILES</b>	<b>ARCHIVE-NAME = ... , NEW-FILE-NAMES = *BY-RULE(NEW-CATALOG-ID=&lt;Kennung des SM-Pubsets&gt; ) ,..</b>
<b>//RESTORE-FILES</b>	<b>ARCHIVE-NAME = ... , NEW-FILE-NAMES = *BY-RULE(NEW-CATALOG-ID=&lt;Kennung des SM-Pubsets&gt; ) ,..</b>		

### 6.4.3.3 Anpassung privater Backup-Archive

Von der Bildung des SM-Pubsets sind alle privaten (SF-) Backup-Archive betroffen, in denen künftig Sicherungen aus dem SM-Pubset abgelegt werden sollen. Werden sie nicht angepasst, so kann nach Bildung des SM-Pubsets zwar noch auf die in ihnen abgelegten Datei- bzw. JV-Sicherungen zugegriffen werden, neue Sicherungen von Dateien/Jobvariablen des SM-Pubsets, sind aber in diese Archive nicht mehr möglich.

Im Unterschied zu den System-Backup-Archiven ist die Ermittlung der betroffenen privaten Archive schwieriger. Eine Unterstützung bietet dabei die HSMS-Anweisung `SHOW-ARCHIVE-ATTRIBUTES`. Vor der Bildung des SM-Pubsets sollte der Systembetreuer die Eigentümer der betroffenen privaten Backup-Archive informieren. Die Anpassungen der privaten Backup-Archive können analog zu den System-Backup-Archiven vorgenommen werden.

### 6.4.4 Anpassung der Langzeitarchive

Langzeitarchive können wie bisher pubset-übergreifend sowohl zum Erstellen als auch zum Wiedereinspielen von Datei-Archivierungen benutzt werden. Bei der Bildung eines SM-Pubsets aus SF-Pubsets sind daher keine Anpassungen der Langzeitarchive erforderlich.

### 6.4.5 Nutzung von ARCHIVE-Funktionen

Sicherungsbestände, die von Dateien auf SF-Pubsets mit ARCHIVE erstellt worden sind, können auf SM-Pubsets entweder mit ARCHIVE oder mit HSMS (Anweisung `IMPORT-FILES`) wiedereingespielt werden.

## 6.5 Rückstieg von einem SM-Pubset auf SF-Pubsets

### 6.5.1 Grundsätzliche Aspekte

Beim Rückstieg von SM-Pubsets auf SF-Pubsets müssen die in [Kapitel „Erzeugen eines SM-Pubsets aus bestehenden SF-Pubsets“ auf Seite 65](#) für den Umstieg behandelten Aspekte (Änderungen der Pfadnamen in Kommandoprozeduren, Anpassen der MRSCAT-Einträge etc.) in „umgekehrter Richtung“ beachtet werden. Der Rückstieg betrifft die Verarbeitungsebene, die Hintergrundebenen und die Backup-Konfiguration der Pubsets. Ein spezieller Fall besteht im Rücksetzen auf den Ausgangszustand vor der Bildung des SM-Pubsets, wobei Änderungen des Dateibestands und Dateibearbeitungen, die nach der Bildung des SM-Pubsets erfolgt sind, unwirksam werden. Es ist vor allem wichtig, wenn beim Umstieg auf SM-Pubsets Probleme auftreten.

### 6.5.2 Anpassung der Verarbeitungsebene

#### Rückkonvertierung

Die folgenden Schritte beziehen sich auf den allgemeinen Fall, dass der SM-Pubset bei der Rückkonvertierung in mehrere SF-Pubsets aufgeteilt wird. Das Umwandeln eines SM-Pubsets in einen SF-Pubset mit gleicher Pubset-Kennung stellt einen besonders einfachen Sonderfall dar.

- Die in der Verarbeitungsebene des SM-Pubsets liegenden Dateien, der Benutzer- und GUARDS-Katalog sowie die Katalogeinträge von auf Hintergrundebenen befindlichen Dateien wie auch von Dateien auf Privatplatten oder Bändern werden gesichert. Dabei ist zu beachten, dass die Dateinamen auch mit der Catid des SF-Pubsets nicht länger als 54 Zeichen werden dürfen, ggf. müssen sie vorher umbenannt werden.
- Die SF-Pubsets werden mit SIR erzeugt.
- Das Backup-Archiv mit den Sicherungen des SM-Pubsets wird für die SF-Pubsets verfügbar gemacht (Importieren des Pubsets, Einspielen des Directory, Definieren eines Backup-Archivs für SF-Pubsets, dabei Zuordnen des zuvor eingespielten Directory).
- Die auf dem SM-Pubset erstellte Sicherung des Benutzerkatalogs wird auf den SF-Pubsets wieder eingespielt. Auf dem SM-Pubset getroffene Festlegungen für die Benutzer, welche auf SF-Pubsets keine Entsprechung haben (z.B. SPACE-Limit für WORK-Dateien) gehen dabei verloren. Damit die wiedereingespielten Benutzerkataloge wirksam werden, müssen die SF-Pubsets exportiert und anschließend rekonstruiert werden. Im Allgemeinen wird eine Nachbearbeitung des Benutzerkatalogs durch den Systembetreuer erforderlich sein (z.B. Anpassung von Space-Limits, Entfernen einzelner Benutzerkennungen).

- Der gesicherte Dateibestand des SM-Pubsets wird auf die SF-Pubsets eingespielt. Dabei muss der Systembetreuer die Verteilung der Dateien auf die einzelnen SF-Pubsets bestimmen. Wurden Dateien auf dem SM-Pubset Dateiattribute zugewiesen, welche auf SF-Pubsets keine Entsprechung haben (z.B. WORK-Datei), werden sie beim Wiedereinspielen in SF-Pubset-Umgebung automatisch auf Dateiattribute abgebildet, die auch für SF-Pubsets gültig sind.

### Rücksetzen

Um auf den Stand der SF-Pubsets vor der Bildung des SM-Pubsets rückzusetzen, werden zunächst die SF-Pubsets mithilfe von SIR wieder erzeugt. Auf sie wird der Sicherungsbestand der SF-Pubsets, der vor der Bildung des SM-Pubsets erstellt worden ist (inklusive Benutzer- und GUARDS-Katalog, Migrations-Directory, etc.), wiedereingespielt.

## 6.5.3 Anpassung der Backup-Archive

### Rückkonvertierung

HSMS unterstützt das Rückkonvertieren eines SM-Pubsets in einen oder mehrere SF-Pubsets durch die Möglichkeit, das Backup-Archiv eines SM-Pubsets in ein Backup-Archiv für SF-Pubsets umzuwandeln, ohne dass Zugriffe auf die Sicherungsdatenträger erforderlich sind.

Dazu wird das Directory auf dem SM-Pubset gesichert und später auf einem - ggf. in der Zwischenzeit durch SIR neu erzeugten - SF-Pubset wiedereingespielt. Dann wird ein Backup-Archiv für SF-Pubsets definiert, dem das wiedereingespielte Directory zugeordnet wird. Hat sich durch die Verlagerung des Directory von dem SM-Pubset auf einen SF-Pubset sein Pfadnamen geändert (z.B. weil sich die Kennung des SF-Pubsets von der des SM-Pubsets unterscheidet), müssen Anpassungen in dem MAREN-Katalog erfolgen. Sie werden durch die DIRCONV-Anweisung UPDATE-VOLUME- CATALOG ermöglicht.

In dem Backup-Archiv des früheren SM-Pubsets sind die Dateisicherungen zunächst noch unter der Catid des SM-Pubsets vermerkt. Durch die DIRCONV-Anweisung RENAME-CATID kann in allen Dateisicherungen global oder für einzelne Benutzer die Catid ausgetauscht werden. Sicherungen, die in dem Backup-Archiv unter einer Catid abgelegt sind, die sich von der Kennung des Pubsets unterscheidet, auf dem sie wiedereingespielt werden, müssen bei dem Backup-Aufruf mit vollständigem Pfadnamen, d.h. mit Angabe der Catid, referenziert werden.

Im speziellen Fall, dass der SM-Pubset in einen SF-Pubset mit gleicher Kennung umgewandelt wird, erübrigen sich die Anpassung des MAREN-Katalogs und der Austausch der Catids in dem Directory.

## Rücksetzen

Um bei Problemen bei dem Übergang auf den SM-Pubset die ursprüngliche Backup-Konfiguration der SF-Pubsets wiederherstellen zu können, werden die Directories der Backup-Archive vor der Bildung des SM-Pubsets gesichert oder auf einen Pubset verlagert, der nicht von der Bildung des SM-Pubsets betroffen ist. Sie können für das Rücksetzen genutzt werden, solange sie mit dem Inhalt der Sicherungsdatenträger konsistent sind, d.h. solange nach der Fixierung der Directories der Inhalt der Datenträger unverändert bleibt. Zum Wiederherstellen eines Backup-Archivs in seinem Zustand vor der Bildung des SM-Pubsets sind folgende Schritte erforderlich:

- Das ursprüngliche Backup-Directory (mit dem Stand vor der Bildung des SM-Pubsets) muss auf einem SF-Pubset wiedereingespielt werden, falls es bei der Bildung des SM-Pubsets verloren gegangen ist.
- Das ursprüngliche Backup-Archiv muss wieder definiert werden, falls die Definition zwischenzeitlich gelöscht wurde. Dabei erfolgt die Zuweisung des wiederhergestellten Directory.
- Falls im MAREN-Katalog der Pfadname des ursprünglichen Directory ersetzt wurde, muss dieser wieder eingetragen werden. Dies ermöglicht die DIRCONV-Anweisung:

```
//UPDATE-VOLUME-  
CATALOG
```

```
DIRECTORY-NAMES = *NONE ,  
NEW-DIRECTORY-NAME=<Name des alten Directory> , ...
```

Das Rücksetzen der Backup-Directories auf den Ausgangszustand vor der Bildung des SM-Pubsets ist nur solange möglich, wie sich der ursprüngliche Sicherungsbestand noch nicht durch Sicherungen für den SM-Pubset verändert hat.

### 6.5.4 Anpassung des Migrationsarchivs

In dem speziellen Fall, dass ein SM-Pubset, in dem keine Dateien auf S1 migriert sind, in genau einen SF-Pubset umgewandelt wird, kann das Migrationsarchiv in-place rückkonvertiert werden. Auch wenn während der Konvertierung der SF-Pubsets Probleme auftreten, können die Migrationsarchive in ihren Ausgangszustand vor der Bildung des SM-Pubsets wiederhergestellt werden, ohne dass Zugriffe auf die Sicherungsdatenträger notwendig sind. Im Fall, dass ein SM-Pubset in mehrere SF-Pubsets aufgespalten wird, oder dass Dateien auf S1 migriert sind, muss die Anpassung durch Sichern und Wiedereinspielen von Dateien erfolgen.

## Rückkonvertierung durch Sichern und Wiedereinspielen von Dateien

Die auf den Hintergrundebenen des SM-Pubsets liegenden Dateien und die Katalogeinträge der migrierten Dateien werden gesichert z.B. im Rahmen der Gesamtsicherung des SM-Pubsets. Nachdem die SF-Pubsets eingerichtet worden sind, die Backup-Archive mit den Sicherungen der Dateien nutzbar gemacht worden sind und für die SF-Pubsets zunächst leere Migrationsarchive eingerichtet sind, wird durch die HSMS-Anweisung REPAIR-CATALOG-BY-RESTORE der Dateibestand der Hintergrundebenen aus den Sicherungen wiederhergestellt. Eine andere Vorgehensweise besteht darin, die früher auf den Hintergrundebenen des SM-Pubsets befindlichen Dateien zunächst in die Verarbeitungsebene der SF-Pubsets einzuspielen und sie von dort durch Migrations-Aufträge auf die Hintergrundebenen zu verlagern. Die Verarbeitungsebene der SF-Pubsets muss dafür genug Platz bieten.

## Rückkonvertierung durch in-place-Anpassungen

Für einen SM-Pubset-, der in genau einen SF-Pubset umgewandelt wird, kann das Migrationsarchiv durch folgende Maßnahmen in-place angepasst werden, wenn keine Dateien auf S1 migriert sind.

- Falls sich die Catid des bisherigen SM-Pubsets von der Catid des künftigen SF-Pubsets unterscheidet, wird sie in dem Directory durch folgende DIRCONV-Anweisung ausgetauscht:

<b>//RENAME-CATID</b>	<b>DIRECTORY-NAME = ... , OLD-CATID = *ALL , NEW-CATID = &lt;Kennung des SF-Pubsets&gt; , ...</b>
-----------------------	---

- Falls sich der Pfadname des Directory bei der Umwandlung des SM-Pubsets in einen SF-Pubset ändert (z.B. weil sich die Kennung des SF-Pubsets von der des SM-Pubsets unterscheidet), muss der neue Pfadname in dem MAREN-Katalog berücksichtigt werden. Dies geschieht durch folgende DIRCONV-Anweisung:

<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = ... , NEW-DIRECTORY-NAME = &lt;Pfadname des Directory auf dem SF-Pubset&gt; , ...</b>
--------------------------------	--

- Auf dem SM-Pubset werden das Migrations-Directory und die Katalogeinträge der migrierten Dateien gesichert.
- Mit SIR wird der SF-Pubset erzeugt.
- Der Dateibestand der S0-Ebene (einschließlich dem Migrations-Directory) und die Katalogeinträge der migrierten Dateien werden auf dem SF-Pubset aus Sicherungen wiederhergestellt.



- Das Migrationsarchiv für den SF-Pubset muss definiert werden und dem SF-Pubset zugewiesen werden. Als Directory wird ihm das wiedereingespielte Migrations-Directory zugewiesen.

## Rücksetzen

Um bei Problemen während des Übergangs auf den SM-Pubset die Migrations-Directories der SF-Pubsets (ohne S1-Ebene) in ihrem Ausgangszustand wiederherstellen zu können, werden die Directories der Migrationsarchive vor der Bildung des SM-Pubsets gesichert oder auf einen Pubset verlagert, der nicht von der Bildung des SM-Pubsets betroffen ist. Sie können für das Rücksetzen genutzt werden, solange sie mit dem Inhalt der Sicherungsdatenträger konsistent sind, d.h. solange auf dem SM-Pubset keine Dateien auf Hintergrundebenen migriert werden. Zum Wiederherstellen eines Migrationsarchivs sind folgende Schritte erforderlich:

- Das ursprüngliche Migrations-Directory (mit dem Stand vor der Bildung des SM-Pubsets) muss auf einem SF-Pubset wiedereingespielt werden, falls es bei der Bildung des SM-Pubsets verloren gegangen ist.
- Das ursprüngliche Migrationsarchiv muss wieder definiert werden. Dabei wird ihm das wiedereingespielte Directory zugewiesen.
- Falls im MAREN-Katalog der Pfadname des ursprünglichen Directory ersetzt wurde (Schritt C7), muss dieser wieder eingetragen werden. Dies wird durch die DIRCONV-Anweisung

<b>//UPDATE-VOLUME-CATALOG</b>	<b>DIRECTORY-NAMES = *NONE , NEW-DIRECTORY-NAME= &lt;Name des alten Directory&gt; , ...</b>
--------------------------------	---

ermöglicht. Das Rücksetzen der Migrations-Directories in den Ausgangszustand vor der Bildung des SM-Pubsets ist nur in der Übergangsphase während der Konvertierung der SF-Pubsets anwendbar.

## 6.6 Exemplarische Umstiegsszenarien und Vorgehensweisen

### 6.6.1 Umstiegsszenario 1

Das Umstiegsszenario 1 ist in [Bild 11](#) skizziert. Es ist dadurch charakterisiert, dass ein SF-Pubset durch einen nahezu eigenschaftsgleichen SM-Pubset ersetzt wird. Dabei wird vorausgesetzt, dass die Migrations- und Backup-Archive des SF-Pubsets schon pubset-bezogen eingerichtet sind. Der SM-Pubset unterscheidet sich in seiner Funktionalität zunächst nur unwesentlich von dem früheren SF-Pubset. Er kann sukzessive erweitert werden (Hinzunahme weiterer Volume-Sets, Einrichten von HSMS-Management-Klassen, etc.), um die für SM-Pubsets vorhandenen Möglichkeiten zu nutzen.

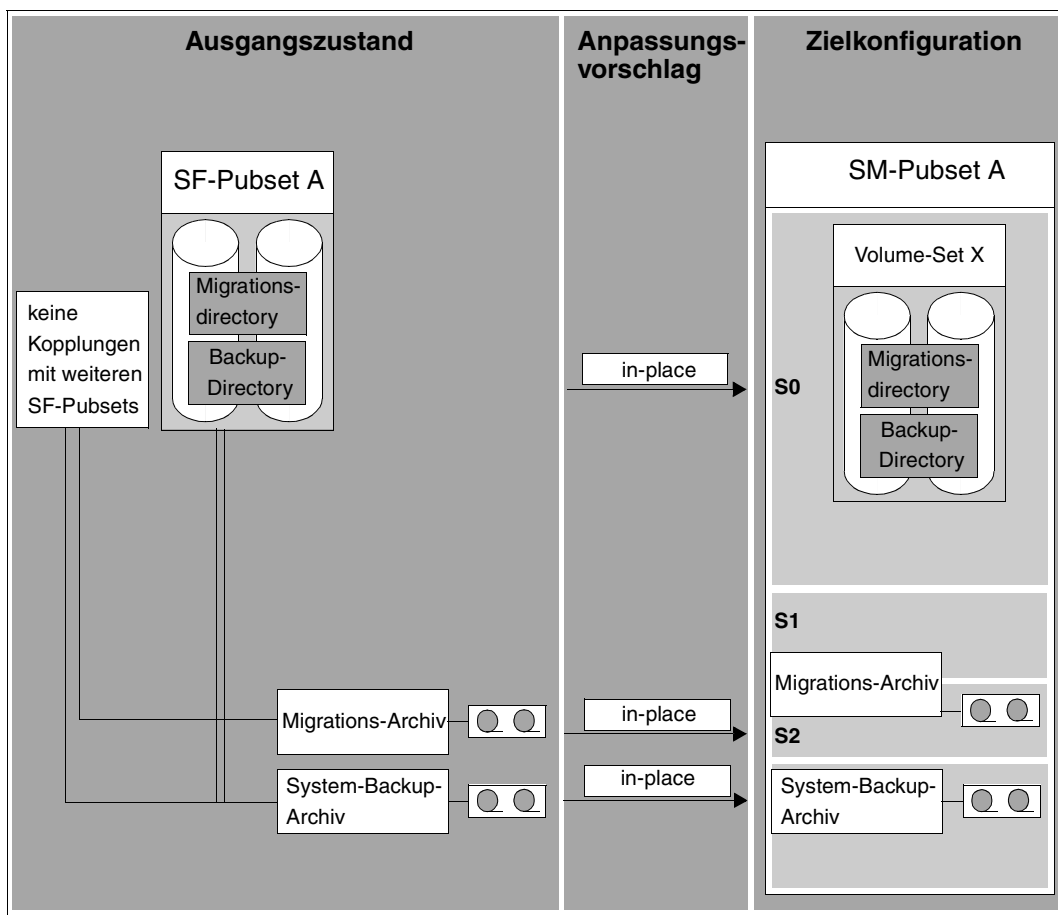


Bild 11: Umstiegsszenario 1

## Ausgangszustand und Zielkonfiguration

### 1. Verarbeitungsebene

Die physikalische Struktur des Volume-Sets entspricht der des SF-Pubsets. Der Dateibestand auf der Verarbeitungsebene bleibt unverändert.

### 2. Hintergrundebenen

Dem SF-Pubset ist eine S2-Ebene, jedoch keine S1-Ebene zugeordnet. Das Migrationsarchiv wird nicht durch andere SF-Pubsets benutzt, das Migrations-Directory liegt auf dem Pubset selbst. Die S2-Ebene geht mit ihrem gesamten Dateibestand in die S2-Ebene des SM-Pubsets über.

### 3. Backup-Konfiguration

Dem SF-Pubset ist ein System-Backup-Archiv zugeordnet, dem ausschließlich Bänder als Sicherungsdatenträger zugeordnet sind. Es wird nicht für andere SF-Pubsets genutzt, das Backup-Directory liegt auf dem SF-Pubset selbst. Das System-Backup-Archiv geht mit seinem kompletten Sicherungsbestand in das System-Backup-Archiv des SM-Pubsets über.

## Vorgeschlagenes Vorgehen

Maßnahmen vor der Anpassung der Verarbeitungsebene	Bemerkungen
Bereinigung von Kollisionen und Überlängen von Pfadnamen (mithilfe von SMPGEN-Checks)	entfällt, da der SM-Pubset die Kennung des SF-Pubsets übernimmt
Anpassen von Kommandoprozeduren und Programmen, die Kommandos/Anweisungen enthalten, die für SM-Pubsets nicht zulässig sind	betrifft den Systembetreuer und die einzelnen Benutzer
Anpassung der privaten Backup-Archive	betrifft den Systembetreuer und die einzelnen Benutzer; das Vorgehen ist abhängig von den speziellen Voraussetzungen (Kopplungen mit anderen SF-Pubsets)
<b>ab jetzt: Pubset-Betrieb im Modus ACCESS-CONTROLLED!</b>	
SMPGEN-Checks durch den Systembetreuer: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET explizit im Prüfmodus (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY)	
generelle Empfehlung: Vollsicherung der SF-Pubsets mit HSMS inklusive der Hintergrundebenen	D1, D2
ggf. FDDRL-Sicherung der SF-Pubsets (schnelles Rücksetzen)	
Ermitteln der Directories des Migrationsarchivs und des System-Backup-Archivs des SF-Pubsets	A2 ,C1
Anpassung des Migrationsarchivs (Directory)	entfällt, da keine Kopplungen mit fremden Pubsets bestehen und der SM-Pubset die Kennung des SF-Pubsets übernimmt
Anpassung des System-Backup-Archivs (Directory)	entfällt, da keine Kopplungen mit fremden Pubsets bestehen und der SM-Pubset die Kennung des SF-Pubsets übernimmt
Bereinigung der HSMS-Definitionen für die SF-Pubsets; ggf erst nach Bildung des SM-Pubsets (Rücksetzen)	A11, D6

Anpassung der Verarbeitungsebene	Bemerkungen
Umbenennung des SF-Pubsets (PVSREN). Die neue Pubset-Kennung (künftige Volume-Set-Kennung) sollte nicht länger sein als die bisherige, um Überlängen von Pfadnamen zu vermeiden.	
Umwandlung des Pubsets mit SMPGEN: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET (implizit im Ausführungsmodus)	
Anpassung der von SMPGEN getroffenen Voreinstellungen	
ggf. Einrichten von Storage-Klassen und Zuweisen von Default-Storage-Klassen	
Anpassung der MRSCAT-Einträge an sonstigen Rechnern	
Maßnahmen nach der Anpassung der Verarbeitungsebene	
Pubset-Inbetriebnahme im Modus ACCESS-CONTROLLED	
Einrichten der HSMS-Umgebung auf dem SM-Pubset	A12 (=C11)
Einrichten des System-Backup-Archivs, Zuordnen des Directory	C13
Einrichten des Migrationsarchivs, Zuordnen des Directory	A14, A15
Festlegen der Migration-Control-Parameter (z.B. Except-File)	
Pubset-Betrieb wieder im allgemein zugänglichen Modus	

## 6.6.2 Umstiegsszenario 2

Im Umstiegsszenario 2 (siehe Bild 12) wird wie im Umstiegsszenario 1 ein SF-Pubset in einen namensgleichen SM-Pubset umgewandelt, der jedoch aus mehreren Volume-Sets besteht. Die Verarbeitungsebene kann daher nicht in-place angepasst werden. Abgesehen davon ist der Umstellungsaufwand (wie in Umstiegs-Szenario 1) gering. Beispielsweise muss die Adressierung der in dem Pubset liegenden Dateien nicht angepasst werden. Dieses Szenario erlaubt z.B einen SF-Pubset mit einer großen Anzahl von Volumes (große Ausfallereinheit) durch einen SM-Pubset aus mehreren Volume-Sets (kleinere Untereinheiten für Ausfall) zu ersetzen. Eine andere Anwendungsmöglichkeit besteht darin, einen SF-Pubset mit dem Format K durch einen SM-Pubset zu ersetzen, der jeweils einen Volume-Set mit dem Format K und einen mit dem Format NK2 enthält. In dem SM-Pubset können die Dateien mit dem Format NK2 auf den Volume-Set mit dem Format NK2 gelegt werden. Sie lassen sich damit im Allgemeinen auf dem SM-Pubset kostengünstiger speichern als auf dem SF-Pubset.

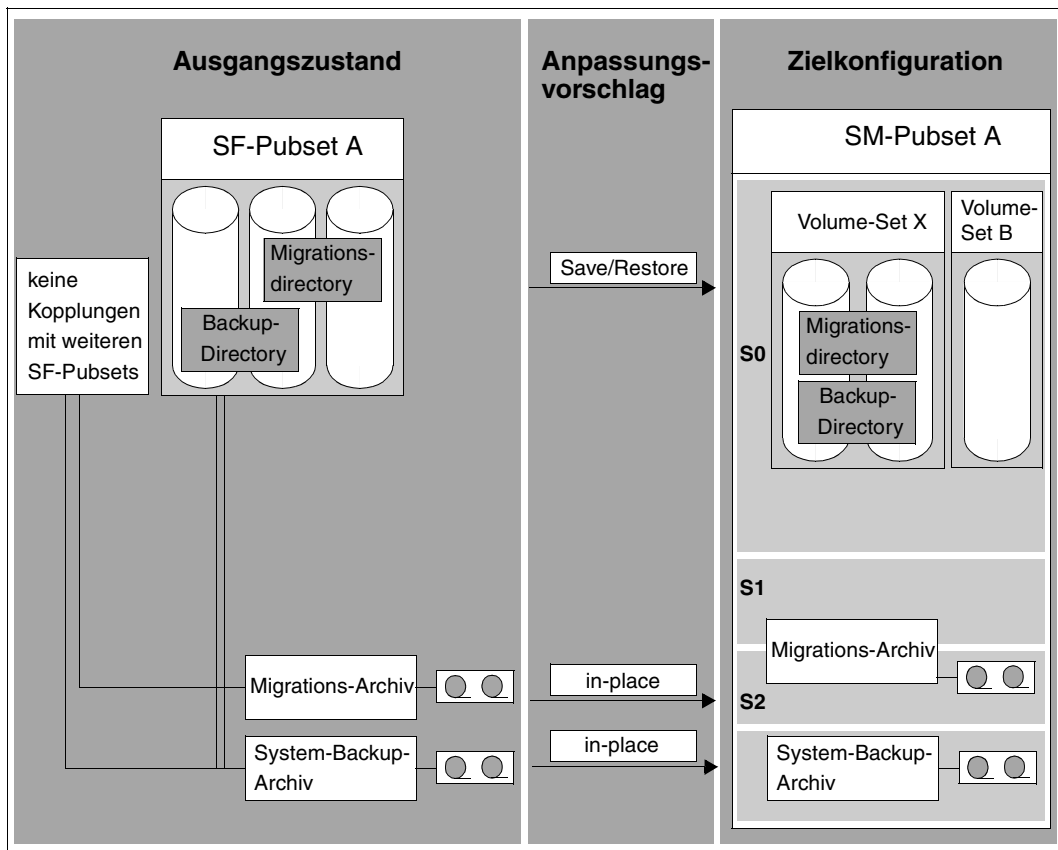


Bild 12: Umstiegsszenario 2

## Ausgangszustand und Zielkonfiguration

### 1. Verarbeitungsebene

Die physikalische Struktur der Verarbeitungsebene wird neu festgelegt. Der Dateibestand auf der Verarbeitungsebene bleibt unverändert.

### 2. Hintergrundebenen

Dem SF-Pubset ist eine S2-Ebene, jedoch keine S1-Ebene zugeordnet. Das Migrationsarchiv wird nicht durch andere SF-Pubsets benutzt, das Migrations-Directory liegt auf dem Pubset selbst. Die S2-Ebene geht mit ihrem gesamten Dateibestand in die S2-Ebene des SM-Pubsets über.

### 3. Backup-Konfiguration

Dem SF-Pubset ist ein System-Backup-Archiv zugeordnet, dem ausschließlich Bänder als Sicherungsdatenträger zugeordnet sind. Es wird nicht für andere SF-Pubsets genutzt, das Backup-Directory liegt auf dem SF-Pubset selbst. Das System-Backup-Archiv geht mit seinem kompletten Sicherungsbestand in das System-Backup-Archiv des SM-Pubsets über.

## Vorgeschlagenes Vorgehen

Maßnahmen vor der Anpassung der Verarbeitungsebene	Bemerkungen
Bereinigung von Kollisionen und Überlängen von Pfadnamen (mithilfe von SMPGEN-Checks)	entfällt, da der SM-Pubset die Kennung des SF-Pubsets übernimmt
Anpassen von Kommandoprozeduren und Programmen, die Kommandos/Anweisungen enthalten, die für SM-Pubsets nicht zulässig sind	betrifft den Systembetreuer und die einzelnen Benutzer
Anpassung der privaten Backup-Archive	betrifft den Systembetreuer und die einzelnen Benutzer; das Vorgehen ist abhängig von den speziellen Voraussetzungen (Kopplungen mit anderen SF-Pubsets)
<b>ab jetzt: Pubset-Betrieb im Modus ACCESS-CONTROLLED!</b>	
SMPGEN-Checks durch den Systembetreuer: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET explizit im Prüfmodus (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY)	
ggf. FDDRL-Sicherung des SF-Pubsets (schnelles Rücksetzen)	
Ermitteln der Directories des Migrationsarchivs und des System-Backup-Archivs des SF-Pubsets	A2, C1
Anpassung des Migrationsarchivs (Directory)	entfällt, da keine Kopplungen mit fremden Pubsets bestehen und der SM-Pubset die Kennung des SF-Pubsets übernimmt
Anpassung des System-Backup-Archivs (Directory)	entfällt, da keine Kopplungen mit fremden Pubsets bestehen und der SM-Pubset die Kennung des SF-Pubsets übernimmt
Bereinigung der HSMS-Definitionen für die SF-Pubsets; ggf erst nach Bildung des SM-Pubsets (Rücksetzen)	A11, D6



Anpassung der Verarbeitungsebene	
Vollsicherung der in der Verarbeitungsebene befindlichen Dateien; dabei sind insbesondere auch die Directory-Dateien des System-Backup-Archivs und des Migrationsarchivs des Pubsets und ggf. auch der MAREN-Katalog zu berücksichtigen; für den MAREN-Katalog ist eine Verlagerung auf einen anderen Pubset in Betracht zu ziehen.	
Erzeugen des SM-Pubsets mit SIR	
Anpassung der von SIR getroffenen Voreinstellungen	
ggf. Einrichten von Storage-Klassen und Zuweisen von Default-Storage-Klassen	
Wiedereinspielen der Vollsicherung der Verarbeitungsebene; dabei auch Wiederherstellen der Directories des System-Backup-Archivs und des Migrationsarchivs	
Anpassung der MRSCAT-Einträge an sonstigen Rechnern	
Maßnahmen nach der Anpassung der Verarbeitungsebene	Bemerkungen
Pubset-Inbetriebnahme im Modus ACCESS-CONTROLLED	
Einrichten der HSMS-Umgebung auf dem SM-Pubset	A12 (=C11)
Einrichten des System-Backup-Archivs, Zuordnen des Directory	C13
Einrichten des Migrationsarchivs, Zuordnen des Directory,	A14, A15
Festlegen der Migration-Control-Parameter (z.B. Except-File)	
<b>Pubset-Betrieb wieder im allgemein zugänglichen Modus</b>	

### 6.6.3 Umstiegsszenario 3

Im Umstiegsszenario 3 (Bild 13) wird aus mehreren SF-Pubsets ein SM-Pubset gebildet. Alle betroffenen SF-Pubsets haben ein gemeinsames Migrationsarchiv und ein gemeinsames System-Backup-Archiv, das nicht durch sonstige Pubsets genutzt wird. Weder das Migrationsarchiv noch das System-Backup-Archiv haben Platten als Datenträger zugeordnet. Die Kennung des SM-Pubsets soll mit der Kennung des Pubsets eines der in ihn eingehenden SF-Pubsets übereinstimmen. Der Umstellungsaufwand hängt wesentlich von dem Umfang ab, in dem Anpassungen von Benutzerdatenstrukturen (z.B. Catid-Ersetzung in Kommandoprozeduren) und Änderungen von Dateinamen (z.B. wegen Namenskollisionen) erforderlich sind.

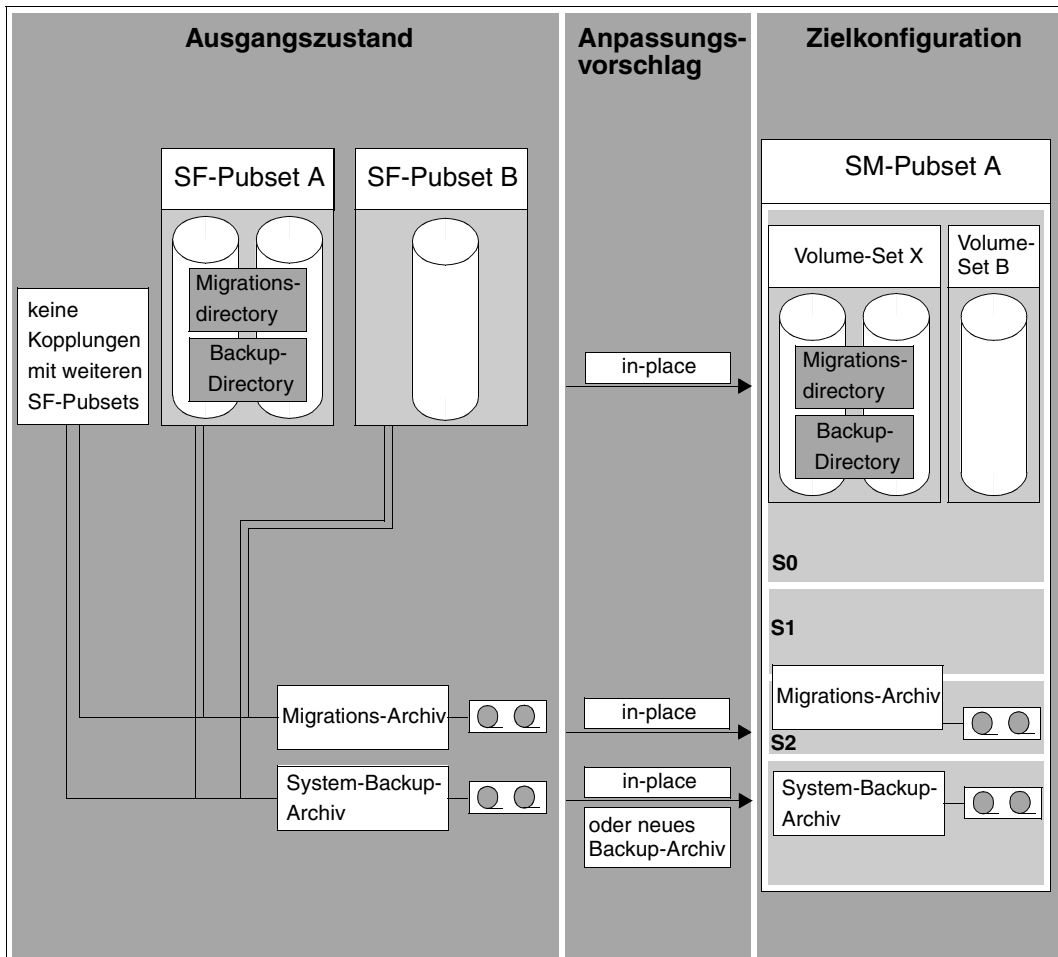


Bild 13: Umstiegsszenario 3

## Ausgangszustand und Zielkonfiguration

### 1. Verarbeitungsebene

Die Verarbeitungsebene des SM-Pubsets ergibt sich durch das Zusammenfassen der Verarbeitungsebenen der einzelnen SF-Pubsets. Der darauf befindliche Dateibestand ändert sich dabei nicht.

### 2. Hintergrundebenen

Allen in den SM-Pubset eingehenden SF-Pubsets ist ein gemeinsames Migrationsarchiv zugeordnet, das nicht für weitere Pubsets genutzt wird. Das Directory befindet sich auf dem SF-Pubset, der Control-Volume-Set wird. Keinem der SF-Pubsets ist ein S1-Pubset zugeordnet. Die S2-Ebene geht mit ihrem gesamten Dateibestand in die S2-Ebene des SM-Pubsets über.

### 3. Backup-Konfiguration

Allen in den SM-Pubset eingehenden SF-Pubsets ist ein gemeinsames Backup-Archiv als System-Backup-Archiv zugeordnet. Dieses wird nicht durch weitere Pubsets verwendet. Als Sicherungsdatenträger dienen nur Bänder. Das Directory befindet sich auf dem SF-Pubset, der Control-Volume-Set wird. Das Backup-Archiv geht mit seinem kompletten Sicherungsbestand in das System-Backup-Archiv des SM-Pubsets über.

## Vorgeschlagenes Vorgehen

Maßnahmen vor der Anpassung der Verarbeitungsebene	Bemerkungen
Bereinigung von Kollisionen und Überlängen von Pfadnamen (mithilfe von SMPGEN-Checks)	betrifft den Systembetreuer und die einzelnen Benutzer; dieser Schritt beinhaltet auch A1 bzw. B1
Berücksichtigung von geänderten Pfadnamen in Kommandoprozeduren, Programmen, GUARDS-Katalogen und sonstigen Benutzerdatenstrukturen; Zuordnen von geeigneten Dateiattributen in den Anweisungen zum Anlegen von Dateien	betrifft den Systembetreuer und die einzelnen Benutzer
Anpassen von Kommandoprozeduren und Programmen, die Kommandos/Anweisungen enthalten, die für SM-Pubsets nicht zulässig sind	betrifft den Systembetreuer und die einzelnen Benutzer
Anpassung der Default-Catids (in allen potenziellen Home-Pubsets)	betrifft nur den Systembetreuer

Maßnahmen vor der Anpassung der Verarbeitungsebene (Forts.)	Bemerkungen	
Anpassung der privaten Backup-Archive	betrifft den Systembetreuer und die einzelnen Benutzer; das Vorgehen ist abhängig von den speziellen Voraussetzungen	
<b>ab jetzt: Pubset-Betrieb im Modus ACCESS-CONTROLLED!</b>		
SMPGEN-Checks durch den Systembetreuer: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET explizit im Prüfmodus (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY)		
Attributierung bestehender Dateien: Dateien mit PERFORMANCE=STD, welche auf SF-Pubsets aus hochperformanten Platten liegen, sollten PERFORMANCE=HIGH oder VERY-HIGH erhalten; das Verfügbarkeitsattribut wird automatisch durch SMPGEN berücksichtigt	kann für alle Benutzerdateien zentral durch den Systembetreuer vorgenommen werden	
generelle Empfehlung: Vollsicherung der SF-Pubsets mit HSMS inklusive der Hintergrundebenen	D1, <b>D2</b>	
ggf. FDDRL-Sicherung der SF-Pubsets (schnelles Rücksetzen)		
Anpassung des Migrationsarchivs (Directories)	A1, A5, A6, A7(optional), A8, A9	
Anpassung der System-Backup-Archive (Directories)	in-place	C1 ,C4, <b>C5</b> , C6, C7, C8, C10
	neues Archiv	(D1, <b>D2</b> s.o.), D3, D4, D7
Bereinigung der HSMS-Definitionen für die SF-Pubsets; ggf. erst nach Bildung des SM-Pubsets (Rücksetzen)	A11, D6	

Anpassung der Verarbeitungsebene	Bemerkungen	
Umbenennung des SF-Pubsets, dessen Kennung sich auf den SM-Pubset übertragen soll (PVSREN). Die neue Pubset-Kennung (künftige Volume-Set-Kennung) sollte nicht länger sein als die bisherige, um Überlängen von Pfadnamen zu vermeiden.		
Umwandlung des Pubsets mit SMPGEN: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET (implizit im Ausführungsmodus)		
Anpassung der von SMPGEN getroffenen Voreinstellungen		
ggf. Einrichten von Storage-Klassen und Zuweisen von Default-Storage-Klassen		
Anpassung der MRSCAT-Einträge an sonstigen Rechnern		
Maßnahmen nach der Anpassung der Verarbeitungsebene		
Pubset-Inbetriebnahme im Modus ACCESS-CONTROLLED		
Einrichten der HSMS-Umgebung auf dem SM-Pubset	A12 (=C11=D8)	
Einrichten des System-Backup-Archivs, Zuordnen des Directory	in-place	C13
	neues Archiv	<b>D9</b> , D10, D11
Einrichten des Migrationsarchivs, Zuordnen des Directory	A14, A15	
Festlegen der Migration-Control-Parameter (wie Except-File)		
Pubset-Betrieb wieder im allgemein zugänglichen Modus		

*Anmerkung*

Die halbfett gekennzeichneten Schritte C5, D2, D9 sollten bei der Wahl der Vorgehensweise (in-place-Anpassung des Backup-Archivs oder neues Backup-Archiv) besonders beachtet werden, da ihre Durchführung schwierig ist oder Zugriffe auf die Bänder erfordert.

### 6.6.4 Umstiegsszenario 4

Umstiegsszenario 4 (Bild 14) beschreibt den schwierigsten Fall für die Bildung eines SM-Pubsets aus SF-Pubsets. Wie im Umstiegsszenario 3 gehen in den SM-Pubset mehrere SF-Pubsets ein. Im Unterschied zu dem Umstiegsszenario 3 sind den einzelnen SF-Pubsets unterschiedliche Migrationsarchive zugeordnet, die auch für SF-Pubsets genutzt werden, welche nicht in den SM-Pubset eingehen. Die SF-Pubsets verfügen neben der S2-Ebene auch über eine S1-Ebene. Auch die System-Backup-Archive haben Kopplungen zu SF-Pubsets, die nicht in den SM-Pubset eingehen. Als Sicherungsdatenträger werden neben Bändern auch Platten (z.B. Privatplatten) benutzt.

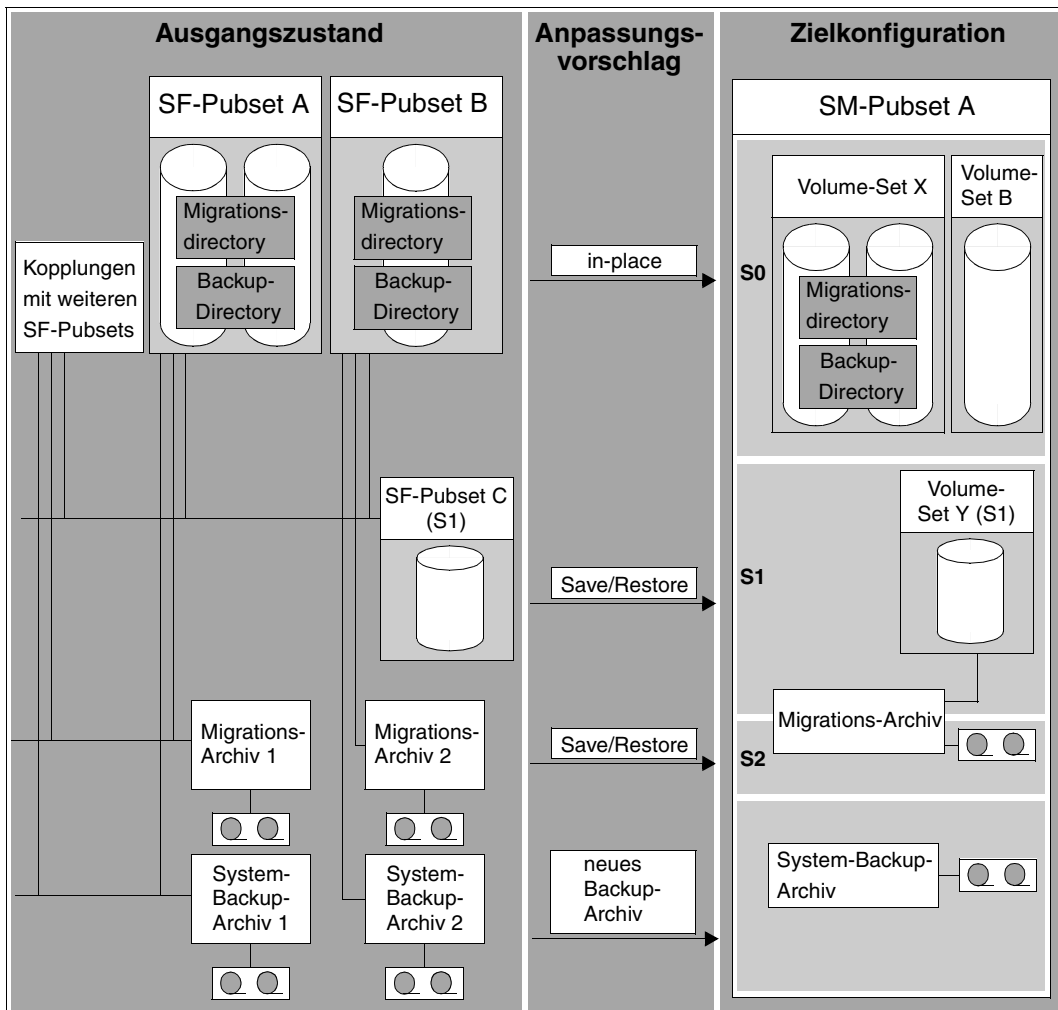


Bild 14: Umstiegsszenario 4

## Ausgangszustand und Zielkonfiguration

### 1. Verarbeitungsebene

Die S0-Ebene des SM-Pubsets ergibt sich durch Zusammenfassung der S0-Ebenen der einzelnen SF-Pubsets. Der darauf befindliche Dateibestand ändert sich dabei nicht.

### 2. Hintergrundebenen

Der auf der S1-Ebene bzw der S2-Ebene liegende Dateibestand der SF-Pubsets liegt nach der Bildung des SM-Pubsets auf der S1-Ebene bzw. S2-Ebene des SM-Pubsets.

### 3. Backup-Konfiguration

Für den SM-Pubset wird ein neues System-Backup-Archiv angelegt. In dieses werden die Sicherungen der SF-Pubsets eingebracht, die unmittelbar vor der Bildung des SM-Pubsets erstellt werden. Zusätzlich wird aus den bisherigen System-Backup-Archiven ein Teil des Sicherungsbestands übernommen. Den Umfang legt der Systembetreiber in Absprache mit den Benutzern fest. Die Archive werden als private Backup-Archive für SF-Pubsets beibehalten. Damit bleibt auch der in ihnen enthaltene Sicherungsbestand, der nicht in das System-Backup-Archiv des SM-Pubsets transferiert wurde, weiterhin zugreifbar.

## Vorgeschlagenes Vorgehen

Maßnahmen vor der Anpassung der Verarbeitungsebene	Bemerkungen
Bereinigung von Kollisionen und Überlängen von Pfadnamen (mithilfe von SMPGEN-Checks)	betrifft den Systembetreiber und die einzelnen Benutzer; dieser Schritt beinhaltet auch A1 bzw. B1
Berücksichtigung von geänderten Pfadnamen in Kommandoprozeduren, Programmen, GUARDS-Katalogen und sonstigen Benutzerdatenstrukturen; Zuordnen von geeigneten Dateiattributen in den Anweisungen zum Anlegen von Dateien	betrifft den Systembetreiber und die einzelnen Benutzer
Anpassen von Kommandoprozeduren und Programmen, die Kommandos/Anweisungen enthalten, die für SM-Pubsets nicht zulässig sind	betrifft den Systembetreiber und die einzelnen Benutzer
Anpassung der Default-Catids (in allen potenziellen Home-Pubsets)	betrifft nur den Systembetreiber

Maßnahmen vor der Anpassung der Verarbeitungsebene (Forts.)	Bemerkungen	
Anpassung der privaten Backup-Archive	betrifft den Systembetreuer und die einzelnen Benutzer; das Vorgehen ist abhängig von den speziellen Voraussetzungen	
<b>ab jetzt: Pubset-Betrieb im Modus ACCESS-CONTROLLED!</b>		
SMPGEN-Checks durch den Systembetreuer: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET explizit im Prüfmodus (OPERATIONAL-MODE=*CHECK-NAME-CONSISTENCY)		
Attributierung bestehender Dateien: Dateien, welche auf SF-Pubsets aus hochperformanten Platten liegen, sollten PERFORMANCE=HIGH oder VERY-HIGH erhalten; das Verfügbarkeitsattribut wird automatisch durch SMPGEN berücksichtigt.	kann für alle Benutzerdateien zentral durch den Systembetreuer vorgenommen werden	
generelle Empfehlung: Vollsicherung der SF-Pubsets mit HSMS inklusive der Hintergrundebenen	D1, <b>D2</b>	
ggf. FDDRL-Sicherung der SF-Pubsets (schnelles Rücksetzen)		
Anpassung der Migrationsarchive (Directories)	in-place	A2, <b>A3, A4</b> , A5, A6, A7 (optional), A8, A9
	Save/Restore	<b>B2</b> (ist in <b>D2</b> enthalten, s.o.), B3 (optional)
Anpassung der System-Backup-Archive (Directories)	in-place	C1 ,C4, C5, C6, C7, C8, C10;
	neues Archiv	(D1, <b>D2</b> s.o.), D3, D4, D7
Bereinigung der HSMS-Definitionen für die SF-Pubsets; ggf. erst nach Bildung des SM-Pubsets (Rücksetzen)	A11, D6	



Anpassung der Verarbeitungsebene	Bemerkungen	
Umbenennung des SF-Pubsets, dessen Kennung sich auf den SM-Pubset übertragen soll (PVSREN). Die neue Pubset-Kennung (künftige Volume-Set-Kennung) sollte nicht länger sein als die bisherige, um Überlängen von Pfadnamen zu vermeiden.		
Umwandlung des Pubsets mit SMPGEN: Aufruf von //CREATE-SYSTEM-MANAGED-PUBSET (implizit im Ausführungsmodus)		
Anpassung der von SMPGEN getroffenen Voreinstellungen		
ggf. Einrichten von Storage-Klassen und Zuweisen von Default-Storage-Klassen		
Anpassung der MRSCAT-Einträge an sonstigen Rechnern		
Maßnahmen nach der Anpassung der Verarbeitungsebene		
Pubset-Inbetriebnahme im Modus ACCESS-CONTROLLED		
Einrichten der HSMS-Umgebung auf dem SM-Pubset	A12 (=B5=C11=D8)	
Einrichten des System-Backup-Archivs, Zuordnen des Directory	in-place	C13
	neues Archiv	<b>D9</b> , D10, D11
Einrichten des Migrationsarchivs, Zuordnen des Directory	in-place	A14, A15, <b>A16</b>
Festlegen der Migration-Control-Parameter (wie Except-File)	Save/Restore	B7, B8, B10, <b>B11</b>
Pubset-Betrieb wieder im allgemein zugänglichen Modus		

*Anmerkung*

Die halbfett gekennzeichneten Schritte A3, A4, A16, B2, B11, D2, D9 sollten bei der Wahl der Vorgehensweise (Anpassung des Backup-Archivs: in-place oder neues Archiv, Anpassung des Migrationsarchivs: in-place oder durch Save/Restore) besonders beachtet werden, da ihre Durchführung schwierig ist oder Zugriffe auf die Bänder erfordert.

### 6.6.5 Umstiegsszenario 5

Umstiegsszenario 5 (Bild 15) beschreibt den Fall, dass ein SF-Pubset in einen bestehenden SM-Pubset integriert werden soll. Die Dateien des SF-Pubsets sollen entsprechend ihrer Lage auf dem SF-Pubset auf die Verarbeitungsebene, die S1-Ebene oder die S2-Ebene des SM-Pubsets gelangen. Der Sicherungsbestand des System-Backup-Archivs soll so weit möglich in das System-Backup-Archiv des SM-Pubsets übernommen werden.

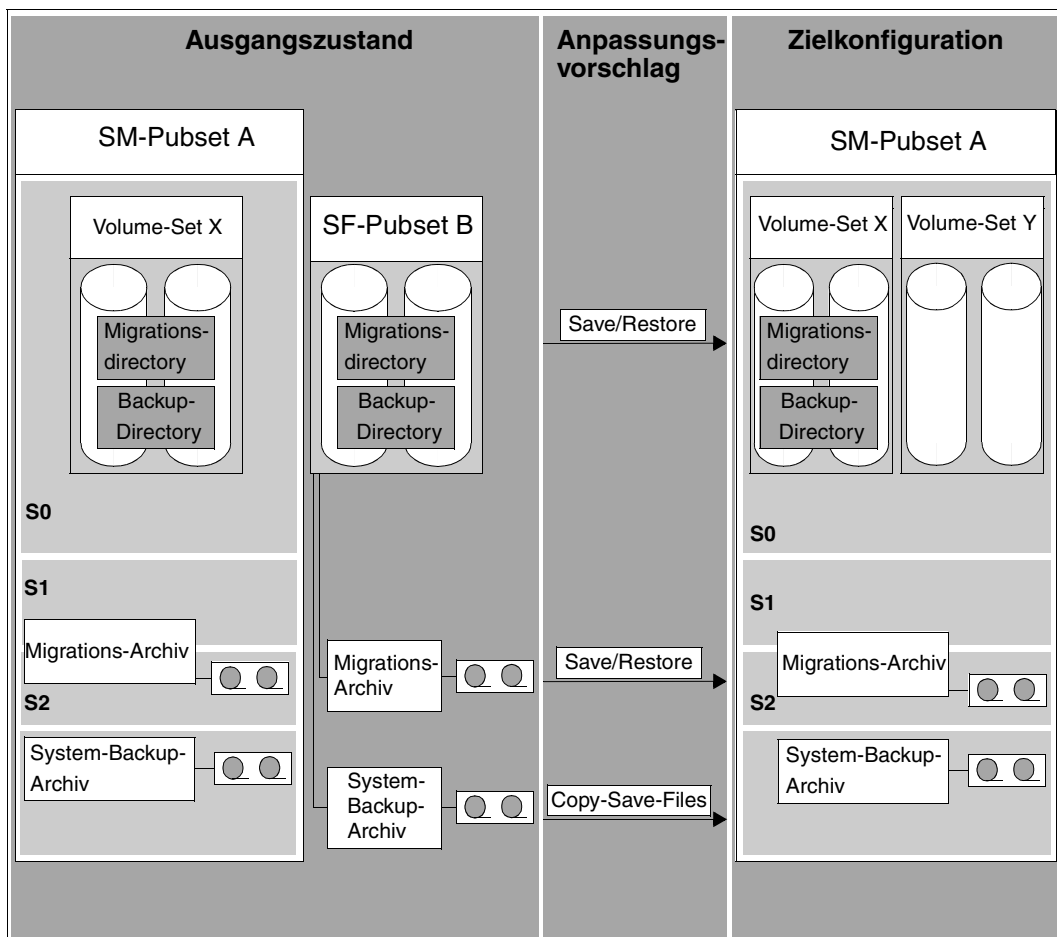


Bild 15: Migrationsszenario 5

### **Vorgeschlagenes Vorgehen**

Die in der Verarbeitungsebene und auf den Hintergrundebenen liegenden Dateien des SF-Pubsets werden gesichert. Die frei gewordenen Platten des SF-Pubsets kann der Systembetreuer verwenden, um die Kapazität des SM-Pubsets zu vergrößern. Dazu können entweder bestehende Volume-Sets durch Hinzunahme von Volumes vergrößert werden oder neue freie Volume-Sets aus den Volumes gebildet werden, die anschließend in den SM-Pubset aufgenommen werden. In beiden Fällen gehen die auf dem SF-Pubset liegenden Dateien verloren. Sie müssen auf dem SM-Pubset durch Wiedereinspielen der Sicherung wiederhergestellt werden. Für das Transferieren der auf den Hintergrundebenen befindlichen Dateien kann an Stelle von Sichern und Wiedereinspielen auch die HSMS-Funktion COPY-SAVE-FILE benutzt werden.

Zum Übertragen des Sicherungsbestands des SF-Pubsets in das System-Backup-Archiv des SM-Pubsets eignet sich die HSMS-Funktion COPY-SAVE-FILE.



---

## 7 Auskunftsfunktionen für SM-Pubsets

Wir beschränken uns auf eine Zusammenstellung von Auskunftsfunktionen, welche Informationen über die Eigenschaften von SM-Pubsets bereitstellen. Detailliertere Informationen finden sich in den jeweiligen Handbüchern.

Auskunftsfunktionen des Pubset-Managements bezüglich der Pubset-Konfiguration und Pubset-Nutzung:

- SHOW-MASTER-CATALOG-ENTRY (für Benutzer und Systembetreuer)
- SHOW-PUBSET-PARAMETERS (nur für Systembetreuer)
- SHOW-PUBSET-CONFIGURATION (für Benutzer und Systembetreuer)
- SHOW-PUBSET-SPACE-ALLOCATION (nur für Systembetreuer)
- SHOW-PUBSET-ATTRIBUTES (nur für Systembetreuer)
- SHOW-PUBSET-FILE-SERVICES (für Benutzer und Systembetreuer)

Ein großer Teil der Informationen, welche die oben aufgeführten Kommandos bereitstellen ist auch über die STAMCE-Programmschnittstelle zugänglich.

Weitere für das Management von SM-Pubsets relevante Auskunftsfunktionen:

- SHOW-USER-ATTRIBUTES
- Auskunftsfunktionen für Storage-Klassen
- Auskunftsfunktionen für die HSMS-Management-Klassen
- Auskunftsfunktionen für HSMS-Konfigurationen (SHOW-ARCHIVE,...)
- SHOW-PUBSET-USAGE (HSMS-Anweisung)
- Auskunftsfunktionen der Utility SPCCNTRL
- Auskunftsfunktionen für GUARDS-Schutzprofile
- Auskunftsfunktionen auf physikalischer Ebene (z.B. für Partitionen im Globalspeicher, DRV-Konfigurationen.).



---

## 8 Änderung der Merkmale von SM-Pubsets

In diesem Abschnitt werden die wichtigsten Möglichkeiten beschrieben, die zum Ändern der Merkmale von SM-Pubsets zur Verfügung stehen

### 8.1 Umbenennen von Pubsets und Volume-Sets

Die Utility PVSREN ermöglicht SM-Pubsets und Volume-Sets, die SM-Pubsets angehören, umzubenennen. Das Umbenennen freier Volume-Sets wird nicht unterstützt.

Die Nutzung von PVSREN erfordert bestimmte Voraussetzungen (z.B. bezüglich Pubset-Zustand, Konfigurationszustand der Volume-Sets, exklusive Belegbarkeit der Volumes, Anzahl der MRSCAT-Einträge). Sie sind im Detail in dem Handbuch „Dienstprogramme“ [1] dargestellt. Insbesondere ist zu beachten, dass die neue Kennung des Pubsets oder des Volume-Sets nicht zu Kollisionen mit den Kennungen anderer Pubsets oder Volume-Sets führt. Dies ist für alle Rechner sicherzustellen, an denen der umbenannte Pubset bzw. der Pubset mit dem umbenannten Volume-Set später in Betrieb genommen werden soll.

Die Grundfunktionalität von PVSREN besteht in der konsistenten Modifikation der internen Pubset-Datenstrukturen, auf die der Anwender keinen direkten Zugriff hat (z.B. SVL, Pubset-Konfigurationsdatei, Dateikatalog). Als zusätzlichen Komfort bietet PVSREN in gewissem Umfang auch Unterstützung für die Anpassung des Pubset-Umfelds, die der Systembetreuer auch direkt, z.B. mit Hilfe von Kommandos vornehmen könnte (Anpassung von MRSCAT-Einträgen, Benutzerkatalogen).

### 8.1.1 Pubset-Umbenennung

Die meisten Aspekte, die bei der Umbenennung eines Pubsets zu berücksichtigen sind, sind für SF-Pubsets und SM-Pubsets gleich:

- Durch die Umbenennung eines Pubsets ändern sich die Pfadnamen von Dateien und Jobvariablen. Verlängert sich die Kennung des Pubsets, so muss sichergestellt sein, dass kein Pfadname anschließend die maximal zulässige Länge überschreitet.
- Um nach der Umbenennung noch auf die migrierten Dateien eines Pubsets zugreifen zu können, muss in den Einträgen des Migrations-Directories die bisherige Pubset-Kennung durch die neue Pubset-Kennung ersetzt werden. Wenn sich durch die Umbenennung eines Pubsets der Pfadname des Migrations-Directories ändert (was bei SM-Pubsets immer der Fall ist), muss in den Datenträgerzuordnungen des MAREN-Katalogs der Pfadname für das Directory ausgetauscht werden.
- Die Maßnahmen für die Anpassung der Backup-Archive sind analog, wobei jedoch der Austausch der Catid in dem Directory nicht zwingend erforderlich ist. Er ist aber dringend zu empfehlen, da sonst der Zugriff auf Sicherungen, die vor der Umbenennung des Pubsets erstellt wurden, kompliziert wird (Adressierung der Sicherungen mit der bisherigen CATID; RESTORE mit RENAME).
- Bestehende Referenzen auf die Pubset-Kennung müssen angepasst werden. Dies betrifft u.a. die MRSCAT-Einträge, die Default-Catids in den Benutzereinträgen potentieller Home-Pubsets, Programmnamen in GUARDS-Profilen, IMON-Deklarationen, sowie Kommandoprozeduren und Benutzerprogramme, in denen Dateien mit vollständigem Pfadnamen (d.h. expliziter Catid-Angabe) adressiert werden.
- Unter den Datenstrukturen, in denen die Kennung des Pubsets vermerkt ist, sind die MRSCAT-Einträge von besonderer Wichtigkeit. Bei SM-Pubsets sind nicht nur die MRSCAT-Einträge des Pubsets, sondern auch die der Volume-Sets zu berücksichtigen. Sie müssen entfernt werden, da sie nach der Pubset-Umbenennung mit dem Pubset-Eintrag nicht mehr konsistent sind. An dem Rechner, an dem PVSREN die Umbenennung ausführt, erfolgt die Anpassung der MRSCAT-Einträge automatisch durch PVSREN, an allen anderen Rechnern (z.B. in einem Shared-Pubset-Verbund) muss sie vom Systembetreuer explizit vorgenommen werden. Erfolgt sie an einem Rechner nicht, scheitert an diesem die nächste Inbetriebnahme.



## 8.1.2 Volume-Set-Umbenennung

Das Umbenennen von Volume-Sets und die Berücksichtigung im Pubset-Umfeld ist spezifisch für SM-Pubsets.

- Wird der Control-Volume-Set umbenannt, so muss an allen Rechnern, an denen für den zugehörigen SM-Pubset ein MRSCAT-Eintrag eingerichtet ist, in diesem der Name des Control-Volume-Sets ausgetauscht werden (Kommando MODIFY-MASTER-CATALOG-ENTRY). An dem Rechner, an dem PVSREN abläuft, erfolgt dies automatisch, an sonstigen Rechnern (z.B. in einem Shared-Pubset-Verbund) muss der Systembetreuer die Anpassung explizit durchführen. Erfolgt sie an einem bestimmten Rechner nicht, scheitert dort die nächste Inbetriebnahme des Pubsets.
- Durch das Umbenennen eines Volume-Sets werden die ihm zugeordneten MRSCAT-Einträge ungültig. An dem Rechner, an dem PVSREN abläuft, wird der ungültige MRSCAT-Eintrag automatisch durch PVSREN gelöscht. An sonstigen Rechnern werden diese Einträge automatisch bei der nächsten Inbetriebnahme des Pubsets entfernt, falls sie der Systembetreuer nicht bereits zuvor durch das Kommando REMOVE-MASTER-CATALOG-ENTRY explizit gelöscht hat.
- Bei der Umbenennung eines S1-Volume-Sets muss die über die HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS getroffene Zuordnung des S1-Volume-Sets angepasst werden.
- Volume-Set-Listen, die den Volume-Set enthalten, der umbenannt wird, sollten angepasst werden.
- In bestimmten Fällen versucht HSMS beim Recall, eine Datei auf den Volume-Set zurück zu bringen, auf dem sie sich vor der Verdrängung befand (z.B. eine Datei mit dem Attribut S0-MIGRATION=\*FORBIDDEN). Wurde dieser Volume-Set zwischenzeitlich umbenannt, kann er nicht mehr als Herkunfts-Volume-Set der Datei identifiziert werden. Dann ermittelt HSMS den Ablageort wie bei einem Recall mit der Angabe NEW-DATA-SUPPORT=\*BEST-VOLUME-SET. Entsprechendes gilt für das Sichern und Wiedereinspielen.
- Die Umbenennung eines Volume-Sets hat keinen Einfluss auf die Pfadnamen von Dateien/Jobvariablen. Im Vergleich zur Änderung der Pubset-Kennung sind daher die Auswirkungen auf Kommandoprozeduren und Benutzerprogramme geringer. Anpassungen sind beispielsweise in Prozeduren notwendig, welche Dateien mit physikalischer Allokierung auf einem bestimmten Volume-Set anlegen.

## 8.2 Hinzunahme und Entfernen von Volume-Sets

Die Plattenkonfiguration eines SM-Pubsets lässt sich ändern, indem komplette Volume-Sets hinzugenommen oder entfernt werden oder indem dem SM-Pubset bereits angehörende Volume-Sets durch Hinzufügen oder Entfernen einzelner Volumes vergrößert oder verkleinert werden.

Die Hinzunahme von Volume-Sets wird sowohl durch SIR im Rahmen der statischen Pubset-Pflege als auch durch Funktionen der dynamischen Pubset-Rekonfiguration unterstützt. Für die statische Pubset-Pflege muss der Pubset außer Betrieb sein, während die dynamische Pubset-Rekonfiguration nur für in Betrieb befindliche Pubsets möglich ist. Das Entfernen von Volume-Sets kann nur mit den Funktionen der dynamischen Pubset-Rekonfiguration erfolgen. Wenn sich die Volume-Set-Konfiguration eines SM-Pubsets ändert, werden dadurch auch globale Pubset-Merkmale wie z.B. die maximale I/O-Länge oder das den Benutzern sichtbare Service-Spektrum des Pubsets tangiert.

### 8.2.1 Pubset-Rekonfiguration und Volume-Set-Konfigurationszustände

Die Pubset-Rekonfiguration auf Volume-Set-Ebene muss im Zusammenhang mit den Änderungen des Konfigurationszustands eines Volume-Sets betrachtet werden. Änderungen des Konfigurationszustands können entweder durch den Systembetreuer oder durch das System verursacht sein, das bei dem Auftreten bestimmter Fehlersituationen automatisch Übergänge in die Zustände „in hold“ oder „defective“ veranlasst. Wir geben zunächst einen Überblick über die bei den Rekonfigurationsmaßnahmen auftretenden Zustandsübergänge. Sie gelten nicht für den Control-Volume-Set. Dieser muss immer im Zustand „normal use“ sein und kann auch nicht aus dem SM-Pubset entfernt werden. In [Abschnitt „Hinzunahme eines Volume-Sets im laufenden Pubset-Betrieb“ auf Seite 150](#) bis [Abschnitt „Pubset-Rekonfiguration bei Problemen bei der Pubset-Inbetriebnahme“ auf Seite 159](#) werden die Funktionen der Pubset-Rekonfiguration im Detail behandelt.

#### Ausgangszustand nach der Pubset-Generierung

Nachdem ein SM-Pubset durch SIR oder SMPGEN erzeugt worden ist, befinden sich alle ihm zugehörigen Volume-Sets in ihrem normalen Nutzungszustand, der als „normal use“ bezeichnet wird.

### **Eintragen eines neuen Volume-Sets in die Konfigurationsdatei des SM-Pubsets**

Durch das Kommando MODIFY-PUBSET-DEFINITION-FILE, Operand VOLUME-SET-ENTRY= \*ADD kann für einen Volume-Set ein neuer Eintrag in der Konfigurationsdatei des SM-Pubsets erzeugt werden. Der Volume-Set ist dann aus Sicht des SM-Pubsets zwar definiert, es ist ihm aber keine physikalische Konfiguration zugeordnet. Dieser Zustand wird als „defined only“ bezeichnet. Er hat die Rolle eines Zwischenzustands, der im Allgemeinen die Phase zwischen zwei direkt aufeinander folgenden Schritten der Pubset-Rekonfiguration überbrückt.

### **Aufnahme eines freien Volume-Sets**

Einem Volume-Set im Zustand „defined only“ können durch das Kommando MODIFY-PUBSET-PROCESSING, Operand VOLUME-SET-SUPPORT=\*ADD physikalische Datenträger zugewiesen werden. Sie müssen dazu bereits in einem freien (und leeren) Volume-Set vorkonfiguriert sein. Ein solcher kann z.B. mit SIR erzeugt werden. Durch die Zuordnung der physikalischen Konfiguration kommt der Volume-Set aus Sicht des SM-Pubsets in den Zustand „normal use“.

### **Übergänge in den Zustand „in hold“**

Ein im Zustand „normal use“ befindlicher Volume-Set kann entweder implizit durch das System (beim Auftreten von Problemen) oder explizit von dem Systembetreuer mit dem Kommando MODIFY-PUBSET-RESTRICTIONS, Operand RESTRICTION=\*PROC-STATE(MODE=\*HOLD) in den Zustand „in hold“ versetzt werden. Der Zustand „in hold“ ergibt sich auch dann, wenn der Systembetreuer bei der Inbetriebnahme des Pubsets angibt, dass der Volume-Set in den Zustand „in hold“ gesetzt werden soll (Kommando IMPORT-PUBSET, Operand IN-HOLD-VOLUME-SET). Der Zustand „in hold“ ist dafür vorgesehen, dass der SM-Pubset nutzbar bleibt, auch wenn einzelne seiner Volume-Sets nicht funktionsunfähig sind, aber trotzdem nicht zwangsentfernt werden sollen (s.u.), weil ihre Funktionalität nur vorübergehend gestört ist.

### Übergänge von „in hold“ nach „normal use“

Durch das Kommando MODIFY-PUBSET-RESTRICTIONS, Operand RESTRICTION=\*PROC-STATE(MODE=\*RESTART) kann der Zustand „in hold“ aufgehoben werden, wenn der Volume-Set wieder intakt ist. Treten bei der Durchführung des Kommandos Probleme auf, kann als Folge der Volume-Set weiterhin im Zustand „in hold“ bleiben oder in den Zustand „defective“ übergehen. Bei erfolgreicher Ausführung wird der Zustand „normal use“ eingenommen. Bei der Inbetriebnahme eines SM-Pubsets setzt das System voraus, dass alle Volume-Sets des SM-Pubsets funktionsfähig sind, mit Ausnahme von denen, für die der Systembetreuer explizit angibt, dass sie anschließend im Zustand „in hold“ sein oder bei der Inbetriebnahme zwangsentfernt werden sollen (Kommando IMPORT-PUBSET, Operanden DEFECT-VOLUME-SET oder IN-HOLD-VOLUME-SET). Sind außer den angegebenen Volume-Sets auch weitere nicht funktionsfähig, wird die Inbetriebnahme abgebrochen. Sie muss von dem Systembetreuer ggf. mit geänderten Angaben wiederholt werden. Durch eine erfolgreiche Pubset-Inbetriebnahme wird für Volume-Sets, die zuvor im Zustand „in hold“ waren, jetzt aber wieder intakt sind, der Zustand „in hold“ aufgehoben, d.h. sie sind anschließend im Zustand „normal use“

### Übergänge in den Zustand „defective“

Der Zustand „defective“ wird durch das System herbeigeführt, wenn es irreparable Inkonsistenzen in den Volume-Set-Metadaten erkennt. Diese können durch Software- oder Hardware-Probleme bedingt sein. Für einen Volume-Set im Zustand „defective“ ist die einzig mögliche Maßnahme das Zwangsentfernen der physikalischen Konfiguration aus dem SM-Pubset.

### Entfernen der physikalischen Volume-Set-Konfiguration

Das physikalische Entfernen eines Volume-Sets kann auf folgende Arten geschehen:

- Die physikalische Konfiguration eines Volume-Sets im Zustand „normal use“, auf dem sich nur der volume-set-spezifische Dateikatalog befindet, wird durch das Kommando MODIFY-PUBSET-PROCESSING, Operand VOLUME-SET-SUPPORT=\*REMOVE(CONDITION=\*EMPTY-VOLUME-SET) aus dem SM-Pubset entfernt. Sie bildet anschließend einen freien (und leeren) Volume-Set.

- Die physikalische Konfiguration eines Volume-Sets im Zustand „defective“ oder „in hold“ kann durch das Kommando MODIFY-PUBSET-PROCESSING, Operand VOLUME-SET-SUPPORT=\*REMOVE(CONDITION=\*VOLUME-SET-DEFECTS) aus dem SM-Pubset zwangsentfernt werden, auch wenn sich auf dem Volume-Set noch Dateien darauf befinden. Das System hinterlegt die Namen dieser Dateien, die dann auch aus dem SM-Pubset verschwinden, in einer Datei. Die physikalische Konfiguration des entfernten Volume-Sets bildet anschließend einen freien, aber nicht leeren Volume-Set. Dessen weitere Verwendbarkeit hängt vom Zustand der zugehörigen Volumes ab. Sind sie noch intakt, kann der ausgegliederte Volume-Set mit Hilfe einer speziellen Inbetriebnahme in einen SF-Pubset umgewandelt werden. Dadurch werden auch die darauf befindlichen Dateien wieder zugreifbar. Die Aufnahme eines freien, aber nicht leeren Volume-Set in einen SM-Pubset ist nicht möglich.
- Das zwangsweise Entfernen eines Volume-Sets kann auch im Rahmen der Inbetriebnahme des SM-Pubsets erfolgen. Der Systembetreuer muss die Volume-Sets, deren physikalische Konfiguration zwangsweise entfernt werden soll, im Kommando IMPORT-PUBSET mit dem Operanden DEFECT-VOLUME-SET spezifizieren. Im Unterschied zum Zwangsentfernen im laufenden Pubset-Betrieb wird dabei nicht der Ausgangszustand „defective“ oder „in hold“ vorausgesetzt.

Beim Entfernen der physikalischen Konfiguration eines Volume-Sets bleibt der Volume-Set-Eintrag in der Konfigurationsdatei erhalten, d.h. der Volume-Set ist (aus Sicht des Pubsets) im Zustand „defined only“.

### **Entfernen eines Volume-Set-Eintrags aus der Konfigurationsdatei des SM-Pubsets**

Durch das Kommando MODIFY-PUBSET-DEFINITION-FILE, Operand VOLUME-SET-ENTRY=\*REMOVE wird der Eintrag eines Volume-Sets aus der Konfigurationsdatei des SM-Pubsets entfernt. Der Volume-Set muss dazu im Zustand „defined only“ sein.

### 8.2.2 Hinzunahme eines Volume-Sets im laufenden Pubset-Betrieb

Für die Hinzunahme eines Volume-Sets zu einem SM-Pubset mithilfe der dynamischen Pubset-Rekonfiguration sind mehrere Schritte erforderlich.

#### Eintragen eines Volume-Sets in die Konfigurationsdatei

Mit dem Kommando `MODIFY-PUBSET-DEFINITION-FILE`, Operand `VOLUME-SET-ENTRY= *ADD(...)` wird ein neuer Volume-Set in die Konfigurationsdatei des SM-Pubsets eingetragen, wobei ihm zunächst keine physikalische Volume-Set-Konfiguration zugewiesen ist. Dieser Schritt wird als logische Hinzunahme eines Volume-Sets bezeichnet. Der Volume-Set kommt dadurch in den Zustand „defined only“.

Beim Eintragen eines Volume-Sets in die Pubset-Konfigurationsdatei wird an allen Rechnern, an denen der Pubset zu diesem Zeitpunkt in Betrieb ist, ein MRSCAT-Eintrag für den Volume-Set angelegt. Voraussetzung ist, dass nicht bereits ein MRSCAT-Eintrag gleicher Kennung existiert, der anderweitig genutzt wird. Als anderweitig benutzt gilt ein MRSCAT-Eintrag dann, wenn er nicht vom passenden Typ (d.h. vom Typ SF- oder SM-Pubset) ist, oder wenn er einem Volume-Set eines anderen SM-Pubsets zugeordnet ist. An den Rechnern, an denen der Pubset zu diesem Zeitpunkt in Betrieb ist, wird damit automatisch durch das System sichergestellt, dass sich keine Namenskollisionen mit der übrigen Konfiguration ergeben. An allen sonstigen Rechnern, an denen die Inbetriebnahme des Pubsets möglich sein soll, muss der Systembetreuer dafür explizit Sorge tragen.

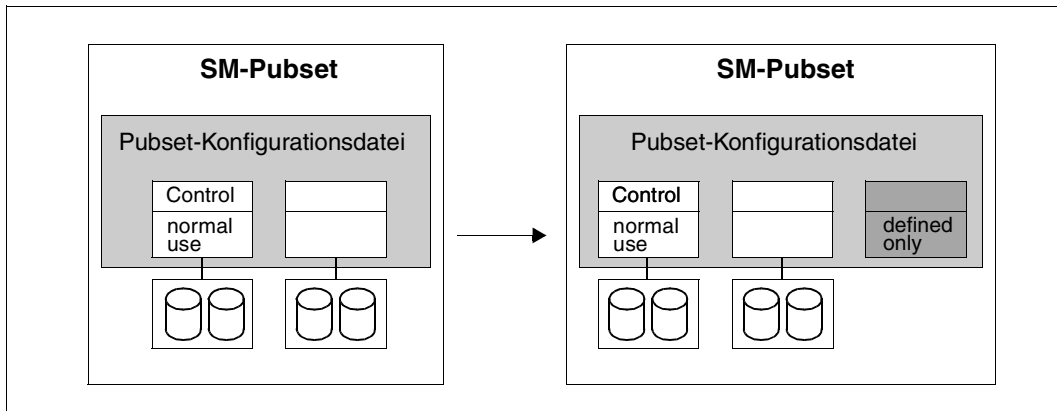


Bild 16: Eintragen eines Volume-Sets in die Konfigurationsdatei

Im Zustand „defined only“ sind nur die Volume-Set-Merkmale definiert, die in der Konfigurationsdatei hinterlegt werden. Zu diesen gehören das Performance-Profil, das Verfügbarkeitsprofil, die Nutzungsart, die Nutzungsrestriktionen, die Sättigungsschwellwerte, die statische Cache-Konfiguration, etc. Sie werden beim Eintragen des Volume-Sets in die Konfigurationsdatei teils durch den Systembetreuer festgelegt, teils mit Standardwerten initialisiert. Anschließend können sie durch den Systembetreuer abgefragt und (im Allgemeinen) auch verändert werden. Die Standardwerte entsprechen weitgehend denen, welche Volume-Sets beim Erzeugen eines SM-Pubsets durch SIR zugewiesen bekommen. Die unmittelbar aus der physikalischen Volume-Set-Konfiguration resultierenden Eigenschaften Format, Größe der Allocation-Unit, Menge der zugehörigen Volumes sind im Zustand „defined only“ (aus Pubset-Sicht) unbestimmt.

Der Zustand „defined only“ ist ein Zwischenzustand, der sich zum Durchführen bestimmter Vorbereitungsmaßnahmen für die spätere Nutzung des Volume-Sets eignet. Sollen beispielsweise auf einem Volume-Set Dateien nur durch physikalische Allokierung angelegt werden können, kann er bereits im Zustand „defined only“ mit der entsprechenden Nutzungsrestriktion belegt werden.

### Hinzunahme eines freien Volume-Sets

Ein im Zustand „defined only“ befindlicher Volume-Set wird mit einer physikalischen Konfiguration ausgestattet, indem ihm durch das Kommando `MODIFY-PUBSET-PROCESSING`, Operand `VOLUME-SET-SUPPORT=*ADD(...)` ein kennungsgleicher freier (und leerer) Volume-Set zugewiesen wird. Wir bezeichnen diesen Schritt als physikalische Hinzunahme eines Volume-Sets.

Ein freier Volume-Set ist ein Volume-Set, der keinem SM-Pubset angehört. Er besteht aus mindestens einem Volume und wird durch seine Kennung, seine Volres, die übrigen ihm zugehörigen Volumes, sein Format und die Größe der Allocation-Unit charakterisiert. Er enthält abgesehen von Sonderfällen, die später betrachtet werden, außer der volume-set-spezifischen Katalogdatei keine weiteren Dateien. Die Volume-Set-Merkmale, die nicht in einem Volume-Set selbst, sondern in der Konfigurationsdatei des SM-Pubsets hinterlegt sind (wie das Performance- und Verfügbarkeitsprofil, die Nutzungsart, Nutzungsrestriktionen, Sättigungsschwellwerte), sind für einen freien Volume-Set undefiniert. Freie leere Volume-Sets können durch SIR erzeugt werden (Anweisung `BEGIN-VOLUME-SET-DECLARATION`, Operand `ACTION=*INSTALL` ohne vorhergehende `DECLARE-PUBSET`-Anweisung). Sie entstehen auch dann, wenn leere Volume-Sets physikalisch aus einem SM-Pubset entfernt werden. Freie Volume-Sets mit „großen“ Volumes können nur in `LARGE-OBJECT` Pubsets aufgenommen werden.

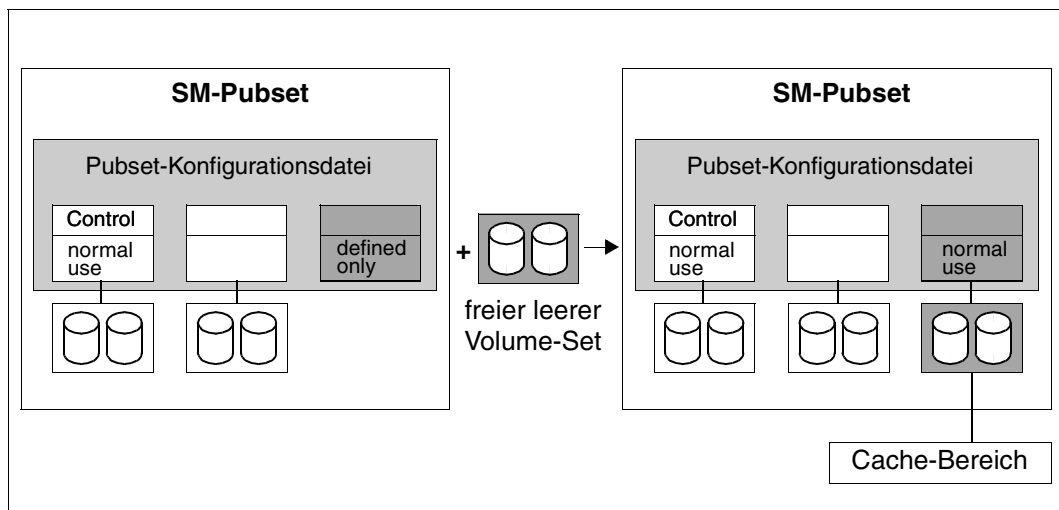


Bild 17: Aufnahme eines freien Volume-Sets in den SM-Pubset

Bei der physikalischen Hinzunahme eines Volume-Sets werden alle seine Volumes an jedem Rechner konnektiert, an dem der SM-Pubset zu diesem Zeitpunkt in Betrieb ist. Die zu dem Volume-Set gehörenden Volumes werden dabei aus dem Volume-Katalog ermittelt, der auf der Volres hinterlegt ist. Die VSN der Volres ergibt sich aus der Kennung des Volume-Sets.

Soll ein Volume-Set entsprechend der in der Konfigurationsdatei hinterlegten Festlegung mit einem Cache betrieben werden (statische Cache-Attribute des Volume-Sets), wird versucht, einen passenden Cache-Bereich bereitzustellen und diesen mit dem Volume-Set zu konnektieren. Ergeben sich dabei Probleme (z.B. Cache-Bereich nicht oder nicht in ausreichender Größe vorhanden), so entscheidet die von dem Systembetreuer für den SM-Pubset getroffene Einstellung der SIZE-TOLERANCE (siehe Kommando MODIFY-PUBSET-CACHE-ATTRIBUTES), ob die physikalische Hinzunahme des Volume-Sets trotzdem durchgeführt oder abgebrochen wird.

Nachdem der Volume-Set physikalisch in den SM-Pubset aufgenommen ist, befindet er sich im Zustand „normal use“. In diesem haben auch die durch die physikalische Konfiguration bestimmten Merkmale des Volume-Sets (aus Pubset-Sicht) definierte Werte.



### 8.2.3 Hinzunahme von Volume-Sets durch SIR

Die Hinzunahme von Volume-Sets zu einem SM-Pubset wird durch die SIR-Anweisung DECLARE-PUBSET zusammen mit den Anweisungen BEGIN / END-VOLUME-SET-DECLARATION ermöglicht. Im Vergleich zur dynamischen Pubset-Rekonfiguration ist eine stärkere Bündelung von Einzelschritten möglich (Bildung des Volume-Sets aus den Volumes, logische und physikalische Hinzunahme des Volume-Sets). Dabei werden verschiedene Anwendungsfälle unterstützt:

#### Der hinzuzunehmende Volume-Set existiert noch nicht als freier Volume-Set

Ausgangsmaterial sind einzelne Volumes, die noch nicht oder nicht passend formatiert sein müssen. Durch die folgenden SIR-Anweisungen wird aus ihnen der Volume-Set gebildet, der in den SM-Pubset aufgenommen wird:

```
//DECLARE-PUBSET  PUBSET-TYPE=*S-M(ACTION=*EXTEND(...))
...
//BEGIN-VOLUME-SET-DECLARATION  ACTION=*INSTALL(...), ...
//CREATE-VOLUME  ...
...
//END-VOLUME-SET-DECLARATION
...
```

#### Ein bereits existierender freier Volume-Set wird in den SM-Pubset aufgenommen

Dieser Anwendungsfall kann mithilfe folgender SIR-Anweisungen durchgeführt werden:

```
//DECLARE-PUBSET  TYPE=*S-M(ACTION=*EXTEND(...))
...
//BEGIN-VOLUME-SET-DECLARATION  ACTION=*ADD(...), ...
//END-VOLUME-SET-DECLARATION
...
```

Die Voraussetzungen und erforderlichen Randbedingungen für die Hinzunahme von Volume-Sets durch SIR sind in dem Handbuch „Systeminstallation“ [3] detailliert dargestellt. Folgende Aspekte sind zu beachten:

- Der SM-Pubset, in den ein neuer Volume-Set hinzugenommen werden soll, darf nicht in Betrieb sein (statische Pubset-Rekonfiguration).
- Der Systembetreuer hat sicherzustellen, dass von allen Rechnern aus, an denen der Pubset in Betrieb genommen werden soll, Zugangspfade zu jedem Volume des aufzunehmenden Volume-Sets existieren. Für den Rechner, an dem SIR zum Ablauf kommt, wird dies implizit überprüft.
- Wird durch SIR ein vorgenerierter Volume-Set aufgenommen, muss dieser als freier Volume-Set zur Verfügung stehen.
- Der Systembetreuer hat sicherzustellen, dass an allen Rechnern, an denen der Pubset in Betrieb genommen wird, die Kennung des aufzunehmenden Volume-Sets nicht mit den Kennungen anderer SF-Pubsets, SM-Pubsets und Volume-Sets kollidiert. An dem Rechner, an dem SIR zum Ablauf kommt, wird dies automatisch überprüft, da für den Volume-Set ein MRSCAT-Eintrag angelegt wird. Dies ist aber nur möglich, wenn nicht bereits ein MRSCAT-Eintrag existiert, dessen Kennung mit der Kennung des Volume-Sets übereinstimmt. Andernfalls wird die Hinzunahme des Volume-Sets zurückgewiesen. An sonstigen Rechnern, an denen es möglich sein soll, den Pubset in Betrieb zu nehmen, muss der Systembetreuer explizit dafür sorgen, dass die Kennung des Volume-Sets nicht zu Kollisionen führt.
- Bei der Aufnahme eines Volume-Sets in den SM-Pubset werden dem Volume-Set auch die Merkmale zugeordnet, die nicht bereits durch seine physikalische Volume-Konfiguration bestimmt sind, wie Performance-Profil, Verfügbarkeitsprofil, Nutzungsart, Nutzungsrestriktionen, Einstellungen der Sättigungsschwellwerte, Festlegung der Cache-Konfiguration, etc. Während der Systembetreuer das Performance- und Verfügbarkeitsprofil sowie die Nutzungsart an der SIR-Oberfläche vorgeben kann (Operand VOLUME-SET-ATTRIBUTES), werden die restlichen dieser Volume-Set-Merkmale durch Zuweisung von Vorgabewerten bestimmt. Die Voreinstellungen sind analog den Initialwerten nach der Generierung eines SM-Pubsets durch SIR. Sie lassen sich nach der Hinzunahme des Volume-Sets durch die Kommandos der dynamischen Pubset-Rekonfiguration für konkrete Anforderungen anpassen.

## 8.2.4 Entfernen von Volume-Sets im laufenden Pubset-Betrieb

Mit Ausnahme des Control-Volume-Sets können die Volume-Sets aus diesem SM-Pubset entfernt werden. Wie die Aufnahme eines Volume-Sets erfordert dies mehrere Schritte.

### 1. Physikalisches Entfernen eines Volume-Sets

Mithilfe des Kommandos MODIFY-PUBSET-PROCESSING, Operand VOLUME-SET-SUPPORT= \*REMOVE(...) kann ein im Zustand „normal use“ befindlicher Volume-Set physikalisch aus dem SM-Pubset entfernt werden. Im Fall, dass der Volume-Set mit einem Cache-Bereich konnektiert ist, wird auch dieser freigegeben. Anschließend ist der Volume-Set (aus Sicht des Pubsets) im Zustand „defined only“, d.h. die Pubset-Konfigurationsdatei enthält für ihn weiterhin einen Eintrag, er verfügt aber im Pubset nicht über eine physikalische Konfiguration. Das Format und die Größe der Allocation-Unit sind für den Volume-Set (aus Pubset-Sicht) anschließend undefiniert, die in der Konfigurationsdatei hinterlegten Volume-Set-Merkmale (wie Performance- und Verfügbarkeitsprofil, Sättigungsschwellwerte) bleiben unverändert bestehen. Durch das Entfernen eines Volume-Sets kann sich der Umfang der durch einen SM-Pubset gebotenen Services reduzieren. Insbesondere ist darauf zu achten, dass das Zurückholen von auf Hintergrundebenen liegenden Dateien beeinträchtigt werden kann (z.B. wenn auf Hintergrundebenen liegende Dateien ein Format haben, das anschließend in der Verarbeitungsebene nicht mehr erfüllbar ist, oder wenn der Volume-Set, der entfernt wird, Herkunftspubset von Dateien mit dem Attribut S0-MIGRATION=\*FORBIDDEN ist). Für das physikalische Entfernen eines Volume-Sets stehen verschiedene Modi zur Verfügung:

#### a) regulärer Modus

Im regulären Modus (Operand CONDITION=\*EMPTY-VOLUME-SET) muss der zu entfernende Volume-Set leer sein. Leer bedeutet dabei, dass sich auf ihm außer dem volume-set-spezifischen Dateikatalog keine weiteren Dateien befinden. Zusätzlich muss der Volume-Set zuvor durch den Systembetreuer mit der Sperre gegen Neuallokierungen versehen worden sein (Kommando MODIFY-PUBSET-RESTRICTIONS, Operand RESTRICTION=\*NEW-FILE-ALLOCATION(MODE=\*NOT-ALLOWED). Diese schützt davor, dass während des Herausnehmens des Volume-Sets neue Dateien angelegt werden. Sie ist außerdem ein Hilfsmittel für den Systembetreuer, um einen Volume-Set zu leeren. Die Vorgehensweise hierfür wird in [Abschnitt „Leeren eines Volume-Sets“ auf Seite 159](#) detaillierter betrachtet. Nachdem ein Volume-Set im regulären Modus physikalisch entfernt wurde, bildet er (außerhalb des SM-Pubsets) einen freien Volume-Set.

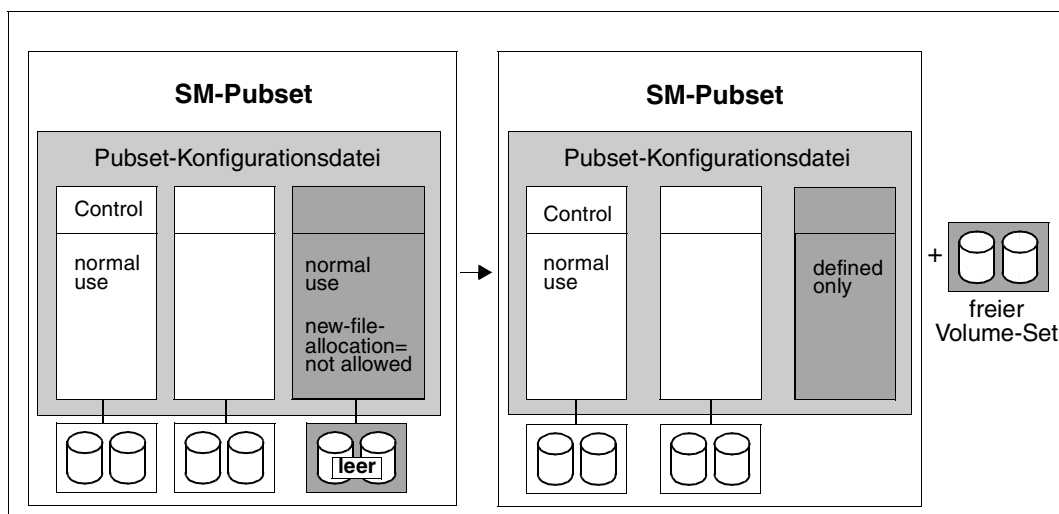


Bild 18: Physikalisches Entfernen eines leeren Volume-Sets im regulären Modus

## b) Zwangsentfernen

Nicht immer ist es dem Systembetreuer möglich, einen Volume-Set zu leeren, z.B. wenn der Volume-Set defekt ist und Plattenzugriffe nicht möglich sind. Um auch in derartigen Situationen das Entfernen von Volume-Sets zu ermöglichen, wird ein Zwangsmodus zur Verfügung gestellt. Er wird durch den Operanden `CONDITION=*VOLUME-SET-DEFECTS` spezifiziert. Die auf dem Volume-Set liegenden Dateien werden dabei implizit aus dem Pubset entfernt. Um die Benutzer darüber zu informieren und ein gezieltes Wiederherstellen der verschwundenen Dateien zu ermöglichen, werden ihre Namen in der Datei:  
`:<catid>:$TSOS.SYS.PUBSET.DEFECT.<volsetid>.<datum>.<zeit>` hinterlegt, die für den Restore mit HSMS verwendet werden kann.

Das Entfernen eines Volume-Sets im Zwangsmodus ist nur möglich, wenn er im Zustand „in hold“ oder „defective“ ist. Der Zustand „in hold“ wird automatisch durch das System beim Erkennen von bestimmten Problemen gesetzt oder kann durch den Systembetreuer durch das Kommando `MODIFYPUBSET-RESTRICTIONS`, Operand `RESTRICTION=*PROCESSING-STATE(MODE=*HOLD)` herbeigeführt werden, der Zustand „defective“ wird ausschließlich durch das System gesetzt. Nach dem Entfernen des Volume-Sets im Zwangsmodus ist der Volume-Set (aus Pubset-Sicht) im Zustand „defined only“.

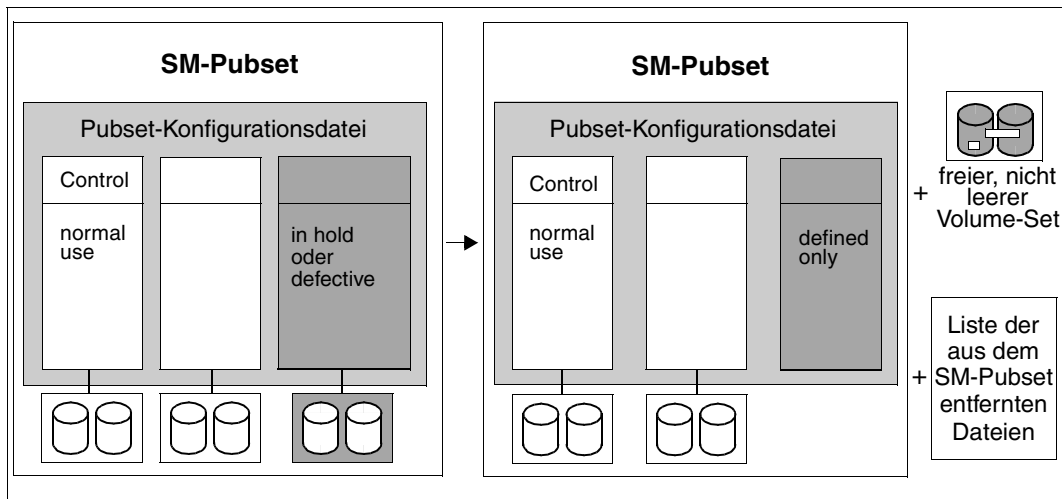


Bild 19: physikalisches Entfernen eines Volume-Sets im Zwangsmodus

Die bisher zu dem Volume-Set gehörenden Volumes bilden einen freien, aber nicht leeren Volume-Set. Für ihre weitere Verwendung gibt es verschiedene Anwendungsfälle.

- Die hauptsächliche Anwendung für die Zwangsentfernung eines Volume-Sets dürfte sein, dass ein Volume ausgefallen ist und infolge dessen ein Volume-Set nicht mehr funktionsfähig ist. In diesem Fall ist der durch die Zwangsentfernung entstandene freie Volume-Set nicht mehr als Einheit nutzbar. Die ihm zugehörigen noch intakten Volumes können nur einzeln weiterverwendet werden, um z.B. mit SIR wieder einen neuen freien (und leeren) Volume-Set zu bilden, der den ausgefallenen Volume-Set ersetzt. Die verloren gegangenen Dateien können aus Sicherungen wiedereingespielt werden.
- Eine weitere Anwendungsmöglichkeit für die Zwangsentfernung eines Volume-Sets besteht darin, dass für seine Volumes eine neue Verwendung vorgesehen wird, dass aber das Leeren des Volume-Sets entweder nicht möglich ist, oder einen zu großen Aufwand verursachen würde. Diese Situation kann sich zum Beispiel ergeben, wenn ein den Benutzern temporär zur Verfügung gestellter Work-Volume-Set wieder aufgelöst werden soll, einzelne Anwendungen aber die Bearbeitung von Dateien, die darauf liegen, nicht beenden oder die Reservierung solcher Dateien (SECURE-RESOURCE-ALLOCATION) nicht aufheben. Auch in diesem Fall ist der durch die Zwangsentfernung entstandene freie nicht leere Volume-Set nicht geeignet, um als Einheit weiterbenutzt zu werden.

- Schließlich bietet die Zwangsentfernung die Möglichkeit, einen Volume-Set „in place“ mit Mitnahme seines Dateibestands aus dem Pubset auszugliedern und anschließend anderweitig zu verwenden. Für die weitere Benutzung muss der entstandene freie nicht leere Volume-Set zuerst im Rahmen einer speziellen Inbetriebnahme in einen SF-Pubset konvertiert werden. Die „in place“- Eingliederung eines SF-Pubsets in einen bereits bestehenden SM-Pubset ist zur Zeit aber nicht möglich.

## 2. Löschen eines Volume-Set-Eintrags aus der Konfigurationsdatei

Im Zustand „defined only“ ist der Volume-Set noch in der Konfigurationsdatei des SM-Pubsets vermerkt. Beabsichtigt der Systembetreuer, dem Volume-Set in Kürze wieder eine physikalische Konfiguration zuzuordnen, kann in der Zwischenzeit der Volume-Set-Eintrag mit den entsprechenden Festlegungen für den Volume-Set beibehalten werden. Dieser Fall ist beispielsweise gegeben, wenn ein defekter Volume-Set durch einen intakten Volume-Set ausgetauscht wird. Soll jedoch ein Volume-Set dauerhaft aus einem SM-Pubset entfernt werden, so ist es empfehlenswert, auch den Eintrag in der Konfigurationsdatei zu löschen.

Dies wird durch das Kommando `MODIFY-PUBSET-DEFINITION-FILE`, Operand `VOLUME-SET-ENTRY=*REMOVE` ermöglicht. Der Volume-Set muss dazu in dem Ausgangszustand „defined only“ sein. Bei der Ausführung des Kommandos werden an allen Rechnern, an denen der Pubset zu diesem Zeitpunkt in Betrieb ist, die MRSCAT-Einträge des Volume-Sets gelöscht. Das Entfernen eines Volume-Set-Eintrags aus der Konfigurationsdatei eines SM-Pubsets wird auch als logisches Entfernen eines Volume-Sets bezeichnet.

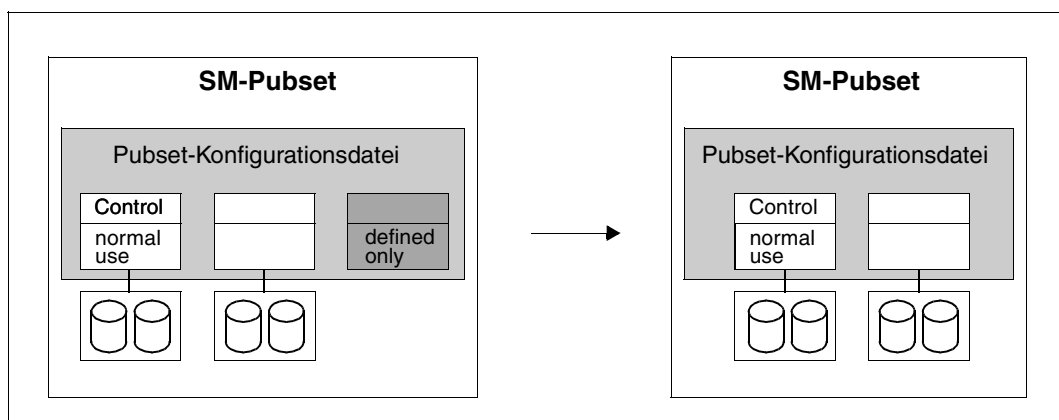


Bild 20: Logische Volume-Set-Entfernung aus einem SM-Pubset

## 8.2.5 Pubset-Rekonfiguration bei Problemen bei der Pubset-Inbetriebnahme

Die Inbetriebnahme eines SM-Pubsets umfasst im Normalfall alle zum SM-Pubset gehörenden Volume-Sets. Sind Einzelne von ihnen nicht zugreifbar, müssen sie vom Systembetreuer speziell berücksichtigt werden, da sonst die Inbetriebnahme scheitert. Es sind wiederum die beiden Fälle zu unterscheiden, ob ein Volume-Set dauerhaft beschädigt ist, oder ob temporäre Beeinträchtigungen vorliegen. Im ersten Fall ist es empfehlenswert, den betroffenen Volume-Set bei der Inbetriebnahme als defekt zu erklären und damit aus dem SM-Pubset zu entfernen. Die Möglichkeit hierzu wird im Kommando IMPORT-PUBSET durch den Operanden DEFECT-VOLUME-SET gegeben. Im zweiten Fall ist es sinnvoll, den Volume-Set im SM-Pubset zu lassen, ihn aber bei der Pubset-Inbetriebnahme in den Zustand „in hold“ zu setzen. Dies lässt sich durch den Operanden IN-HOLD-VOLUME-SET erreichen.

## 8.2.6 Leeren eines Volume-Sets

Es wird exemplarisch dargestellt, wie der Systembetreuer vorgehen kann, um einen Volume-Set eines SM-Pubsets zu leeren. Für die Durchführung dieser Maßnahmen muss der Pubset in Betrieb sein.

### 1. Einschränken neuer Benutzeraktivitäten auf dem Volume-Set

Zunächst belegt der Systembetreuer den Volume-Set durch das Kommando MODIFY-PUBSET-RESTRICTIONS mit folgenden Sperren:

- a) Sperre gegen das Anlegen neuer Dateien

```
RESTRICTION=*NEW-FILE-ALLOCATION(MODE=*NOT-ALLOWED)
```

Danach können Dateien, die auf dem Volume-Set liegen, normal bearbeitet werden, es werden jedoch keine neuen Dateien auf dem Volume-Set angelegt.

- b) Sperre des Volume-Set-Access

```
RESTRICTION=*VOLUME-SET-ACCESS(MODE=*ADMINISTRATOR-ONLY)
```

Dabei ist auf die Reihenfolge zu achten. Für das Setzen der Volume-Set-Access-Sperre wird vorausgesetzt, dass die Sperre gegen das Anlegen neuer Dateien bereits gesetzt ist.

Nach dem Setzen der Sperren sind für Benutzer ohne TSOS-Privileg die Zugriffsmöglichkeiten auf Dateien des Volume-Sets stark eingeschränkt. Laufende Bearbeitungen von Dateien können noch weitergeführt werden, während der Beginn neuer Bearbeitungen und das Anlegen neuer Dateien verhindert wird. Auch das Löschen der auf dem Volume-Set befindlichen Dateien ist Benutzern ohne TSOS-Privileg nicht mehr erlaubt. Wenn gewünscht wird, dass die Benutzer den Systembetreuer beim Leeren des Volume-Sets unterstützen, indem sie nicht mehr benötigte Dateien löschen, müssen sie vor dem Setzen der Volume-Set-Access-Sperre dazu aufgefordert werden.

## 2. Entfernen der Dateien

Die auf dem Volume-Set liegenden Dateien werden von dort entfernt, indem sie gelöscht, auf eine Hintergrundebene verdrängt oder innerhalb der Verarbeitungsebene auf einen anderen Volume-Set umverlagert werden.

Die Umverlagerung innerhalb der Verarbeitungsebene kann im Allgemeinen dadurch erfolgen, dass der Systembetreuer die Dateien zunächst auf eine Hintergrundebene migriert (durch die HSMS-Anweisung MIGRATE-FILES mit Angabe des Selektionskriteriums VOLUME-SET-ID) und sie anschließend von dort wieder in die Verarbeitungsebene zurückholt. Dabei ist zu beachten, dass für das Zurückholen in dem SM-Pubset geeignete Volume-Sets zur Verfügung stehen müssen, auf denen die Dateiattribute der verdrängten Dateien erfüllbar sind.

Für die Migrierbarkeit einer Datei auf eine Hintergrundebene müssen bestimmte Voraussetzungen erfüllt sein, die ggf. zuvor hergestellt werden müssen. Die nicht migrierbaren Dateien eines Volume-Sets kann der Systembetreuer dadurch feststellen, dass er zunächst einen Migrations-Auftrag für alle auf dem Volume-Set liegenden Dateien initiiert und sich anschließend durch den Einblick in die HSMS-REPORT-FILE oder durch das Kommando SHOW-FILE-ATTRIBUTES (Selektionskriterium VOLUME-SET-ID) darüber informiert, welche Dateien dabei nicht berücksichtigt worden sind. Nachdem auch für diese die Voraussetzungen für die Migrierbarkeit hergestellt sind, wird erneut ein Migrations-Auftrag gestartet. Allerdings ist zu beachten, dass für bestimmte Dateien die Migration auf Hintergrundebenen prinzipiell nicht möglich ist. Für sie muss die Umverlagerung auf einem anderen Weg vorgenommen werden.



Das Verlagern einer Datei kann aus folgenden Gründen nicht möglich sein:

- a) Die Datei wird gerade bearbeitet und ist durch einen SECURE-Lock reserviert oder mit sonstigen Locks versehen. Durch das Kommando SHOW-FILE-LOCK kann sich der Systembetreuer über die für eine Datei gesetzten Locks informieren. Außerdem können durch das Kommando SHOW-PUBSET-CONFIGURATION die Tasks und die zugehörigen Userids ausgegeben werden, die für den Volume-Set eine Belegung halten. Aus diesen Informationen muss der Systembetreuer die auf dem Volume-Set aktiven Anwendungen ermitteln und sie dazu bewegen, die jeweiligen Dateien zu schließen und Reservierungen oder sonstige Locks zurückzugeben. Eine radikale Maßnahme, die den gleichen Effekt hat, aber nur in Sonderfällen akzeptabel sein dürfte, besteht darin, alle den Volume-Set belegenden Benutzertasks abzubrechen.
- b) Die Datei darf generell nicht migriert werden. Beispiele hierfür sind Dateien mit dem Attribut AVAILABILITY=\*HIGH, temporäre Dateien, Arbeitsdateien, mit der Sperre MIGRATION=\*FORBIDDEN versehene Dateien, in der Except-Files für Migration aufgeführte Dateien, etc. In diesem Fall ist ein zwischen Systembetreuer und Benutzern detailliert abgestimmtes Vorgehen notwendig. Ein möglicher Weg besteht darin, die betroffenen Dateien zunächst zu sichern, anschließend zu löschen und später wiedereinzuspielen. Beim Wiedereinspielen müssen für die Dateiattribute passende Volume-Sets bereitstehen. Vor der Durchführung dieser Maßnahmen müssen die Eigentümer der Dateien darüber informiert werden, dass vorübergehend Sonderzustände auftreten, in denen die Dateien nicht in dem Pubset katalogisiert und damit auch nicht zugreifbar sind. Im Fall, dass die Migrationssperren mit physikalischer Allokierung in Zusammenhang stehen, sollten die Eigentümer der Dateien ggf. auch den neuen Ablageort physikalisch vorgeben können. Diese Wünsche muss der Systembetreuer berücksichtigen, wenn er die Dateien wiedereinspielt.

## 8.3 Erweitern eines SM-Pubsets mit SF-Pubsets

Mit SMPGEN kann ein SM-Pubset mit einem oder mehreren (nicht leeren) SF-Pubsets erweitert werden. Bei der Aufnahme der SF-Pubsets in den SM-Pubset erfolgt eine Konvertierung in äquivalente Volume-Sets. Vorgehensweise und Randbedingungen, die für das Zusammenführen von SF-Pubsets zu einem SM-Pubset beschrieben sind, gelten hier sinngemäß (siehe [Kapitel „Erzeugen eines SM-Pubsets aus bestehenden SF-Pubsets“ auf Seite 65ff](#)).

Mit dieser Funktion erhält die Zuordnung von Datenbeständen zu SM-Pubsets zusätzliche Flexibilität:

- Die Migration zu SM-Pubsets kann stufenweise erfolgen, indem die SF-Pubsets sukzessive in einen SM-Pubset aufgenommen werden.
- Volume-Sets lassen sich zwischen SM-Pubsets transferieren:  
In einem ersten Schritt werden die Volume-Sets mit einem Spezial-Import (/IMPORT-PUBSET mit USE=\*EXCLUSIVE(CONVERT-VOLUME-SET=\*YES)) in SF-Pubsets konvertiert. Die hier zu beachtenden Randbedingungen sind bei dem Kommando-IMPORT-PUBSET im Handbuch „Kommandos“ [\[10\]](#) beschrieben. In einem weiteren Schritt können die so erzeugten SF-Pubsets in einen anderen SM-Pubset aufgenommen werden.  
Die Durchführung einer Datensicherung für die SF-Pubsets ist hier ebenso zu empfehlen (vgl. [Kapitel „Erzeugen eines SM-Pubsets aus bestehenden SF-Pubsets“ auf Seite 65](#)).

## 8.4 Hinzunahme und Entfernen von Volumes

Die wichtigsten Funktionen für die Hinzunahme und das Entfernen von Volumes werden durch die Kommandos der dynamischen Pubset-Rekonfiguration geboten. Zusätzlich können die Pubsets durch SIR und bei der Pubset-Inbetriebnahme durch die Hinzunahme weiterer Volumes vergrößert werden.

### 8.4.1 Hinzunahme von Volumes durch SIR

#### Volume-Hinzunahme zu Volume-Sets

Zu einem Volume-Set eines SM-Pubsets können durch folgende SIR-Anweisungen weitere Volumes hinzugenommen werden.

<code>//DECLARE-PUBSET TYPE=*S-M(ACTION=*EXTEND(...))</code>
<code>//BEGIN-VOLUME-SET-DECLARATION ACTION=*EXTEND</code>
<code>//CREATE-VOLUME</code>
<code>...</code>
<code>//END-VOLUME-SET-DECLARATION</code>
<code>....</code>

Auch freie Volume-Sets lassen sich durch Hinzunahme weiterer Volumes vergrößern. In diesem Fall entfällt die Anweisung DECLARE-PUBSET.

#### Voraussetzungen

Bei der Hinzunahme von Volumes durch SIR zu Volume-Sets sind folgende Punkte zu beachten:

1. Zu den Volumes müssen von allen Rechnern aus Zugangspfade existieren, an denen es möglich sein soll, den SM-Pubset in Betrieb zu nehmen. An dem Rechner, an dem die Erweiterung durch SIR vorgenommen wird, wird dies durch das System überprüft.
2. Der Plattentyp der Volumes muss eine Initialisierung/Formatierung mit einem Format und einer Allocation-Unit zulassen, welche mit dem Format und der Größe der Allocation-Unit des Volume-Sets übereinstimmen. Auch bezüglich der Eigenschaften Dualisierung durch DRV und Volume-Emulation im Globalspeicher muss die Homogenität des Pubsets gewahrt werden.

3. Für die Dualisierung der Volumes durch RAID oder REMOTE-COPY wird die Homogenität nicht durch das System erzwungen. Der Systembetreuer muss selbst entscheiden, ob ein bezüglich dieser Eigenschaften inhomogener Pubset den gewünschten Verfügbarkeitsanforderungen genügt.
4. In dem Volume-Set darf nicht bereits ein Volume mit gleicher VSN vorhanden sein. Die Hinzunahme von Volumes darf auch nicht zu einer Überschreitung der für den jeweiligen Volume-Set maximal möglichen Anzahl von Volumes führen. Diese ist von der Länge der Kennung des SF-Pubsets bzw. Volume-Sets und der Größe der Allocation-Unit abhängig. Sie kann unter bestimmten Umständen erhöht werden, indem der Volume-Set durch PVSREN umbenannt wird.
5. Handelt es sich bei dem neu hinzukommenden Volume um ein „großes“ Volume, so muss das zu erweiternde Pubset ein LARGE-OBJECTS Pubset sein. Andernfalls wird die Erweiterung mit einer entsprechenden Meldung abgelehnt.

*Hinweis*

SIR stellt automatisch sicher, dass die VSNs der hinzugenommenen Volumes mit der Kennung des Volume-Sets entsprechend der Namenskonvention für Volumes zusammenpassen.

## 8.4.2 Hinzunahme von Volumes bei der Pubset-Inbetriebnahme

Bei der Inbetriebnahme eines SM-Pubsets im Modus „exclusive“ oder „shared-master“ werden alle Volumes in der Konfiguration des Rechners ermittelt, deren VSN entsprechend der Namenskonvention für Volumes zu einem Volume-Set des Pubset passt, die aber nicht in dem Volume-Katalog auf der Volres eingetragen sind. Für jedes dieser Volumes wird beim Operator/Systembetreuer nachgefragt, ob es in den Pubset aufgenommen werden soll.

Voraussetzung für die Aufnahme eines Volumes ist wiederum, dass die vom System erzwungenen Homogenitätsanforderungen bezüglich Format, Allocation-Unit, DRV und Volume-Emulation im Globalspeicher erfüllt sind. „Große“ Volumes werden nur dann aufgenommen, wenn das zu erweiternde SM-Pubset ein LARGE-OBJECTS Pubset ist. Der Systembetreuer muss ebenso dafür Sorge tragen, dass von allen anderen Rechnern, an denen die Inbetriebnahme des Pubsets möglich sein soll, Zugangspfade zu den neu hinzukommenden Volumes existieren.

Ein Volume, das neu hinzugenommen wird, wird als leer betrachtet, d.h. bestimmte Metadatenbereiche (SVL, Header der F5-Etikette) müssen korrekt versorgt sein, der übrige Inhalt des Volumes wird ignoriert. Passend initialisierte leere Volumes können mit Hilfe der Utility VOLIN erzeugt werden.

### 8.4.3 Hinzunahme von Volumes durch dynamische Pubset-Rekonfiguration

Das Kommando MODIFY-PUBSET-PROCESSING erlaubt durch den Operanden VOLUME-SET-SUPPORT=\*MODIFY(VOLUME-ASSIGNMENT=\*ADD(...),...) im laufenden Pubset-Betrieb einen Volume-Set durch Hinzunahme eines Volumes zu erweitern. Dabei sind wie bei der Volume-Hinzunahme durch SIR und bei der Pubset-Inbetriebnahme bestimmte Aspekte zu beachten:

- Um zu einem Volume-Set ein Volume hinzunehmen zu können, muss er sich im Zustand „normal use“ befinden.
- Zu dem neu hinzukommenden Volume müssen von allen Rechnern aus Zugangspfade existieren, an denen es möglich sein soll, den SM oder SF-Pubset in Betrieb zu nehmen. An den Rechnern, an denen der Pubset bei der Ausführung des Kommandos in Betrieb ist, wird dies durch das System automatisch überprüft. Für sonstige Rechner muss der Systembetreuer dafür explizit Sorge tragen.
- Das Volume muss geeignet initialisiert und formatiert sein:
  - Die VSN des aufzunehmenden Volumes muss entsprechend der Namenskonvention für Volumes mit der Kennung des Volume-Sets zusammenpassen.
  - Format und Größe der Allocation-Unit des Volumes müssen mit dem Format und der Allocation-Unit des Volume-Sets übereinstimmen.
  - Die Homogenität der Volume-Set-Eigenschaften bezüglich Dualisierung durch DRV und Volume-Emulation im Globalspeicher muss gewahrt werden.
  - Für die Dualisierung der Volumes durch RAID oder REMOTE-COPY wird die Homogenität eines Volume-Sets nicht durch das System erzwungen. Es ist Aufgabe des Systembetreuers darauf zu achten, dass nur solche Volumes hinzugenommen werden, die den für den Volume-Set vorgesehenen Verfügbarkeits-Level nicht vermindern.
  - Handelt es sich bei dem neu hinzukommenden Volume um ein „großes“ Volume, so muss das zu erweiternde Pubset ein LARGE-OBJECTS Pubset sein. Andernfalls wird die Erweiterung mit einer entsprechenden Meldung abgelehnt.
- Wenn die maximale I/O-Länge des neu hinzukommenden Volumes kleiner ist als die der übrigen Volumes des Volume-Sets, verringert sich die maximale I/O-Länge des Volume-Sets. Die privilegierten Anwendungen, welche die maximale I/O-Länge ausnützen (ARCHIVE, CCOPY), berücksichtigen diese Änderung.
- In dem Volume-Set darf nicht bereits ein Volume mit gleicher VSN vorhanden sein. Die Hinzunahme von Volumes darf auch nicht zu einer Überschreitung der für den jeweiligen Volume-Set maximal möglichen Anzahl von Volumes führen.

Da die Hinzunahme eines Volumes den freien Platz auf einem Volume-Set vergrößert, hat sie (entschärfende) Auswirkungen auf die Sättigungssituation. Als Folge kann sich der bisher erreichte Sättigungsgrad erniedrigen.

Wenn der Platz auf dem neu hinzugekommenen Volume zunächst nicht automatisch durch das System vergeben wird, kann es in bestimmten Anwendungsfällen erwünscht sein möglichst gut geeigneten Platz durch physikalische Allokierung bereitstellen zu können. Dies kann beispielsweise bei der Erweiterung bestimmter Systemdateien nötig sein. Zu diesem Zweck bietet das Kommando MODIFY-PUBSET-PROCESSING den Operanden ALLOCATION-ON-VOLUME=\*NOT-ALLOWED. Auf dem neu hinzugekommenen Volume wird dann solange kein Platz belegt, bis der Systembetreuer mit dem Kommando MODIFY-PUBSET-RESTRICTIONS die Allokierungssperre vollständig aufhebt oder durch die weichere Sperre ALLOCATION-ON-VOLUME=\*PHYSICAL-ONLY ersetzt.

#### 8.4.4 Entfernen eines leeren Volumes aus einem SF-Pubset oder Volume-Set

Das Kommando MODIFY-PUBSET-PROCESSING erlaubt durch den Operanden VOLUME-SET-SUPPORT=\*MODIFY(VOLUME-ASSIGNMENT=\*REMOVE(...),...) im laufenden Pubset-Betrieb ein Volume aus einem Volume-Set zu entfernen.

Für das Entfernen eines Volumes müssen bestimmte Voraussetzungen erfüllt sein:

1. Der SM-Pubset, dem der betroffene Volume-Set angehört, muss an dem Rechner der Kommandoingabe in Betrieb sein.
2. Der Volume-Set, der verkleinert wird, muss er sich im Zustand „normal use“ befinden.
3. Bei dem zu entfernenden Volume darf es sich nicht um die Volres des Volume-Sets handeln.
4. Das zu entfernende Volume muss leer sein. Damit nicht anschließend wieder Platz auf dem Volume vergeben wird, sollte es bereits beim Leeren mit der Allokierungssperre ALLOCATION-ON-VOLUME=\*NOT-ALLOWED versehen werden. Die Hilfsmittel zum Leeren von Volumes werden anschließend dargestellt.
5. Das Entfernen des Volumes wird nicht erlaubt, wenn sich durch die Verminderung der Kapazität des Volume-Sets eine Überschreitung des Saturation-Level-4-Werts ergäbe oder wenn der Saturation-Level-4 bereits überschritten ist. Dadurch wird vermieden, dass der Systembetreuer unbeabsichtigt einen problematischen Sättigungszustand für den Volume-Set bzw. SF-Pubset herbeiführt. Wünscht er aber trotzdem das Volume zu entfernen, kann er den Saturation-Level-4 zuvor passend einstellen (Kommando MODIFY-SPACE-SATURATION-LEVELS).

## 8.4.5 Leeren eines Volumes

Um ein Volume zu leeren, müssen alle Dateien, welche auf dem Volume Platz belegen, verlagert oder gelöscht werden. Für Systemdateien wie die Katalogdateien, den Benutzerkatalog, den Storage-Klassen-Katalog, etc. ist dies im Allgemeinen nicht möglich.

Bei SM-Pubsets sind die Pubset-Metadaten - abgesehen von den Katalogdateien der Volume-Sets - auf den Control-Volume-Set konzentriert, wobei sie sich dort über alle Volumes des Control-Volume-Sets verteilen können. Für den Control-Volume-Set ist daher das Leeren und Entfernen der Volumes nur in speziellen Anwendungsfällen möglich. Die übrigen Volume-Sets eines SM-Pubsets enthalten als einzige Systemdatei den volume-set-spezifischen Dateikatalog. Dessen Lage kann beim Erzeugen des Volume-Sets durch SIR und auch bei dem späteren Erweitern durch physikalische Allokierung so beeinflusst werden, dass er auf bestimmte Volumes des Volume-Sets begrenzt bleibt. Volumes, die aus dem Volume-Set entfernbar sein sollen, können damit von Datei-Extents für Systemdateien frei gehalten werden.

Folgende Schritte sind zum Leeren eines Volumes nötig:

- Durch das Kommando `MODIFY-PUBSET-RESTRICTIONS` belegt der Systembetreuer das Volume mit der generellen Allokierungssperre (`ALLOCATION-ON-VOLUME =*NOT-ALLOWED`). Diese bewirkt, dass auf dem Volume belegter Platz freigegeben werden kann, neuer Platz aber nicht bereitgestellt wird. Falls die sich dadurch ergebende Verminderung des verfügbaren Platzes auf den nicht gesperrten Volumes eine Überschreitung des Saturation-Level-4 zur Folge hätte, wird das Kommando zurückgewiesen. Dadurch wird vermieden, dass der Systembetreuer unbeabsichtigt einen problematischen Sättigungszustand für den Volume-Set bzw. SF-Pubset herbeiführt. Wünscht er aber trotzdem das Volume zu entfernen, kann er den Saturation-Level-4 zuvor passend einstellen (Kommando `MODIFY-SPACE-SATURATION-LEVELS`).
- Alle geöffneten Dateien, welche Platz auf dem zu leerenden Volume belegen, werden geschlossen.
- Die geöffneten Dateien, die auf dem zu leerenden Volume Platz belegen, werden mit Hilfe der Utility `SPACEOPT` innerhalb des Volume-Sets passend umverlagert.

## 8.5 Ändern der Eigenschaftsprofile für die Volume-Set-Selektion

Die Eigenschaftsprofile (siehe [Abschnitt „Eigenschaftsprofile für die Volume-Set-Selektion“ auf Seite 23](#)) der zu einem Pubset gehörenden Volume-Sets (mit einer funktionsfähigen physikalischen Konfiguration) werden von dem System zur Bestimmung des geeigneten Ablageorts einer Datei herangezogen. Sie bestimmen auch den Umfang des Service-Angebots, das den Benutzern durch einen SM-Pubset bereitgestellt wird. Änderungen der Eigenschaftsprofile führen damit im Allgemeinen auch automatisch zu einer Änderung des Service-Angebots des Pubsets. Die Durchführung der Änderungen erfolgt für die Eigenschaften, die sich direkt aus der physikalischen Konfiguration des Volume-Sets ergeben, und die Eigenschaften, die der Systembetreuer explizit zu beschreiben hat, unterschiedlich.

### Änderung des Formats und der Allocation-Unit

Das Format und die Größe der Allocation-Unit sind an die physikalische Volume-Set-Konfiguration gekoppelt. Sie können nur durch eine Neugenerierung des Volume-Set geändert werden.

### Änderung des Performance- und Verfügbarkeitsprofils

Das Performance- und das Verfügbarkeitsprofil eines Volume-Set werden durch den Systembetreuer vorgegeben und können im laufenden Pubset-Betrieb durch das Kommando MODIFY-PUBSET-DEFINITION-FILE, Operand VOLUME-SET-ENTRY=\*MODIFY geändert werden. Dies gibt dem Systembetreuer die Möglichkeit, die Eigenschaften von Volume-Sets im Laufe der Zeit neu zu bewerten. Beim Neuanlegen von Dateien und dem Zurückholen von migrierten Dateien in die Verarbeitungsebene kommt die neue Festlegung unmittelbar zum Tragen. Sie sollten aber auch in der Verteilung der bereits existierenden und in der Verarbeitungsebene liegenden Dateien berücksichtigt werden. In gewissem Umfang wird der Systembetreuer zur Umverteilung der Dateien gezwungen:

1. Wenn ein Volume-Set, der bisher als hoch verfügbar eingestuft worden ist, die von ihm erwartete Verfügbarkeit nicht bietet, sollten auf ihm liegende Dateien mit dem Attribut AVAILABILITY=\*HIGH auf tatsächlich hochverfügbare Volume-Sets umverlagert werden. Um den Systembetreuer zu diesem Schritt zu zwingen, wird die Änderung des Verfügbarkeitsprofils von HIGH auf STD für einen Volume-Set nur dann gestattet, wenn sich auf ihm keine Datei mit dem Attribut AVAILABILITY=\*HIGH befindet. Die Umverteilung von Dateien mit hohen Verfügbarkeitsanforderungen kann erfolgen, indem sie zunächst gesichert und anschließend wiedereingespielt werden. Eine Verlagerung mithilfe der HSMS-Funktionen Migrate und Recall ist nicht möglich, da Dateien mit AVAILABILITY=\*HIGH nicht auf Hintergrundebenen migriert werden können. Damit während der Verlagerung der Dateien das Systems nicht neue Dateien mit hoher Verfügbarkeitsanforderung auf dem Volume-Set anlegt, muss der Systembetreuer diesen mit der Sperre NEW-FILE-ALLOCATION=\*NOT-ALLOWED (normaler Volume-Set)



bzw. NEW-FILE-ALLOCATION=\*PHYSICAL-ONLY (Control-Volume-Set) versehen. Nachdem dem Volume-Set AVAILABILITY=\*STD zugewiesen werden konnte, kann die Sperre kann wieder rückgesetzt werden.

2. Änderungen des Performance-Profiles sowie die Erhöhung des Verfügbarkeitsprofils von STD nach HIGH sind jederzeit möglich. Die Dateien behalten dabei ihre bisherige Lage auf den Volume-Sets bei, auch wenn ihre aktuelle Verteilung innerhalb des SM-Pubsets im Hinblick auf die geänderten Volume-Set-Attribute nicht mehr optimal ist. Es liegt bei dem Systembetreuer, ob bzw. wann er den SM-Pubset durch eine Reorganisation wieder in einen besseren Zustand bringt.

Auf die Dateiattribute bestehender Dateien hat die Änderung des Performance- und Verfügbarkeitsprofils eines Volume-Sets keine Auswirkungen. Dies gilt auch für Dateien, die durch physikalische Allokierung auf den Volume-Set gelegt wurden und durch die Migrationssperre S0-MIGRATION=\*FORBIDDEN an diesen gebunden sind. Allerdings sollten sich die Eigentümer solcher Dateien Gedanken machen, ob ein Volume-Set mit einer anderen Bewertung weiterhin den Anforderungen der gezielt auf ihm angelegten Dateien gerecht wird.

## 8.6 Ändern der Nutzungsart und der Nutzungsrestriktionen

Die Nutzungsart eines Volume-Set kann durch das Kommando MODIFY-PUBSET-DEFINITION-FILE, Operand VOLUME-SET-ENTRY=\*MODIFY im laufenden Pubset-Betrieb geändert werden. Voraussetzung ist, dass der Volume-Set im Zustand „defined only“ ist. Insbesondere können sich dann auf dem Volume-Set keine Dateien befinden. Dadurch wird sichergestellt, dass die Änderung der Nutzungsart nicht zu unerlaubten Kombinationen von Datei- und Volume-Set-Attributen führt. Beispielsweise dürfen Arbeitsdateien nicht auf Volume-Sets mit der Nutzungsart STANDARD liegen.

Die Nutzungsrestriktionen für Volume-Sets bzw. für einzelne Volumes von Volume-Sets können durch den Systembetreuer im laufenden Pubset-Betrieb mithilfe des Kommandos MODIFY-PUBSET-RESTRICTIONS geändert werden. Dabei sind folgende Punkte zu beachten:

1. Für freie Volume-Sets sind Nutzungsrestriktionen weder auf Volume-Set-Ebene noch auf Volume-Ebene definiert. Sie können folglich auch nicht geändert werden.
2. Nutzungsrestriktionen auf Volume-Set-Ebene sind abgesehen von der Sperre „in hold“ auch dann definiert und modifizierbar, wenn sich der Volume-Set im Zustand „defined only“ befindet und ihm damit zur Zeit keine physikalische Konfiguration zugeordnet ist.
3. Nutzungsrestriktionen für Volumes können nur für Volumes geändert werden, die Pubsets angehören.

4. Durch das Setzen der Volume-Set-Sperren `RESTRICTION=NEW-FILE-ALLOCATION (MODE=*NOT-ALLOWED)` oder `RESTRICTION=NEW-FILE-ALLOCATION (MODE=*PHYSICAL-ONLY)` wird die Menge der allgemein nutzbaren Volume-Sets verkleinert. Dies kann zu einer Reduzierung des durch den Pubset gebotenen und allgemein verfügbaren Service-Angebots führen. Ergäbe sich als Konsequenz, dass anschließend das voreingestellte Dateiformat des SM-Pubsets (current-Wert) auf keinem der allgemein nutzbaren Volume-Sets erfüllbar wäre, wird das Setzen dieser Sperren nicht zugelassen. Für die Sperre „in hold“ besteht ein ähnliches Problem. Da diese aber primär zum Sperren defekter Volume-Sets dient und in Fehlersituationen auch automatisch durch das System gesetzt werden muss, wird in diesem Fall das voreingestellte Dateiformat (current-Wert) durch das System neu ermittelt (siehe [Abschnitt „Voreinstellungen für Pubset-Space“ auf Seite 37](#)). Bei dem Setzen der Sperren sind auch die auf Hintergrundebene migrierten Dateien zu berücksichtigen, für die bei dem Zurückholen in die Verarbeitungsebene noch ein geeigneter, nicht gesperrter Volume-Set vorhanden sein muss.

5. Die Sperre `RESTRICTION=*VOLUME-SET-ACCESS(MODE=*ADMINISTRATOR-ONLY)` ist als Verschärfung der Sperre `RESTRICTION=NEW-FILE-ALLOCATION (MODE=*NOT-ALLOWED)` zu betrachten. Dies spiegelt sich in folgender Relation wider:

`VOLUME-SET-ACCESS = *ADMINISTRATOR-ONLY`

impliziert:

`NEW-FILE-ALLOCATION = *NOT-ALLOWED`

Modifikationen, welche zur Verletzung dieser Relation führen würden, werden zurückgewiesen.

6. Durch das Setzen der Volume-Sperren `ALLOCATION-ON-VOLUME=*NOT-ALLOWED` und `ALLOCATION-ON-VOLUME=*PHYSICAL-ONLY` reduzieren sich die allgemein verfügbaren Volumes und damit der allgemein verfügbare Platz eines Volume-Sets. Das Setzen der Sperre wird zurückgewiesen, wenn (für den allgemein verfügbaren Platz) bereits der Saturation-Level-4 (current-Wert) überschritten ist oder das Setzen der Sperre zu seiner Überschreitung führen würde. Soll die Sperre trotzdem gesetzt werden, muss der Systembetreuer zuvor die Saturation-Level-Werte passend einstellen.

## 8.7 Ändern der Cache-Konfiguration von Volume-Sets

Es ist zwischen den defined-Werten und der aktuellen Cache-Konfiguration eines Volume-Sets zu unterscheiden. Änderungen der defined-Werte können durch das Kommando `MODIFY-PUBSET-CACHE-ATTRIBUTES` erfolgen. Der SM-Pubset, dem der Volume-Set angehört, muss dabei importiert sein (Unterschied zu SF-Pubsets!), der Volume-Set selbst muss sich im Zustand „normal use“ oder „defined only“ befinden. Die neuen defined-Werte kommen automatisch bei der nächsten Inbetriebnahme des Pubsets zum Tragen, wenn für die einzelnen Volume-Sets Cache-Bereiche bereitgestellt werden. Außerdem werden durch die Kommandos `STOP-PUBSET-CACHING`, `START-PUBSET-CACHING` und `MODIFY-PUBSET-PROCESSING` Möglichkeiten bereitgestellt, damit sich Modifikationen der defined-Werte bereits in der laufenden Pubset-Session auf die aktuelle Cache-Konfiguration auswirken. Dabei gibt es folgende verschiedene Anwendungsfälle:

### 1. Neueinrichten einer Cache-Konfiguration

Für einen Volume-Set (im Zustand „normal use“), dem bisher, d.h. seit der Inbetriebnahme des Pubsets oder seit seiner physikalischen Aufnahme in den SM-Pubset, noch keine aktuelle Cache-Konfiguration zugewiesen war, soll die Cache-Nutzung aktiviert werden.

Dies lässt sich erreichen, indem zunächst die defined-Werte entsprechend der gewünschten Cache-Konfiguration festgelegt werden und anschließend durch das Kommando `START-PUBSET-CACHING` das Caching für den Volume-Set aktiviert wird.

### 2. Dynamische Anpassung der Cache-Größe

Für einen Volume-Set, für den bereits Caching aktiv ist, soll die Größe des Cache-Bereichs dynamisch angepasst werden. Das Cache-Medium und die Cache-Betriebsparameter bleiben gleich.

Hierzu wird zunächst der defined-Wert für die Cache-Größe des Volume-Sets neu festgelegt, dann das Caching für den Volume-Set deaktiviert (Kommando `STOP-PUBSET-CACHING`) und schließlich das Caching für den Volume-Set wieder eingeschaltet (Kommando `START-PUBSET-CACHING`).

### 3. Änderung des Cache-Mediums

Nachdem ein Volume-Set mit einem Cache-Bereich konnektiert wurde, können für ihn das aktuell benutzte Cache-Medium und der Cache-Betriebsparameter VOLATILITY nicht mehr geändert werden, bis der Pubset außer Betrieb genommen wird oder der Volume-Set physikalisch aus dem Pubset entfernt wird. Für den Systembetreuer äußert sich dies darin, dass das Kommando START-PUBSET-CACHING nur dann akzeptiert wird, wenn zu diesem Zeitpunkt die defined-Werte für die Cache-Konfiguration mit der Vorgeschichte des Volume-Sets konsistent sind.

Um für einen Volume-Set das aktuelle Cache-Medium oder die Festlegung für VOLATILITY im laufenden Pubset-Betrieb zu ändern, sind daher folgende Schritte erforderlich:

- Erneute Festlegung der defined-Werte für die Cache-Konfiguration
- Leeren des Volume-Sets
- Physikalisches Entfernen des Volume-Sets (regulärer Modus)
- Physikalisches Wiederaufnehmen des Volume-Sets

Für die Änderung der Cache-Konfiguration hat der Systembetreuer dafür zu sorgen, dass die entsprechenden Cache-Medien in ausreichendem Umfang zur Verfügung stehen. Durch die Einstellung SIZE-TOLERANCE wird dem System mitgeteilt, wie es sich verhalten soll, wenn es bei der Pubset-Inbetriebnahme, der physikalischen Aufnahme eines Volume-Sets oder bei der Ausführung des Kommandos START-PUBSET-CACHING die Vorgaben entsprechend den defined-Werten nicht oder nur teilweise erfüllen kann.

## 8.8 Ändern der Sättigungsschwellwerte von Volume-Sets

Mithilfe des Kommandos `MODIFY-SPACE-SATURATION-LEVELS` können die Sättigungsschwellwerte für die einzelnen Volume-Sets eines SM-Pubsets geändert werden. Der SM-Pubset muss dabei importiert sein (Unterschied zu SF-Pubsets!). Durch den Operanden `SCOPE` wird festgelegt, ob sich die vorgenommenen Änderungen nur auf die `current`-Werte (steuern die aktuelle Sättigungsüberwachung), nur auf die `defined`-Werte (dienen zur Initialisierung der `current`-Werte) oder auf beide Werte beziehen. Daraus ergeben sich Unterschiede für den Beginn und die Dauer des Zeitraums, in dem sich die Änderungen auf die Sättigungsüberwachung auswirken:

- Bei der Angabe `SCOPE=*TEMPORARY` werden nur die `current`-Werte geändert. Sie ist nur erlaubt, wenn der Volume-Set nicht im Zustand „defined only“ ist. Die Änderungen beeinflussen damit die Sättigungsüberwachung ab sofort, wirken aber nicht länger als bis zum Ende der Pubset-Session bzw. bis zum physikalischen Entfernen des Volume-Sets aus dem SM-Pubset. Bei der nächsten Inbetriebnahme des Pubsets bzw. der physikalischen Hinzunahme des Volume-Sets werden die `current`-Werte aus den nicht geänderten `defined`-Werten bestimmt.
- Bei der Angabe `SCOPE=*NEXT-PUBSET-SESSION` werden nur die `defined`-Werte geändert. Sie ist auch dann erlaubt, wenn der Volume-Set im Zustand „defined only“ ist. Die Änderungen beeinflussen die Sättigungsüberwachung erst ab der nächsten Inbetriebnahme des Pubsets oder ab dem nächsten physikalischen Hinzunehmen des Volume-Sets, wenn die `current`-Werte aus den geänderten `defined`-Werten bestimmt werden. Die Änderungen bleiben wirksam, bis der Systembetreuer wieder erneut explizit ändert.
- Bei der Angabe `SCOPE=*PERMANENT` werden im Allgemeinen sowohl die `current`- als auch die `defined`-Werte geändert, für Volume-Sets im Zustand „defined only“ beziehen sich die Modifikationen allerdings nur auf die `defined`-Werte, da die `current`-Werte in diesem Fall nicht definiert sind. Die Änderungen beeinflussen die Sättigungsüberwachung damit sofort bzw. sobald der Volume-Set physikalisch in den SM-Pubset aufgenommen wird. Die Dauer der Wirksamkeit ist nicht begrenzt (sofern sie der Systembetreuer später nicht mehr ändert).

## 8.9 Ändern der Pubset-Space-Voreinstellungen

Mithilfe des Kommandos MODIFY-PUBSET-SPACE-DEFAULTS können für einen SM-Pubset die Voreinstellungen für Pubset-Space (Zuweisung einer Voreinstellung für das Dateiformat und für die Größe von Primär-, Sekundär- und Maximalallokierung) geändert werden. Der Pubset muss dabei importiert sein (Unterschied zu SF-Pubsets!). Für das voreingestellte Dateiformat werden nur Werte zugelassen, die auf dem SM-Pubset erfüllbar sind. Durch den Operanden SCOPE (Werte: TEMPORARY, PERMANENT, NEXT-PUBSET-SESSION) kann die Gültigkeitsdauer der Einstellungen spezifiziert werden. Die Bedeutung der Werte ist analog dem Kommando MODIFY-SPACE-SATURATION-LEVELS. Abhängig von der gewünschten Gültigkeitsdauer werden nur die defined-Werte, nur die current-Werte oder beide Werte geändert.

Der dynamische Wert für das voreingestellte Dateiformat wird in bestimmten Fällen auch implizit durch das System geändert. Gelangt ein Volume-Set in den Zustand „in hold“ oder „defective“, so wird überprüft, ob der dynamische Werte für das voreingestellte Dateiformat in dem SM-Pubset immer noch erfüllbar ist. Ist dies nicht der Fall, ermittelt das System einen in dem SM-Pubset erfüllbaren current-Wert, der dem bisherigen current-Wert möglichst nahe kommt (bei bisherigem current-Wert K wird NK2 bevorzugt; ist auch NK2 nicht erfüllbar, wird NK4 gewählt; ist der bisherige current-Wert NK2, ist nur NK4 erfüllbar). Wenn der bisherige current-Wert in dem SM-Pubset wieder erfüllbar ist (z.B. weil der Volume-Set wieder in einem funktionsfähigen Zustand ist oder ein neuer Volume-Set aufgenommen wurde), erfolgt kein automatisches Rückschalten. Soll der bisherige current-Wert wieder eingestellt werden, muss der Systembetreuer dies explizit veranlassen.

## 8.10 Ändern von Storage-Klassen und Volume-Set-Listen

Durch das Bereitstellen von Storage-Klassen kann der Systembetreuer die in einem SM-Pubset verfügbaren Storage-Services benennen und den Benutzern über eine komfortable Schnittstelle zugänglich machen. Storage-Klassen in Verbindung mit Volume-Set-Listen erlauben dem Systembetreuer die Realisierung bestimmter Strategien für die Speichernutzung. Einem bestimmten Benutzer kann mit den Kommandos der Benutzerverwaltung eine Storage-Klasse als Default-Storage-Klasse zugewiesen werden. Diese kommt zum Tragen, wenn der Benutzer für eine Datei nicht explizit einen Storage-Service anfordert. Diese Nutzungsmöglichkeiten werden in [Abschnitt „Ablageortrelevante Dateiattribute und zugewiesene Voreinstellungen“ auf Seite 210](#) detailliert erläutert. Zum Einrichten, Verändern und Löschen von Storage-Klassen und Volume-Set-Listen werden durch das Storage-Klassen-Management Kommandos zur Verfügung gestellt, die Zuordnung einer Default-Storage-Klasse und eine spätere Änderung dieser Zuordnung wird durch das Kommando MODIFY-USER-PUBSET-ATTRIBUTES ermöglicht. Ein Bedarf für die Änderung von Storage-Klassen, Volume-Set-Listen oder Default-Storage-Klassen ergibt sich im Allgemeinen dann, wenn für einen SM-Pubset die Strategien der Speichernutzung geändert werden sollen, was wiederum oft mit dem Hinzunehmen oder Entfernen bestimmter Volume-Sets in Zusammenhang stehen dürfte. Bei ihrer Durchführung hat der Systembetreuer zu beachten, dass zwischen Kommandoprozeduren, Dateien, Storage-Klassen, Volume-Set-Listen, Volume-Sets und Default-Storage-Klassen Abhängigkeiten bestehen, die dadurch tangiert werden können. Dabei gelten folgende Regeln:

### 1. In Dateien referenzierte Storage-Klassen

Das Anlegen einer Datei mit Zuweisung einer nicht existierenden Storage-Klasse wird abgewiesen. Das Löschen einer Storage-Klasse kann damit die Ablauffähigkeit von Kommandoprozeduren beeinträchtigen. Storage-Klassen können gelöscht werden, auch wenn in Dateien noch Referenzen auf sie bestehen. Eine Datei, der eine nicht definierte Storage-Klasse zugeordnet ist, wird bei der erneuten Ablageortbestimmung (bei einem Recall) wie eine Datei behandelt, der keine Storage-Klasse zugewiesen ist.

### 2. Als Default-Storage-Klassen zugeordnete Storage-Klassen

Das Zuordnen einer Default-Storage-Klasse ist auch dann möglich, wenn diese nicht definiert ist. Das Löschen einer Storage-Klasse wird auch dann erlaubt, wenn sie noch bestimmten Benutzern als Default-Storage-Klasse zugewiesen ist. Dem Systembetreuer wird aber dringend empfohlen zu vermeiden, dass nicht definierte Storage-Klassen Benutzern als Default-Storage-Klassen zugeordnet sind. Dies hat nämlich zur Folge, dass die entsprechenden Benutzer keine Dateien anlegen können, für welche die Dateiattribute nicht explizit (durch Direktattributierung oder explizite Angabe einer Storage-Klasse) spezifiziert werden. Die Ablauffähigkeit von Kommandoprozeduren kann dadurch erheblich beeinträchtigt werden.

### 3. In Storage-Klassen referenzierte Volume-Set-Listen

Beim Anlegen oder Ändern von Storage-Klassen ist es nicht möglich, diesen nicht definierte Volume-Set-Listen zuzuordnen. Das Löschen einer Volume-Set-Liste ist aber auch dann erlaubt, wenn sie noch in Storage-Klassen referenziert wird. Eine Storage-Klasse, der eine nicht definierte Volume-Set-Liste zugeordnet ist, wird bei dem Anlegen von Dateien und bei dem Recall wie eine Storage-Klasse ohne Volume-Set-Liste behandelt.

### 4. In Volume-Set-Listen referenzierte Volume-Sets

Beim Anlegen oder Modifizieren von Volume-Set-Listen dürfen auch Volume-Sets referenziert werden, die in der Konfiguration des SM-Pubsets nicht definiert sind, sich in dem Zustand „defined only“, „in hold“ oder „defective“ befinden oder mit einer der Sperren NEW-FILE-ALLOCATION=\*NOT-ALLOWED oder NEW-FILE-ALLOCATION=\*PHYSICAL-ONLY versehen sind. Auch wird beim physikalischen Entfernen eines Volume-Sets oder beim Setzen der Allokierungs-Sperren nicht überprüft, ob der Volume-Set in einer Volume-Set-Liste referenziert wird. Referenzen auf solche Volume-Sets werden bei der Bestimmung des Ablageorts von Dateien ignoriert.

Um das Allokierungsverhalten überschaubar zu halten, wird dem Systembetreuer dringend empfohlen, nicht konsistente Referenzen auf Storage-Klassen, Volume-Set-Listen, Volume-Sets zu vermeiden. Insbesondere sollten beim Löschen dieser Objekte auch alle auf sie bestehenden Referenzen bereinigt werden. Das Ermitteln von Referenzen wird durch die Kommandos SHOW-FILE-ATTRIBUTES (Selektionskriterium STORAGE-CLASS), SHOW-STORAGE-CLASS (Selektionskriterium VOLUME-SET-LIST) und SHOW-VOLUME-SET-LIST (Selektionskriterium VOLUME-SET) unterstützt. Direkte Informationsfunktionen zum Aufdecken inkonsistenter Referenzen stehen nicht zur Verfügung.



## 8.11 Ändern sonstiger Pubset-Merkmale

### Ändern der HSMS-Konfiguration und der HSMS-Management-Klassen

Die durch HSMS bereitgestellten Möglichkeiten, um für einen SM-Pubset Backup-Archive, Hintergrundebenen und HSMS-Management-Klassen modifizieren zu können, werden in dem Benutzerhandbuch „HSMS“ [4] dargestellt. Für die Anpassung der S1-Ebene bietet die dynamische Pubset-Rekonfiguration Basismechanismen (Hinzufügen, Entfernen, Vergrößern und Verkleinern eines S1-Volume-Sets).

### Ändern von Benutzereinträgen und Benutzergruppen

Die Möglichkeiten zum Einrichten, Ändern und Entfernen von Benutzereinträgen werden in dem Handbuch „Einführung in die Systembetreuung“ [5] beschrieben. Dort werden auch die besonderen Bedingungen, die sich für das Ändern der Benutzerkontingente von SM-Pubsets aus ihrer komplexeren Struktur ergeben, erläutert.

### Ändern von Schutzprofilen

Die Möglichkeiten zum Einrichten, Ändern und Entfernen von GUARDS-Schutzprofilen sind im dem Handbuch „SECOS“ [2] beschrieben.

### Ändern der physikalischen Volume-Set-Verfügbarkeit

Bezüglich der Möglichkeiten, die Verfügbarkeit eines Volume-Sets auf physikalischer Ebene zu beeinflussen, wird auf die Handbücher „SHC-OSD V3.0“ [8] und „DRV“ [6] verwiesen.

### Ändern der Festlegungen für die Nutzung eines Pubsets an einem bestimmten Rechner

Die Festlegungen, welche sich auf die Nutzung eines Pubsets an einem bestimmten Rechner beziehen (Größe der Pufferbereiche für das Katalog-Management, EAM-Parameter, Pubset-Zugangs-Kontrolle, etc.) können für SM-Pubsets wie bisher bei SF-Pubsets durch das Kommando MODIFY-MASTER-CATALOG-ENTRY modifiziert werden.

### Ändern der Festlegungen für die Nutzung eines Pubsets im Rechnerverbund

Die Festlegungen, welche sich auf die Nutzung des Pubsets im Rechnerverbund beziehen, können für SM-Pubsets wie für SF-Pubsets durch die Kommandos MODIFY-MASTER-CATALOG-ENTRY und SET-PUBSET-ATTRIBUTES geändert werden.



---

## 9 Management des Lebenszyklus einer Datei

Der Benutzer ist die Instanz, welche für eine Datei die Anforderungen bezüglich Verarbeitung, Sicherung, Zeitpunkt und Dauer der Verarbeitungsphasen, etc. kennt und deswegen in der Lage sein soll, ihren Lebenszyklus entsprechend zu steuern. Der Systembetreuer ist die Instanz, welche über die in einem Rechenzentrum vorhandenen Ressourcen Bescheid weiß, in der Benutzung der Kommandos und Utilities für die Datenadministration speziell ausgebildet ist und über globale, d.h. benutzerübergreifende Informationen verfügt. Er ist daher besonders geeignet, die konkreten Maßnahmen durchzuführen, welche die Anforderungen der Benutzer bestmöglich realisieren.

SMS in BS2000/OSD bietet die Möglichkeit, die Schnittstellen zwischen Systembetreuer und Benutzer so zu gestalten, dass sie die skizzierte Rollenverteilung ermöglichen. Außerdem erlaubt es eine hohe Automatisierung der Aufgaben des Systembetreuers. Dabei ist berücksichtigt, dass auch in der SMS-Welt noch gezielte Eingriffsmöglichkeiten auf physikalischer Ebene erwünscht sein können. Obwohl diese Anforderung zunächst der SMS-Idee konträr scheint, sind die SM-Pubsets in BS2000/OSD so konzipiert, dass auch Benutzer mit derartigen Wünschen eingebunden werden können.

Im Folgenden werden die Aufgabenbereiche Nutzung der Verarbeitungsebene, Nutzung der Hintergrundebenen, Dateisicherung, Freigabe nicht benötigten Platzes unter den Aspekten logische Schnittstellen zwischen Benutzer und Systembetreuer, Automatisierbarkeit und direkte Steuerungsmöglichkeiten für die Benutzer genauer betrachtet.

## 9.1 Nutzung der Verarbeitungsebene

Für die Nutzung der Verarbeitungsebene sind folgende Aspekte wesentlich:

- Einer Datei wird beim Anlegen Platz in der Verarbeitungsebene (S0-Ebene) zugewiesen. Für die Dauer der Bearbeitung kann diese nicht verlassen werden. Wird auf die Datei für längere Zeit nicht zugegriffen, ist es für eine effiziente Ressourcennutzung oft sinnvoll, sie auf eine kostengünstigere Hintergrundebene zu migrieren. Für eine spätere Verarbeitung muss die migrierte Datei wieder in die Verarbeitungsebene zurückgeholt werden.
- Die Medien der Verarbeitungsebene können sich in ihren Eigenschaften stark unterscheiden. Sie sind als Ablageort für eine Datei unterschiedlich geeignet, je nachdem welche Eigenschaften der Benutzer für die Datei wünscht. Bei der Platzbereitstellung in der S0-Ebene kommt daher der Bestimmung des geeigneten Ablageorts eine große Bedeutung zu. Für die Auswahl des Ablageorts wesentliche Eigenschaften der Speichermedien sind das Performance-Verhalten, die Ausfallsicherheit, das Format (K, NK2, NK4) und die Größe der Allocation-Unit (6K, 8K, 64K). Sie korrespondieren mit den Dateieigenschaften gewünschte Performance, gewünschte Ausfallsicherheit, gewünschtes Dateiformat und Größe der Datei. Ferner ist bei der Ablageortbestimmung für eine Datei zu berücksichtigen, ob es sich um eine Arbeitsdatei handelt, die auf speziell für Arbeitsdateien vorgesehene Speichermedien gelangen soll.
- Im Laufe des Bestehens einer Datei können sich die von dem Benutzer gewünschten Dateieigenschaften ändern. Ein bisher gut geeigneter Ablageort in der S0-Ebene kann sich dann als unvorteilhaft erweisen. In diesem Fall ist es günstig, die Datei innerhalb der S0-Ebene auf ein besser geeignetes Medium zu verlagern.

## 9.1.1 Bestimmung des Ablageorts einer Datei bei SM-Pubsets – Übersicht

Um die Wünsche der Benutzer nach möglichst gut geeigneten Ablageorten der Dateien in der Verarbeitungsebene abbilden zu können, stehen verschiedene Funktionen zur Verfügung:

### Direktattributierung

Zur Grundfunktionalität eines SM-Pubsets, die automatisch durch das System bereitgestellt wird, gehört eine Standardschnittstelle für die Benutzer, welche die physikalische Struktur des Pubsets auf die logische Ebene abbildet. Die durch den Pubset gebotenen Möglichkeiten werden dabei als Kombinationen von Dateiattributen sichtbar, welche den Dateien durch die Benutzer zugewiesen werden können.

### Storage-Klassen ohne Volume-Set-Listen

Auf SM-Pubsets können Storage-Klassen eingerichtet werden. Ein Anwendungsfall sind Storage-Klassen ohne Volume-Set-Listen. Mit ihnen kann den Benutzern eine funktional der Direktattributierung weitgehend äquivalente, in der Anwendung aber komfortablere Schnittstelle zur Verfügung gestellt werden. Im Unterschied zur Direktattributierung erfordern sie Vorleistungen durch den Systembetreuer. Er muss die Storage-Klassen selbst einrichten und pflegen.

### Storage-Klassen mit Volume-Set-Listen

Storage-Klassen mit Volume-Set-Listen erlauben dem Systembetreuer, das System bei der Auswahl der Volume-Sets zu beeinflussen und in einem SM-Pubset spezifische Strategien für die Speichernutzung zum Einsatz zu bringen. Dadurch lässt sich u.a. erreichen, dass die Ressourcen in einem SM-Pubset differenzierter genutzt werden können, als dies mit Direktattributierung möglich ist, ohne dass die Trennung von logischer Sicht (für die Benutzer) und physikalischer Sicht (für den Systembetreuer) aufgegeben werden muss.

### Physikalische Allokierung

In einem SM-Pubset kann einzelnen Benutzern die Möglichkeit zur physikalischen Allokierung gegeben werden. Sie haben damit direkte Einflussmöglichkeiten auf den Ablageort der Dateien.

Im Folgenden werden diese Möglichkeiten und ihre Koexistenz innerhalb eines SM-Pubsets ausführlicher dargestellt.

## 9.1.2 Direktattributierung

In einem SM-Pubset kann der Systembetreuer jedem Benutzer, der nicht die Möglichkeit zur physikalischen Allokierung hat, die Möglichkeit der Direktattributierung entziehen, indem er dem Benutzer eine Default-Storage-Klasse zuweist.

In diesem Abschnitt wird der spezielle Anwendungsfall betrachtet, dass für einen SM-Pubset ausschließlich Direktattributierung genutzt wird (d.h. ein SM-Pubset ohne Storage-Klassen, Volume-Set-Listen und physikalische Allokierung). **Bild 21** verdeutlicht die Relation zwischen Systembetreuer, Benutzer und System bei dieser Nutzung.

Einem SF-Umfeld, in dem der Systembetreuer den Benutzern mehrere SF-Pubsets mit unterschiedlichen Eigenschaften zur Verfügung stellt, entspricht ein SM-Umfeld mit einem SM-Pubset, der aus Volume-Sets mit unterschiedlichen Eigenschaften besteht. Beim Einrichten bzw. Erweitern des SM-Pubsets ordnet der Systembetreuer den einzelnen Volume-Sets Eigenschaftsprofile für Performance und Verfügbarkeit zu und bestimmt außerdem, ob es sich um Volume-Sets für Arbeitsdateien handelt (Nutzungsart). Zusammen mit der Größe der Allocation-Unit und dem Format, die den Volume-Sets bei der Initialisierung bzw. Formatierung aufgeprägt werden, sind dadurch die für die Volume-Set-Selektion relevanten Eigenschaften der Volume-Sets festgelegt.

Die Volume-Sets eines SM-Pubsets, die sich im Zustand „normal use“ befinden und nicht mit Allokierungssperren belegt sind, bestimmen zusammen mit ihren Eigenschaftsprofilen den Umfang der Storage-Services, die der SM-Pubset den Benutzern zur Verfügung stellt. Ein Storage-Service wird durch eine bestimmte Kombination der Dateiattribute Performance (d.h. PERFORMANCE, USAGE, DISK-WRITE), Verfügbarkeit, Format, Nutzungsart repräsentiert. Hinsichtlich der Qualität, die ein SM-Pubset für einen bestimmten Storage-Service bietet, ist zwischen verschiedenen Stufen zu unterscheiden: nicht erfüllbar, erfüllbar, nahezu optimal unterstützt, optimal unterstützt. Das Kommando SHOW-PUBSET-FILE-SERVICES zeigt den Benutzern die Storage-Services eines SM-Pubsets und den Grad ihrer Erfüllbarkeit an. Wenn ein Pubset durch Rekonfiguration geändert wird, z.B. wenn der Systembetreuer Volume-Sets physikalisch hinzunimmt, entfernt oder mit Allokierungssperren versieht, überträgt sich dies automatisch auf das durch SHOW-PUBSET-FILE-SERVICES angezeigte verfügbare Service-Spektrum des SM-Pubsets.

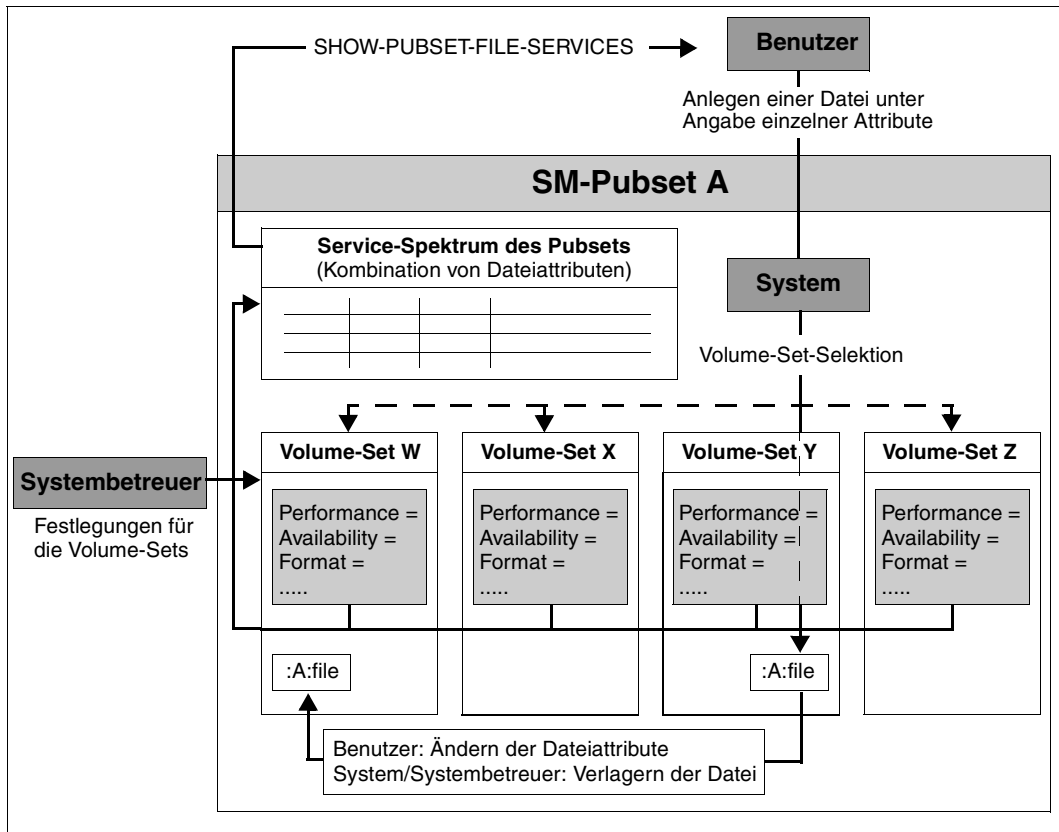


Bild 21: Nutzung der Verarbeitungsebene eines SM-Pubsets (Grundfunktionalität)

Ein Benutzer muss nicht wissen, aus welchen Volume-Sets ein SM-Pubset aufgebaut ist. Er ermittelt beim Anlegen einer Datei denjenigen Storage-Service aus dem Service-Spektrum des SM-Pubsets, der seine Anforderungen am besten abdeckt, und ordnet ihn der Datei durch Angabe der entsprechenden einzelnen Dateiattribute (Direktattributierung) zu. Das System (Volume-Set-Selektion) bestimmt daraus den für die Datei am besten geeigneten Volume-Set, wobei neben den Eigenschaften der Volume-Sets auch die aktuelle Platzbelegung berücksichtigt wird. Volume-Sets, die nicht im Zustand „normal use“ sind oder mit einer Allokierungssperre belegt sind, werden nicht herangezogen. Ein für die Volume-Set-Selektion relevantes Kriterium ist die Größe der Datei. Damit sie durch das System richtig eingeschätzt werden kann, sollte der Benutzer bereits beim Anlegen der Datei einen Wert für Primärallokierung angeben, der an dem künftigen Platzbedarf orientiert ist. Generell sollten die Benutzer durch sinnvolle Platzanforderungen dazu beitragen, dass der Pubset in einem ordentlichen Zustand gehalten werden kann. Ein ungünstiges Verhalten besteht z.B. in häufigen Sekundärallokierungen. Sie erhöhen die Anzahl der Datei-Extents und tragen zur Zersplitterung des freien Speicherplatzes bei.

Ändern sich die Anforderungen für eine bereits existierende Datei, kann der Benutzer dies dadurch berücksichtigen, dass er die für den Ablageort relevanten Attribute der Datei entsprechend anpasst (Kommando MODIFY-FILE-ATTRIBUTES) und ihr dadurch einen anderen Storage-Service aus dem Service-Spektrum des SM-Pubsets zuweist. Im Fall, dass die neuen Dateiattribute mit den Eigenschaften des Volume-Sets, auf dem die Datei bisher liegt, nicht verträglich sind, wird sie automatisch durch das System verlagert (z.B. neues Dateiautribut AVAILABILITY=\*HIGH, Verfügbarkeit des Volume-Sets AVAILABILITY=\*STD). Andernfalls behält die Datei ihren Ablageort zunächst bei. Die geänderten Attribute kommen dann zum Tragen, wenn die Datei nach dem Verdrängen auf eine Hintergrundebene in die Verarbeitungsebene zurückgeholt wird und dabei der Volume-Set für die Datei erneut ermittelt wird. Dies kann z.B. im Rahmen einer Pubset-Reorganisation erfolgen, die der Systembetreuer für eine effiziente Nutzung der Pubset-Ressourcen regelmäßig vornehmen sollte. Die Verlagerung einer Datei innerhalb des SM-Pubsets hat keine Auswirkung auf ihren Pfadnamen.

Durch Benutzerkontingente kann der Systembetreuer den von den einzelnen Benutzern belegten Platz kontrollieren. Entsprechend den unterschiedlichen Dateiattributen gibt es für SM-Pubsets verschiedene Kontingentarten (siehe auch [Kapitel 11](#), [Seite 259](#)).

### 9.1.3 Storage-Klassen ohne Volume-Set-Listen

Storage-Klassen ohne Volume-Set-Listen sind benannte Kombinationen von Dateiattributen, die vom Systembetreuer pubset-spezifisch eingerichtet werden. Die Definitionen für die Storage-Klassen werden im Storage-Klassen-Katalog des SM-Pubsets hinterlegt (siehe [Tabelle 2](#)).

Storage-Klassen-Katalog								
Name der Storage-Klasse	Preformat	Performance-relevante Dateiattribute			Availability	Work	zugeordnete Volume-Set-Liste	Kommentar
		Performance	Usage	Disk-Write				
.....	.....	.....	.....	.....	.....	.....	.....	.....
STANDARD	K	STD	.....	.....	STD	NO	keine	.....
HIGHAVAIL	K	STD	.....	.....	VERY-HIGH	NO	keine	.....
HIGHPRF1	NK2	HIGH	READ-WRITE	IMMEDIATE	STD	NO	keine	.....
HIGHPRF2	NK2	HIGH	READ-WRITE	BY-CLOSE	STD	NO	keine	.....
WORK	K	STD	.....	.....	STD	YES	keine	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....

Tabelle 2: Beispiele für Storage-Klassen ohne Volume-Set-Listen

[Bild 22](#) illustriert die Rolle der Storage-Klassen ohne Volume-Set-Listen für das Zusammenwirken von Benutzer, Systembetreuer und System.



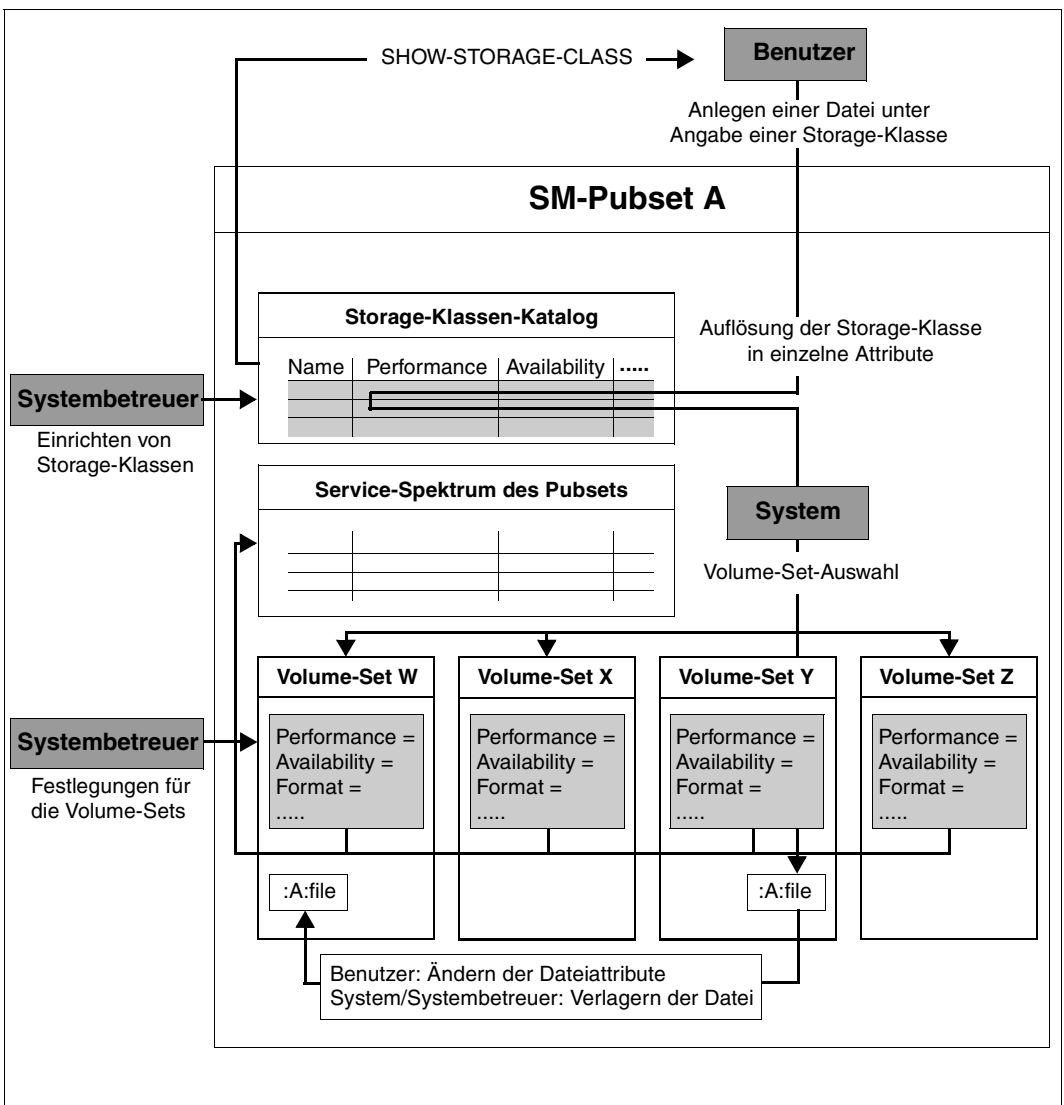


Bild 22: SM-Pubset mit Storage-Klassen ohne Volume-Set-Listen

Sinnvollerweise werden vom Systembetreuer nur Storage-Klassen vorgesehen, deren Attribute durch den SM-Pubset erfüllt werden können. Er kann dies mithilfe des Kommandos SHOW-PUBSET-FILE-SERVICES überprüfen. Die Benutzer können sich über die vorhandenen Storage-Klassen und deren Eigenschaften durch das Kommando SHOW-STORAGE-CLASS informieren. Beim Anlegen von Dateien können die Storage-Klassen alternativ zu der Direktattributierung verwendet werden, um den gewünschten Storage-Service zu spezifizieren. Sich ändernden Anforderungen für bereits bestehende Dateien kann durch die Zuweisung von anderen Storage-Klassen Rechnung getragen werden.

Für die Auswahl des Ablageorts, der Steuerung der Verlagerung von Dateien und der Platzkontingentierung sind Storage-Klassen ohne Volume-Set-Listen der Direktattributierung weitgehend äquivalent. Sie bieten aber den Vorteil, dass die Kommandoprozeduren der Benutzer übersichtlich und kompakt gehalten werden können. Im Vergleich zu der Direktattributierung führen sie zu einer stärkeren Kopplung der Kommandoprozeduren an bestimmte Pubsets, die möglicherweise nicht erwünscht ist. Kommandoprozeduren, in denen Storage-Klassen referenziert werden, sind nämlich nur in SM-Pubsets ablauffähig, in denen diese auch definiert sind. Storage-Klassen erfordern im Unterschied zu der Direktattributierung auch eine aufwändigere Pflege durch den Systembetreuer. Wird die Konfiguration eines SM-Pubset geändert, sollte der Systembetreuer dafür sorgen, dass die Auswirkungen auf das Service-Angebot der Pubsets auch in den zur Verfügung gestellten Storage-Klassen berücksichtigt werden. Das Löschen von Storage-Klassen sollte der Systembetreuer in Absprache mit den Benutzern vornehmen, da diese möglicherweise Kommandoprozeduren anpassen müssen.

Eine spezielle, aber sehr wichtige Anwendungsmöglichkeit für die Storage-Klassen besteht in ihrer Verwendung als Default-Storage-Klassen. Wenn ein Benutzer für eine Datei den gewünschten Storage-Service nicht explizit spezifiziert (d.h. durch Angabe einer Storage-Klasse oder von Einzelattributen), überprüft das System, ob dem Benutzer eine Default-Storage-Klasse zugewiesen ist. Ist dies der Fall, wird die Benutzeranforderung genauso behandelt, als ob der Datei diese Storage-Klasse explizit zugewiesen wäre. Im Fall, dass Storage-Klassen ausschließlich als Default-Storage-Klassen verwendet werden, gehen sie nicht in Kommandoprozeduren ein. Sie führen in diesem Fall auch nicht zu Kopplungen von Kommandoprozeduren an bestimmte SM-Pubsets.

Storage-Klassen können durch die Zuordnung von GUARDS-Profilen auf einzelne Benutzer eingeschränkt werden. Diese Funktion dient vor allem zum Ressourcenschutz bei Storage-Klassen mit zugeordneten Volume-Set-Listen (siehe [Abschnitt „Storage-Klassen mit Volume-Set-Listen“ auf Seite 187](#)). Für Storage-Klassen ohne Volume-Set-Listen ist die Verwendung von GUARDS-Profilen nur bei denjenigen Benutzern relevant, denen durch eine Default-Storage-Klasse die Möglichkeit zur Direktattributierung entzogen wurde..

## 9.1.4 Storage-Klassen mit Volume-Set-Listen

Indem der Systembetreuer auf einem SM-Pubset Volume-Set-Listen einrichtet und sie mit Storage-Klassen verknüpft, kann er die Volume-Set-Selektion des Systems beeinflussen. Die Volume-Set-Listen werden in dem Volume-Set-Listen-Katalog des SM-Pubsets hinterlegt (siehe [Tabelle 3](#) und [Tabelle 4](#)).

Storage-Klassen-Katalog								
Name der Storage-Klasse	Preformat	Performance-relevante Dateiattribute			Availability	Work	zugeordnete Volume-Set-Liste	Kommentar
		Performance	Usage	Disk-Write				
.....	.....	.....	.....	.....	.....	.....	.....	.....
STDUSR1	K	STD	.....	.....	STD	NO	USR1LST	.....
STDUSR2	K	STD	.....	.....	STD	NO	USR2LST	.....
SPCPRF1	K	VERY-HIGH	WRITE	IMMEDIATE	STD	NO	GSVS1LST	xxxx
HIGHPRF2	K	VERY-HIGH	READ		STD	NO	GSVS2LST	xxxx
AVAIL	K	STD	.....	.....	HIGH	NO	keine	.....
WORK	K	STD	.....	.....	STD	YES	keine	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....

Tabelle 3: Beispiel für Definition von Storage-Klassen

Volume-Set-Listen-Katalog	
Name der Volume-Set-Liste	zugehörige Volume-Sets
.....	
USR1LST	VS1, VS2
USR2LST	VS3, VS4
GSVS1LST	GVS1
GSVS2LST	GVS2
.....	.....

Tabelle 4: Beispiel für Volume-Set-Listen

In einem SM-Pubset mit Volume-Set-Listen sind zwei Kategorien von Volume-Sets zu unterscheiden:

1. Volume-Sets, die in (mindestens) einer der Volume-Set-Listen enthalten sind, die der Systembetreuer eingerichtet hat (Anmerkung: Die vom Systembetreuer definierten Volume-Set-Listen müssen nicht durchschnittsfrei sein).
2. Volume-Sets, welche keiner der durch den Systembetreuer definierten Volume-Set-Listen angehören. Diese Volume-Sets werden als der Default-Volume-Set-Liste zugehörig betrachtet, die dadurch implizit definiert wird. Bei SM-Pubsets, auf denen keine Volume-Set-Listen definiert sind, gehören alle Volume-Sets zur Default-Volume-Set-Liste.

**Bild 23** veranschaulicht, wie in einem SM-Pubset mit Volume-Set-Listen die Benutzeranforderungen bei der Auswahl des Volume-Sets berücksichtigt werden. Dabei sind folgende Fälle zu unterscheiden:

#### 1. Angabe einer Storage-Klasse mit Volume-Set-Liste

Ordnet ein Benutzer einer Datei beim Anlegen eine Storage-Klasse zu, ermittelt das System zunächst aus der Definition der Storage-Klasse die Kombination der Dateiattribute, welche sie repräsentiert. Ist die Storage-Klasse (entsprechend ihrer Definition) mit einer Volume-Set-Liste verknüpft, werden alle zugehörigen Volume-Sets bestimmt. Aus diesen wählt das System denjenigen als Ablageort für die Datei aus, der die der Storage-Klasse zugeordneten Dateiattribute am besten erfüllt. Dabei wird auch die Belegungssituation auf den einzelnen Volume-Sets berücksichtigt. Mit einer Allokierungssperre versehene Volume-Sets oder Volume-Sets, die nicht im Zustand „normal use“ sind, werden nicht herangezogen. Kann die Datei auf keinem Volume-Set der Volume-Set-Liste angelegt werden, bezieht das System auch alle übrigen Volume-Sets des SM-Pubsets in die Volume-Set-Auswahl mit ein, die im Zustand „normal use“ und nicht mit einer Allokierungssperre versehen sind. Beim Einrichten bzw. Ändern einer Storage-Klasse hat der Systembetreuer darauf zu achten, dass er ihr eine Kombination von Dateiattributen zuordnet, die durch Volume-Sets ihrer Volume-Set-Liste erfüllt werden können. Geschieht dies nicht, kann sich ein schwer zu überschauendes Systemverhalten bei der Volume-Set-Auswahl ergeben, da auf Volume-Sets anderer Volume-Set-Listen ausgewichen werden muss. Für die Ermittlung, welche Kombinationen von Dateiattributen durch eine Volume-Set-Liste unterstützt werden, bietet das Kommando SHOW-PUBSET-FILE-SERVICES (Operand VOLUME-SET-LIST) Unterstützung.

## 2. Angabe einer Storage-Klasse ohne Volume-Set-Liste

Storage-Klassen ohne Volume-Set-Liste werden bei der Volume-Set-Auswahl genau so wie Storage-Klassen mit Volume-Set-Listen behandelt, wobei jedoch an Stelle einer vom Systembetreuer eingerichteten Volume-Set-Liste die implizit definierte Volume-Set-Liste herangezogen wird. Entsprechend hat der Systembetreuer beim Einrichten bzw. Ändern einer Storage-Klasse ohne Volume-Set-Liste darauf zu achten, dass die ihr zugeordnete Kombination von Dateiattributen durch Volume-Sets der Default-Volume-Set-Liste erfüllt werden kann. Zum Ermitteln der Kombinationen von Dateiattributen, welche auf den Volume-Sets der Default-Volume-Set-Liste möglich sind, kann das Kommando `SHOW-PUBSET-FILE-SERVICE` (Operand `VOLUME-SET-LIST=*NONE`) benutzt werden.

## 3. Direktattributierung

Auch bei Direktattributierung erfolgt die Volume-Set-Auswahl genauso, als ob die Default-Volume-Set-Liste als Volume-Set-Liste vorgegeben wäre. Unter deren Volume-Sets, d.h. allen Volume-Sets die keiner vom Systembetreuer definierten Volume-Set-Liste angehören, wird derjenige ausgewählt, der die vom Benutzer angegebenen Attribute am besten erfüllt. Wird dabei kein passender Volume-Set gefunden, auf dem die Datei angelegt werden kann, werden auch die übrigen Volume-Sets des SM-Pubsets in die Auswahl einbezogen, die im Zustand „normal use“ und nicht mit einer Allokierungssperre versehen sind. Die Benutzer, denen die Möglichkeit der Direktattributierung nicht entzogen wurde, sollten vom Systembetreuer angehalten werden, sich bei Direktattributierung auf die Kombinationen von Dateiattributen zu beschränken, die durch die Volume-Sets der Default-Volume-Set-Liste unterstützt werden. Sie können sich darüber durch das Kommando `SHOW-PUBSET-FILE-SERVICE` informieren. Werden bei der Direktattributierung andere Kombinationen von Dateiattributen angegeben, können die vom Systembetreuer eingesetzten Allokierungsstrategien gestört werden. Die Dateien gelangen dann auf Volume-Sets, die explizit definierten Volume-Set-Listen angehören, und damit im Allgemeinen für eine spezielle Nutzung vorgesehen sind.

#### 4. Änderung des gewünschten Storage-Services

Wird einer bestehenden Datei eine andere Storage-Klasse zugewiesen oder werden die Dateiattribute durch Direktattributierung geändert, kann dies implizit auch die Zuordnung einer anderen Volume-Set-Liste bewirken. Im Fall, dass die neuen Dateiattribute mit dem Volume-Set, auf dem die Datei bisher liegt, nicht verträglich sind, wird die Datei automatisch durch das System auf einen anderen Volume-Set verlagert. Dabei kommt auch die neue Volume-Set-Liste zum Tragen. Andernfalls behält die Datei ihren bisherigen Ablageort zunächst bei. Die geänderten Attribute und die neue Volume-Set-Liste werden erst dann wirksam, wenn die Datei nach dem Verdrängen auf eine Hintergrundebene in die Verarbeitungsebene zurückgeholt wird und dabei der für die Datei am besten geeignete Volume-Set erneut ermittelt wird (z.B. bei einer Pubset-Reorganisation)

Wenn auf einem SM-Pubset keine Volume-Set-Listen eingerichtet sind, gehören automatisch alle Volume-Sets des SM-Pubsets zu der Default-Volume-Set-Liste. Die in den vorangegangenen Abschnitten dargestellten Szenarien können damit als Spezialfall eines SM-Pubsets mit genau einer Volume-Set-Liste betrachtet werden, der alle Volume-Sets angehören und die bei jeder Volume-Set-Auswahl herangezogen wird.

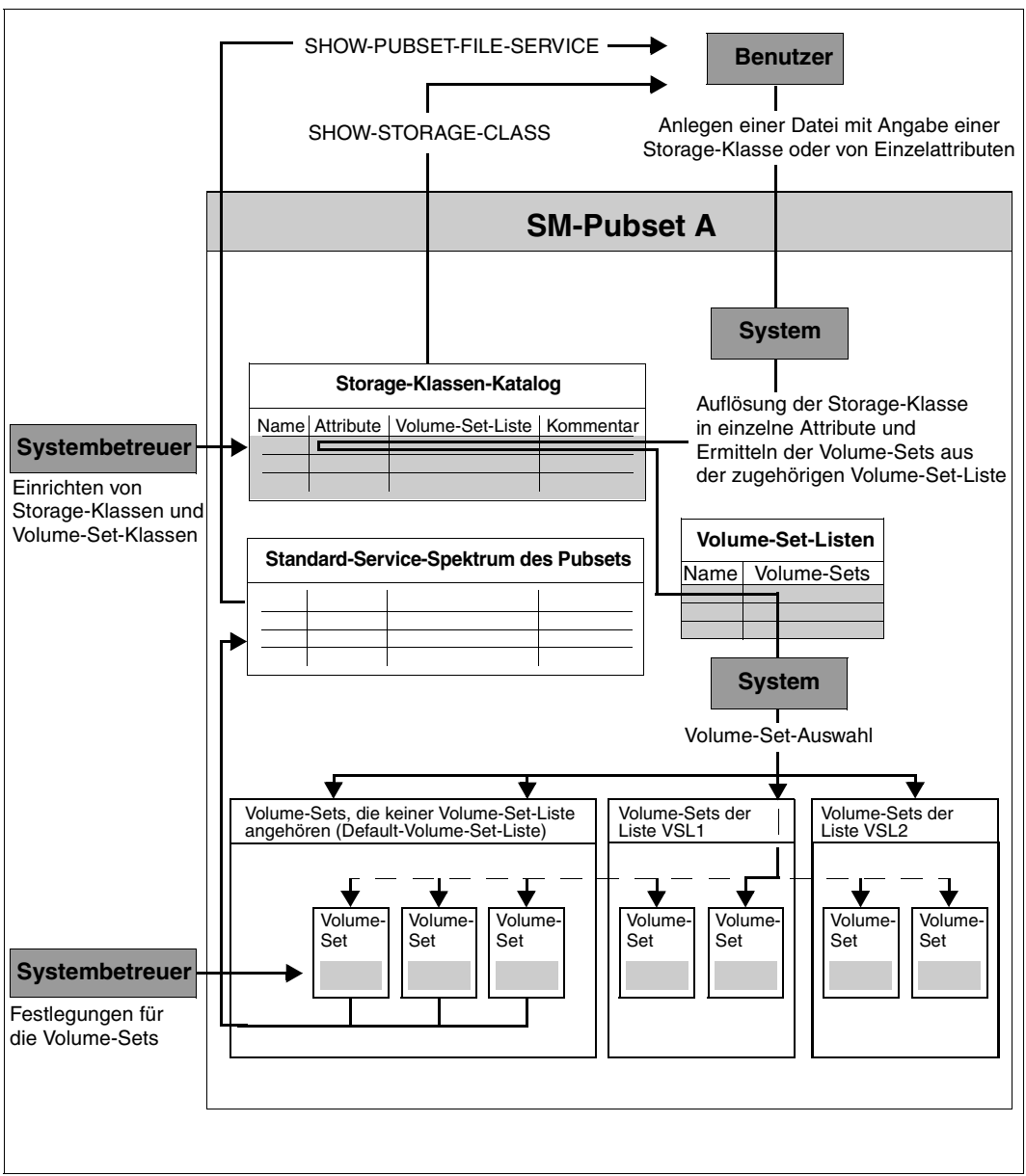


Bild 23: SM-Pubset mit Storage-Klassen mit Volume-Set-Listen

## Exemplarische Anwendungen für Storage-Klassen mit Volume-Set-Listen

### *Differenzierte Ressourcennutzung*

In den Eigenschaftsprofilen der Volume-Sets, welche der Systembetreuer für die Volume-Set-Selektion bereitstellt, können die unterschiedlichen Merkmale der einzelnen Medien nur bis zu einem bestimmten Detaillierungsgrad berücksichtigt werden. Beispielsweise wird nicht unterschieden, ob eine sehr hohe Performance durch die Zuschaltung eines schreibsicheren GS-Caches oder durch die Verwendung hochperformanter Volumes, wie im Globalspeicher emulierter Volumes, erreicht wird. Entsprechend sind auch die Dateiattribute, welche zur Spezifikation der Anforderungen für eine Datei zur Verfügung stehen, in ihrer Genauigkeit begrenzt. Als Folge davon können bei Direktattributierung bzw. bei Verwendung von Storage-Klassen ohne Volume-Set-Listen die unterschiedlichen Eigenschaften der einzelnen Medien nur bis zu einem bestimmten Detaillierungsgrad gezielt genutzt werden. Dies ist insbesondere für große SM-Pubsets, die gleichzeitig viele verschiedene Medien umfassen, problematisch.

Um eine weitergehende Differenzierung zu erreichen, gruppiert der Systembetreuer die Volume-Sets mit unterschiedlichen Eigenschaften in Volume-Set-Listen. Den gewünschten Detaillierungsgrad kann er selbst bestimmen. Soll beispielsweise zwischen Volume-Sets aus Fast-Disks und Volume-Sets aus im Globalspeicher emulierten Volumes unterschieden werden (siehe [Bild 23](#)), wird für beide Arten von Volume-Sets jeweils eine eigene Volume-Set-Liste gebildet. Anschließend richtet der Systembetreuer Storage-Klassen ein, denen die Volume-Set-Listen zugeordnet werden, z.B. eine Storage-Klasse SPCPRF1 und SPCPRF2. Je nachdem, welche der beiden Storage-Klassen einer Datei zugewiesen werden, wird diese bevorzugt auf im Globalspeicher emulierten Volumes angelegt und erhält das für das jeweilige Medium spezifische Performance-Verhalten.

Wenn Storage-Klassen mit Volume-Set-Listen (wie GSVS1LST und GSVS2LST) als Mittel der Service-Differenzierung benutzt werden, sind die Services durch die zugeordneten Dateiattribute nicht ausreichend charakterisiert, da diese die Unterschiede zwischen den Services nicht im Detail zum Ausdruck bringen. Die über die Dateiattribute nicht mehr darstellbaren Service-Merkmale müssen vom Systembetreuer in dem Kommentarteil der Storage-Klasse beschrieben werden. Sie werden den Benutzern durch das Kommando SHOW-STORAGE-CLASS sichtbar. Bei der Beschreibung der Storage-Klassen sollte sich der Systembetreuer auf die Eigenschaften beschränken, welche die Benutzer für die Auswahl einer geeigneten Storage-Klasse benötigen, d.h. die physikalische Sicht des Systembetreuers sollte soweit möglich auf eine logische Benutzer-Schnittstelle abgebildet werden.



Die Benutzerkontingente für belegten Platz berücksichtigen die unterschiedlichen Qualitäten der durch einen SM-Pubset gebotenen Storage-Services nur in dem Umfang, in dem sich diese in den Dateiattributen für Performance und Verfügbarkeit widerspiegeln. Eine weitergehende Differenzierung wird dadurch ermöglicht, dass durch GUARDS-Schutzprofile die Verwendung einer Storage-Klasse auf bestimmte Benutzer eingeschränkt werden kann. Auf Volume-Sets, welche Volume-Set-Listen angehören, gelangen damit im Normalfall nur Dateien der Benutzer, welche Storage-Klassen verwenden dürfen, denen die entsprechenden Volume-Set-Listen zugewiesen sind. Die Grenzen der Volume-Set-Listen werden aber dann ignoriert, wenn andernfalls Benutzeranforderungen - z.B. auf Grund der aktuellen Belegungssituation - zurückgewiesen werden müssten. Dies ist auch dann der Fall, wenn ein Benutzer durch Direktattribute eine Kombination von Dateiattributen spezifiziert, welche durch Volume-Sets der Default-Volume-Set-Liste nicht erfüllbar ist. Das System zieht dann auch Volume-Sets in die Volume-Set-Selektion ein, welche den vom Systembetreuer definierten Volume-Set-Listen angehören. Die Benutzer sollten solche Angaben vermeiden, da sie dadurch die vom Systembetreuer beabsichtigten Strategien der Speichernutzung unterlaufen. Ein ähnliches Problem ergibt sich, wenn der Systembetreuer Storage-Klassen ungünstig definiert, z.B. indem er einer Storage-Klasse ohne Volume-Set-Liste eine Kombination von Dateiattributen zuweist, die auf der Default-Volume-Set-Liste nicht erfüllbar ist. Die Administration der Storage-Klassen erfordert daher große Sorgfalt.

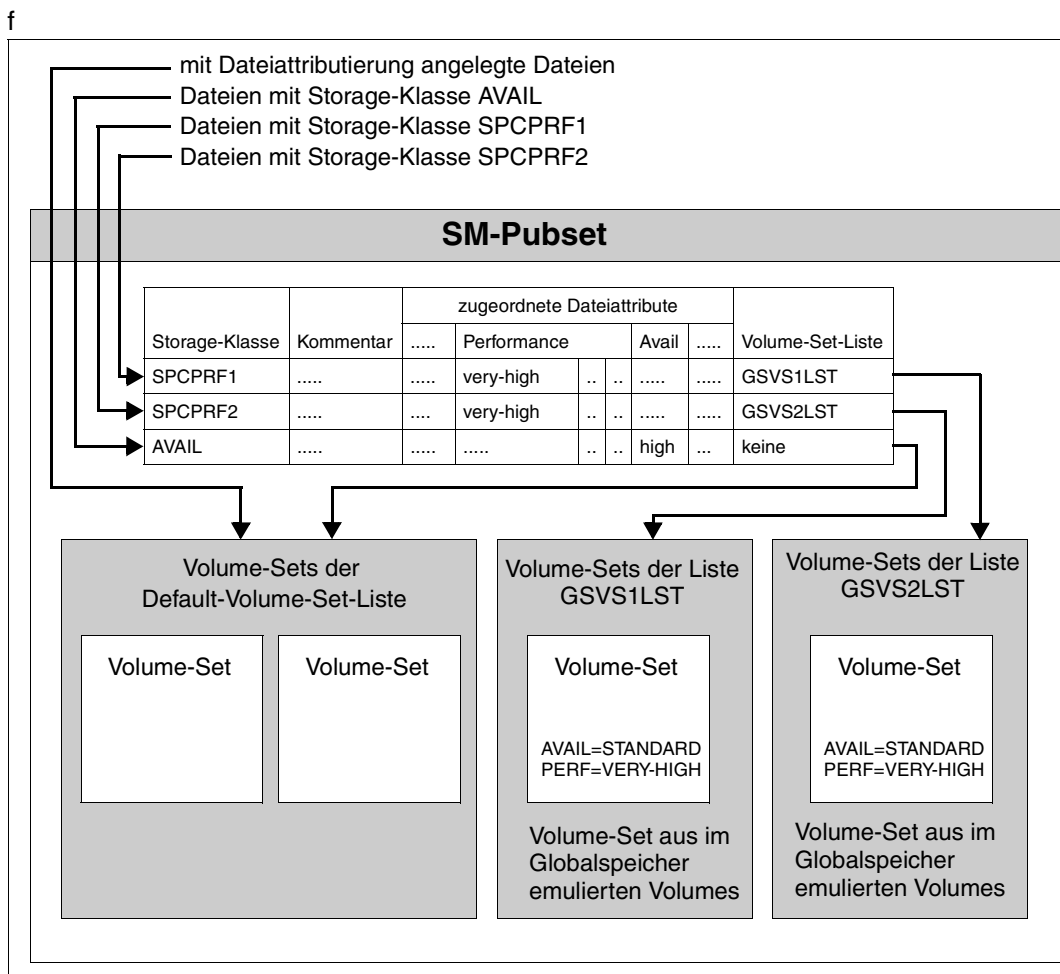


Bild 24: Differenzierte Ressourcennutzung durch Storage-Klassen mit Volume-Set-Listen

Die Volume-Sets des SM-Pubsets, deren Eigenschaften auch ohne besondere Vorkehrungen durch die Volume-Set-Selektion ausreichend berücksichtigt werden, brauchen nicht in Volume-Set-Listen aufgenommen werden. Sie gehören damit der Default-Volume-Set-Liste an und dienen automatisch als Ablageort für Dateien, deren Eigenschaften durch Direktattributierung spezifiziert werden. Für die komfortable Nutzung der durch die Default-Volume-Set-Liste bereitgestellten Services kann der Systembetreuer zusätzlich auch Storage-Klassen ohne Volume-Set-Listen einrichten. Direktattributierung, Storage-Klassen ohne Volume-Set-Listen und Storage-Klassen mit Volume-Set-Listen können damit innerhalb eines SM-Pubsets gleichzeitig so eingesetzt werden, dass sie sich gegenseitig ergänzen.

Die Benutzer benötigen keine Kenntnisse über die Abbildung der Storage-Services auf die Volume-Sets des SM-Pubsets, insbesondere benötigen sie keine Information über Volume-Set-Listen und deren Zuordnung zu Storage-Klassen. Sie orientieren sich über die Kommandos SHOW-PUBSET-FILE-SERVICES und SHOW-STORAGE-CLASS über das Service-Angebot des SM-Pubsets, wobei sie auch die jeweiligen Storage-Klassen-Beschreibungen beachten müssen. Durch das Kommando SHOW-STORAGE-CLASS werden einem Benutzer nur die Storage-Klassen angezeigt, für die er eine GUARDS-Berechtigung besitzt.

### *Separierung der Pubset-Benutzer*

Storage-Klassen mit Volume-Set-Listen sind sehr flexibel verwendbar, um spezifische Strategien der Platzbelegung innerhalb eines SM-Pubsets zum Einsatz zu bringen. Das folgende Beispiel soll dies zeigen, auch wenn es sich dabei um keinen typischen Anwendungsfall handelt:

Es wird ein SM-Pubset betrachtet, der aus einer großen Zahl eigenschaftsgleicher „normaler“ Volume-Sets und einer kleineren Menge von Volume-Sets mit speziellen Eigenschaften besteht (siehe [Bild 25](#)). Während für Dateien mit speziellen Anforderungen der optimale Ablageort weitgehend festgelegt ist, besteht für Dateien ohne besondere Anforderungen eine größere Anzahl gleichwertiger Alternativen. Der Systembetreuer wünscht nun die Volume-Set-Selektion so zu beeinflussen, dass die Standard-Dateien der einzelnen Benutzer nach Möglichkeit nur auf bestimmte Volume-Sets gelangen. Auf diese Weise soll erreicht werden, dass beim Ausfall eines normalen Volume-Sets die Anzahl der betroffenen Benutzer gering ist.

Dazu werden zunächst verschiedene Volume-Set-Listen mit „normalen“ Volume-Sets gebildet. Anschließend werden Storage-Klassen eingerichtet, denen die Volume-Set-Listen zugeordnet werden. Jede dieser Storage-Klasse repräsentiert damit eine Menge von normalen Volume-Sets. Je nachdem, auf welche Volume-Sets die Standard-Dateien eines Benutzers bevorzugt gelangen sollen, wird ihm eine entsprechende Default-Storage-Klasse zugewiesen. Werden für eine Datei keine expliziten Anforderungen gestellt (durch Direktattribute bzw. durch explizite Angabe einer Storage-Klasse), wird ihr die Default-Storage-Klasse ihres Eigentümers zugeordnet. Sie gelangt damit bevorzugt auf einen Volume-Set aus der Volume-Set-Menge des jeweiligen Benutzers. Dateien mit besonderen Anforderungen gelangen auf die Volume-Sets mit speziellen Eigenschaften. Diese enthalten daher im Allgemeinen Dateien vieler unterschiedlicher Benutzer. Bei einem Ausfall ist der Schaden breit gestreut. Der Systembetreuer sollte daher in Betracht zu ziehen, sie besonders ausfallsicher zu konfigurieren.

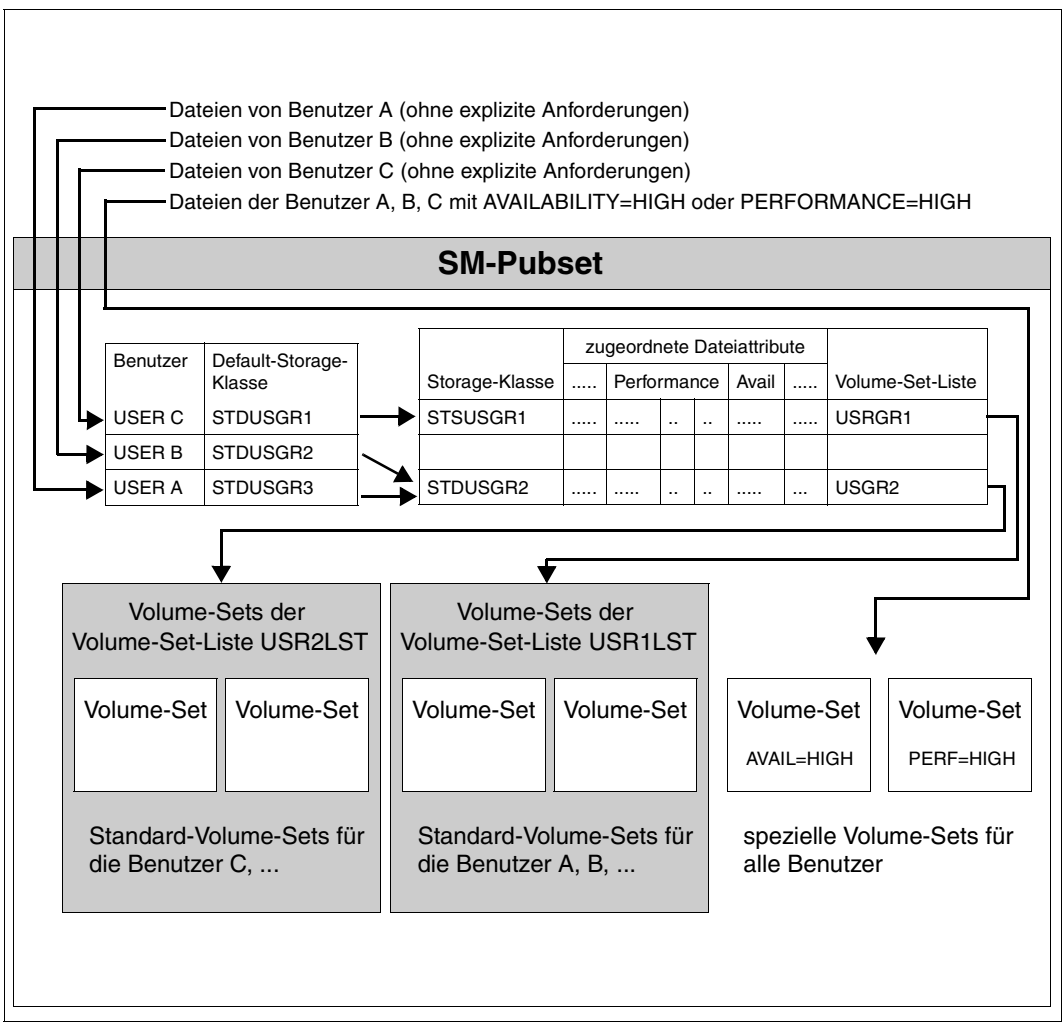


Bild 25: Benutzerseparierung durch Volume-Sets mit Volume-Set-Listen

## 9.1.5 Physikalische Allokierung

SM-Pubsets unterstützen insbesondere auch die physikalische Allokierung. Dies mag überraschen, da das primäre Ziel von SMS darin besteht, den Benutzern zur Beschreibung ihrer Anforderungen Schnittstellen auf logischer Ebene zu bieten und ihnen dadurch die Details der physikalischen Konfiguration zu verbergen. Es ist aber davon auszugehen, dass in speziellen Anwendungsfällen auch künftig der Wunsch besteht, bei der Bestimmung des Dateiablageorts Kriterien zu berücksichtigen, die auf logischer Ebene mit den gegebenen Möglichkeiten nicht ausreichend unterstützt werden können. Ein Beispiel dafür ist die gezielte Verteilung von Datenbankdateien auf unterschiedliche Ausfalleinheiten. Indem die Benutzer selbst die Anordnung der Dateien vorgeben, lassen sich auch solche Aspekte berücksichtigen. Dabei ist es wichtig, dass die Benutzer nicht durch parallele Systemaktivitäten gestört werden, während sie Platz durch physikalische Allokierung anfordern. Auf SF-Pubsets ist die physikalische Allokierung zwar möglich, aber wegen der fehlenden Abschottung paralleler Platzbelegungen durch das System schwer einzusetzen. Im Allgemeinen wird ihre Nutzung auf den Systembetreuer beschränkt. Für Anwendungen, deren Dateien gezielt auf bestimmten Platten angelegt werden sollen, werden häufig Privatplatten herangezogen.

Die physikalische Allokierung in einem SM-Pubset ist im Unterschied zur Nutzung von Privatplatten dadurch charakterisiert, dass sie dort keine separate Welt darstellt, sondern voll in das Konzept des SM-Pubsets eingebettet ist. Abgesehen von der Vorgabe des Ablageorts werden vom System angelegte Dateien und vom Benutzer physikalisch angelegte Dateien einheitlich behandelt, sonstige funktionale Unterschiede z.B. bezüglich der Verwendung von Schutzprofilen, Nutzung von Hardware, Dateiadressierung bestehen nicht. Die einheitliche Behandlung erleichtert den Benutzern die gleichzeitige Nutzung von Dateien, die physikalisch allokiert werden, und sonstigen Dateien. Oft besteht nämlich nur für einzelne Dateien der Bedarf für physikalische Allokierung, während für die übrigen Dateien die Platzbereitstellung durch das System bevorzugt wird. Es ist auch einfach möglich, für bereits physikalische allokierte Dateien die Bindung an ihren Ablageort aufzuheben, wenn dies nützlich erscheint.

Die Integration der physikalischen Allokierung in die Nutzung eines SM-Pubset wird durch folgende Funktionen ermöglicht:

- Durch Allokierungssperren auf einzelnen Volumes oder Volume-Sets können gegenseitige Störungen zwischen dem System und den Benutzern, welche physikalische Allokierung nutzen, vermieden werden.
- Einzelnen Anwendungen können innerhalb des SM-Pubsets Volume-Sets durch Nutzung der physikalischen Allokierung zugeordnet werden, auf welche nur Dateien der entsprechenden Anwendungen gelangen. Für die einzelnen Anwendungen bilden diese Volume-Sets abgegrenzte Ausfalleinheiten, d.h. für eine Anwendung ergeben sich nur dann Auswirkungen, wenn Platten ihres Volume-Sets ausfallen (sofern der Control-Volume-Set des SM-Pubsets funktionsfähig ist). Der Ausfall der übrigen Volumes betrifft die Anwendung nicht. Ein SM-Pubset kann damit beliebig groß werden, ohne dass

sich das Ausfallrisiko für die Anwendung erhöht, d.h. die Größe einer Ausfalleinheit und die Größe der Adressierungseinheit können (wie bei Privatplatten) voneinander entkoppelt festgelegt werden.

- Für Dateien, die durch physikalische Allokierung angelegt werden, ergeben sich (im Unterschied zu Privatplatten) hinsichtlich der übrigen Pubset-Funktionalität (Nutzung von GUARDS-Schutzprofilen, Ressourcenschutz durch Benutzerkontingente, Einschränkungen hinsichtlich Hardware, etc.) keine Einschränkungen. Auch Funktionen wie die Verdrängung der Dateien auf Hintergrundebenen können genutzt werden, falls die Benutzer dies wünschen.
- Die Grenzen zwischen Systemautomatismen und physikalischer Allokierung sind nicht starr, sondern können flexibel angepasst werden. Für Benutzer, die in einem SM-Pubset bisher physikalische Allokierung verwendet haben, und künftig stärker von der SMS-Funktionalität Gebrauch machen wollen, ist ein einfacher (und stufenloser) Übergang möglich. Er kann innerhalb des SM-Pubsets erfolgen, ohne dass Dateien verlagert werden müssen oder die Adressierung der Dateien sich ändert.
- Da die physikalische Allokierung starke Einflussmöglichkeiten auf die Platzbelegung eines SM-Pubsets bietet, kann sie auf bestimmte Benutzer eingeschränkt werden.

### **Detaillierungsgrad für die Vorgabe des Ablageorts**

Mithilfe der Operanden VOLUME-SET, VOLUME und SPACE=\*ABSOLUTE(...) in den Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES ist es den Benutzern möglich, bei dem Anlegen oder bei dem Erweitern einer Datei ihren Ablageort innerhalb eines Pubsets auf verschiedenen Detaillierungsstufen zu spezifizieren:

- physikalische Allokierung auf Block-Ebene  
Der Benutzer bestimmt sowohl das Volume als auch die Blöcke innerhalb dieses Volumens, auf dem Platz bereitgestellt werden soll.
- physikalische Allokierung auf Volume-Ebene  
Der Benutzer bestimmt, auf welchen Volumes des Pubsets die Datei angelegt werden soll. Er interessiert sich aber nicht dafür, wo innerhalb der Volumes der Platz bereitgestellt wird.
- physikalische Allokierung auf Volume-Set-Ebene  
Der Benutzer bestimmt, auf welchem Volume-Set die Datei abzulegen ist. Er interessiert sich aber nicht dafür, wo innerhalb des Volume-Sets dies geschieht.

## Berechtigung zur physikalische Allokierung

Das TSOS-Privileg berechtigt generell auf allen Benutzerkennungen zur physikalischen Allokierung. Zusätzlich kann der Systembetreuer weiteren Benutzern durch die Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES (Operand PHYSICAL-ALLOCATION) die Berechtigung zur physikalischen Allokierung geben. Sie bedeutet, dass für die entsprechenden Benutzerkennungen (Userid im Pfadnamen) auf allen Platten des Pubsets Dateien physikalisch angelegt werden dürfen. Beim Einrichten einer Benutzerkennung erfolgt die Initialisierung „nicht berechtigt zur physikalischen Allokierung“. Die physikalische Allokierung auf Privatplatten wie auch auf den Platten von Work-Volume-Sets ist allen Benutzern erlaubt.

## Informationsfunktionen für Benutzer der physikalischen Allokierung

Für die physikalische Allokierung werden im Allgemeinen detailliertere Kenntnisse der physikalischen Konfiguration des SM-Pubsets benötigt, die Anzeige des Service-Angebots auf logischer Ebene ist im Allgemeinen nicht ausreichend. Hierfür stehen Auskunftsfunktionen zur Verfügung. Diese Funktionen sind zum Teil nur dem Systembetreuer (TSOS-Privileg) verfügbar. Ggf. muss sich ein Benutzer, der physikalisch Platz belegen möchte, aber nicht über das TSOS-Privileg verfügt, mit dem Systembetreuer in Verbindung setzen.

- Die Benutzer (auch ohne TSOS-Privileg) können durch das Kommando SHOW-PUBSET-CONFIGURATION Einblick in die physikalische Konfiguration des Pubsets gewinnen. Die bereitgestellte Information ist in Informationsblöcke gruppiert.
  - Der Informationsblock SUMMARY stellt Übersichtsinformationen für den Pubset bereit.
  - Der Informationsblock PHYSICAL-CONFIGURATION zeigt die Volume-Sets des SM-Pubsets, die ihnen zugehörigen Volumes, sowie den Device-Typ und die Allokierungsbeschränkungen für die einzelnen Volumes an.
  - Über den Informationsblock VOLUME-SET-PARAMETERS werden für die einzelnen Volume-Sets Nutzungsart (STANDARD, HSMS-CONTROLLED, WORK), Allokierungs- und Nutzungsbeschränkungen sowie die Eigenschaftsprofile für die Volume-Set-Selektion (Performance, Verfügbarkeit, Struktur, Allocation-Unit) ausgegeben. Die physikalische Konfiguration, welche die vom Systembetreuer vorgenommenen Festlegungen für Performance und Verfügbarkeit realisiert, wird nur teilweise angezeigt (z.B. Cache-Zuordnung oder im Globalspeicher emulierte Volumes).
- Das Kommando SHOW-MASTER-CATALOG-ENTRY kann auch auf SM-Pubsets und Volume-Sets angewandt werden. Es ist auch den Benutzern erlaubt.

- Für die physikalische Allokierung auf Blockebene ist die Kenntnis der Platzbelegung bis auf die Ebene von Blöcken erforderlich. Informationen werden durch die Utility SPCCNTRL bereitgestellt, deren Nutzung aber dem Systembetreuer (TSOS-Privileg) vorbehalten ist.
- Benutzer-Programme erhalten durch die STAMCE-Schnittstelle Kenntnisse über den physikalischen Aufbau von SM-Pubsets. Pubset-globale Zustände (accessible, inaccessible,...) werden bei SM-Pubsets über den MRSCAT-Eintrag des Pubsets angezeigt. Konfigurationsdaten bezüglich Cache-Zuordnungen, Format etc., sind bei SM-Pubsets volume-set-spezifisch und finden sich in den MRSCAT-Einträgen der einzelnen Volume-Sets. Um zu ermitteln, welche Volume-Sets zu einem SM-Pubset gehören, bietet die STAMCE-Schnittstelle geeignete Selektionskriterien. Ferner sind in den MRSCAT-Einträgen der Volume-Sets entsprechende Verweise enthalten.

### Koexistenz von physikalischer Allokierung und Systemautomatismen

Bild 26 zeigt die Koexistenz von physikalischer Allokierung und Systemautomatismen beim Anlegen einer Datei. Für Benutzer, die physikalisch allokierten möchten, ist es oft wichtig, für eine bestimmte Zeit auf einem oder mehreren Volumes die ausschließliche Kontrolle über die Allokierungen zu besitzen. Wir betrachten zwei Beispiele:

- kurzfristige Nutzung von Volumes für physikalische Allokierung

Es kann wichtig sein, einzelne Dateien (z.B. Systemdateien) gezielt auf Volumes zu platzieren, die im Normalfall nicht für physikalische Allokierung reserviert sind. Ein Beispiel hierfür ist die gezielte Erweiterung des Dateikatalogs. Der Allokierung dürfte dabei meist eine Analyse des bereits belegten Platzes (z.B. mithilfe von SPCCNTRL) durch den Systembetreuer vorausgehen, um einen möglichst geeigneten Ablageort zu ermitteln. Führt das System unterdessen andere Allokierungen auf den betroffenen Volumes aus, so kann das Ergebnis der Analyse zunichte gemacht werden, und die geplante physikalische Allokierung scheitert.

Um dies zu vermeiden, wird mit dem Kommando MODIFY-PUBSET-RESTRICTIONS Operand RESTRICTION=\*ALLOCATION-ON-VOLUME(MODE=\*PHYSICAL-ONLY(VOLUME=...)) die Möglichkeit geboten, auf einzelnen Volumes eines SM-Pubsets alle Allokierungen (inklusive Sekundärallokierung) mit Ausnahme der physikalischen Allokierungen zu verhindern. Die Sperre kann vor der Durchführung der Platzanalyse gesetzt und nach der Durchführung der Allokierung wieder aufgehoben werden. Setzen und Rücksetzen sind nur dem Systembetreuer gestattet, da sich Auswirkungen auf die gesamte Pubset-Nutzung ergeben (z.B. Überwachung der Sättigungsschwellwerte). Auch die von SPCCNTRL gebotenen Auskunftsfunktionen über die Belegung der Volumes auf Blockebene sind nur dem Systembetreuer zugänglich. Wünschen Benutzer ohne TSOS-Privileg auf einem ansonsten allgemein genutzten Volume die Lage einzelner Dateien gezielt vorzugeben, sollten sie die Platzbelegung in Zusammenarbeit mit dem Systembetreuer vornehmen.



- benutzerkontrollierte Domänen

Ein andere Nutzungsart für physikalische Allokierung, die bei Privatplatten häufig sein dürfte, besteht darin, dass bestimmte Volumes längerfristig für die Dateien einer bestimmten Anwendung reserviert sein sollen. Erreicht wird dies dadurch, dass auf diese Volumes nur Dateien gelangen, die von den Benutzern explizit dort angelegt werden. Die gezielte Verteilung der Dateien über die Volumes kann beispielsweise zur Bildung bestimmter voneinander abgegrenzter Ausfalleinheiten dienen. Bei dieser Nutzung ist die gezielte Vorgabe des Ablageorts innerhalb der Menge der für die Anwendung reservierten Volumes meist ohne Bedeutung, der Zwang zu einer detaillierteren Angabe dürfte sogar oft unerwünscht sein. Auch sollen die Dateien während der Verarbeitung bei Bedarf automatisch durch das System vergrößert werden (Sekundärallokierung), allerdings soll dabei die Platzvergabe innerhalb der der Anwendung zugeordneten Volume-Menge erfolgen.

In SM-Pubsets können solche Anforderungen erfüllt werden, indem der Systembetreiber für eine Anwendung einen eigenen Volume-Set bereitstellt und durch das Kommando `MODIFY-PUBSET-RESTRICTIONS` mit der Allokierungsrestriktion `RESTRICTION=*NEW-FILE-ALLOCATION (MODE=*PHYSICAL-ONLY)` belegt. Sie bewirkt, dass auf diesen Volume-Set nur Dateien gelangen, welche ein Benutzer durch physikalische Adressierung dort anlegt. Für Sekundärallokierungen ergeben sich keine Einschränkungen. Gibt ein Benutzer den Ablageort auf Volume-Set-Ebene vor, nutzt er implizit die Systemroutinen zur Platzbereitstellung innerhalb des Volume-Sets. Bei Bedarf kann er den Ablageort aber auch detaillierter auf der Ebene der einzelnen Volumes oder auf Blockebene spezifizieren. Da der Benutzer die Kontrolle über die Platzvergabe auf dem Volume-Set hat, kann er selbst verhindern, dass er bei detaillierter Platzbestimmung durch parallele Systemaktivitäten gestört wird. Beeinträchtigungen durch andere zur physikalischen Allokierung berechnete Benutzer müssen durch Absprachen verhindert werden. Wenn die Dateien einer Anwendung auf einen bestimmten Volume-Set konzentriert werden, ist sie vom Ausfall sonstiger Volumes nicht betroffen. Im Fall, dass bestimmte Dateien einer Anwendung auf voneinander unabhängige Ausfalleinheiten verteilt werden sollen, müssen mehrere Volume-Sets für sie vorgesehen werden.

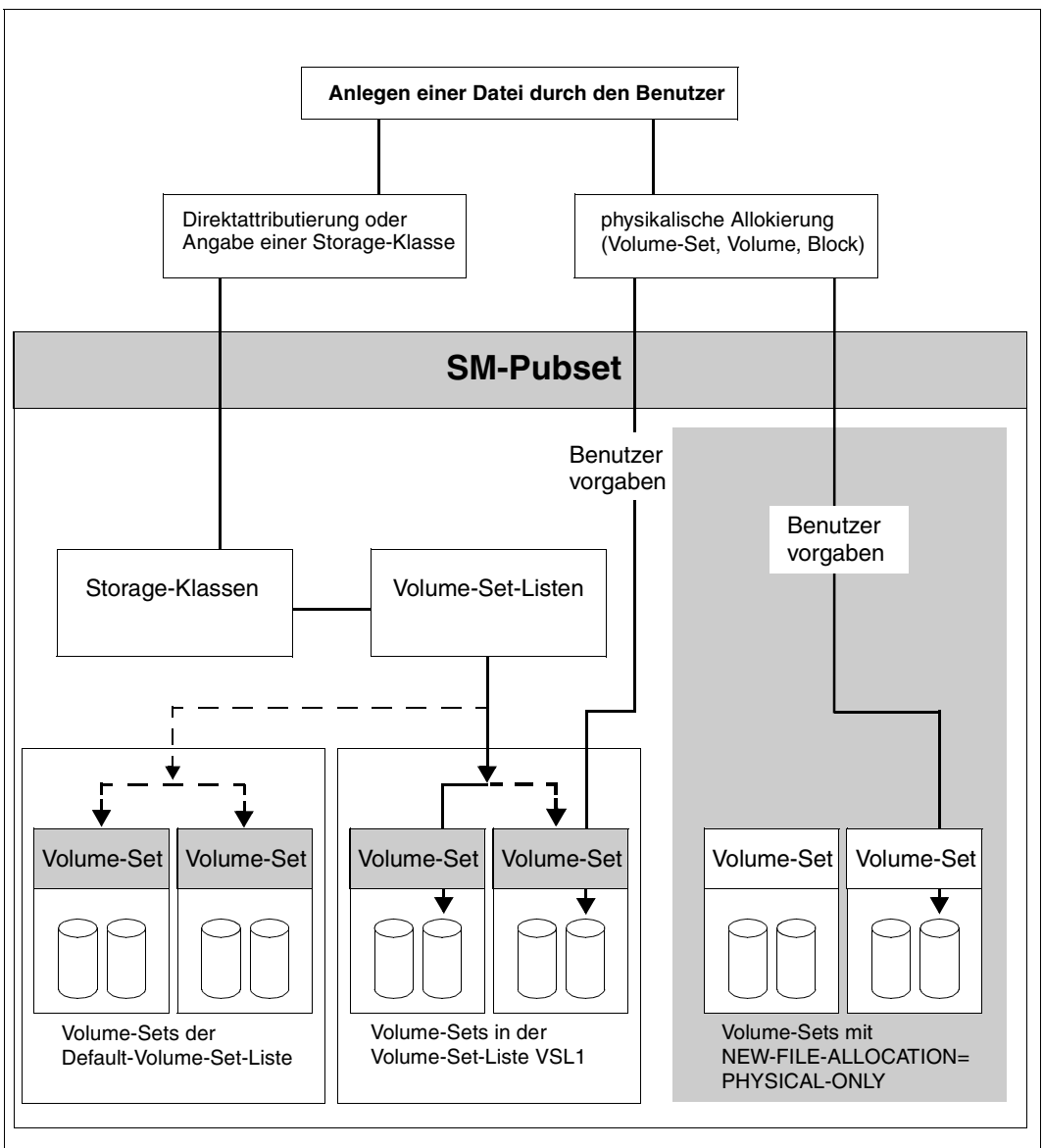


Bild 26: Koexistenz von physikalischer Allokierung und Systemautomatismen beim Anlegen einer Datei

Das Setzen der Allokierungssperre `NEW-FILE-ALLOCATION=*PHYSICAL-ONLY` ist auch für Volume-Sets möglich, die in Volume-Set-Listen enthalten sind. Da die Volume-Sets in diesem Fall bei der Volume-Set-Selektion durch das System nicht herangezogen werden, hat das Setzen der Sperre die gleiche Wirkung, als ob der Volume-Set aus der Volume-Set-Liste entfernt würde. Bei der Ermittlung der Kombinationen von Dateiattributen, welche von den Volume-Sets einer Volume-Set-Liste unterstützt werden (durch das Kommando `SHOW-PUBSET-FILE-SERVICES`, Operand `VOLUME-SET-LIST`), bleiben daher gegen Neuallokierung gesperrte Volume-Sets unberücksichtigt. Gleiches gilt auch für die Ermittlung des über Direktattributierung zugänglichen Service-Spektrums des SM-Pubsets, welches sich aus der Default-Volume-Set-Liste ergibt. Wird die Sperre gegen Neuallokierung rückgesetzt, kommt die Zugehörigkeit zu einer Volume-Set-Liste sowohl in den Auskunftsfunktionen als auch in dem Allokierungsverhalten des Systems wieder zum Tragen.

### **Rolle der Dateiattribute bei physikalischer Allokierung**

Bei der physikalischen Allokierung bestimmt der Benutzer selbst den Volume-Set, auf dem eine Datei angelegt wird. Für die Volume-Set-Selektion durch das System ist daher eine Spezifikation bzgl. Performance, Verfügbarkeit u.a. ohne Bedeutung. Die Dateiattribute erfüllen aber noch weitere Funktionen, die auch für Dateien, die durch physikalische Allokierung angelegt sind, wichtig sind:

- Ist ein Volume-Set mit einem Cache ausgestattet, so steuern die performance-bezogenen Dateiattribute die Cache-Nutzung.
- Das Format des Volume-Sets, auf dem eine Datei liegt, legt im Allgemeinen nicht eindeutig das Format der auf ihm liegenden Dateien (K, NK2, NK4) fest. Zur genauen Format-Spezifikation sind zusätzliche Benutzerangaben erforderlich.
- Aus den Dateiattributen für Performance, Verfügbarkeit, Arbeitsdatei geht hervor, welche Benutzerkontingente für den auf dem Pubset belegten Platz belastet werden.

Aus diesem Grund müssen für Dateien auch dann die Dateiattribute für Performance, Verfügbarkeit, Arbeitsdatei spezifiziert werden, wenn sie durch physikalische Allokierung angelegt werden. Den Besonderheiten der physikalischen Allokierung wird durch eine spezielle Validierung und Zuweisung von Default-Werten der Benutzerangaben Rechnung getragen:

- Wenn der Benutzer eine Datei auf einen Volume-Set mit bestimmten Eigenschaften legt, bringt er dadurch implizit bestimmte Anforderungen für die Datei zum Ausdruck. Spezifiziert er die Dateiattribute nicht explizit, werden sie automatisch unter Berücksichtigung der Eigenschaften des vorgegebenen Volume-Sets bestimmt.

- Es wird verhindert, dass ein Benutzer durch physikalische Allokierung die Kontingentüberwachung unterläuft, beispielsweise indem einer Datei, die auf einem hochverfügbaren Volume-Set angelegt wird, das Attribut AVAILABILITY=\*STD zugewiesen wird. Hierzu werden gegebenenfalls Benutzerangaben modifiziert.
- Im Fall, dass die von dem Benutzer spezifizierten Attribute mit den Eigenschaften des vorgegebenen Volume-Sets nicht verträglich sind, wird die Benutzeranforderung zurückgewiesen.
- Die Spezifikation der ablageortrelevanten Dateiattribute ist bei physikalischer Allokierung nur durch Direktattributierung möglich. Physikalische Allokierung und die Benutzung von Storage-Klassen schließen sich gegenseitig aus.

Tabelle 5 verdeutlicht diese Prinzipien für das Dateiattribut AVAILABILITY.

<b>AVAILABILITY des Volume-Sets, auf dem die Datei angelegt wird</b>	<b>vom Benutzer spezifiziertes Dateiattribut AVAILABILITY</b>	<b>resultierender Wert des Dateiattributs AVAILABILITY</b>
HIGH	nicht explizit angegeben HIGH STD	HIGH (a) HIGH HIGH (b)
STD	nicht explizit angegeben HIGH  STD	STD Angabe wird zurückgewiesen, da sie mit dem Volume-Set nicht verträglich ist (c) STD

Tabelle 5: Ermittlung des Dateiattributs AVAILABILITY bei physikalischer Allokierung

## Physikalische Allokierung und automatische Verlagerung von Dateien

Dateien, welche durch physikalische Allokierung angelegt worden sind, müssen bei der längerfristigen Dateiverdrängung auf Hintergrundebenen und bei der Verlagerung der Dateien innerhalb der Verarbeitungsebene speziell behandelt werden, damit die vom Benutzer gewünschte Fixierung an einen bestimmten Ablageort nicht unwirksam wird. Erreicht wird dies dadurch, dass beim Neuanlegen einer Datei durch physikalische Allokierung die Migrationssperren S0-MIGRATION und MIGRATE entsprechend initialisiert werden. Diese erfolgt abhängig von dem Detaillierungsgrad, mit dem der Benutzer den Ablageort spezifiziert:

- Im Fall, dass der Benutzer nur den Volume-Set vorgibt, wird davon ausgegangen, dass die Datei während der Verarbeitungsphase immer auf dem vorgegebenen Volume-Set liegen soll, dass sie aber innerhalb des Volume-Sets z.B. im Rahmen einer Reorganisation verlagert werden darf. In inaktiven Phasen darf die Datei auf Hintergrundebenen verdrängt werden, sie muss aber bei einem späteren Zurückholen in die Verarbeitungsebene wieder auf den ursprünglichen Volume-Set gelangen. Diesem Anforderungsprofil entsprechen die Voreinstellungen S0-MIGRATION=\*FORBIDDEN sowie MIGRATE=\*ALLOWED, die automatisch erfolgen.
- Erfolgt die Vorgabe des Ablageorts auf Volume- oder Block-Ebene, wird angenommen, dass die Datei auch innerhalb des Volume-Sets nicht verlagert werden darf. Auch die Verdrängung auf Hintergrundebenen ist nicht sinnvoll. Um sicherzustellen, dass die Datei bei dem Zurückholen wieder auf den ursprünglichen Platz gelangt, müsste dieser für die Datei reserviert bleiben, während sie sich auf einer Hintergrundebene befindet. Da der Platz nicht anderweitig nutzbar wäre, bringt die kostengünstigere Speicherung auf der Hintergrundebene keinen Nutzen. Aus diesen Gründen erfolgen bei physikalischer Allokierung auf Volume-Ebene oder Blockebene die Voreinstellungen S0-MIGRATION=\*FORBIDDEN, MIGRATE=\*FORBIDDEN.

Die Migrationssperren S0-MIGRATION und MIGRATE können auch explizit durch die Kommandos CREATE-FILE bzw. MODIFY-FILE-ATTRIBUTES vorgegeben oder modifiziert werden, falls die automatischen Voreinstellungen nicht passen oder später geändert werden sollen. Da den Benutzern dadurch ähnlich starke Einflussmöglichkeit auf die Platznutzung eines Pubsets gegeben wird wie durch physikalische Allokierung, sind die Einstellungen S0-MIGRATION=\*FORBIDDEN und MIGRATE=\*FORBIDDEN an die Berechtigung zur physikalischen Allokierung geknüpft.

Die Berücksichtigung der Migrationssperren bei einem Migrations-Auftrag erfordert keine besonderen Maßnahmen durch den Systembetreuer. Sie erfolgt automatisch durch HSMS, welches Dateien mit MIGRATE-FORBIDDEN von der Migration auf Hintergrundebenen ausnimmt und auf Hintergrundebenen befindliche Dateien mit der Sperre S0-MIGRATION-FORBIDDEN beim Zurückholen in die Verarbeitungsebene wieder auf ihren Herkunfts-Volume-Set zurückbringt, sofern dieser in dem Pubset noch vorhanden ist.

## Übergang von physikalischer Allokierung zur Nutzung von SMS-Funktionen

Die physikalische Allokierung gibt dem Benutzer erweiterte Kontrollmöglichkeiten, beschränkt aber die Leistungen und den Komfort, die ihm durch den Systembetreuer bzw. das System geboten werden. Es kann sich daher der Wunsch ergeben, die benutzerkontrollierte Datenadministration aufzugeben bzw. einzuschränken und in verstärktem Umfang SMS-Funktionen zu nutzen. Innerhalb eines SM-Pubsets kann der Übergang auf folgende Weise geschehen:

- Die Kommandoprozeduren sind so anzupassen, dass keine physikalische Allokierung mehr erfolgt und auch nicht explizit Migrationssperren gesetzt werden. Beim Anlegen der Dateien müssen diesen geeignete Storage-Services (durch Zuordnung geeigneter Dateiattribute oder Storage-Klassen) zugewiesen werden, damit sie durch das System auf passenden Volume-Sets angelegt werden. Für bereits existierende Dateien sind die Migrationssperren rückzusetzen.
- Waren Volume-Sets bisher mit Allokierungssperren belegt, um sie der Anwendung zur alleinigen Nutzung bereitzustellen, können diese von dem Systembetreuer aufgehoben werden. Die betroffenen Volume-Sets sind anschließend allgemein nutzbar. Da die Migrationssperren für die darauf befindlichen Dateien rückgesetzt worden sind, sind die Dateien nicht mehr an den Volume-Set gebunden. Sie können damit auch auf andere Volume-Sets des SM-Pubsets gelangen.

## 9.1.6 Privatplattenablösung durch SM-Pubsets

Oft steht die physikalische Allokierung im Zusammenhang mit Privatplattennutzung und wird zur gezielten Dateiverteilung genutzt um die Ausfalleinheiten, die bei einem Plattenausfall betroffen sind, möglichst klein zu halten. Eine Anwendung, deren Dateien auf Privatplatten liegen, ist nur vom Ausfall der Platten betroffen, auf denen ihre Dateien liegen. Indem ihre Dateien auf wenige Platten konzentriert werden, kann das Risiko, durch Plattenausfälle beeinträchtigt zu werden, gering gehalten werden. Kleine Ausfalleinheiten bedingen aber nicht auch kleine Adressierungseinheiten. Dateien, die auf verschiedenen Privatplatten liegen, können über eine gemeinsame Catid adressiert werden (indem sie in einen bestimmten SF-Pubset importiert werden), auch wenn sie auf Platten liegen, die verschiedenen Ausfalleinheiten zugeordnet sind. Die Größe der Ausfalleinheiten ist von der Größe der Adressierungseinheiten entkoppelbar.

SF-Pubsets bieten hierfür keine gleichwertige Alternative. Für Anwendungen, deren Dateien auf einem SF-Pubset liegen, ist jede Platte des SF-Pubsets ausfallkritisch. Die Verwendung großer Pubsets erhöht folglich das Ausfallrisiko für alle Anwendungen, die ihre Dateien auf dem Pubset ablegen. Wird aber an Stelle eines großen SF-Pubsets eine größere Anzahl kleiner SF-Pubsets eingesetzt, um die Dateien auf kleinere und voneinander unabhängige Ausfalleinheiten zu verteilen, müssen für die Adressierung der Dateien verschiedene Catids benutzt werden, was im Allgemeinen unerwünscht ist. Eine große Anzahl kleiner SF-Pubset ist auch aufwändig zu administrieren, weil jeder SF-Pubset eigens gepflegt werden muss (Benutzerkennungen, GUARDS-Schutzprofile, etc.).

SM-Pubsets mit dem gesamten Pubset als Adressierungseinheit, den einzelnen Volume-Sets als separaten Ausfalleinheiten und der physikalischen Allokierung als Instrument der Dateiverteilung erfüllen die Anforderungen der Privatplattenbenutzer. Da die für Privatplatten bestehenden Einschränkungen nicht für sie gelten, bieten sie einen attraktiven Anreiz für die Privatplattenablösung. Der Übergang auf SM-Pubsets erzwingt aber nicht die Privatplattenablösung. Aus Gründen der Kompatibilität und damit eine kontinuierliche Umstellung möglich ist, werden die Privatplatten durch SM-Pubset in der selben Weise unterstützt wie durch SF-Pubsets.

Im Folgenden soll an einem Beispiel für Privatplattenablösung dargestellt werden, wie sich dabei eine bestehende Privatplattenkonfiguration auf eine Volume-Set-Konfiguration innerhalb eines SM-Pubsets abbilden lässt. Zur Illustration des Beispiels dient [Bild 27](#).

### *Ausgangskonfiguration*

In der Ausgangskonfiguration werden die Privatplatten in Verbindung mit dem SF-Pubset X genutzt. Die auf ihnen liegenden Dateien werden in den Katalog des SF-Pubsets importiert. Für ihre Adressierung werden Pfadnamen verwendet, in welche die Kennung von Pubset X als Catid eingeht. Die Dateien der Anwendung A werden beliebig über mehrere Volumes verteilt. Die Dateien von Anwendung B werden so angeordnet, dass bei dem Ausfall eines der beiden Volumes nicht alle Dateien gleichzeitig betroffen sind.

Zielkonfiguration

In der Zielkonfiguration ist der SF-Pubset X durch den SM-Pubset X ersetzt. Der Control-Volume-Set des SM-Pubset ist besonders ausfallsicher konfiguriert. Der SM-Pubset enthält einen Volume-Set für die Anwendung A, der aus drei Volumes besteht, und zwei Volume-Sets für die Anwendung B mit jeweils einem Volume. Die Volume-Sets der Anwendungen A und B sind mit der Sperre NEW-FILE-ALLOCATION=\*PHYSICAL-ONLY versehen. Dass die einzelnen Anwendungen jeweils nur ihre eigenen Volume-Sets nutzen und sich gegenseitig nicht stören, wird durch Absprache erreicht. Beim Anlegen der Dateien wird jeweils der gewünschte Volume-Set spezifiziert.

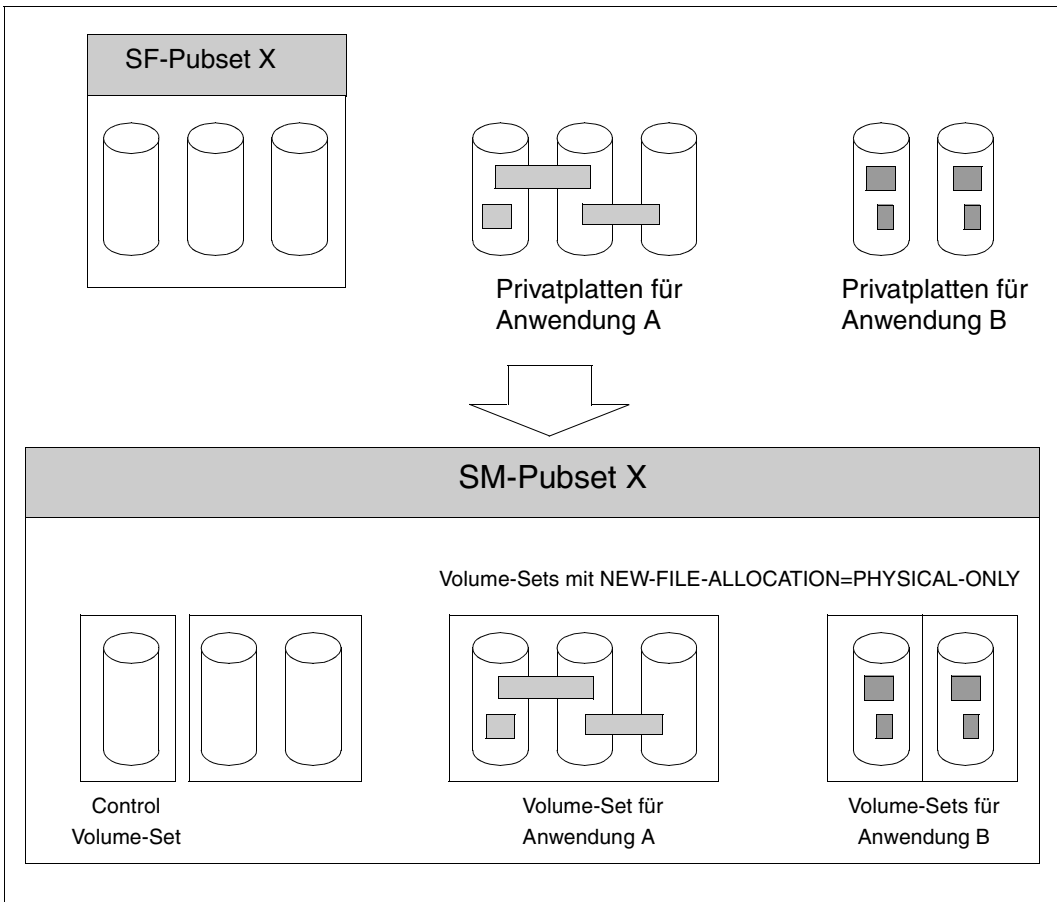


Bild 27: Beispiel für die Privatplattenablösung durch SM-Pubsets



*Für den Umstieg erforderliche Schritte*

Der Übergang von der Ausgangskonfiguration in die Zielkonfiguration kann in folgenden Schritten erfolgen:

- Zunächst wird der SF-Pubset z.B. mit SMPGEN in einen SM-Pubset umgewandelt. Die Nutzung der Privatplatten ändert sich dadurch nicht.
- Anschließend werden die auf den Privatplatten liegenden Dateien gesichert.
- Aus den bisherigen Privatplatten werden mit SIR freie Volume-Sets gebildet, die in den SM-Pubset aufgenommen und mit einer Sperre gegen Neuallokierung versehen werden. Die auf den Privatplatten liegenden Dateien gehen dabei verloren.
- Die Dateisicherungen für die einzelnen Anwendungen werden auf den jeweiligen Volume-Sets eingespielt (Angabe des Volume-Sets beim Restore).
- Die Adressierung der Dateien ändert sich nicht. Die Kommandoprozeduren müssen im Allgemeinen aber angepasst werden. Beispielsweise sind Anweisungen, in welchen bisher die Privatplatten als Ablageort vorgegeben werden, durch Anweisungen zu ersetzen, in welche die entsprechenden Volume-Sets eingehen.

## 9.1.7 Ablageortrelevante Dateiattribute und zugewiesene Voreinstellungen

Bei den für die Auswahl des geeigneten Ablageorts einer Datei relevanten Dateiattributen handelt es sich um Performance-Attribute (PERFORMANCE, USAGE, DISK-WRITE), Verfügbarkeit, Format, Arbeitsdatei (WORK-FILE) sowie das implizite Attribut Dateigröße.

### Festlegen der Dateiattribute beim Neuanlegen einer Datei (CREATE-FILE)

Wenn eine Datei angelegt wird, werden für die Bestimmung der Dateiattribute und ggf. die Zuordnung einer Storage-Klasse zunächst die expliziten Angaben der Benutzer herangezogen. Sind diese nicht vollständig, werden sie durch das System ergänzt. In einzelnen Fällen werden die expliziten Angaben durch das System modifiziert. Insgesamt wird bei SM-Pubsets für die Bestimmung der Dateiattribute ein sehr differenziertes Verfahren angewandt, das durch den Systembetreuer mithilfe der Default-Storage-Klassen und durch das (Pubset-) voreingestellte File-Format pubset- und benutzerspezifisch beeinflusst werden kann. Dies ermöglicht eine flexible Nutzung der verschiedenen Services eines SM-Pubsets, ohne dass die Kommandoprozeduren der Benutzer angepasst werden müssen. Auch der Umstieg von SF-Pubsets auf einen SM-Pubset wird erleichtert, da für Dateiattribute Voreinstellungen möglich sind, welche den Kommandoprozeduren und Benutzerprogrammen auf SM-Pubsets ein zu den SF-Pubsets (weitgehend) kompatibles Systemverhalten gewährleisten. Wie die Zuweisung von Voreinstellungen erfolgt ist im Allgemeinen auch für die Benutzer von Interesse. Sie können sich durch die Kommandos SHOW-USER-ATTRIBUTES (Default-Storage-Klasse, DMS-TUNING-RESOURCES), SHOW-STORAGE-CLASS (Definition der Storage-Klasse) und SHOW-PUBSET-CONFIGURATION (Voreinstellung des File-Format des Pubsets) über die für sie relevanten Parameter für Zuweisung von Voreinstellungen informieren.

Die Ermittlung der Dateiattribute ergibt sich aus einem Zusammenspiel von Storage-Klassen, Default-Storage-Klassen und den zugewiesenen Voreinstellungen der Einzelattribute.

Wenn dem Benutzer die Möglichkeit der Direktattributierung entzogen wurde, werden bei CREATE-FILE die Angaben PERFORMANCE, USAGE, DISK-WRITE, AVAILABILITY, WORK-FILE=\*NO und STORAGE-CLASS=\*NONE ignoriert.

Das Kommando CREATE-FILE erlaubt für die Zuordnung der Dateiattribute folgende Alternativen:

*STORAGE-CLASS=\*NONE (Direktattributierung)*

In diesem Fall liegt Direktattributierung vor, d.h. der Benutzer spezifiziert die Dateiattribute WORK-FILE, PERFORMANCE, USAGE, DISK-WRITE, FILE-PREFORMAT und AVAILABILITY einzeln. Den nicht explizit spezifizierten Attributen werden die jeweiligen attributspezifischen Voreinstellungen zugewiesen. Die Benutzerangaben für Performance werden ggf. durch das System unter Berücksichtigung der Benutzerberechtigung DMS-TUNING-RESOURCES modifiziert. In Verbindung mit der Direktattributierung ist auch die Angabe der Operanden VOLUME-SET, VOLUME, DEVICE-TYPE und S0-MIGRATION möglich, welche für die physikalische Allokierung von Bedeutung sind. Sie beeinflussen auch die Zuweisung von Voreinstellungen der Attribute PERFORMANCE, AVAILABILITY und WORK-FILE und führen ggf. zu Modifikationen von expliziten Benutzerangaben für diese Attribute.

*STORAGE-CLASS=<composed-name> (explizite Angabe einer Storage-Klasse)*

Der Benutzer spezifiziert die Dateiattribute durch Angabe einer Storage-Klasse, die vom Systembetreuer eingerichtet sein muss. Die Datei erhält für die Attribute WORK-FILE, PERFORMANCE, USAGE, DISK-WRITE, FILE-PREFORMAT und AVAILABILITY die in der Storage-Klasse hinterlegten Werte, wobei der Wert für PERFORMANCE ggf. unter Berücksichtigung der Einstellung von DMS-TUNING-RESOURCES reduziert wird. Die Angabe einer Storage-Klasse und eine gleichzeitige physikalische Allokierung sind nicht möglich und werden durch die Kommandosyntax nicht zugelassen. Die für die physikalische Allokierung relevanten Dateiattribute erhalten implizit folgende Zuweisungen: VOLUME-SET=\*STD (keine Volume-Set-Angabe), VOLUME=\*STD (keine Volume-Angabe), DEVICE-TYPE=\*BY-VOLUME (keine Angabe des Device-Typs), S0-MIGRATION=\*ALLOWED.

*STORAGE-CLASS=\*STD (keine explizite Angabe von Attributen, Voreinstellungen)*

Diese Zuweisung erfolgt automatisch, wenn die Dateiattribute weder durch die Angabe der einzelnen Werte noch durch die Angabe einer Storage-Klasse explizit spezifiziert werden. Ist dem Eigentümer der Datei eine vom Systembetreuer eingerichtete Storage-Klasse als Default-Storage-Klasse zugeordnet, wird diese der Datei zugewiesen. Die Bestimmung der Dateiattribute erfolgt analog zur expliziten Zuordnung einer Storage-Klasse, d.h. die Dateiattribute WORK-FILE, USAGE, DISK-WRITE, FILE-PREFORMAT und AVAILABILITY werden unverändert aus der Definition der Default-Storage-Klasse übernommen, der dort hinterlegte Wert für Performance wird bei der Übernahme ggf. unter Berücksichtigung der Festlegung von DMS-TUNING-RESOURCES reduziert. Wenn dem Eigentümer der Datei keine Default-Storage-Klasse zugewiesen ist, werden die einzelnen Dateiattribute über das jeweilige attributspezifische Verfahren zur Bestimmung der Voreinstellungen zugewiesen.

Der Fall, dass eine in einem Benutzereintrag als Default-Storage-Klasse eingetragene Storage-Klasse nicht definiert ist, wird von dem System als inkonsistent bewertet und führt zum Zurückweisen des CREATE-FILE-Kommandos.

Für Dateien, die in einem SM-Pubset katalogisiert sind, aber weder auf der Verarbeitungsebene noch auf einer Hintergrundebene Platz belegen, sind die Storage-Klassen-Zuordnung und die ablageortrelevanten Attribute mit Ausnahme der Performance-Attribute nicht definiert. Sie werden erst dann bestimmt, wenn für die bisher nur katalogisierte Datei erstmalig Platz angefordert wird (z.B. durch das Kommando MODIFY-FILE-ATTRIBUTES). Dabei wird das gleiche Verfahren angewandt wie beim Neuanlegen einer Datei durch CREATE-FILE.

### Ändern der Dateiattribute (MODIFY-FILE-ATTRIBUTES)

Das Ändern der Dateiattribute und der Storage-Klassen-Zuordnung wird durch das Kommando MODIFY-FILE-ATTRIBUTES ermöglicht. Wie bei dem Kommando CREATE-FILE gibt es auch für MODIFY-FILE-ATTRIBUTES mehrere Alternativen, die über den Operanden STORAGE-CLASS ausgewählt werden. Wenn dem Benutzer die Möglichkeit der Direktattributierung entzogen wurde, werden bei MODIFY-FILE-ATTRIBUTES die Angaben PERFORMANCE, USAGE, DISK-WRITE, AVAILABILITY, WORK-FILE=\*NO, STORAGE-CLASS=\*NONE und S0-MIGRATION=\*ALLOWED ignoriert.

*STORAGE-CLASS=\*UNCHANGED (Voreinstellung)*

Abgesehen von der Dateigröße können keine ablageortrelevanten Dateiattribute spezifiziert werden, d.h. sie behalten ihre bisherigen Werte bei. Auch eine bestehende Zuordnung einer Storage-Klasse bleibt unverändert.

*STORAGE-CLASS=\*NONE (Direktattributierung)*

Eine bisher für die Datei bestehende Storage-Klassen-Zuordnung wird aufgehoben. Von den Benutzern explizit angegebene Dateiattribute werden entsprechend geändert, wobei in bestimmten Fällen die Angaben für Performance und Verfügbarkeit automatisch modifiziert werden (in Abhängigkeit von der DMS-TUNING-RESOURCES-Berechtigung oder bei Dateien, die durch das Attribut S0-MIGRATION=\*FORBIDDEN an einen bestimmten Volume-Set gebunden sind). Nicht explizit spezifizierte Attribute bleiben im Allgemeinen unverändert. Einen Sonderfall bildet die Änderung des Attributs S0-MIGRATION=\*ALLOWED in S0-MIGRATION=\*FORBIDDEN. Sie kann eine implizite Anpassung der Attribute Performance und Verfügbarkeit nach sich ziehen.

*STORAGE-CLASS=composed-name (explizite Angabe einer Storage-Klasse)*

Der Datei wird die angegebene Storage-Klasse zugeordnet, die Dateiattribute werden gemäß der Definition der Storage-Klasse neu bestimmt. Das dabei angewandte Verfahren ist gleich wie beim Kommando CREATE-FILE mit expliziter Angabe einer Storage-Klasse.

*STORAGE-CLASS=\*STD (explizite Zuordnung der Default-Storage-Klasse)*

Diese Angabe führt nur dann zur Änderung der Dateiattribute, wenn dem Eigentümer der Datei eine Default-Storage-Klasse zugewiesen ist und diese der Datei nicht bereits bisher als Storage-Klasse zugeordnet war. In diesem Fall wird der Datei die Default-Storage-Klasse des Eigentümers als Storage-Klasse zugewiesen und die Dateiattribute werden wie bei CREATE-FILE mit STORAGE-CLASS=\*STD entsprechend der Definition der Storage-Klasse neu bestimmt.

*STORAGE-CLASS=\*UPDATE (Aktualisieren der Dateiattribute)*

Die Zuordnung der Storage-Klasse ändert sich dabei nicht, die Dateiattribute werden neu bestimmt, wobei die aktuelle Definition der Storage-Klasse herangezogen wird. Dadurch wird es ermöglicht, Diskrepanzen zwischen der einer Datei zugeordneten Storage-Klasse und den Dateiattributen zu beseitigen. Diese können sich dadurch ergeben haben, dass zwischenzeitlich die Definition der Storage-Klasse geändert worden ist. Dabei werden die Attribute der Dateien, denen die Storage-Klasse zugeordnet ist, nicht automatisch angepasst.

### **Dateiattribute an der FILE-Schnittstelle**

Auch die FILE-Programmschnittstelle<sup>1)</sup> unterstützt die Direktattributierung, die explizite Storage-Klassen-Angabe und die Verwendung von Default-Storage-Klassen als alternative Möglichkeiten für die Zuordnung der Dateiattribute. Gegenüber dem CREATE-FILE-Kommando ist die Weichenstellung zwischen Direktattributierung und der Zuordnung der Default-Storage-Klasse etwas anders. Für den Fall, dass dem Benutzer die Möglichkeit der Direktattributierung entzogen wurde sowie für den Fall, dass beim Anlegen der Datei keine explizite Angabe zum Operanden STOCLAS erfolgt, und keine ablageortrelevanten Dateiattribute außer BLKCTRL und BLKSIZE spezifiziert sind, wird überprüft, ob dem Dateieigentümer eine Default-Storage-Klasse zugewiesen ist. Ist dies der Fall, werden die Dateiattribute Performance, Verfügbarkeit, Arbeitsdatei entsprechend der Definition der Default-Storage-Klasse festgelegt (wie bei STORAGE-CLASS=\*STD), in die Bestimmung des Preformats fließen jedoch auch die BLKSIZE- und BLKCTRL-Angaben des Benutzers ein. Ist der Default-Storage-Klasse eine Volume-Set-Liste zugeordnet, wird diese bei der Auswahl des Ablageorts berücksichtigt. Die spezielle Behandlung der Format-Angaben durch die FILE-Programmschnittstelle ermöglicht, dass für Dateien, welche durch bestehende Programme mit expliziter Angabe von BLKSIZE oder BLKCTRL angelegt werden, die Zuordnung der Dateiattribute und damit die Verteilung innerhalb des SM-Pubsets beeinflusst werden kann, ohne dass die Programme geändert werden müssen.

Im Folgenden werden die einzelnen Dateiattribute näher erläutert.

---

<sup>1)</sup> Das ISP-Kommando FILE verhält sich - insbesondere hinsichtlich der BLKCTRL- und BLKSIZE-Angaben - analog zur FILE-Programmschnittstelle. Ausnahme: Für das Anlegen einer Datei über das FILE-Kommando sind die Möglichkeiten für die Attributierung eingeschränkt, z.B. ist die explizite Angabe einer Storage-Klasse nicht möglich.

## Performance-Attribute

Die Dateiattribute PERFORMANCE, USAGE, DISK-WRITE erlauben für Dateien die Vorgabe der gewünschten Performance.

Performance-Attribute	Werte
PERFORMANCE	STD, HIGH, VERY-HIGH
USAGE (Operationen, für die erhöhte Performance gewünscht wird)	READ nur Leseoperationen WRITE nur Schreiboperationen READ-WRITE Lese- und Schreiboperationen
DISK-WRITE (erhöhte Performance auch auf Kosten der Schreibsicherheit)	IMMEDIATE nicht gewünscht BY-CLOSE gewünscht

Tabelle 6: Bedeutung der Performance-Attribute

Bei SM-Pubsets bewirken die durch die Kommandos CREATE-FILE, MODIFY-FILE-ATTRIBUTES zugeordneten statischen Werte der Performance-Attribute zunächst, dass beim Anlegen der Datei ein geeigneter Volume-Set ausgesucht wird. Gelangt eine Datei mit erhöhten Performance-Anforderungen auf einen Volume-Set mit einem Cache, wird beim Eröffnen der Datei bestimmt, in welchem Umfang der Cache genutzt wird. Dabei können die für die Cache-Nutzung maßgeblichen Werte der Performance-Attribute durch den Benutzer (Kommando ADD-FILE-LINK bzw. FCB-Schnittstelle) oder durch den Systembetreuer (DMS-TUNING-RESOURCES) gegenüber den statischen Werten dynamisch reduziert werden. Auf den Ablageort der Datei hat die dynamische Modifikation der Performance-Werte keine Auswirkung. Sie ist daher für Dateien, die auf Volume-Sets ohne Cache liegen, wirkungslos, da deren Performance-Verhalten bereits vollständig durch den Ablageort bestimmt wird.

Die Erfüllung der Performance-Angaben wird dem Benutzer nicht garantiert. Ein Volume-Set, der die gewünschte Performance nicht oder nicht ausreichend bietet, wird als verträglich mit der Benutzeranforderung betrachtet und als Ablageort für die Datei herangezogen, wenn kein besser geeigneter gefunden werden kann. Eine geforderte Schreibsicherheit wird immer gewährleistet, ggf. mit Verzicht auf erhöhte Performance.

Der maximale Wert für Performance, der für die Dateien eines bestimmten Benutzers zulässig ist, kann durch den Systembetreuer mithilfe des Benutzerrechts DMS-TUNING-RESOURCES eingeschränkt werden. Dieses erlaubt die Einstellungen NONE, CONCURRENT-USE, EXCLUSIVE-USE. Sie sind mit dem maximalen Performance-Wert entsprechend [Tabelle 7](#) korreliert: Weist ein Benutzer einer Datei durch die Kommandos

CREATE-FILE oder MODIFY-FILE-ATTRIBUTES einen höheren Performance-Wert zu, als ihm erlaubt ist, wird die Anforderung nicht zurückgewiesen, sondern auf den maximal zulässigen Wert reduziert.

Werte für DMS-TUNING-RESOURCES	erlaubte Performance-Attribute
NONE	STD
CONCURRENT-USE	STD, HIGH,
EXCLUSIVE-USE	STD, HIGH, VERY-HIGH

Tabelle 7: Relation zwischen DMS-TUNING-RESOURCES und Performance-Attributen

Damit bei physikalischer Allokierung die Wirkung der Benutzerkontingente nicht umgangen werden kann - z.B. indem ein Benutzer eine Datei mit dem Attribut PERFORMANCE=\*STD physikalisch an einen Volume-Set bindet, der aus hochperformanten Volumes besteht - werden ggf. die von den Benutzern spezifizierten expliziten Performance-Werte angehoben. Auch in die Bestimmung der Voreinstellung gehen bei physikalischer Allokierung die Eigenschaften des vorgegebenen Volume-Sets ein. Dies wird durch [Tabelle 8](#) verdeutlicht. Entsprechend wird bei Dateien, die bereits physikalisch an einen bestimmten Volume-Set gebunden sind (S0-MIGRATION=\*FORBIDDEN) die Reduzierung des Performance-Werts durch das System überwacht und ggf. ignoriert. Wenn eine Datei durch das Setzen der Migrationssperre S0-MIGRATION=\*FORBIDDEN explizit an den Volume-Set gebunden, auf dem sie zurzeit liegt, wird der bisherige Performance-Wert der Datei ggf. automatisch erhöht.

PERFORMANCE-Spektrum des Volume-Sets	vom Benutzer spezifizierter Wert für das Dateiattribut PERFORMANCE	resultierender Wert für das Dateiattribut PERFORMANCE
STD [,HIGH] [,VERY-HIGH]	nicht explizit angegeben STD HIGH VERY-HIGH	STD STD HIGH VERY-HIGH
HIGH [,VERY-HIGH]	nicht explizit STD HIGH VERY-HIGH	HIGH HIGH HIGH VERY-HIGH
VERY-HIGH	nicht explizit STD HIGH VERY-HIGH	VERY-HIGH VERY-HIGH VERY-HIGH VERY-HIGH

Tabelle 8: Bestimmung der Performance-Attribute bei physikalischer Allokierung

Das automatische Anheben des Performance-Attributes bei physikalischer Allokierung erfolgt ohne Berücksichtigung der DMS-TUNING-RESOURCES-Berechtigung des Dateieigentümers. Da die DMS-TUNING-RESOURCES-Berechtigung bei SM-Pubsets vor allem für den Ressourcenschutz für Caches geeignet ist (siehe [Kapitel „Pubset-Überwachung und Pubset-Pflege“ auf Seite 245](#)), ist dies im Hinblick auf den Ressourcenschutz unproblematisch. Volume-Sets mit Caches verfügen im Allgemeinen über ein Performance-Spektrum, das auch den Wert STD enthält. In diesem Fall erfolgt keine automatische Anhebung (siehe [Tabelle 8](#)).

## Verfügbarkeit

Zur Spezifikation der gewünschten Verfügbarkeit dient das Dateiattribut AVAILABILITY. Eine Datei mit AVAILABILITY=\*HIGH gelangt nur auf einen Volume-Set, dem durch den Systembetreuer das Verfügbarkeitsprofil AVAILABILITY=\*HIGH zugewiesen wurde. Die Kombination aus dem Dateiattribut AVAILABILITY=\*HIGH und das Volume-Set-Attribut AVAILABILITY=\*STD wird von der Volume-Set-Selektion als unverträglich bewertet. Dass ein Volume-Set mit dem Attribut AVAILABILITY=\*HIGH auch tatsächlich eine hohe Verfügbarkeit bietet, muss der Systembetreuer sicherstellen.

Für Arbeitsdateien und temporäre Dateien ist das Dateiattribut AVAILABILITY=\*HIGH nicht erlaubt. Attributzuweisungen, sowie Umwandlungen von permanenten Dateien in temporäre Dateien, durch welche diese Bedingung durchbrochen würde, werden zurückgewiesen. Eine spezielle Behandlung erfolgt in dem Fall, dass einem Benutzer eine Default-Storage-Klasse zugewiesen ist, für die AVAILABILITY=\*HIGH festgelegt ist. Das Anlegen einer temporären Datei, bei dem die Attribute nicht explizit spezifiziert werden (STORAGE-CLASS=\*STD), wird erlaubt, wobei die Dateiattribute mit Ausnahme von AVAILABILITY entsprechend der Definition der Default-Storage-Klasse festgelegt werden. Für die Verfügbarkeit erfolgt die implizite Zuweisung von AVAILABILITY=\*STD. Die Sonderbehandlung der Default-Storage-Klasse ist darin begründet, dass für temporäre Dateien auf den SM-Pubsets ein zu den SF-Pubsets kompatibles Verhalten gewährleistet werden soll, auch wenn Default-Storage-Klassen mit AVAILABILITY=\*HIGH verwendet werden.

Wie bei den Performance-Attributen wird beim Anlegen einer Datei durch physikalische Allokierung der Defaultwert für Verfügbarkeit unter Berücksichtigung der Eigenschaften des jeweiligen Volume-Sets bestimmt. Auch werden explizite Benutzervorgaben ggf. modifiziert (siehe [Tabelle 9](#)). Entsprechendes gilt auch für Modifikationen des Verfügbarkeits-Attributs von bestehenden Dateien, die bereits bisher physikalisch an einen bestimmten Volume-Set gebunden sind (S0-MIGRATION=\*FORBIDDEN) und das Setzen der Migrationssperre S0-MIGRATION=\*FORBIDDEN.



AVAILABILITY des Volume-Sets, auf dem die Datei angelegt wird	vom Benutzer spezifizierter Wert für das Dateiattribut AVAILABILITY	resultierender Wert für das Dateiattribut AVAILABILITY
HIGH	nicht explizit angegeben	HIGH
	HIGH	HIGH
	STD	HIGH
STD	nicht explizit	STD
	HIGH	Angabe unverträglich
	STD	STD

Tabelle 9: Ermittlung des Attributs AVAILABILITY bei physikalischer Allokierung

## Arbeitsdateien

Das Nutzungsmodell für Arbeitsdateien (d.h. Dateien mit dem Attribut WORK-FILE=\*YES) besteht darin, dass sie ausschließlich auf Work-Volume-Sets (d.h. Volume-Sets mit der Nutzungsart WORK) gelangen, welche der Systembetreuer den Benutzern nur für beschränkte Zeit zur Verfügung stellt. Die Zeitdauer muss zwischen dem Systembetreuer und den Benutzern vereinbart werden. Nach Ablauf der festgelegten Frist hat der Systembetreuer das Recht, den Work-Volume-Set zu entfernen und dazu alle darauf befindlichen Dateien zu löschen. Dies kann gegebenenfalls durch Zwangsentfernen des Volume-Sets erfolgen (Operand CONDITION=\*VOLUME-SET-DEECTS im Kommando MODIFY-PUBSET-PROCESSING). Dabei werden die auf dem Volume-Set liegenden Dateien implizit gelöscht. Werden zu diesem Zeitpunkt auf dem Volume-Set liegende Dateien durch Benutzer-tasks bearbeitet, kann sie der Systembetreuer durch die Option TERMINATE-JOB=\*YES zwangsterminieren lassen. Da auf einen Work-Volume-Set ausschließlich Arbeitsdateien gelangen, sind davon nur Benutzer betroffen, welche die mit dem Systembetreuer bestehende Vereinbarung nicht einhalten. Da umgekehrt Arbeitsdateien immer auf Work-Volume-Sets angelegt und auch nicht auf Hintergrundebenen verdrängt werden, kann die gesamte durch Arbeitsdateien belegte Kapazität zurückgewonnen werden, indem die Work-Volume-Sets den Benutzern entzogen werden.

Beim Anlegen einer Datei durch CREATE-FILE sind für das Attribut WORK-FILE die Angaben \*STD, \*YES und \*NO möglich. Die Angabe \*STD ist bei nicht-physikalischer Allokierung mit \*NO gleichbedeutend. Bei physikalischer Allokierung wird das WORK-FILE-Attribut aus den Benutzerangaben und der Nutzungsart des Volume-Sets entsprechend [Tabelle 10](#) ermittelt. Nachdem das Attribut WORK-FILE für eine Datei festgelegt ist, kann es durch das Kommando MODIFY-FILE-ATTRIBUTES nicht mehr geändert werden.

Nutzungsart des Volume-Sets, auf dem die Datei angelegt wird	vom Benutzer spezifizierter Wert für das Dateiattribut WORK-FILE	resultierender Wert für das Dateiattribut WORK-FILE
nicht WORK	STD NO YES	NO NO Benutzerangabe unverträglich
WORK	STD NO YES	YES Benutzerangabe unverträglich YES

Tabelle 10: Bestimmung des WORK-FILE-Attributs bei physikalischer Allokierung

Arbeitsdateien und Work-Volume-Sets bieten eine Alternative für eine verbreitete Verwendung von Privatplatten als Arbeitsplatten. Beim Anlegen einer Datei auf einer Privatplatte muss der Benutzer die VSN angeben. Um die bestehenden Kommandoprozeduren auf SM-Pubsets zur Anwendung bringen zu können, genügt es, die VSN der Privatplatte durch die VSN eines Volumes des Work-Volume-Sets zu ersetzen. Das WORK-FILE-Attribut wird automatisch passend bestimmt. Um ein Verhalten wie bei der Benutzung von privaten Arbeitsplatten zu gewährleisten, wird für die physikalische Allokierung auf Work-Volume-Sets das Recht zur physikalischen Allokierung nicht benötigt.

### Dateiformat

Das Dateiformat für die Dateiverarbeitung erlaubt die Werte (K, NK2, NK4). Es bestehen jedoch Unterschiede darin, wie eine Datei zu ihrem Format gelangt. Bei SM-Pubsets ist es wichtig, dass die Benutzer bereits dann Information über das Dateiformat bereitstellen können, wenn für die Datei erstmalig Platz angefordert wird. Sie kann dann bei der Auswahl des geeigneten Volume-Sets berücksichtigt werden.

Dies wird für die Dateien auf SM-Pubsets durch das Dateiattribut „vorläufiges Dateiformat“ (Preformat) ermöglicht. Es wird einer Datei zugeordnet, wenn für sie erstmalig Platz bereitgestellt wird (siehe Kommandos CREATE-FILE, MODIFY-FILE-ATTRIBUTES sowie FILE-Programmschnittstelle). Im Allgemeinen erfolgt dies beim Neuanlegen einer Datei. Aber auch der Fall, dass für eine bisher nur katalogisierte Datei durch das Kommando MODIFY-FILE-ATTRIBUTES Platz angefordert wird, gilt als erstmalige Platzbereitstellung. Für eine Datei, die bereits über Platz verfügt, ist die Änderung des zugeordneten Preformats nicht möglich und wird zurückgewiesen. Für Dateien, die über keinen Platz verfügen, d.h. nur katalogisiert sind, ist das Preformat undefiniert. Für Dateien auf SF-Pubsets sind Preformat-Angaben ohne Relevanz und werden ignoriert. Im Fall, dass ein SM-Pubset durch Konvertierung von SF-Pubsets mithilfe von SMPGEN gebildet wird, haben die darauf befindlichen Dateien den speziellen Preformat-Wert NONE. Er hat die gleiche Bedeutung, als ob der Datei das Format des Volume-Sets, auf dem sie liegt, als Preformat zugeordnet wäre.

Erfolgt die Platzbereitstellung im Rahmen der Kommandos CREATE-FILE oder MODIFY-FILE-ATTRIBUTES, bestehen folgende Alternativen:

- Der Benutzer spezifiziert das (komplette) Preformat durch Angabe des Direktattributs FILE-PREFORMAT oder durch explizite Zuweisung einer Storage-Klasse.
- Das System ermittelt eine Voreinstellung für das Preformat. Dabei werden das Default-File-Format des Pubsets, das in der Default-Storage-Klasse des Benutzers hinterlegte Preformat und bei physikalischer Allokierung das Format des Volume-Sets, auf den die Datei gelangt, herangezogen.

Das Verfahren zur Ermittlung des Preformats wird in [Tabelle 11](#) veranschaulicht. Der von dem Benutzer angeforderte Platz wird durch die Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES nur dann bereitgestellt, wenn das Preformat in Kombination mit den anderen Dateiattributen (z.B. Verfügbarkeit) auf dem Pubset erfüllt werden kann, d.h. die Preformat-Angabe wird in diesem Fall verbindlich erfüllt.

Für die FILE-Programmschnittstelle<sup>1)</sup> wurde ein gegenüber CREATE-FILE anderes Verhalten festgelegt. Dies geschah damit Dateibearbeitungen, die ursprünglich für SF-Pubsets konzipiert sind, ohne Änderungen (bzw. möglichst geringen Änderungen) auf SM-Pubsets genutzt werden können. Zur Bestimmung des Preformats werden die Angaben zu Dateiformat (BLKCTRL) und Blocklänge (BLKSIZE) herangezogen. Für Dateien auf SM-Pubsets ist es günstig, wenn das gewünschte Format bereits beim Anlegen der Datei vollständig spezifiziert wird. Für die Dateibearbeitung auf SF-Pubsets sind auch die im Folgenden dargestellten Möglichkeiten für die Formatspezifikation erlaubt, die für SM-Pubsets nicht günstig sind, aber aus Kompatibilitätsgründen auch für sie zugelassen werden:

- Die Benutzer sind nicht gezwungen bereits beim Anlegen einer Datei Angaben zu Dateiformat (BLKCTRL) und Blocklänge (BLKSIZE) zu machen, die das später bei der Verarbeitung gewünschte Format vollständig und verbindlich bestimmen.
- Selbst Angaben, von denen sicher ist, dass sie auf dem entsprechenden Pubset nicht erfüllbar sind, werden nicht zurückgewiesen. Die Validierung der Angaben für das Format erfolgt erst beim Eröffnen der Datei.
- Neben Benutzerangaben, die das Format eindeutig beschreiben und Benutzerangaben, bei denen das Format vollständig offen bleibt, sind auch Angaben möglich, aus denen das Format nur teilweise hervorgeht. Letztere müssen sinnvoll komplettiert werden.

---

<sup>1)</sup> bzw. das ISP-Kommando FILE

Für SM-Pubsets ungünstige Benutzerangaben müssen bei best möglicher Wahrung der Kompatibilität gegenüber dem Verhalten für SF-Pubsets im Hinblick auf das Preformat und damit auf die Volume-Set-Auswahl „möglichst sinnvoll“ interpretiert werden. [Tabelle 12 auf Seite 222](#) zeigt für die FILE-Programmschnittstelle<sup>1)</sup>, wie das Preformat aus den Angaben zu Dateiformat (BLKCTRL) und Blocklänge (BLKSIZE) abgeleitet wird.

Wenn es sich bei einem FILE-Aufruf herausstellt, dass das aus den Benutzerangaben ermittelte Preformat nicht erfüllt werden kann, wird (im Unterschied zu dem Kommando CREATE-FILE) der Datei ein anderes Preformat zugeordnet, das auf dem Pubset erfüllt werden kann und dem ursprünglich ermittelten Preformat möglichst ähnlich ist. Dies ist z.B. der Fall, wenn in dem SM-Pubset kein geeigneter Volume-Set vorhanden ist oder der Volume-Set, den der Benutzer durch physikalische Allokierung vorgegeben hat das Preformat nicht unterstützt.

Ist das ursprünglich ermittelte Preformat K, so wird bevorzugt NK2 herangezogen, ist aber auch NK2 nicht möglich, wird NK4 gewählt. Ist das ursprünglich ermittelte Preformat NK2, ist NK4 die einzige Alternative.

Benutzer, die sich über die zugewiesene Voreinstellung für die relevanten Parameter des File-Formats informieren wollen, können über die Kommandos SHOW-PUBSET-CONFIGURATION (Voreinstellung des File-Format des Pubsets), SHOW-USER-ATTRIBUTES mit Operand INFORMATION=\*PUBSET-ATTRIBUTES (Default-Storage-Klasse) und SHOW-STORAGE-CLASS (Preformat in der Default-Storage-Klasse) zusätzliche Informationen abfragen.

---

<sup>1)</sup> Für das ISP-Kommando FILE ist das Vorgehen analog, wobei das Kommando nur einen Teil der Funktionalität der FILE-Programmschnittstelle enthält (z.B. keine Angabe von Storage-Klassen).

Bedingungen		resultierender Wert für das Dateiattribut Preformat	
<b>Benutzerangabe für STORAGE-CLASS im CREATE-FILE:</b>			
*NONE Direktattributierung	<b>Angabe für FILE-PREFORMAT im CREATE-FILE:</b>		
	K	K	
	NK2	NK2	
	NK4	NK4	
	BY-PUBSET-DEFAULT (Voreinstellung)	<b>VOLUME- oder VOLUME-SET-Angabe im CREATE-FILE:</b>	
	ja	Preformat wie Format des Volume-Sets	
	nein	Preformat wie Default-File- Format des Pubsets	
<composed-name> Angabe einer Storage-Klasse	<b>Preformat in Storage-Klasse:</b>		
	K	K	
	NK2	NK2	
	NK4	NK4	
	BY-PUBSET-DEFAULT	Preformat wie Default-File- Format des Pubsets	
*STD (Voreinstellung) keine expliziten Angaben für Dateiattribute	<b>Ist dem Benutzer eine Default-Storage-Klasse zuge- ordnet:</b>		
	ja	<b>Preformat in der Default- Storage-Klasse:</b>	
		K	K
		NK2	NK2
		NK4	NK4
		BY-PUBSET-DEFAULT	Preformat wie Default-File- Format des Pubsets
nein		Preformat wie Default-File- Format des Pubsets	

Tabelle 11: Ermittlung des Preformats bei der erstmaligen Platzanforderung für eine Datei durch CREATE-FILE oder MODIFY-FILE-ATTRIBUTES

Bedingungen		resultierendes Preformat	
<b>BLKCTRL:</b>			
PAMKEY		K	
DATA2K		NK2	
DATA4K		NK4	
DATA/NO	<b>BLKSIZE:</b>		
	(STD,n) n ungerade		NK2
	(STD,n) n gerade		NK4
	sonst	<b>physikalische Allokierung:</b>	
	nein	<b>Angaben einer Storage-Klasse:</b>	
	nein	<b>Default-Storage-Klasse zugewiesen</b>	
	nein	<b>Pubset-Default-File-Format</b>	
		K oder NK2	NK2
		NK4	NK4
	ja	<b>Preformat in Default-Storage-Klasse:</b>	
		K oder NK2	NK2
		NK4	NK4
		BY-PUBSET-DEFAULT	<b>Pubset-Default-File-Format:</b>
		K oder NK2	NK2
		NK4	NK4
ja	<b>Preformat in der Storage-Klasse:</b>		
	K oder NK2	NK2	
	NK4	NK4	
	BY-PUBSET-DEFAULT	<b>Pubset-Default-File-Format:</b>	
	K oder NK2	NK2	
	NK4	NK4	
ja	<b>Format des Volume-Sets, auf dem die Datei angelegt wird:</b>		
	K oder NK2	NK2	
	NK4	NK4	

Tabelle 12: Ermittlung des Preformats bei der erstmaligen Platzanforderung für eine Datei über die FILE-Programmschnittstelle

Bedingungen		resultierendes Preformat				
not specified	<b>physikalische Allokierung:</b>					
	nein	<b>Angabe einer Storage-Klasse:</b>				
		nein	<b>Default-Storage-Klasse zugewiesen</b>			
			nein		Pubset-Default	
			ja	<b>Preformat in Default-Storage-Klasse:</b>		
				K		K
				NK2		NK2
				NK4		NK4
				BY-PUBSET-DEFAULT		Pubset-Default
			ja	<b>Preformat in Storage-Klasse:</b>		
				K		K
				NK2		NK2
				NK4		NK4
BY-PUBSET-DEFAULT		Pubset-Default				
ja		Volume-Set-Format				

Tabelle 12: Ermittlung des Preformats bei der erstmaligen Platzanforderung für eine Datei über die FILE-Programmschnittstelle

Die Bezeichnung Preformat bringt zum Ausdruck, dass es für die Datei zunächst nur einen vorläufigen Charakter hat. Die endgültige Festlegung des Formats erfolgt (bei SM-Pubsets wie auch bisher bei SF-Pubsets) für eine Datei erst, wenn sie für ihre Erstellung eröffnet wird. Bei dem für SF-Pubsets bestehenden Verfahren im Rahmen der OPEN-Behandlung sind die wesentlichen Eingangsgrößen das Format des SF-Pubsets, auf dem die Datei liegt, und die zum OPEN-Zeitpunkt in der Task-File-Table (versorgt über ADD-FILE-LINK, FILE-Programmschnittstelle<sup>1)</sup>) bzw. in dem File-Control-Block (FCB) von dem Benutzer aktuell hinterlegten Werte für Dateiformat (BLOCK-CONTROL-INFO bzw. BLKCTRL), Blocklänge (BUFFER-LENGTH bzw. BLKSIZE) und Zugriffsmethode (ACCESS-METHOD bzw. FCBTYP). In dem speziellen Fall, dass der Benutzer bei dem Erstellen der Datei auf die explizite Formatvorgabe verzichtet, wird das (Pubset-) Format des SF-Pubsets, auf dem die Datei liegt, als Dateiformat (K, NK2, NK4) übernommen. Für SM-Pubsets, die über kein Pubset-Format verfügen, wird dieses Verfahren modifiziert, wobei die Rolle des Formats des SF-Pubsets weitgehend durch das Preformat der Datei eingenommen wird (siehe [Bild 28](#)).

<sup>1)</sup> bzw. ISP-Kommando FILE

Dabei wird zwischen folgenden beiden Ausgangssituationen unterschieden:

- Eine Datei wird erstmalig erstellt, d.h. sie verfügt über kein CREATION-DATE. Hier wird die Rolle des SF-Pubset-Formats bei einem SM-Pubset durch das Preformat der Datei übernommen, das der Datei zuvor (d.h. bei der erstmaligen Platzbereitstellung) zugeordnet wurde. Verzichtet der Benutzer bei der Dateieröffnung auf explizite Vorgaben zu Dateiformat (BLOCK-CONTROL-INFO bzw. BLKCTRL) und Blocklänge (BUFFER-LENGTH bzw. BLKSIZE), wird das Preformat als Dateiformat übernommen.
- Der Inhalt für eine bereits existierende Datei wird neu erstellt, wobei der Platz, den die Datei bisher belegt hat, dieser weiterhin zugeordnet bleiben soll. In diesem Fall geht nicht das Preformat, sondern das bisherige Format in das Verfahren der Formatbestimmung ein. Wenn der Benutzer bei der Dateieröffnung auf explizite Vorgaben zu Dateiformat (BLOCK-CONTROL-INFO bzw. BLKCTRL) und Blocklänge (BUFFER-LENGTH bzw. BLKSIZE) verzichtet, wird das bisherige Dateiformat (K, NK2, NK4) beibehalten. In diesem Anwendungsfall ist sichergestellt, dass das Format (der neu zu erstellenden Datei) mit dem bisherigen Ablageort verträglich ist, und der bisher der Datei zugeordnete Platz weiterhin verwendet werden kann.

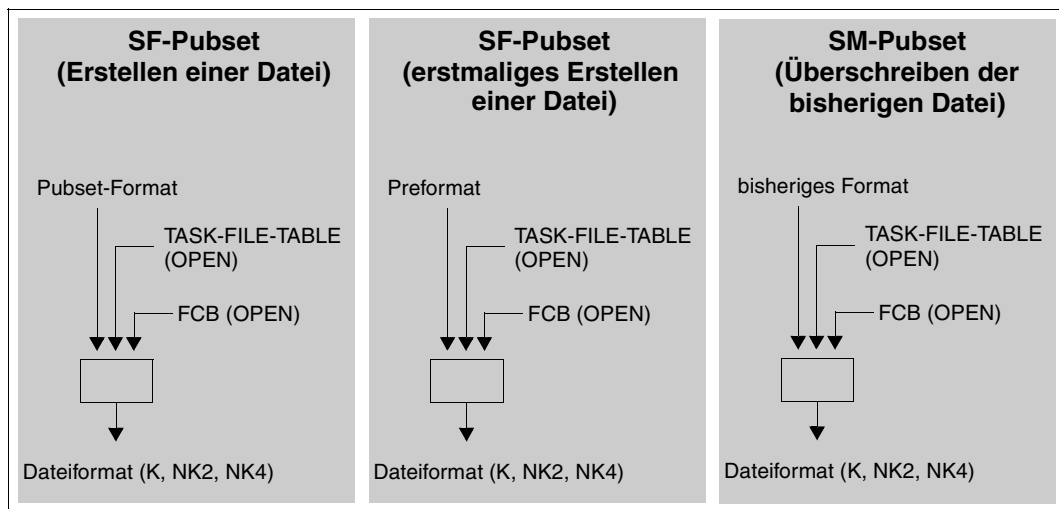


Bild 28: Ermittlung des Formats einer Datei durch die OPEN-Behandlung bei SF- und SM-Pubsets



Bei einer expliziten Angabe von Dateiformat und Blocklänge zum OPEN-Zeitpunkt kann es sich für eine in einem SM-Pubset liegende Datei ergeben, dass der Volume-Set, auf dem der Datei bisher Platz zugewiesen war, mit dem gewünschten Dateiformat nicht verträglich ist. In diesem Fall erfolgt während der OPEN-Behandlung eine implizite Verlagerung auf einen passenden Volume-Set, wenn ein solcher vorhanden ist und die Verlagerung erlaubt ist (S0-MIGRATION). Um die Belastung des Systems gering zu halten und die Risiken bei einer möglichen Verlagerung (Platzengpässe etc.) auszuschalten, wird den Benutzern dringend empfohlen, für Dateien eines SM-Pubsets bereits zum Zeitpunkt der erstmaligen Platzbereitstellung das für die spätere Verarbeitung gewünschte Format als Preformat vorzugeben.

In dem speziellen Fall, dass ein Benutzer sowohl beim Anlegen als auch beim Erstellen einer Datei keine expliziten Vorgaben für das Format trifft, sondern dessen Bestimmung vollständig dem System überlässt, ergibt sich folgendes Verhalten:

- Für Dateien, die nicht über physikalische Allokierung angelegt werden, ergibt sich das Format in der Regel direkt aus dem Default-File-Format des Pubsets.
- Eine Ausnahme hiervon ist nur der Fall, dass dem Dateieigentümer eine Storage-Klasse zugewiesen ist, in welcher ein expliziter Wert für das Preformat eingetragen ist (d.h. nicht der Wert BY-PUBSET-DEFAULT) und die Default-Storage-Klasse beim Anlegen der Datei zum Tragen kommt (d.h. keine Direktattributierung bezüglich anderer Dateiattribute, wie beispielsweise Performance). In diesem Fall bestimmt der Wert aus der Default-Storage-Klasse das Dateiformat.
- Für Dateien, die durch physikalische Allokierung explizit auf einem bestimmten Volume-Set angelegt werden, ergibt sich das Format direkt aus dem Format dieses Volume-Sets.

Benutzerprogramme, welche Dateien von SF-Pubset bearbeiten, verwenden das Pubset-Format auf verschiedene Weise. Ein Anwendungsfall besteht darin, dass das Pubset-Format über die STAMCE-Programmschnittstelle ermittelt wird, um zunächst festzustellen, welche Dateiformate auf dem Pubset möglich sind. Aus diesen wird durch entsprechende Angaben von Dateiformat und Blocklänge das gewünschte Format ausgewählt. Dabei ist es in einzelnen Anwendungsfällen möglich, dass das gewünschte Format nur partiell spezifiziert wird, und die komplette Bestimmung den Verfahren zur Ermittlung der Voreinstellung dem System überlassen wird. Dabei werden bestimmte Regeln für die Zuweisung von Voreinstellungen vorausgesetzt. Für SM-Pubsets ist dieses Verhalten nicht optimal, da in diesen mehrere Volume-Sets mit unterschiedlichen Formaten vorhanden sein können.

Damit die bestehenden Benutzerprogramme auf SM-Pubsets ohne Änderungen ablauffähig bleiben, wird an der STAMCE-Programmschnittstelle mit dem Layout für BS2000/OSD-Versionen < V3.0 ein SM-Pubset wie ein SF-Pubset dargestellt, dessen Format gleich dem Default-File-Format des SM-Pubsets ist.

Ein SM-Pubset verhält sich bei der Bestimmung des Formats nahezu identisch wie ein entsprechender SF-Pubset. Es sei denn die Dateien werden durch physikalische Allokierung auf Volume-Sets angelegt, deren Format vom Default-File-Format des Pubsets abweicht, oder es werden Default-Storage-Klassen genutzt. Ein Unterschied besteht z.B. darin, dass auf einem SF-Pubset mit dem Format NK2 generell keine K-Datei angelegt werden kann, während dies auf einem SM-Pubset mit dem Default-File-Format NK2 möglich ist, wenn er einen K-Volume-Set enthält. Insbesondere ist in dem speziellen Fall, dass alle Volume-Sets eines SM-Pubsets das gleiche Format haben, hinsichtlich der Format-Bestimmung vollständige Kompatibilität zu SF-Pubset gewährleistet.

Die oben erwähnten Abweichungen, die sich bei Verwendung von Default-Storage-Klassen und physikalischer Allokierung gegenüber dem bestehenden Verfahren zur Bestimmung von Voreinstellungen ergeben, dürften im Allgemeinen für die Ablauffähigkeit bestehender Benutzerprogramme keine Probleme verursachen. Denn diese dürften meist nicht das bestehende Verfahren zur Bestimmung der Voreinstellungen in allen Details voraussetzen.

Eine Anwendungsmöglichkeit für das Pubset-Default-File-Format im Zusammenspiel mit Default-Storage-Klassen ist z.B. die Umstellung der Dateibearbeitung von K nach NK2. Die Umstellungsphase ist dadurch charakterisiert, dass für einige Programme bei der Bestimmung der Voreinstellung der Dateiattribute das gleiche Verhalten wie auf SF-Pubsets mit dem Format K geboten werden muss. Dazu wird für das Pubset-Default-File-Format der Wert K gewählt und Benutzern, die solche Programme verwenden, entweder keine Default-Storage-Klasse oder eine Default-Storage-Klasse mit PREFORMAT=BY-PUBSET-DEFAULT zugewiesen. Für die übrigen Benutzer ist die Zuordnung einer Default-Storage-Klasse mit dem Preformat NK2 möglich.

## Dateiattribut S0-MIGRATION

Durch das Dateiattribut S0-MIGRATION=\*FORBIDDEN wird eine Datei physikalisch an einen bestimmten Volume-Set gebunden. Es verhindert aber nicht, dass eine Datei auf eine Hintergrundebene verdrängt wird. Beim Zurückholen in die Verarbeitungsebene gelangt sie wieder auf ihren Herkunfts-Volume-Set. Innerhalb von diesem wird ihre Lage aber neu bestimmt. Für die Zuweisung S0-MIGRATION=\*FORBIDDEN wird das Recht zur physikalischen Allokierung benötigt. Beim Anlegen einer Datei wird das Attribut S0-MIGRATION abhängig davon bestimmt, ob die Datei physikalisch angelegt wird (siehe [Tabelle 13](#)).

Bedingungen	resultierender Wert für das Dateiattribut S0-MIGRATION
<b>physikalische Allokierung</b>	
ja	FORBIDDEN
nein	ALLOWED

Tabelle 13: Bestimmung des Attributs S0-MIGRATION beim Anlegen einer Datei

Der Wert für S0-MIGRATION kann für eine bestehende Datei mithilfe des Kommandos MODIFY-FILE-ATTRIBUTES explizit geändert werden. Eine Zuweisung von zusätzlichem Platz für eine bereits angelegte Datei führt jedoch zu keiner impliziten Änderung des Attributs S0-MIGRATION, auch wenn sie über physikalische Allokierung erfolgt.

## Dateiattribut MIGRATE

Das Attribut MIGRATE erlaubt die Werte \*STD, \*INHIBITED (weiche Sperre gegen Hintergrundverdrängung durch HSMS) und \*FORBIDDEN (harte Migrationssperre). Die harte Sperre kann im Unterschied zur weichen vom Systembetreuer nicht außer Kraft gesetzt werden. Beim Anlegen einer Datei wird das Attribut MIGRATE abhängig davon bestimmt, ob die Datei physikalisch angelegt wird (siehe [Tabelle 14](#)). Wird bei physikalischer Allokierung der Ablageort auf Volume- oder Blockebene vorgegeben, ist davon auszugehen, dass die Datei den ihr zugewiesenen Ablageort in der Verarbeitungsebene dauerhaft beibehalten soll. Würde sie auf eine Hintergrundebene verdrängt, könnte nicht gewährleistet werden, dass sie bei einem späteren Recall wieder auf ihren ursprünglichen Ablageort kommt. Durch die implizite Zuweisung MIGRATE=\*FORBIDDEN wird dies verhindert.

Bedingungen		resultierender Wert für das Dateiattribut MIGRATE	
<b>MIGRATE-Angabe in CREATE-FILE</b>			
STD	<b>physikalische Allokierung auf Volume- oder Blockebene:</b>		
	ja		FORBIDDEN
	nein	<b>Dateityp:</b>	
		permanente Datei	ALLOWED
temporäre Datei	INHIBITED		
*ALLOWED	<b>Dateityp:</b>		
	permanente Datei	ALLOWED	
	temporäre Datei	Angabe nicht erlaubt	
*INHIBITED	<b>Dateityp:</b>		
	permanente Datei	INHIBITED	
	temporäre Datei	INHIBITED	
*FORBIDDEN	<b>Dateityp:</b>		
	permanente Datei	FORBIDDEN	
	temporäre Datei	INHIBITED	

Tabelle 14: Bestimmung des Attributs MIGRATE beim Anlegen einer Datei

Wenn der Benutzer die Fixierung an den Ablageort aufheben möchte, kann er das Attribut MIGRATE durch das Kommando MODIFY-FILE-ATTRIBUTES entsprechend ändern. Für die Zuweisung von MIGRATE=\*FORBIDDEN wird das Recht zu physikalischer Allokierung benötigt, unabhängig davon, ob sie explizit oder implizit (durch physikalische Allokierung) erfolgt.

Arbeitsdateien, temporäre Dateien und Dateien mit hohen Verfügbarkeitsanforderungen können generell nicht auf Hintergrundebenen gebracht werden. Dies wird durch HSMS unabhängig von den MIGRATE-Zuweisungen gewährleistet. Temporäre Dateien erhalten zusätzlich beim Anlegen, bzw. beim Umwandeln von permanenten Dateien in temporäre Dateien die Zuweisung MIGRATE=\*INHIBITED, sofern der Benutzer keinen anderen Wert vorgibt.

### Dateigröße

Die Größe einer Datei, welche für die Auswahl eines Volume-Sets mit einer passenden Allocation-Unit von Bedeutung ist, ergibt sich automatisch aus den diesbezüglichen Benutzerangaben beim Anlegen oder Vergrößern der Datei.

## 9.2 Nutzung der Hintergrundebenen für Dateiverdrängung

Die Hintergrundebenen eignen sich als kostengünstiger Ablageort für Dateien, auf die voraussichtlich längere Zeit nicht mehr zugegriffen wird und für die bei dem nächsten Zugriff von dem Benutzer eine längere Zugriffszeit in Kauf genommen wird. Außerdem können die Hintergrundebenen für die Reorganisation der S0-Ebene oder beim Auftreten akuter Speicherprobleme in der S0-Ebene auch temporär als Ablage für Dateien benutzt werden, die nicht die Bedingung der langfristigen Verdrängbarkeit erfüllen. Diese Dateien sollten nach Durchführung der Reorganisation bzw. nach der Entschärfung der Speicherengpässe (aus Sicht des Benutzers) wieder automatisch in die Verarbeitungsebene zurückkommen. Bei der Verlagerung von Dateien auf Hintergrundebenen ist zu beachten, dass bestimmte Dateien, wie Systemdateien, davon ausgeschlossen werden müssen.

### 9.2.1 Schnittstelle Benutzer/Systembetreuer

Der Benutzer hat die Kenntnis darüber, welche seiner Dateien sich für langfristige Verdrängung auf Hintergrundebenen eignen. Die Organisation und Durchführung der Migration ist in einem SMS-administrierten Rechenzentrum Aufgabe des Systembetreuers, da hierfür zentrale Ressourcen (wie Bandgeräte) bereitgestellt werden müssen und die Migration besonders effizient durchgeführt werden kann, wenn sie benutzerübergreifend erfolgt. Daher muss zwischen Benutzer und Systembetreuer eine einfach handhabbare Schnittstelle vorhanden sein, die es den Benutzern erlaubt, ihre Anforderungen zu artikulieren, und die dem Systembetreuer die Möglichkeit bietet, die unterschiedlichen Benutzerwünsche mit geringem administrativen Aufwand zu berücksichtigen. Hierfür werden durch die HSMS-Management-Klassen spezielle Funktionen bereitgestellt.

*Für die Verdrängung auf Hintergrundebenen relevante Kriterien*

Wenn auf eine Datei länger nicht zugegriffen wurde, wird sie oft auch anschließend für längere Zeit nicht bearbeitet. Daher eignet sich die Frist, die seit dem letzten Zugriff auf eine Datei vergangen ist, als Kriterium, um die Migrierbarkeit von Dateien automatisch zu bestimmen. Allerdings können für die einzelnen Dateien unterschiedliche Werte geeignet sein. Auch können sich für eine bestimmte Datei im Laufe ihres Bestehens verschieden lange Fristen als günstig erweisen. Da der Benutzer am besten das zeitliche Raster kennt, in dem er seine Dateien benötigt, sollte er die Zeit, nach der eine Datei als migrierbar bewertet wird, dateispezifisch beeinflussen können.

Im Fall, dass für einen SM-Pubset eine S1- und eine S2-Ebene vorhanden ist, muss entschieden werden, auf welche von ihnen eine Datei verdrängt wird. Die S1-Ebene ist gegenüber der S2-Ebene dadurch charakterisiert, dass die darauf befindlichen Dateien schneller in die Verarbeitungsebene zurückgebracht werden können und damit bei dem nächsten Zugriff eine kürzere Zugriffszeit ermöglicht wird. Die Speicherung in der S2-Ebene ist hingegen kostengünstiger. Für Dateien, die zunächst auf die S1-Ebene gekommen sind, kann es erwünscht sein, dass sie nach einer bestimmten Zeit von dort auf die S2-Ebene verdrängt werden. Für die Auswahl der Hintergrundebene und die Verlagerung der Dateien von der S1-Ebene auf die S2-Ebene sollten die Benutzer dateispezifische Anforderungen zum Ausdruck bringen können.

Weitere Größen, die bei der Bestimmung der Verdrängbarkeit dateispezifisch berücksichtigt werden sollten, sind die Größe und die Extent-Anzahl der Dateien.

*HSMS-Management-Klassen*

Die HSMS-Management-Klassen unterstützen das Zusammenwirken zwischen Systembetreuer, Benutzer und System bei der Benutzung der Hintergrundebenen. Dies wird in [Bild 29](#) illustriert.

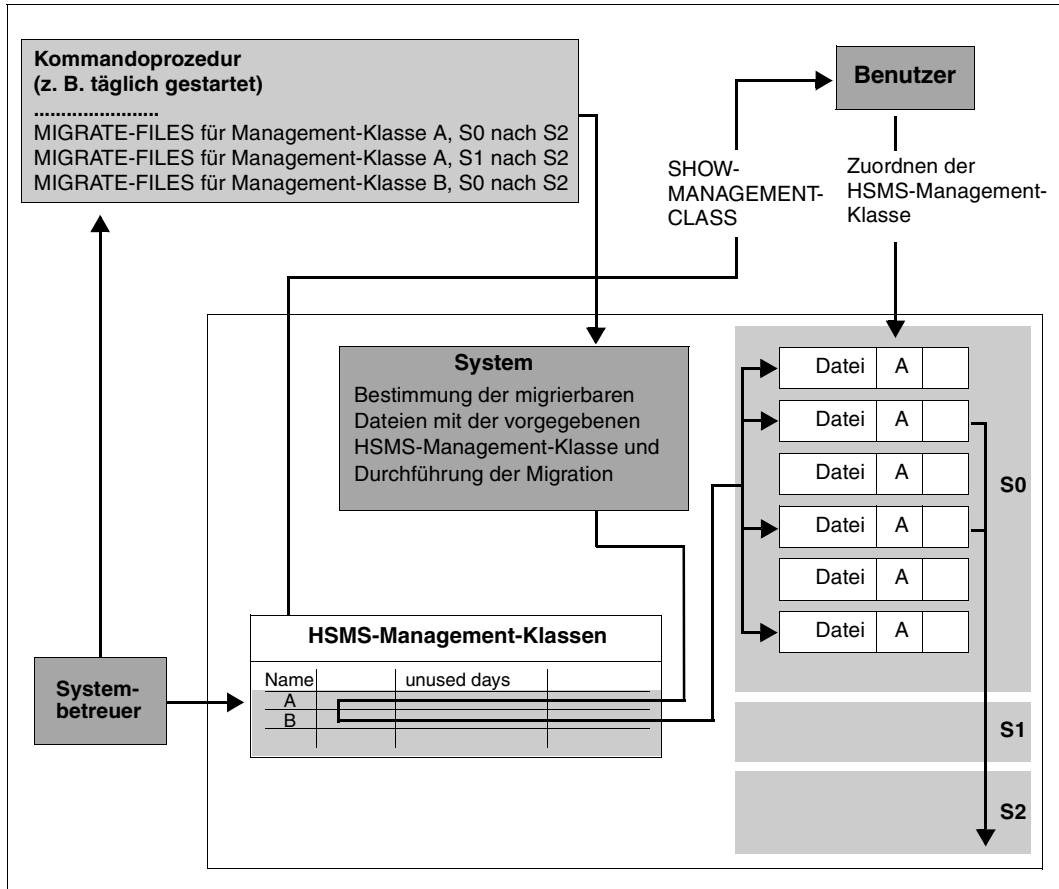


Bild 29: Zusammenwirken zwischen Systembetreuer, Benutzer und System bei der Verdrängung von Dateien auf die Hintergrundebenen

Die HSMS-Management-Klassen werden vom Systembetreuer eingerichtet (siehe HSMS-Anweisungen CREATE-MANAGEMENT-CLASS, MODIFY-MANAGEMENT-CLASS). Dabei werden ihnen Attribute zugeordnet, welche den verschiedenen, auf dem SM-Pubset unterstützten Anforderungsprofilen für Dateiverdrängung entsprechen.

Zur Steuerung der Dateiverdrängung von der S0-Ebene dienen folgende Attribute:

- UNUSED-DAYS (Frist, nach der die Datei als verdrängbar gilt)
- MAXIMUM-SIZE, MINIMUM-SIZE, MINIMUM-EXTENTS (Einschränkungen der Migrierbarkeit in Abhängigkeit von der Dateigröße, sowie der Extent-Anzahl der Datei)
- TO-STORAGE (Hintergrundebene, auf die Datei verdrängt werden soll)

Zur Steuerung der Dateiverdrängung von der S1-Ebene auf die S2-Ebene dienen folgende Attribute:

- MINIMUM-DAYS-ON-S1 und MAXIMUM-DAYS-ON-S1
- MINIMUM-SIZE
- TO-STORAGE=\*S2-STORAGE-LEVEL

Ein Benutzer kann sich über die ihm erlaubten HSMS-Management-Klassen und deren Attribute durch die HSMS-Anweisung SHOW-MANAGEMENT-CLASS informieren. Seine Wünsche hinsichtlich der Dateiverdrängung auf Hintergrundebenen bringt er dadurch zum Ausdruck, dass er durch das Kommando MODIFY-FILE-ATTRIBUTES den einzelnen Dateien die jeweils am besten geeignete HSMS-Management-Klasse zuweist. Die Verwendung der einzelnen HSMS-Management-Klassen kann durch GUARDS-Profile auf bestimmte Benutzer beschränkt werden.

Es ist empfehlenswert, dass der Systembetreuer einige spezielle HSMS-Management-Klassen vorsieht, z.B. eine Klasse „KEINE LÄNGERFRISTIGE MIGRATION“ (durch Wahl eines sehr hohen Wert für UNUSED-DAYS) oder eine Klasse „SOFORT MIGRIERBAR“ (UNUSED-DAYS=0, MAXIMUM-SIZE=\*NONE, MINIMUM-SIZE=\*NONE, MINIMUM-EXTENTS=1,..). Wünscht ein Benutzer, dass eine Datei möglichst bald verdrängt wird, z.B. weil er weiß, dass er diese Datei ab sofort längere Zeit nicht mehr benötigt, so kann er dies durch die Zuordnung der HSMS-Management-Klasse „SOFORT MIGRIERBAR“ erreichen. Der Speicherplatz in der S0-Ebene wird damit nicht unnötig lange belegt.



*Migrationssperren*

Die Reorganisation der S0-Ebene kann dadurch erfolgen, dass der Systembetreuer Dateien, die nicht auf optimal geeigneten Volume-Sets liegen, auf Hintergrundebenen migriert. Anschließend werden die Dateien wieder in die Verarbeitungsebene zurückgeholt, wobei ihre Verteilung neu festgelegt wird. Für eine effiziente Reorganisation ist es sinnvoll, dass auch Dateien auf Hintergrundebenen verlagert werden, welche zurzeit nicht die Kriterien einer längerfristigen Verdrängbarkeit erfüllen. Die in den HSMS-Management-Klassen hinterlegten Attribute für die längerfristige Hintergrundverdrängung können in diesem Fall ignoriert werden. Für die Dauer der Reorganisation ist der Zugriff auf die involvierten Dateien eingeschränkt. Eine analoge Situation ergibt sich, wenn bei temporären Speicherengpässen in der Verarbeitungsebene Dateien vorübergehend auf die Hintergrundebenen ausgelagert werden.

Bestimmte Dateien dürfen grundsätzlich nicht, d.h. auch nicht kurzfristig für eine Pubset-Reorganisation oder bei vorübergehenden Speicherengpässen auf eine Hintergrundebene gebracht werden. Sie müssen von den Benutzern entsprechend gekennzeichnet werden. Dies wird durch die harte Migrationssperre ermöglicht, die durch die Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES (Operand MIGRATE=\*FORBIDDEN) gesetzt werden kann. Die harte Migrationssperre wird durch HSMS automatisch berücksichtigt und kann auch durch den Systembetreuer nicht außer Kraft gesetzt werden. Sie ist z.B. wichtig im Zusammenhang mit physikalischer Allokierung, da auf Block- oder Volume-Ebene physikalisch allokierte Dateien fest an ihren Ablageort fixiert sind und diesen auch kurzfristig nicht verändern dürfen. Die harte Migrationssperre kann ferner dazu dienen, bestimmte Dateien zu kennzeichnen, die aus prinzipiellen Erwägungen die S0-Ebene nicht verlassen. In dieser Eigenschaft ist sie eine Alternative zur Benutzung von Except-File-Listen. Während diese von dem Systembetreuer (HSMS-Verwalter) in Absprache mit den Benutzern gepflegt werden müssen, kann das Setzen der harten Migrationssperre von den Benutzern selbst vorgenommen werden. Da sie dadurch ähnlich massive Einflussmöglichkeiten auf die Platzbelegung eines Pubsets erhalten wie bei physikalischer Allokierung, müssen sie zur physikalischen Allokierung berechtigt sein.

Neben der harten Migrationssperre gibt es auch eine weiche Migrationssperre (Dateiattribut `MIGRATE=*INHIBITED`). Sie kann durch den Systembetreuer außer Kraft gesetzt werden (siehe HSMS-Anweisung `MODIFY-HSMS-PARAMETERS`, Operand `MIGRATION-CONTROL`). Ihre Bedeutung ist nicht fest vorgegeben, sondern hängt davon ab, unter welchen Umständen sie der Systembetreuer berücksichtigt bzw. ignoriert. Die weiche Migrationssperre kann auf folgende Weise behandelt werden: Für Dateien, die mit einer weichen Migrationssperre versehen sind, erfolgt keine längerfristige Hintergrundverdrängung, unabhängig davon, ob ihnen eine HSMS-Management-Klasse zugeordnet ist, und wie die ggf. zugeordnete HSMS-Management-Klasse definiert ist. Bei kurzfristigen Verdrängungen, z.B. im Rahmen der Pubset-Reorganisation oder bei temporären Speicherengpässen wird die weiche Migrationssperre nicht berücksichtigt. Sie ist damit in ihrer Bedeutung weitgehend der Zuordnung einer HSMS-Management-Klasse „KEINE LÄNGERFRISTIGE MIGRATION“ (siehe oben) äquivalent. Im Gegensatz zu den HSMS-Management-Klassen können die Migrationssperren sowohl für in SF-Pubsets wie auch für in SM-Pubsets liegende Dateien genutzt werden. Sie bieten damit für SF-Pubsets den Benutzern ein einfaches Instrument zur Steuerung der Verdrängung auf Hintergrundebenen und sind auch bei SM-Pubsets sinnvoll, wenn die Einrichtung von HSMS-Management-Klassen als zu aufwändig betrachtet wird.

#### *Querbeziehungen zwischen Migrierbarkeit und anderen Dateiattributen*

Temporäre Dateien, Arbeitsdateien (Dateiattribut `WORK-FILE=*YES`) sowie Dateien mit hohen Verfügbarkeitsanforderungen (Dateiattribut `AVAILABILITY=*HIGH`) sind generell nicht auf Hintergrundebenen migrierbar. Für temporäre Dateien und Arbeitsdateien ist dies darin begründet, dass bei ihnen eine kurze Lebensdauer angenommen wird. Dateien mit hohen Verfügbarkeitsanforderungen werden deswegen nicht auf Hintergrundebenen gebracht, weil nicht davon ausgegangen werden kann, dass die für die Hintergrundebenen nutzbaren kostengünstigen Medien eine entsprechende Ausfallsicherheit bieten können. Als Konsequenz dieser Einschränkungen wird auch verhindert, dass ein Benutzer einer migrierten Datei das Attribut `AVAILABILITY=*HIGH` zuordnet bzw. eine migrierte Datei in eine temporäre Datei umwandelt. Zur Änderung dieser Attribute muss die Datei zuvor in die Verarbeitungsebene zurückgebracht werden.

## 9.2.2 Durchführung der Dateimigration auf die Hintergrundebenen

In diesem Abschnitt wird die Dateimigration vor allem im Zusammenhang mit der längerfristigen Dateiverdrängung betrachtet. Die kurzfristige Dateimigration bei temporären Speichereingpässen und bei der Pubset-Reorganisation wird vor allem in [Kapitel „Ressourcenkontrolle für SM-Pubsets“](#) behandelt.

HSMS-Management-Klassen ermöglichen nicht nur, ein differenziertes Service-Angebot für die Nutzung der Hintergrundebenen zu definieren, sondern bieten dem Systembetreuer auch komfortable Funktionen, dieses mit geringem Aufwand zu realisieren. Dabei kommt der Automatisierbarkeit der Migrationsaufträge eine große Bedeutung zu. Im Folgenden werden die durch BS2000 und insbesondere HSMS bereitgestellten Instrumente für die Automatisierung der Dateimigration dargestellt.

### *automatische Aktivierung von Migrationsaufträgen*

Eine zentrale Voraussetzung für die Automatisierbarkeit besteht darin, dass Migrationsaufträge automatisch angestoßen werden können. Dies kann z.B. mithilfe von ENTER-Jobs erreicht werden:

- Durch die REPEAT-JOB-Funktionalität können ENTER-Prozeduren für Migrationsaufträge in Zeitintervallen angestartet werden, die durch den Anwender vorgegeben werden können.
- Um zu bewirken, dass beim Eintreten von bestimmten Sättigungssituationen automatisch bestimmte Dateien auf Hintergrundebenen verdrängt werden, können die von HSMS angebotenen Jobvariablen zum Anzeigen der erreichten Sättigungswerte benutzt werden. Bei SF-Pubsets wirken diese pubset-spezifisch, bei SM-Pubsets erfolgt die Sättigungsüberwachung für jeden Volume-Set getrennt. Entsprechend sind bei SM-Pubsets die Jobvariablen für die Sättigungsüberwachung volume-set-spezifisch ausgeprägt. Über CJC (Funktionalität des Produkts Jobvariable) kann ein ENTER-Job in einen Wartezustand versetzt werden, der aufgehoben wird, wenn die Sättigungs-Jobvariablen bestimmte Werte annehmen. Nach dem Wecken aus dem Wartezustand können durch den ENTER-Job Migrationsaufträge gestartet werden.
- Auf Programmebene stehen Schnittstellen zur Verfügung, welche Zustandsänderungen der Jobvariablen für Sättigungsüberwachung sowie das Eintreten bestimmter zeitlicher Ereignisse signalisieren (Kalenderfunktionen). Zusammen mit den von HSMS bereitgestellten Programmschnittstellen zum Aktivieren von Migrations-Aufträgen gibt dies dem Systembetreuer die Möglichkeit, Programme zu entwickeln und in ENTER-Jobs zum Ablauf zu bringen, die ereignisgesteuert die Verdrängung auf die Hintergrundebenen veranlassen.

*Bestimmung der migrierbaren Dateien*

Der Systembetreuer darf mit der Überwachung der komplexen Kriterien für die Migrierbarkeit nicht belastet werden, sondern muss diese dem System übertragen können. Dies wird durch den Parameter MANAGEMENT-CLASS in der HSMS-Anweisung MIGRATE-FILES ermöglicht.

Wird ein Migrations-Auftrag für in der Verarbeitungsebene liegende Dateien initiiert (FROM-STORAGE= \*S0-LEVEL), bei dem eine bestimmte HSMS-Management-Klasse angegeben ist, so werden automatisch alle Dateien erfasst, die folgende Bedingungen erfüllen:

- Den Dateien ist die vorgegebene HSMS-Management-Klasse zugewiesen.
- Die Dateien erfüllen die in der HSMS-Management-Klasse spezifizierten Kriterien bezüglich der Verdrängung von der Verarbeitungsebene (UNUSED-DAYS, MINIMUM-SIZE, MAXIMUM-SIZE, MINIMUM-EXTENTS).
- Die Dateien sind auch nicht durch sonstige Bedingungen - wie beispielsweise Migrationssperren - von der Migration ausgeschlossen (s.u.).

Die Hintergrundebene, auf welche die Datei gebracht wird, ergibt sich aus der Definition der HSMS-Management-Klasse (Operand TO-STORAGE).

Sonstige Bedingungen, welche die Hintergrundmigration einer Datei nicht erlauben, sind u.a.:

- Es handelt sich um eine Arbeitsdatei, eine temporäre Datei oder eine Datei mit hohen Verfügbarkeitsanforderungen.
- Es handelt sich um eine Datei der Benutzerkennung SYSHSMS oder um eine Systemdatei.
- Die Datei ist in der Except-File-Liste des SM-Pubsets (HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS, Operanden MIGRATION-CONTROL, EXCEPT-FILE) oder in einer bei dem Migrations-Auftrag explizit angegebenen Except-File-Liste (siehe HSMS-Anweisung MIGRATE-FILES, Operanden S0-STORAGE-LEVEL, EXCEPT-FILE-NAMES) enthalten.
- Es handelt sich um die Datei einer File-Generations-Gruppe.
- Die Datei ist zurzeit eröffnet.
- Die Datei ist mit einem SECURE-Lock versehen.
- Die Datei muss repariert werden.
- Die Datei ist mit einer harten Migrationssperre (MIGRATE=\*FORBIDDEN) belegt.
- Die Datei ist mit einer weichen Migrationssperre (MIGRATE=\*INIBITED) versehen, welche auf dem entsprechenden SM-Pubset nicht ignoriert werden soll (HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS, Operanden MIGRATION-CONTROL, FILE-INHIBIT).

- Der Datei ist eine Backup-Klasse verschieden von E zugeordnet, es gibt keine aktuelle Sicherung der Datei und auf dem entsprechenden SM-Pubset dürfen aber nur aktuelle gesicherte Dateien migriert werden (HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS, Operanden MIGRATION-CONTROL, BACKUP-MANDATORY).
- Die Verdrängung auf Hintergrundebenen ist auf dem entsprechenden SM-Pubset nicht erlaubt (siehe HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS, Operanden MIGRATION-CONTROL, MIGRATION).
- Die Verdrängung auf Hintergrundebenen ist auf dem entsprechenden SM-Pubset auf die S2-Ebene beschränkt (HSMS-Anweisung MODIFY-SM-PUBSET-PARAMETERS, Operanden MIGRATION-CONTROL, MIGRATION), in der HSMS-Management-Klasse ist aber die S1-Ebene als gewünschte Hintergrundebene spezifiziert.

Die Verdrängung von der S1-Ebene auf die S2-Ebene erfolgt durch einen Migrations-Auftrag mit der Angabe FROM-STORAGE=\*S1-STORAGE-LEVEL. Wird dabei eine HSMS-Management-Klasse spezifiziert, werden alle auf der S1-Ebene befindlichen Dateien erfasst, welche gleichzeitig folgende Bedingungen erfüllen:

- Den Dateien ist die vorgegebene HSMS-Management-Klasse zugewiesen.
- Die Dateien erfüllen die in der HSMS-Management-Klasse spezifizierten Kriterien bezüglich der Verdrängung von der S1-Ebene (MINIMUM-DAYS-ON-S1, MAXIMUM-DAYS-ON-S1 sowie MINIMUM-SIZE).
- Die Dateien sind auch nicht durch sonstige Bedingungen (z.B. durch bestimmte Einstellungen von MIGRATION-CONTROL) von der Migration ausgeschlossen.

Um das für die einzelnen HSMS-Management-Klassen festgelegte Verdrängungsverhalten zu erzielen, ist es ausreichend, für jede HSMS-Management-Klasse einen (für HSMS-Management-Klassen mit direkter Verdrängung von der Verarbeitungsebene auf die S2-Ebene) oder zwei (für HSMS-Management-Klassen mit zweistufiger Verdrängung über die S1-Ebene) Migrations-Aufträge vorzusehen. Es ist dafür zu sorgen, dass die Migrationsaufträge genügend oft (z.B. täglich) gestartet werden. Je häufiger die Migrations-Aufträge durchgeführt werden, umso kürzer ist die durchschnittliche Zeit, in der Dateien noch Platz in der Verarbeitungsebene oder S1-Ebene belegen, nachdem für sie die Bedingung für die Migrierbarkeit eingetreten ist. Die in den Definitionen der HSMS-Management-Klassen festgelegten Fristen, anhand derer die Migrierbarkeit der Dateien bestimmt wird, brauchen bei der Häufigkeit der Migrationsaufträge nicht berücksichtigt werden.

Damit die Kommandoprozeduren mit den Migrationsaufträgen zu den gewünschten Zeiten ausgeführt werden können, muss der Systembetreuer natürlich auch sicherstellen, dass jeweils die erforderliche HSMS-Umgebung eingerichtet und betriebsfähig ist und dass die benötigten Betriebsmittel (Bänder, Bandgeräte etc.) bereitgestellt werden können. Durch den Einsatz von roboterbedienten Bandarchiven vereinfachen sich die hierfür erforderlichen Vorbereitungen.

*Benutzergesteuerte Dateimigration*

In einem SMS-administrierten Rechenzentrum ist die Organisation und Durchführung der Dateimigration Aufgabe des Systembetreuers (HSMS-Verwalters). Der Systembetreuer kann aber auch Benutzern das Recht einräumen, Migrationsaufträge selbst zu initiieren.

### 9.2.3 Recall von Dateien in die Verarbeitungsebene

Die Initiative für den Recall von langfristig verdrängten Dateien liegt im Allgemeinen beim Benutzer. Wie bei den Migrationsaufträgen muss der Systembetreuer dafür sorgen, dass die für die Ausführbarkeit der Recalls erforderliche HSMS-Umgebung eingerichtet und betriebsfähig ist und dass die erforderlichen Betriebsmittel (Bänder, Bandgeräte etc.) bereitgestellt werden können.

Der Recall einer Datei kann explizit oder implizit erfolgen:

- Ein expliziter Recall wird durch den HSMS-Aufruf `RECALL-MIGRATED-FILES` bewirkt.
- Ein impliziter Recall wird dann durchgeführt, wenn ein Benutzer eine auf einer Hintergrundebene liegende Datei eröffnet oder durch das Kommando `SECURE-RESOURCE-ALLOCATION` reserviert. Implizite Recalls von S2-Ebene können durch den Systembetreuer rechner-spezifisch eingeschränkt werden (HSMS-Anweisung `MODIFY-HSMS-PARAMETERS`, Operand `MIGRATION-CONTROL`).

Bereits bei SF-Pubsets können sich beim Zurückholen einer migrierten Datei verschiedene Probleme ergeben. Beispielsweise kann auf dem Pubset ein Sättigungszustand bestehen oder das Platzkontingent des Benutzers kann nicht ausreichend sein. Bei SM-Pubsets können noch komplexere Situationen auftreten, da die Benutzerkontingente differenzierter sind, und an Stelle der globalen Sättigungsüberwachung bei SF-Pubsets der Sättigungszustand auf den einzelnen Volume-Sets berücksichtigt wird. Außerdem ist nicht sichergestellt, dass für eine migrierte Datei in dem SM-Pubsets immer ein mit ihren Dateiattributen verträglicher Volume-Set existiert, auf den sie zurückgebracht werden kann - beispielsweise wenn nach der Hintergrundverdrängung aller K-Dateien der einzige K-Volume-Set durch Pubset-Rekonfiguration aus dem Pubset entfernt oder für Allokierung gesperrt wird. Es obliegt dem Systembetreuer, den Pubset in einem Zustand zu halten, dass im Normalfall der Recall von Dateien nicht scheitert, sofern der Benutzer über ausreichende Benutzerkontingente verfügt. In Sondersituationen, z.B. beim Auftreten von Fehlersituationen, Engpässen etc. kann dies nicht immer garantiert werden. Um sicherzustellen, dass die für eine Verarbeitung benötigten Dateien bereitgestellt werden können, und um unerwünschte Recall-Wartezeiten während der Verarbeitung zu vermeiden, sollte sie der Benutzer zuvor durch das Kommando `SECURE-RESOURCE-ALLOCATION` in die Verarbeitungsebene zurückzuholen. Sie sind dadurch anschließend auch vor der Migration auf Hintergrundebenen geschützt. Wenn für die Verarbeitung mehrere migrierte Dateien benötigt werden, ist es sinnvoll, mit `SECURE-RESOURCE-ALLOCATION` Sammelaufträge zu erzeugen, da sie eine besonders effiziente Durchführung der Recalls ermöglichen.

## 9.2.4 Einfluss von Migrate und Recall auf Dateieigenschaften

Die Migration einer Datei auf eine Hintergrundebene und ein anschließender Recall lassen die Eigenschaften einer Datei im Allgemeinen unverändert, wobei folgende Ausnahmen bestehen:

### *Lage der Datei (Extent-Listen)*

Sowohl der Ablageort der Datei in der Verarbeitungsebene, d.h. der Volume-Set, auf den die Datei gelangt, als auch die Lage der Datei innerhalb des Volume-Sets wird neu bestimmt. Dabei wird der Volume-Set ausgewählt, auf dem die Dateiattribute am besten erfüllt werden, falls nicht der Benutzer bei dem Aufruf RECALL-MIGRATED-FILES den Volume-Set explizit spezifiziert (Operand NEW-DATA-SUPPORT) oder die Datei auf den Volume-Set gebracht werden muss, auf dem sie sich vor der Hintergrundverdrängung befand (Dateiattribut S0-MIGRATION=\*FORBIDDEN). Für die explizite Vorgabe des Volume-Sets muss der Benutzer zur physikalischen Allokierung berechtigt sein. Im Unterschied zu dem Kommando CREATE-FILE führt die explizite Volume-Set-Vorgabe für die Datei nicht zu S0-MIGRATION=\*FORBIDDEN. Soll die Datei auf dem vorgegebenen Volume-Set fixiert werden, muss dies der Anwender im Anschluss an den Recall durch das Kommando MODIFY-FILE-ATTRIBUTES veranlassen. Die erneute Bestimmung des Volume-Sets im Rahmen eines Migrate-/Recall-Zyklus ist ein Instrument, um ungünstige Dateiverteilungen, bei denen Dateiattribute und Volume-Set-Attribute nicht harmonisieren, zu verbessern. Eine Ursache für eine ungünstige Verteilung besteht darin, dass Änderungen der Dateiattribute durch die Benutzer nicht zu einer unmittelbaren Verlagerung auf einen besser geeigneten Volume-Set führen, sofern nicht eine Unverträglichkeit der neuen Dateiattribute mit dem bisherigen Volume-Set dies erzwingt. Ein Benutzer kann nach einer Attributänderung die Verlagerung der Datei auf einen geeigneten Volume-Set dadurch forcieren, dass er für eine möglichst schnelle Verdrängung der betroffenen Datei auf eine Hintergrundebene sorgt (z.B. durch einen expliziten Migrations-Auftrag für dazu berechtigte Benutzer oder durch Zuordnung einer entsprechenden HSMS-Management-Klasse). Derartige Reorganisationsmaßnahmen durch die Benutzer sind aber nur in Sonderfällen sinnvoll, im Normalfall sollte die Pubset-Reorganisation zentral durch den Systembetreuer durchgeführt werden.

### *Dateigröße*

Wenn eine Datei auf eine Hintergrundebene migriert wird, wird vermerkt, wie viel Platz sie ursprünglich in der Verarbeitungsebene belegt hat. Beim Recall einer migrierten Datei wird versucht, in der Verarbeitungsebene wieder genau so viel Platz bereitzustellen. Dabei kann es sich als notwendig erweisen, den Wert so aufzurunden, dass er ein Vielfaches der Allocation-Unit des Volume-Sets bildet, auf den die Datei gebracht wird.

Bestimmte für den Ablageort in der Verarbeitungsebene relevante Dateiattribute (Performance-Attribute, Dateigröße, Storage-Klassen-Zuordnung, etc.) können durch den Benutzer geändert werden, während die Datei migriert ist. Die Modifikationen kommen erst dann zum Tragen, wenn die Datei wieder in die Verarbeitungsebene zurückgebracht wird.

## 9.3 Lebensdauer von Dateien

Die Initiative für das Löschen einer Datei liegt im Allgemeinen bei den Benutzern. Es existieren aber auch Mechanismen, durch welche der Benutzer das Löschen von Dateien an die Instanzen System bzw. Systembetreuer delegieren kann:

### Temporäre Dateien

Im Normalfall werden am Ende einer Benutzer-Session alle temporären Dateien, die in dieser Session angelegt wurden und noch nicht durch den Benutzer selbst gelöscht sind, automatisch durch das System entfernt.

### Arbeitsdateien

Für Arbeitsdateien ist es sinnvoll, dass zwischen dem Systembetreuer und den Benutzern ein Zeitpunkt vereinbart wird, ab dem der Systembetreuer zum Entfernen oder ggf. Zwangsentfernen mit Task-Terminierung der Work-Volume-Sets berechtigt ist. Beim Entfernen gehen die auf den Work-Volume-Sets befindlichen Arbeitsdateien verloren.

### FREE-FOR-DELETION-Attribut

Mit diesem Attribut (Kommandos CREATE-FILE, MODIFY-FILE-ATTRIBUTES) hat der Benutzer die Möglichkeit, einer Datei eine Frist zuzuordnen, nach deren Ablauf sie gelöscht werden kann. Da Dateien mit abgelaufener Frist durch die Selektionskriterien von SHOW-FILE-ATTRIBUTES und DELETE-FILE erfasst werden, können auf einfache Weise Aufräumprozeduren erzeugt werden. Die Aufräumaktivitäten können sowohl dezentral durch die einzelnen Benutzer, als auch zentral durch den Systembetreuer durchgeführt werden. Zentral durchgeführte Löschkaktionen erlauben beispielsweise dem Systembetreuer beim Auftreten von Speicherengpässen freien Platz zu gewinnen.

### RETENTION-PERIOD

Im Zusammenhang mit der Lebensdauer von Dateien ist auch die RETENTION-PERIOD von Bedeutung. Sie bietet einen Schutz dafür, dass Dateien nicht versehentlich vorzeitig gelöscht werden, stellt aber nicht automatisch sicher, dass für migrierte Dateien auch die auf den Hintergrundebenen befindlichen Dateiinhalte bis zum Ablauf der Retention-Period erhalten bleiben. Für Letzteres muss der Systembetreuer durch passende Festlegungen der Retention-Period der Save-Files Vorkehrungen treffen.



## 9.4 Sichern und Wiedereinspielen von Dateien

### 9.4.1 Anwendungsszenarien für Dateisicherung

Für die Durchführung der Dateisicherung sind verschiedene Anwendungsfälle zu unterscheiden:

- Ein Anwendungsfall besteht darin, dass ein Benutzer starke Einflussmöglichkeiten auf Sicherungszeitpunkt, Auswahl der Sicherungsbänder, Sicherungsverfahren etc. benötigt. Dies ist beispielsweise nötig um alle zu einer Applikation gehörenden Dateien gemeinsam zu einem bestimmten Konsistenzpunkt zu sichern und dabei die Sicherungen auf möglichst wenige Bänder zu streuen, sodass bei Bedarf ein schneller Restore möglich ist. Hierfür sind private Backup-Archive geeignet, die der Benutzer selbst pflegt und für die er die Backup-Aufträge selbst initiiert. Dabei steht dem Benutzer das volle HSMS-Funktionsspektrum für Backup zur Verfügung.
- In einem anderen Anwendungsfall ist den Benutzern der genaue Sicherungszeitpunkt nicht wichtig. Sie wollen auch möglichst wenig in die technische Durchführung der Backups involviert sein. Allerdings sind dateispezifische Einflussmöglichkeiten auf die Sicherungshäufigkeit sowie die Lebensdauer der Sicherungen erwünscht. Hierfür eignen sich zentral geführte Standard-Backup-Archive, die der Systembetreuer pflegt. Die Benutzer artikulieren gegenüber dem Systembetreuer die Wünsche hinsichtlich der Sicherungshäufigkeit, Lebensdauer der Dateisicherungen, Sicherungsverfahren (Vollsicherung/Teilsicherung) etc. Der Systembetreuer kümmert sich um eine den Benutzerwünschen entsprechende Durchführung der Sicherungen in ein Standardarchiv.

## 9.4.2 Benutzer-Systembetreuer-Schnittstelle

In einem SMS-administrierten Rechenzentrum kommt dem zweiten Anwendungsfall besondere Bedeutung zu. Wir wollen im Folgenden die Möglichkeiten skizzieren, die durch das System sowie durch HSMS für seine Unterstützung geboten werden.

Als Schnittstellen, die dem Anwender gestatten, seine Wünsche bezüglich der Durchführung der Sicherungen zu artikulieren, stehen die SAVE-Dateiattribute und die HSMS-Management-Klassen (nur bei SM-Pubsets) zur Verfügung. Beide lassen sich einer Datei durch die Kommandos CREATE-FILE und MODIFY-FILE-ATTRIBUTES zuweisen:

- Die SAVE-Attribute erlauben die Spezifikation einer Backup-Klasse und die Angabe, ob bei der nächsten Sicherung eine Differenzsicherung (nur seit der letzten Sicherung geänderte Blöcke) oder eine Vollsicherung erfolgen soll. Die Backup-Klasse der Datei legt fest, bei welchen Sicherungsläufen diese mit berücksichtigt wird. Sie beeinflusst damit die Sicherungsfrequenz. Die Bedeutung, welche einer Backup-Klasse konkret zukommt, ist davon abhängig, wie sie bei der Durchführung der Sicherungen berücksichtigt wird, insbesondere wie oft der Systembetreuer Sicherungsläufe für die entsprechende Backup-Klasse initiiert. Damit die Backup-Klasse von den Benutzern sinnvoll genutzt werden kann, ist bezüglich ihrer konkreten Bedeutung zwischen dem Systembetreuer und den Benutzern eine Vereinbarung erforderlich.
- Die HSMS-Management-Klassen können alternativ zu den Backup-Klassen benutzt werden, um die Sicherungshäufigkeit für Dateien festlegen zu können. Diese ist aber kein fest vorgegebener Bestandteil der HSMS-Management-Klasse, sondern muss analog wie bei einer Backup-Klasse zwischen Systembetreuer und Benutzer vereinbart werden. Der Systembetreuer kann den Service, den eine HSMS-Management-Klasse für die Sicherung darstellt, in dem Kommentarteil beschreiben. Zusätzlich enthalten die HSMS-Management-Klassen (zur Zeit) als einziges vorgegebenes sicherungsrelevante Attribut die RETENTION-PERIOD. Sie wird von dem Systembetreuer durch die HSMS-Anweisungen CREATE-/MODIFY-MANAGEMENT-CLASS für die verschiedenen HSMS-Management-Klassen eines SM-Pubsets festgelegt. Die Benutzer können sich durch die HSMS-Anweisung SHOW-MANAGEMENT-CLASS über die vom Systembetreuer festgelegten Merkmale der einzelnen HSMS-Management-Klassen informieren.

### 9.4.3 Durchführung der Dateisicherung durch den Systembetreuer

Folgende Möglichkeiten vereinfachen dem Systembetreuer die Aufgabe, die mit den Benutzern vereinbarten Services zu realisieren:

- Um Sicherungsläufe in verschiedenen Zeitrastern zu veranlassen, bieten sich analoge Automatisierungsverfahren an, wie im Zusammenhang mit Hintergrundverdrängung betrachtet wurden (ENTER-Prozeduren, Kalenderfunktionen etc.).
- Dass dabei jeweils die passenden Dateien erfasst werden, wird durch verschiedene Selektionskriterien unterstützt, die für (HSMS-) Backup-Aufträge zur Verfügung stehen, z.B. SELECT-FILES, MAXIMUM-BACKUP-CLASS, MANAGEMENT-CLASS. Um beispielsweise zu erreichen, dass Dateien mit BACKUP-CLASS=A täglich, Dateien mit BACKUP-CLASS=B wöchentlich, Dateien mit einer BACKUP-CLASS=C monatlich gesichert werden, können täglich Migrations-Aufträge mit MAXIMUM-BACKUP-CLASS=A, wöchentlich Aufträge MAXIMUM-BACKUP-CLASS=B, monatlich Aufträge mit MAXIMUM-BACKUP-CLASS=C gestartet werden. Durch die Angabe SELECT-FILE=\*MODIFIED-FILES kann verhindert werden, dass bei verschiedenen Sicherungsläufen unnötig oft identische Sicherungskopien entstehen.
- Wird bei BACKUP-FILES eine HSMS-Management-Klasse angegeben, so werden alle Dateien gesichert, die der entsprechenden HSMS-Management-Klasse angehören und nicht auf Grund anderer Selektionskriterien ausscheiden. Die Sicherungskopien gelangen in ein Save-File, für dessen Retention-Period automatisch durch HSMS die in der HSMS-Management-Klasse spezifizierte Retention-Period herangezogen wird.

### 9.4.4 Durchführung des Wiedereinspielens

Die Initiative für den Restore einer Datei kann abhängig von dem Anwendungsfall bei dem Systembetreuer oder dem Benutzer liegen. Im Fall, dass bei dem Ausfall von Platten Dateien zerstört worden sind, ist es sinnvoll, dass der Systembetreuer als Service für die Benutzer Sicherungen wieder einspielt. Falls ein Benutzer durch eigene Fehler Dateien zerstört hat, liegt die Initiative für den Restore bei den Benutzern.

## 9.5 Langzeitarchivierung

Langzeitarchive dienen als Behälter für Sicherungskopien von Dateien, die langfristig (z.B. auf Grund gesetzlicher Vorschriften) aufbewahrt werden müssen und auf die mit großer Wahrscheinlichkeit nur noch selten bzw. keine Zugriffe erfolgen. Die Initiative für die Durchführung der Langzeitarchivierungen liegt bei dem Benutzer. Entsprechendes gilt für einen Restore einer Datei aus einem Langzeitarchiv. Für die Delegation der technischen Durchführung der Langzeitarchivierung an den Systembetreuer sind zurzeit keine speziellen Hilfsmittel vorgesehen, sie kann jedoch auf Basis spezieller Absprachen erfolgen, z.B. der Systembetreuer veranlasst bei dem automatischen Löschen von Dateien mit abgelaufener FREE-FOR-DELETION-Frist das Erstellen von Sicherungskopien in ein Langzeitarchiv.

---

# 10 Pubset-Überwachung und Pubset-Pflege

## 10.1 Übersicht

Um einen Pubset in einem guten Zustand zu halten, sollte die Pubset-Pflege möglichst kontinuierlich durchgeführt werden. Wesentliche Voraussetzungen dafür sind, dass sie im laufenden Pubset-Betrieb erfolgen kann und dass möglichst geringe Störungen für die Benutzer entstehen. Das System stellt hierfür insbesondere bei SM-Pubsets spezielle Möglichkeiten zur Verfügung.

Die Pubset-Überwachung und -Pflege umfasst das Erkennen und Beheben folgender Situationen:

- Speicherengpässe in der S0-Ebene
- schlechter Organisationszustand der S0-Ebene (Fragmentierung des Speicherplatzes und Extent-Zersplitterung der Dateien, irrtümlich als belegt registrierter Platz, ungünstige Verteilung der Dateien auf die einzelnen Volume-Sets)
- Diskrepanzen zwischen den in den Eigenschaftsprofilen der Volume-Sets beschriebenen Performance- und Verfügbarkeitseigenschaften und dem realen Verhalten der Volume-Sets
- schlechter Organisationszustand der Hintergrundebenen und der Backup-Archive
- Speicherengpässe auf den Hintergrundebenen und in den Backup-Archiven
- Plattendefekte; der Systembetreuer sollte bereits im Voraus ein Recovery-Konzept für den Fall entwickeln, dass ein SM-Pubset nach einem teilweisen oder kompletten Ausfall in einem möglichst aktuellen Zustand wiederhergestellt werden kann. Die Realisierung eines Recovery-Konzepts erfordert im Allgemeinen bereits schon dann ständige Vorkehrungen, wenn der Pubset noch intakt ist und normal betrieben wird.
- Austausch von Hardware; Ersetzen veralteter Speichermedien

In den kommenden Abschnitten werden einige der oben aufgeführten Problemkreise eingehender betrachtet, wobei wir uns auf die SMS-spezifischen Aspekte beschränken, d.h. die Darstellung von Probleme und Vorgehensweisen zur Behebung, die für SM-Pubsets spezifisch sind.

## 10.2 Speicherengpässe in der S0-Ebene

Speicherengpässe in der S0-Ebene werden immer als Speicherengpässe auf einzelnen Volume-Sets sichtbar. Dabei muss zwischen verschiedenen Situationen unterschieden werden:

1. Speicherengpässe, die sich unter Beibehaltung der gewünschten Allokierungsstrategien durch Umverteilung der Dateien innerhalb des SM-Pubsets (auch mit Einbeziehung der Hintergrundebenen) entschärfen lassen. Beibehaltung der Allokierungsstrategien bedeutet dabei, dass bei der Umverteilung u.a.folgende Werte nicht außer Kraft gesetzt werden dürfen:
  - bestehende Allokierungssperren,
  - die den Dateien zugeordneten Storage-Klassen mit Volume-Set-Listen,
  - die Verträglichkeit von Dateiattributen und Volume-Set-Attributen,
  - die vom Administrator definierten Schwellwerte für Speichersättigung,
  - die mit den Anwendern vereinbarten Fristen für Dateiverdrängung auf Hintergrundebenen.
2. Speicherengpässe, die durch Umverteilung allein nicht behebbar sind. Sie können auch eintreten, wenn auf dem SM-Pubset insgesamt noch ausreichend freier Platz vorhanden zu sein scheint. Ihre Ursache besteht letztlich in nicht ausreichenden Pubset-Ressourcen. Sind diese Engpässe dauerhaft, können sie letztlich nur durch ein ressourcensparenderes Benutzerverhalten oder das Bereitstellen von zusätzlichen Ressourcen behoben werden. Kurzfristige Engpässe lassen sich oft durch spezielle Maßnahmen ohne gravierende Beeinträchtigungen der Benutzer überbrücken.

### **Beispiele für Engpässe, die durch Umverteilung behebbar sind**

Das System trägt bei der Auswahl der Volume-Sets der Belegungssituation zwar Rechnung, indem es den überschrittenen Sättigungsschwellwert berücksichtigt, und versucht extrem ungleiche Belegungen der einzelnen Volume-Sets zu vermeiden. Trotzdem können volume-set-lokale Engpässe nicht immer verhindert werden, wie folgende Beispiele zeigen:

1. Die Volume-Sets eines SM-Pubsets befinden sich alle in einem hohen Sättigungszustand, die Belegung ist zunächst ausbalanciert. Anschließend werden von den Benutzern Dateien gelöscht, wobei die einzelnen Volume-Sets zufallsbedingt sehr unterschiedlich betroffen sind.
2. Auf einem Volume-Set werden einzelne Dateien durch die Benutzer nach dem Anlegen z.B. durch viele Sekundärallokierungen unerwartet stark vergrößert (ungünstiges Benutzerverhalten). Auf dem Volume-Set befinden sich Dateien, die unter Beibehaltung der gewünschten Allokierungsstrategie innerhalb des SM-Pubsets ebenso gut auf andere, nicht so stark belastete Volume-Sets oder auf die Hintergrundebenen verlagert werden könnten.
3. Als Reaktion auf eine globale Sättigungssituation wird ein zusätzlicher Volume-Set in den Pubset aufgenommen. Die Verteilung auf den übrigen Volume-Sets bleibt zunächst gleich.

### **Beispiele für Engpässe, die in unzureichenden Ressourcen begründet sind**

In den beiden folgenden Beispielen gehen wir davon aus, dass die Dateiverdrängung auf Hintergrundebenen als Mittel zur Platzgewinnung in der Verarbeitungsebene bereits ausgeschöpft ist. Auf den überlasteten Volume-Sets liegen nur Dateien, die nicht (längerfristig) verdrängbar sind.

1. Der Volume-Set, auf dem der Engpass auftritt, hat als einziger Volume-Set des SM-Pubsets das Format K. Auf ihm liegen ausschließlich K-Dateien.
2. Eine bestimmter Volume-Set ist ausschließlich für Platzanforderungen bestimmter Benutzer vorgesehen. Die Kapazität des Volume-Sets ist aber nicht ausreichend, um deren Bedürfnisse abzudecken. Die Verwendung einzelner Volume-Sets als Ressourcen für bestimmte Benutzer lässt sich mit Hilfe von Storage-Klassen, Volume-Set-Listen und der Zuordnung von Default-Storage-Klassen erreichen.

In beiden Beispielen scheidet die Umverteilung der Dateien innerhalb der Verarbeitungsebene aus, weil entweder die bestehende Pubset-Konfiguration dies nicht zulässt, oder weil sonst vom Systembetreuer mit Hilfe der Storage-Klassen, Volume-Set-Listen, Allokierungssperren etc. implementierte Allokierungsstrategien durchbrochen werden müssten.

## Erkennen von Speicherengpässen

Da Speicherengpässe äußerst unangenehme Auswirkungen haben können, z.B. den Abbruch von laufenden Dateiverarbeitungen, wenn Sekundärallokierungen scheitern, ist es eine wichtige Aufgabe des Systembetreuers, sie zu vermeiden oder sie ggf. schnell zu entschärfen. Für die effiziente Durchführung der erforderlichen Maßnahmen ist es wichtig, dass das Erkennen der Engpässe und das Anstarten von Gegenmaßnahmen auch automatisiert werden kann.

Das Kommando `MODIFY-SPACE-SATURATION-LEVELS` erlaubt dem Systembetreuer das Einstellen von Sättigungsschwellwerten für die einzelnen Volume-Sets des SM-Pubsets. Der höchste überschrittene Sättigungsschwellwert beschreibt die aktuelle Sättigungssituation. Die eingestellten Schwellwerte werden mit dem Kommando `SHOW-PUBSET-PARAMETERS` und der aktuelle Sättigungszustand wird durch das Kommando `SHOW-PUBSET-SPACE-ALLOCATION` angezeigt. Für das automatische Erkennen von Speicherengpässen eignen sich HSMS-Jobvariablen, in denen Informationen über den Sättigungszustand eines SF-Pubsets bzw. von Volume-Sets hinterlegt werden. Sie können in Prozeduren und in Programmen abgefragt werden. Außerdem besteht für den Systembetreuer die Möglichkeit, sich in Programmen oder Kommandoprozeduren das Überschreiten der Schwellwerte durch Jobvariablen-Events signalisieren zu lassen. Damit können Kommandoprozeduren oder Programme eingesetzt werden, welche beim Eintreten von Speicherengpässen automatisch Gegenmaßnahmen ergreifen. Wünscht der Systembetreuer genauere Informationen über die Volume-Set-Belegung, kann er dazu das Kommando `SHOW-PUBSET-SPACE-ALLOCATION` verwenden.

## Maßnahmen zum Verhindern und Beheben von Speicherengpässen

Speicherengpässe werden dadurch verhindert bzw. behoben, indem auf den betroffenen Volume-Sets wieder ausreichend freier Platz geschaffen wird. Dies wird durch verschiedene Arten von Maßnahmen ermöglicht:

- Freigabe von bisher belegtem Platz durch Umverlagerung oder Löschen von Dateien
- Bereitstellen zusätzlicher Kapazitäten



*Möglichkeiten für die Verlagerung von Dateien*

1. Platzgewinnung durch Verdrängung langfristig verdrängbarer Dateien auf Hintergrundebenen

Als eine der ersten Maßnahmen für Platzgewinnung bietet es sich an, die in der Verarbeitungsebene liegenden Dateien, welche die zwischen Benutzern und Systembetreuer vereinbarten Kriterien für Hintergrundverdrängung erfüllen, auf die Hintergrundebenen zu migrieren. Wenn HSMS-Management-Klassen verwendet werden, um differenzierte Bedingungen für die Verdrängung festzulegen, kann der Systembetreuer für jede HSMS-Management-Klasse einen Migrations-Auftrag starten. Es werden dann genau die Dateien erfasst, welche die festgelegten Kriterien für die Verdrängung erfüllen.

2. Platzgewinnung auf einem Volume-Set durch Umverteilung von Dateien innerhalb der S0-Ebene

Die Umverteilung von Dateien zwischen den einzelnen Volume-Sets der Verarbeitungsebene kann erfolgen, indem die Dateien der Volume-Sets, auf denen Speicherengpässe bestehen, durch einen Migrate-Aufruf auf eine Hintergrundebene gebracht (Operand VOLUME-SET-ID in der HSMS-Anweisung MIGRATE-FILES) und anschließend durch einen Recall-Auftrag wieder in die Verarbeitungsebene zurückgeholt werden. Für die Benutzer können sich während der Umverteilens kurzfristige Beeinträchtigungen bei Dateizugriffen ergeben.

3. Platzgewinnung durch Verdrängung von Dateien, die nicht längerfristig migrierbar sind

In Sättigungssituationen kann es sich als notwendig erweisen, dass für eine bestimmte Zeit auch Dateien auf Hintergrundebenen migriert werden, welche nicht die Kriterien für längerfristige Verdrängung erfüllen (z.B. wenn die Zeit seit dem letzten Zugriff kürzer ist als die in der HSMS-Management-Klasse definierte Frist oder eine Datei mit der weichen Migrationssperre `MIGRATE=*INHIBITED` versehen ist). Das Verdrängen von Dateien, welche nicht die Kriterien der längerfristigen Verdrängbarkeit erfüllen, bedeutet eine Service-Verschlechterung für die Benutzer, da sich beim Zugriff lange Wartezeiten ergeben können. Der Systembetreuer sollte daher diese Maßnahmen nur in Ausnahmesituationen ergreifen. Auch sollten Dateien, die nur zur Überbrückung von Speicherengpässen auf Hintergrundebenen verdrängt sein sollen, durch den Systembetreuer automatisch wieder in die Verarbeitungsebene geholt werden, sobald eine Entspannung eingetreten ist.

*Bereitstellen zusätzlicher Speicherkapazitäten*

Für einen SM-Pubset kann durch die Hinzunahme weiterer Volume-Sets oder durch Vergrößern bestehender Volume-Sets zusätzliche Speicherkapazität bereitgestellt werden. Die dynamische Pubset-Rekonfiguration erlaubt, diese Maßnahmen im laufenden Pubset-Betrieb vorzunehmen.

Eine besonders kritische Situation besteht dann, wenn für Dateien, die während ihrer Verarbeitung stark anwachsen, nicht ausreichend Platz auf dem Volume-Set vorhanden ist, und alle Möglichkeiten, durch Umverlagern von Dateien Platz zu gewinnen, bereits ausgeschöpft sind. Da geöffnete Dateien nicht auf andere Volume-Sets verlagert werden können, hilft hier nur die Vergrößerung des Volume-Sets durch Volume-Hinzunahme. Derartig kritischen Engpässe sollte bereits im Vorfeld entgegengewirkt werden, z.B. indem Platzreserven für Sekundärallokierungen vorgesehen werden. Die SM-Pubsets bieten hierfür verschiedene Möglichkeiten. Beispielsweise kann ein Volume-Set beim Überschreiten eines bestimmten Sättigungsschwellwerts mit der Sperre NEW-FILE-ALLOCATION=\*PHYSICAL-ONLY belegt werden. Wird der Schwellwert wieder unterschritten, kann die Sperre wieder aufgehoben werden. Mit den HSMS-Jobvariablen für das Anzeigen von Sättigungssituationen lässt sich das Setzen bzw. Rücksetzen der Schwelle durch Kommando-prozeduren automatisieren. Wenn die Sperre gesetzt ist, weicht das System beim Anlegen neuer Dateien auf die übrigen Volume-Sets des SM-Pubsets aus. Der auf dem Volume-Set noch vorhandene freie Platz wird damit ausschließlich für Vergrößerungen bestehender Dateien herangezogen.

## 10.3 Qualität der Storage-Services

Jeder SM-Pubset bietet entsprechend seiner Volume-Set-Konfiguration für bestimmte Services eine optimale Unterstützung, erlaubt aber auch das Anlegen von Dateien, deren Dateiattribute möglich sind, aber nicht optimal erfüllt werden können. Beispielsweise können Dateien mit dem Attribut `PERFORMANCE=*VERY-HIGH` auch in einem SM-Pubset angelegt werden, dessen Volume-Sets alle das Volume-Set-Attribut `PERFORMANCE=*STD` haben. Das Kommando `SHOW-PUBSET-FILE-SERVICES` gibt Auskunft über die auf einem Pubset möglichen Storage-Services und den Grad ihrer Erfüllbarkeit.

Bei den optimal erfüllbaren Services sollten sich die Benutzer darauf verlassen können, dass die Dateien mit den entsprechenden Attributen die gewünschten Eigenschaften erhalten, bei den möglichen, aber nicht optimal unterstützten Services müssen Einschränkungen bezüglich der Qualität akzeptiert werden. Entsprechendes gilt auch für spezielle Storage-Services, welche durch Volume-Set-Listen realisiert werden. Spezielle Storage-Services werden den Benutzern durch Storage-Klassen zugänglich gemacht, wobei der Systembetreuer den Service in dem Kommentarteil der Storage-Klasse beschreibt. Die Benutzer sollten davon ausgehen können, dass Dateien durch die Zuordnung solcher Storage-Klasse Eigenschaften erhalten, welche der Beschreibung der Storage-Klasse entsprechen.

Damit die Storage-Services durch den Pubset in ausreichender Qualität realisiert werden, d.h. dass die vom Benutzer spezifizierten Dateiattribute mit den realen Eigenschaften der Dateien übereinstimmen, hat der Systembetreuer sich um folgende Fragen zu kümmern:

- Liegen die Dateien auf Volume-Sets, deren Eigenschaftsprofil den vom Benutzer spezifizierten Eigenschaften der Dateien (Dateiattributen) optimal entspricht? Im Fall, dass ein Storage-Service mithilfe einer Volume-Set-Liste realisiert wird, ist zusätzlich wichtig, dass sich die Datei auf einem Volume-Set der entsprechenden Volume-Set-Liste befindet.
- Entsprechen die vom Systembetreuer festgelegten Volume-Set-Attribute für Performance und Verfügbarkeit den realen physikalischen Eigenschaften der Volume-Sets? Ist dies nicht der Fall, basiert die Volume-Set-Auswahl auf falschen Kriterien. Ein analoges Problem besteht für spezielle Storage-Services, welche durch Volume-Set-Listen realisiert sind. In diesem Fall muss sichergestellt werden, dass die Eigenschaften der Volume-Sets in der entsprechenden Volume-Set-Liste der vom Systembetreuer erstellten Beschreibung der Storage-Klasse genügen.

Die beiden Punkte werden im Folgenden ausführlicher behandelt.

### 10.3.1 Optimaler Ablageort von Dateien

Die einzelnen Volume-Sets eines SM-Pubsets lassen sich bezüglich ihrer Eignung als Ablageort für eine bestimmte Datei bewerten, wobei die Eigenschafts-Profile der Volume-Sets, Volume-Set-Listen, Allokierungssperren und der Konfigurationszustand einfließen. Ein Volume-Set wird dann nicht als optimaler Ablageort für eine Datei betrachtet, wenn sich für einen anderen Volume-Set des SM-Pubsets eine bessere Bewertung ergibt.

Für die durch den SM-Pubset optimal unterstützten Services ist mindestens ein Volume-Set vorhanden, der die entsprechenden Dateiattribute vollständig erfüllt. Bei den möglichen, aber nicht optimal unterstützten Services ist dies nicht der Fall. Hier wird der Volume-Set als optimal bewertet, der den gewünschten Dateiattributen am ehesten entspricht. Beispielsweise gilt im Hinblick auf Performance ein Volume-Set mit PERFORMANCE=\*STD als optimaler Ablageort einer Datei mit dem Dateiaattribut PERFORMANCE=\*VERY-HIGH, wenn in dem SM-Pubset kein Volume-Set mit PERFORMANCE=\*HIGH oder PERFORMANCE=\*VERY-HIGH vorhanden ist.

#### Ursachen für die nicht optimale Ablage von Dateien

Dass sich Dateien in einem SM-Pubset nicht auf optimal geeigneten Volume-Sets befinden, kann folgende Ursachen haben:

- Der Pubset bietet nicht ausreichend Ressourcen. Steht auf den als optimal bewerteten Volume-Sets nicht ausreichend Platz für alle Dateien zur Verfügung, für die dieser Service angefordert wird, zieht das System bei der Bestimmung des Volume-Sets, auf dem eine Datei angelegt wird, auch nicht optimal bewertete Volume-Sets mit ein. Deswegen werden durch Speicherengpässe nicht optimale Belegungen forciert.
- Ändert sich die Volume-Set-Konfiguration eines Pubsets, z.B. durch Hinzunehmen neuer Volume-Sets, Aufheben von Allokierungssperren, Ändern von Performance- und Verfügbarkeitsprofilen etc., können sich bisher optimal geeignete Volume-Sets als nicht mehr optimal erweisen, da bessere Alternativen zur Verfügung stehen.
- Ändert ein Benutzer den gewünschten Storage-Service für eine Datei, z.B. indem er ihr eine andere Storage-Klasse zuordnet, wird der bisherige Ablageort beibehalten, falls er mit den neuen Attributen verträglich ist, auch wenn er nicht mehr optimal geeignet ist.

Wenn der Systembetreuer einer Storage-Klasse eine andere Volume-Set-Liste zuordnet oder die zugeordnete Volume-Set-Listen ändert, hat dies zunächst keine Auswirkung auf den Ablageort der Dateien.

## Auswirkungen eines nicht optimalen Ablageorts

Befinden sich Dateien nicht auf den optimal geeigneten Volume-Sets, hat dies folgende Auswirkungen:

- Wenn sich Dateien auf Volume-Sets befinden, welche die Dateiattribute nicht ausreichend erfüllen, ergibt sich eine Beeinträchtigung für die den Benutzern angebotene Service-Qualität. Dies ist beispielsweise der Fall, wenn eine Datei mit dem Dateiattribut `PERFORMANCE=*HIGH` auf einen Volume-Set mit dem Volume-Set-Attribut `PERFORMANCE=*STD` gelegt wird, obwohl eine bessere Alternative möglich wäre.
- Wenn Dateien auf Volume-Sets gelegt werden, die den von den Benutzern gewünschten Service übererfüllen, ergibt sich eine schlechte Nutzung der in dem Pubset vorhandenen Ressourcen. Diese Situation besteht z.B. dann, wenn eine Datei mit dem Dateiattribut `AVAILABILITY=*STD` auf einen Volume-Set mit dem Volume-Set-Attribut `AVAILABILITY=*HIGH` gelangt.

## Hilfsmittel zum Erkennen von Dateien mit ungünstigem Ablageort

Über die HSMS-Anweisung `SELECT-FILE-NAMES` (Operand `ALLOCATION-QUALITY=*BEST-VOLUME-SET` oder `*NOT-BEST-VOLUME-SET`) ist es dem Systembetreuer möglich, sich über die Dateien zu informieren, für welche der Ablageort optimal bzw. nicht optimal ist.

## Pubset-Reorganisation

Sind die Ressourcen eines SM-Pubsets im Hinblick auf die Benutzeranforderungen nicht ausreichend, müssen durch den Systembetreuer zusätzlich Ressourcen bereitgestellt werden (z.B. durch Hinzunahme weiterer Volume-Sets oder das Vergrößern bestehender Volume-Sets). Sind die Ressourcen zwar ausreichend, aber die Dateiverteilung innerhalb des SM-Pubsets ungünstig, muss der Pubset reorganisiert werden. Das Vorgehen ist wie bei der Reorganisation der Verarbeitungsebenen bei ungünstiger Belastung der einzelnen Volume-Sets und kann zusammen mit dieser erfolgen. Dateien mit nicht optimalem Ablageort werden auf Hintergrundebenen migriert. Zum Erfassen der Dateien bietet die `MIGRATE`-Anweisung den Operanden `DATA-LOCATION=*NOT-BEST-VOLUME-SET`. An Stelle dessen kann bei dem Migrations-Auftrag auch eine durch die HSMS-Anweisung `SELECT-FILE-NAMES` erzeugte Liste angegeben werden. Anschließend werden die Dateien wieder in die Verarbeitungsebene zurückgeholt. Dabei wird ihr Ablageort erneut bestimmt.

### 10.3.2 Mangelhafte Volume-Set-Qualität

Die Übereinstimmung zwischen dem von einem Volume-Set erwarteten Verhalten (entsprechend dem Eigenschaftsprofil des Volume-Sets oder der Beschreibungen in den Storage-Klassen) und seinem wirklichen Verhalten ist sehr wichtig. Längerfristige Diskrepanzen zwischen beschriebenem und wirklichem Verhalten bewirken, dass die Volume-Set-Selektion des Systems von falschen Voraussetzungen ausgeht und damit die Benutzeranforderungen falsch auf die physikalische Konfiguration abbildet. Dies reduziert die Qualität der vom Pubset gebotenen Storage-Services.

Die Möglichkeiten zur Überprüfung, ob ein Volume-Set tatsächlich die von ihm erwarteten Eigenschaften bietet, hängen sehr stark von seiner jeweiligen physikalischen Konfiguration ab. Wir verweisen daher auf die jeweiligen Auskunftsfunktionen für RAID, DRV, etc., in denen diese ausführlicher dargestellt sind. Stellt der Systembetreuer fest, dass ein Volume-Set nicht den Festlegungen in seinem Eigenschaftsprofil genügt, muss er versuchen, die Diskrepanz zu beheben. Die dazu erforderlichen Maßnahmen sind wiederum spezifisch für das jeweilige Speichermedium. Sind sie nicht erfolgreich, müssen die Eigenschaftsprofile korrigiert werden. Bietet beispielsweise ein Volume-Set mit dem Volume-Set-Attribut AVAILABILITY=\*HIGH nicht die gewünschte Ausfallsicherheit, sollte eine Rückstufung auf AVAILABILITY=\*STD vorgenommen werden. Die Dateien mit hohen Verfügbarkeitsanforderungen müssen dabei auf andere Volume-Sets gebracht werden, die Ausfallsicherheit bieten.

## 10.4 Vorkehrungen für die Wiederherstellbarkeit eines SM-Pubsets

Wenn ein Volume eines SM-Pubsets ausfällt, sind die Auswirkungen davon abhängig, welchem Volume-Set es angehört. Handelt es sich nicht um den Control-Volume-Set, ist nur der jeweilige Volume-Set selbst betroffen. Die auf ihm befindlichen Dateien sind verloren. Er kann als defekter Volume-Set aus dem SM-Pubset entfernt werden. Dabei wird eine Liste der zerstörten Dateien erstellt. Mit dieser Liste können die Dateien aus Sicherungen wiedereingespielt werden. Der Betrieb des übrigen SM-Pubset muss nicht unterbrochen werden, er ist auch in seiner sonstigen Funktionalität nicht eingeschränkt, da die zentralen Pubset-Metadaten auf dem Control-Volume-Set liegen. Gehört das ausgefallene Volume jedoch dem Control-Volume-Set an, ist ein ordnungsgemäßer Pubset-Betrieb nicht mehr möglich. Im Folgenden soll ausführlicher betrachtet werden, wie im letzten Fall der SM-Pubset mit möglichst geringen Verlusten wiederhergestellt werden kann.

Wegen der schwerwiegenden Auswirkungen, die der Ausfall des Control-Volume-Sets nach sich zieht, wird empfohlen, für ihn besonders ausfallsichere Datenträger, z.B. gespiegelte Platten zu verwenden. Auch sollte seine Plattenkonfiguration möglichst kompakt gehalten werden. Je geringer die Anzahl seiner Platten ist, um so geringer ist das Ausfallrisiko des Volume-Set. Der Ausfall des Control-Volume-Sets kann damit zwar äußerst unwahrscheinlich gemacht werden, trotzdem lässt er sich nicht prinzipiell ausschließen. Um bei seinem Auftreten den Umfang des Schadens (d.h. den Verlust von Benutzerdateien) möglichst gering zu halten, werden spezielle Funktionen für die Wiederherstellung des SM-Pubsets zur Verfügung gestellt. Dass der SM-Pubset auch möglichst schnell so wiederhergestellt kann, dass er wie vor dem Ausfall nutzbar ist und die Ausfallszeit gering ist, setzt zusätzlich bestimmte Vorkehrungen des Systembetreuers voraus, die bereits im normalen Pubset-Betrieb im Hinblick auf einen möglichen Ausfall zu erfolgen haben.

Die Basisfunktionalität für die Pubset-Wiederherstellung wird durch eine Konversionsfunktion bei der Pubset-Inbetriebnahme und die Utility SMPGEN bereitgestellt. Beide zusammen erlauben, den SM-Pubset aus den übrigen nicht ausgefallenen Volume-Sets in einem rudimentären Zustand zu rekonstruieren, wobei die Benutzerdateien in ihrem aktuellen Zustand zum Zeitpunkt des Ausfalls erhalten bleiben. Für den rekonstruierten Pubset bestehen zunächst noch Einschränkungen. Sie betreffen beispielsweise den Zugriff auf Dateien, die auf Hintergrundebenen verdrängt sind, die Nutzung von Dateisicherungen, die vor dem Pubset-Ausfall erstellt wurden, den Ressourcenschutz durch GUARDS-Profile etc. Um sie aufzuheben und den SM-Pubset anschließend wieder ordnungsgemäß betreiben zu können, sind weitere Wiederherstellungsschritte notwendig.

Die Vorkehrungen, die der Systembetreuer im normalen Pubset-Betrieb im Hinblick auf einen Ausfall des Control-Volume-Sets treffen sollte, lassen sich am besten anhand der für die Wiederherstellung des SM-Pubsets erforderlichen Schritte verdeutlichen. Sie zeigen am besten den Verlust der Funktionalität und die Schwierigkeiten, die sich ergeben, wenn die Vorkehrungen nicht oder nur unzureichend erfolgt sind. Wir betrachten die Schritte für die Wiederherstellung an dem exemplarischen Fall eines SM-Pubsets, auf dessen Control-Volume-Set ausschließlich Pubset-Metadaten liegen (empfohlenes Szenario insbesondere für große SM-Pubsets). Die Benutzerdaten befinden sich auf Daten-Volume-Sets, sofern sie nicht auf Hintergrundebenen verdrängt sind. Der SM-Pubset soll auch einen S1-Volume-Set beinhalten, der sowohl für Dateiverdrängung als auch für Backup eingesetzt wird.

Nach dem Ausfall des Control-Volume-Sets sind die darauf befindlichen Daten zerstört, die übrigen Volume-Sets sind mit ihrem Dateibestand noch verfügbar, aber nicht direkt nutzbar (Zustand: freie, aber nicht leere Volume-Sets). Die Volume-Sets enthalten für die auf ihnen liegenden Dateien auch die Katalogeinträge mit den Dateiattributen, die für ihre Nutzung im SM-Pubsets relevant sind, wie Verfügbarkeit, Performance, Arbeitsdatei, zugeordnete Storage-Klasse, zugeordnete Management-Klasse etc. Jeder Volume-Set enthält auch rudimentäre Informationen über die Pubset-Benutzer, von denen Dateien auf ihm liegen.

### **Wiederherstellung des rudimentären SM-Pubsets mit den Benutzerdateien in der S0-Ebene**

Für die Wiederherstellung des SM-Pubsets muss der Systembetreuer zunächst in einem Zwischenschritt SF-Pubsets bilden. Für den künftigen Control-Volume-Set empfiehlt es sich, mit SIR einen neuen SF-Pubset zu erzeugen, die übrigen SF-Pubsets erhält er dadurch, dass er die Volume-Sets des ausgefallenen SM-Pubsets im Rahmen einer speziellen Pubset-Inbetriebnahme in SF-Pubsets umwandelt. Dabei bleiben die darauf befindlichen Dateien erhalten. Auch Dateiattribute, die nur für SM-Pubsets relevant sind (wie die einer Datei zugeordnete Storage-Klasse), gehen nicht verloren, sind aber auf den SF-Pubsets nur latent vorhanden. Die Benutzerkataloge werden aus den Benutzerinformationen, die auf den Volume-Sets hinterlegt sind, rekonstruiert. Obwohl die SF-Pubsets anschließend einzeln in Betrieb genommen werden können, wird i.A. davon abgeraten. Sie sollten hauptsächlich als Zwischenprodukte für die Wiederherstellung des gesamten SM-Pubsets gesehen werden.

Im nächsten Schritt hat der Systembetreuer die SF-Pubsets mit Hilfe von SMPGEN zum SM-Pubset zusammenzufassen. Der SF-Pubset, der aus dem früheren S1-Volume-Set hervorgegangen ist, muss SMPGEN speziell bekannt gegeben werden, damit er diese Rolle wieder erhält. Das Ergebnis ist ein funktionsfähiger SM-Pubset, der alle Benutzerdateien in ihrem aktuellen Zustand vor dem vorausgegangenen Pubset-Ausfall enthält, dem aber noch der Teil der früheren Pubset-Infrastruktur fehlt, die auf dem zerstörten Control-Volume-Set hinterlegt war. Ein Blick auf die Systemdateien, die auf dem Control-Volume-Set angelegt werden, zeigt uns, welche Daten davon betroffen sind.



## Einrichten der HSMS-Umgebung für den SM-Pubset

Auf dem Control-Volume-Set liegen auch die Metadaten-Dateien von HSMS. Dies ist für die Directories zwar nicht zwingend, wird aber empfohlen. Von diesen sind die Steuerdatei und die Directories für Migrations- und Backup-Archive von besonderer Bedeutung. Die Steuerdatei enthält die Konfigurationsdaten der für den SM-Pubset eingerichteten HSMS-Umgebung (Archivdefinitionen, Konfigurationsparameter, etc.). Als Folge ihres Verlusts muss die HSMS-Umgebung auf dem rekonstruierten SM-Pubset wieder erneut eingerichtet werden. Ferner sind die verloren gegangenen Directories aus Sicherungen oder Kopien wiederherzustellen. Diese sollten möglichst aktuell sein. Für ein Backup-Archive lässt sich dies beispielweise erreichen, indem bei Sicherungen von Benutzerdateien zuletzt immer auch das Directory des Backup-Archivs mitgesichert wird. Eine andere Möglichkeit besteht darin, auf einem anderen Volume-Set des SM-Pubsets immer aktuelle Kopien der Directories zu halten. Directories, für die keine Sicherungen vorhanden sind, lassen sich in Notfall auch aus den Save-Files mit den Dateisicherungen oder den migrierten Dateien rekonstruieren, jedoch ist dieses Verfahren äußerst mühsam.

## Wiederherstellen der Hintergrundebenen

Für die Dateien, die beim Ausfall des Control-Volume-Sets auf Hintergrundebenen verdrängt waren, sind zwar die auf den Bändern der S2-Ebene oder dem S1-Volume-Set abgelegten Dateieinhalte noch verfügbar, durch den Ausfall des Control-Volume-Sets sind aber auch das Directory des Migrations-Archivs (s.o.) und die Katalogdatei für migrierte Dateien verloren gegangen. Die Katalogdatei für migrierte Dateien wird bei der Bildung des SM-Pubsets mit SMPGEN wieder automatisch angelegt, ist aber zunächst leer. Sie muss durch das Wiedereinspielen von gesicherten Katalogeinträgen für die migrierten Dateien wieder gefüllt werden. Dass im Notfall möglichst aktuelle Sicherungen verfügbar sind, erfordert aktive Vorkehrungen des Systembetreuers im normalen Pubset-Betrieb.

## Wiederherstellen der Pubset-Konfigurationsdaten

Die Konfigurationsdatei enthält alle wesentlichen Konfigurationsparameter für den SM-Pubset. Diese umfassen auch die Eigenschaftsprofile, Nutzungsarten, Nutzungsrestriktionen, Sättigungsschwellwerte für die einzelnen Volume-Sets sowie auch pubset-globale Festlegungen, wie die Voreinstellungen für den Pubset-Space. Durch den Ausfall des Control-Volume-Sets gehen diese Einstellungen verloren. Bei der Bildung des SM-Pubsets mit SMPGEN werden die Konfigurationsparameter überwiegend mit Standardwerten belegt, für einige erlaubt SMPGEN auch Angaben durch den Systembetreuer. Die Konfigurationsparameter korrespondieren mit dem vom Anwender gewünschten Nutzungsszenario. Um den Pubset wie vor dem Ausfall betreiben zu können, müssen sie nach der Rekonstruktion des SM-Pubsets mit den Kommandos der dynamischen Pubset-Rekonfiguration wieder passend eingestellt werden. Für die Konfigurationsdatei gibt es keine direkte Sicherungsmöglichkeit. Damit im Bedarfsfall die alten Einstellungen wieder in Erfahrung gebracht werden können, wird empfohlen, im normalen Pubset-Betrieb nach Änderungen von

Konfigurationseinstellungen ihre aktuellen Werte mit den Pubset-Auskunftsfunktionen in S-Variablen auszugeben und in einer Datei abzulegen. Die Datei soll natürlich nicht auf dem Control-Volume-Set angelegt werden. Mit Kommandoprozeduren kann dieses Vorgehen automatisiert werden. Die Einstellung der Nutzungsart HSMS-CONTROLLED erfordert spezielle Beachtung. Als Voreinstellung erhalten bei der Pubset-Bildung mit SMPGEN alle Volume-Sets die Nutzungsart STANDARD. Nach der Pubset-Bildung ist eine Änderung nur für Volume-Sets im Zustand „defined only“ möglich. Erhält ein früherer S1-Volume-Set bei der Pubset-Bildung die Nutzungsart STANDARD, kann er später nicht mehr passend umdefiniert werden, ohne seinen Bestand von S1-Sicherungen und migrierten Dateien zu verlieren. Es ist daher darauf zu achten, dass für einen solchen Volume-Sets bereits bei der Pubset-Bildung mit SMPGEN die Nutzungsart HSMS-CONTROLLED vorgegeben wird.

### **Pubset-Benutzerkatalog**

Bei der Bildung des SM-Pubsets wird zwar der Benutzer-Katalog wieder erzeugt, die auf dem früheren SM-Pubset eingerichteten Benutzerkontingente, Benutzerberechtigungen für physikalische Allokierung und Zuordnungen für Default-Storage-Klassen sind darin aber nicht enthalten. Da es sich dabei um Festlegungen handelt, die für die Nutzung des SM-Pubsets von außerordentlicher Bedeutung sein können - z.B. kann der Systembetreuer durch Zuordnung von Default-Storage-Klassen die Allokierungsstrategie innerhalb eines SM-Pubsets wesentlich beeinflussen -, sollten diese Informationen in möglichst aktuellem Zustand wie zum Zeitpunkt des Pubset-Ausfalls z.B. aus einer Sicherung des Pubset-Benutzer-Katalogs wiederherstellbar sein.

### **Einrichten des GUARDS- Katalog**

Durch den Ausfall des Control-Volume-Sets geht der GUARDS-Katalog verloren. Um den Dateischutz wieder zu etablieren, ist eine Wiederherstellung aus Sicherungen notwendig.

### **Storage-Klassen, Volume-Set-Listen und Management-Klassen**

Durch den Ausfall des Control-Volume-Sets gehen die auf dem SM-Pubset eingerichteten Storage-Klassen und Volume-Set-Listen verloren. Falls der Systembetreuer sie verwendet, um bestimmte Strategien der Speicherplatznutzung auf dem SM-Pubset zum Einsatz zu bringen, muss er sie nach dem Ausfall des Control-Volume-Sets wieder in dem gewünschten Zustand einrichten. Gleiches gilt auch für die Management-Klassen.

---

# 11 Ressourcenkontrolle für SM-Pubsets

Für die benutzerbezogene Ressourcenüberwachung auf SM-Pubsets gibt es folgende Funktionen:

- Benutzerkontingente für den durch Dateien belegten Platz
- DMS-TUNING-RESOURCES
- GUARDS-Schutz für Storage-Klassen und HSMS-Management-Klassen
- Berechtigung zur physikalischen Allokierung
- Maximalanzahl für katalogisierte Dateien und Jobvariable

Sie werden in den folgenden Abschnitten ausführlicher behandelt.

## 11.1 Benutzerkontingente für belegten Platz

Das für SF-Pubsets bestehende Konzept der Benutzerkontingente, das dem Systembetreiber erlaubt, den von den einzelnen Benutzern auf einem Pubset belegten Platz zu kontrollieren und zu begrenzen, findet auch bei den SM-Pubsets Anwendung. Dabei wird den Besonderheiten der SM-Pubsets durch eine größere Differenzierung der Kontingente Rechnung getragen. Bei SM-Pubsets, für die die differenzierte Nutzung und Kontrolle der Ressourcen nicht von Bedeutung ist, können die Benutzerkontingente wie bei SF-Pubsets administriert werden

Auf SF-Pubsets erfolgt eine separate Kontingentierung des von permanenten und temporären Dateien belegten Platzes (PERM-SPACE-Kontingent, TEMP-SPACE-Kontingent). Die Space-Kontingentierung beschränkt sich auf den in der Verarbeitungsebene belegten Plattenspeicherplatz. Sie umfasst folgende (pubset-spezifisch wirkende) Funktionen:

- Registrierung des von den einzelnen Benutzern belegten Platzes für permanente Dateien und temporäre Dateien (Space-Used-Erfassung)
- Möglichkeiten für den Systembetreiber, für die einzelnen Benutzer obere Schranken für den durch permanente bzw. temporäre Dateien belegten Platz festzulegen (Space-Limits)

- Möglichkeiten für den Systembetreuer, für die einzelnen Benutzer festzulegen, wie das System bei Überschreitung der Space-Limits reagieren soll (Public-Space-Excess). Diese Festlegungen wirken für das PERM-SPACE-Kontingent und das TEMP-SPACE-Kontingent identisch.

Das Kontingentkonzept für SM-Pubsets hat folgende Erweiterungen:

- Es gibt zusätzlich den Kontingenttyp WORK-SPACE für den durch Arbeitsdateien (Dateiattribut WORK-FILE) belegten Platz.
- Die PERM-SPACE-, TEMP-SPACE- und WORK-SPACE-Kontingente sind in sich strukturiert. Durch Unterkontingente wird berücksichtigt, dass der belegte Platz hinsichtlich Verfügbarkeit und Performance von unterschiedlicher Qualität sein kann.
- Auch der auf der S2-Hintergrundebene belegte Platz wird bei der Einteilung in Kontingente miteinbezogen.

Wie bei SF-Pubsets können benutzerspezifische Überschreitungsregeln definiert werden. Diese wirken für alle Kontingente bzw. Unterkontingente des SM-Pubsets gleich.

### 11.1.1 Kontingentstruktur für permanente Dateien auf SM-Pubsets

Der Kontingentstruktur für permanente Dateien in einem SM-Pubset liegt die Idee zu Grunde, dass die Qualität des in einem SM-Pubsets vorhandenen Platzes durch verschiedene Detaillierungsgrade charakterisiert werden kann. Diese werden durch [Bild 30](#) illustriert:

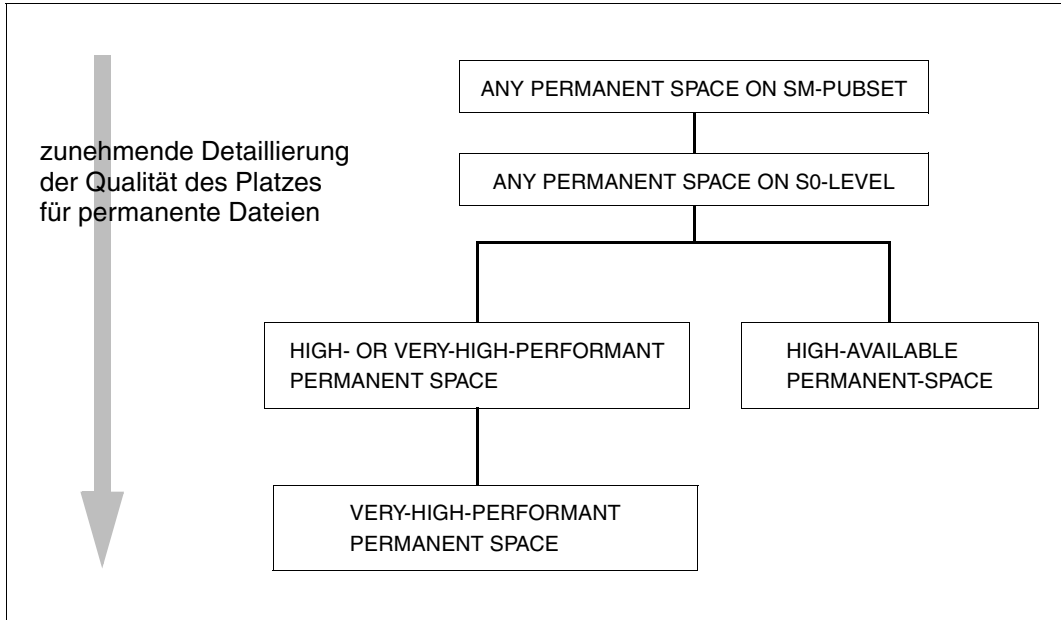


Bild 30: Detaillierungsgrade für die Qualität des Platzes für permanente Dateien

Die verschiedenen Detaillierungsgrade für die Qualität des Platzes werden dadurch berücksichtigt, dass einzelne Kontingente Unterkontingente für ihre höherwertigen Anteile enthalten. Für die permanenten Dateien eines SM-Pubsets sind die einzelnen Kontingente und ihre gegenseitigen Beziehungen in [Bild 31](#) dargestellt:

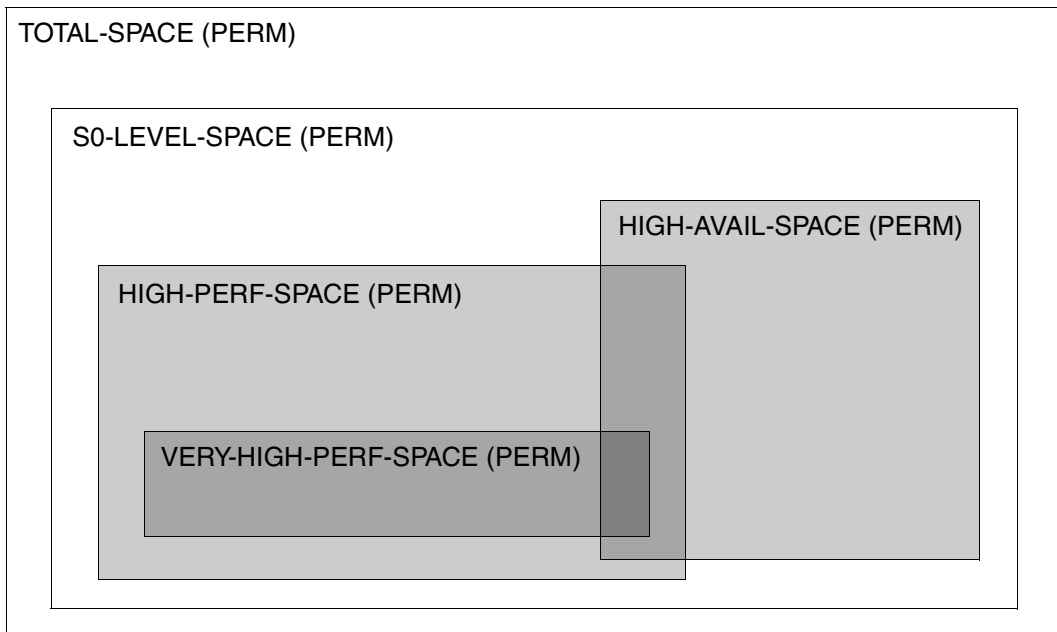


Bild 31: Kontingentstruktur für permanente Dateien in einem SM-Pubset

Der Platz, den eine Datei belegt, führt zu einer Belastung der Benutzerkontingente. Die betroffenen Benutzerkontingente ergeben sich aus den Dateieigenschaften. In [Bild 32](#) ist für permanente Dateien der Zusammenhang zwischen Dateiattributen und Kontingenten dargestellt, wobei zu beachten ist, dass die Belastung eines Kontingents implizit auch eine Belastung aller Kontingente bewirkt, welche dieses einschließen. Beispielsweise bewirkt das Anlegen einer Datei mit den Attributen `AVAILABILITY=*HIGH` und `PERFORMANCE=*HIGH`, dass neben den Belegtwerten der Kontingente für `HIGH-AVAIL-SPACE` und `HIGH-PERF-SPACE` auch die für `S0-LEVEL-SPACE` und `TOTAL-SPACE` erhöht werden.

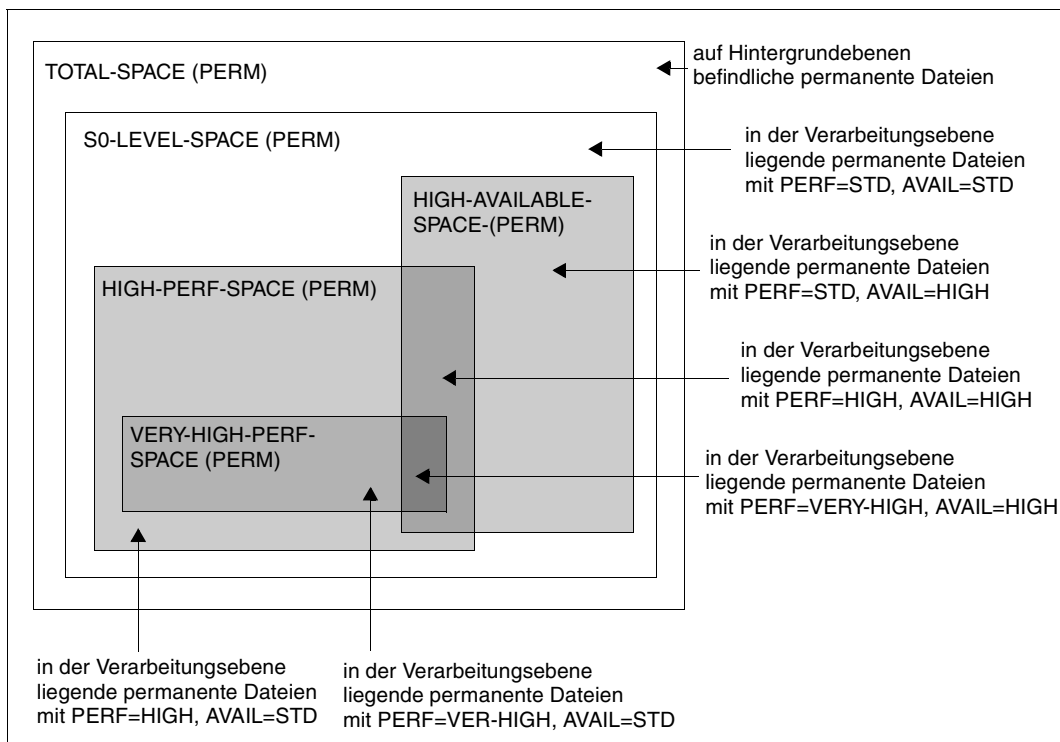


Bild 32: Zuordnung von Dateiattributen und Kontingenten

Bild 32 korrespondiert mit den folgenden Definitionen für die einzelnen Kontingente:

### TOTAL-SPACE-Kontingent für permanente Dateien

Diesem Kontingent wird der Platz für alle auf dem Pubset liegenden permanenten Dateien (eines Benutzers) zugerechnet, unabhängig davon ob die Dateien auf einer Hintergrundebene oder auf der S0-Ebene liegen und unabhängig davon, welche Performance- und Verfügbarkeitsattribute ihnen zugeordnet sind. Als Maßeinheit dient nicht der auf den Hintergrundebenen real belegte Platz, sondern der Platz, den die Datei nach dem Zurückbringen in die S0-Ebene (Recall) in dieser belegt würde.

### S0-LEVEL-SPACE-Kontingent für permanente Dateien

Der Anteil des TOTAL-SPACE-Kontingents, der sich aus dem in der S0-Ebene belegten Platz ergibt, bildet das S0-LEVEL-SPACE-Unterkontingent. Ihm wird der Platz aller in der S0-Ebene des Pubsets liegenden permanenten Dateien (eines Benutzers) zugerechnet - unabhängig davon, welche Performance- und Verfügbarkeitsattribute ihnen zugeordnet sind.

### **HIGH-PERF-SPACE-Kontingent für permanente Dateien**

Dies ist ein Unterkontingent des S0-LEVEL-SPACE-Kontingents. Ihm wird der Platz für jede in der S0-Ebene des Pubsets liegende permanente Datei (eines Benutzers) zugerechnet, der das Performance-Attribut PERFORMANCE=\*HIGH bzw. PERFORMANCE=\*VERY-HIGH zugewiesen ist.

### **VERY-HIGH-PERF-SPACE-Kontingent für permanente Dateien**

Es ist ein Unterkontingent des HIGH-PERF-SPACE-Kontingents für permanente Dateien. Ihm wird der Platz für jede in der S0-Ebene des Pubsets liegende permanente Datei (eines Benutzers) zugerechnet, der das Performance-Attribut PERFORMANCE=\*VERY-HIGH zugeordnet ist.

### **HIGH-AVAILABLE-SPACE-Kontingent für permanente Dateien**

Es ist ein Unterkontingent des S0-LEVEL-SPACE-Kontingents. Ihm wird der Platz für jede in der S0-Ebene des Pubsets liegende permanente Datei (eines Benutzers) zugerechnet, der das Attribut AVAILABILITY=\*HIGH zugeordnet ist.

Jedes Kontingent wird durch seinen Maximalwert (Space-Limit) und den aktuellen Belegungswert (Space-Used) charakterisiert. Der aktuelle Belegungswert der Kontingente ergibt sich aus der Größe der Dateien eines Benutzers und deren Dateiattributen. Das System überwacht bei dem Anlegen und Vergrößern von Dateien, dass die Maximalwerte nicht überschritten werden, sofern keine Sonderbedingung (Public-Space-Excess) vorliegt, welche das Überschreiten gestattet.

Ein Kontingent, das ein höherwertiges Unterkontingent enthält, kann von einem Benutzer ausgeschöpft werden, ohne dass das Unterkontingent voll beansprucht sein muss. In [Bild 33](#) wird dies verdeutlicht. In dem darin gezeigten Beispiel ist das Kontingent für TOTAL-SPACE vollständig ausgeschöpft, nicht jedoch das in diesem enthaltene Kontingent für S0-LEVEL-SPACE. Das Anlegen neuer Dateien und das Erweitern von Dateien ist in dieser Situation nicht möglich, da dies zu einer Überschreitung des Maximalwerts für das TOTAL-SPACE-Kontingent führen würde. Für das Verdrängen von Dateien aus der Verarbeitungsebene auf eine Hintergrundebene bestehen keine Beschränkungen, da dabei der Belegtwert für das TOTAL-SPACE-Kontingent unverändert bleibt, der Belegtwert des S0-LEVEL-SPACE-Kontingents kleiner wird, und die Belegtwerte der darin enthaltenen Unterkontingente reduziert werden oder gleich bleiben. Auch das Zurückholen von Dateien aus einer Hintergrundebene in die Verarbeitungsebene ist in der dargestellten Situation möglich, allerdings nur in dem Umfang, in dem die Maximalwerte des S0-LEVEL-SPACE-Kontingents und dessen Unterkontingente nicht überschritten werden. Beispielsweise würde der RECALL für eine Datei mit dem Attribut PERFORMANCE=\*VERY-HIGH scheitern, da das VERY-HIGH-PERF-SPACE-Kontingent bereits voll ausgeschöpft ist.



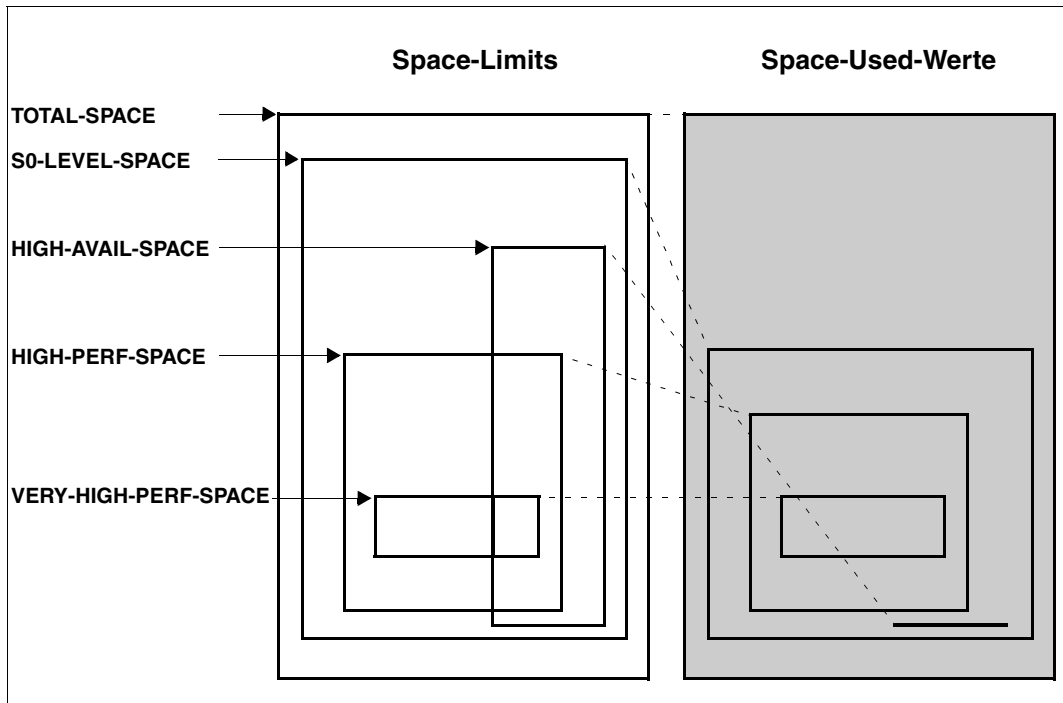
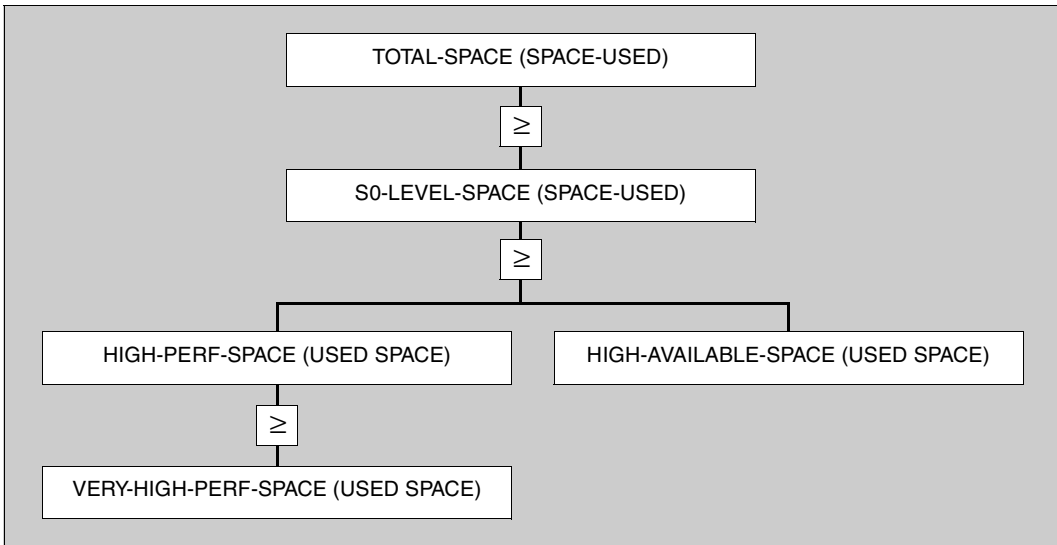
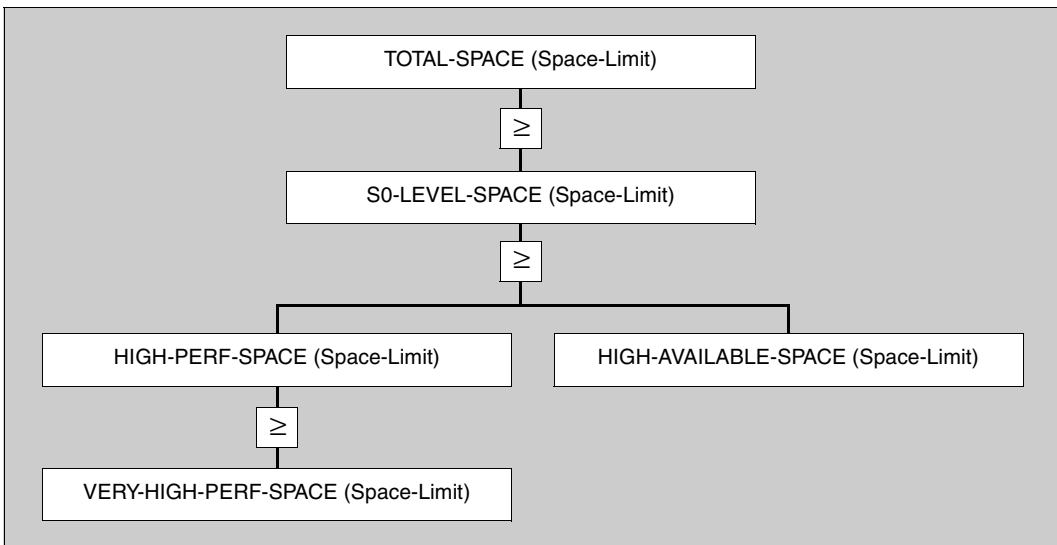


Bild 33: Beispiel für die Nutzung der Kontingente für permanente Dateien

Die Beziehung zwischen den Kontingenten und den Unterkontingenten spiegelt sich in Ungleichungen wider, denen die Werte für den belegten Platz eines Benutzers genügen:



Die Maximalwerte für die Kontingente sind nur dann sinnvoll, wenn sie analoge Ungleichung erfüllen:



Bei der Festlegung der Maximalwerte für die Kontingente muss der Systembetreuer diese Ungleichungen beachten. Ihre Einhaltung wird durch das System überwacht. Die Kommandooberfläche unterstützt den Systembetreuer beim Einstellen der Maximalwerte. Durch Verfahren, mit welchen die Werte der Voreinstellungen bestimmt werden, wird das Einhalten der Regeln vereinfacht.

### 11.1.2 Kontingente für temporäre Dateien und Work-Dateien auf SM-Pubsets

Temporäre Dateien und Arbeitsdateien können nicht auf Hintergrundebenen migriert werden und können nicht das Dateiattribut AVAILABILITY=\*HIGH erhalten. Die Unterscheidung zwischen den Kontingenten für TOTAL-SPACE und S0-LEVEL-SPACE erübrigt sich damit, und für das HIGH-AVAILABLE-SPACE-Kontingent besteht kein Bedarf. Daher ergibt sich für temporäre Dateien und WORK-Dateien im Vergleich zu permanenten Dateien eine vereinfachte Kontingentstruktur, die in veranschaulicht ist.

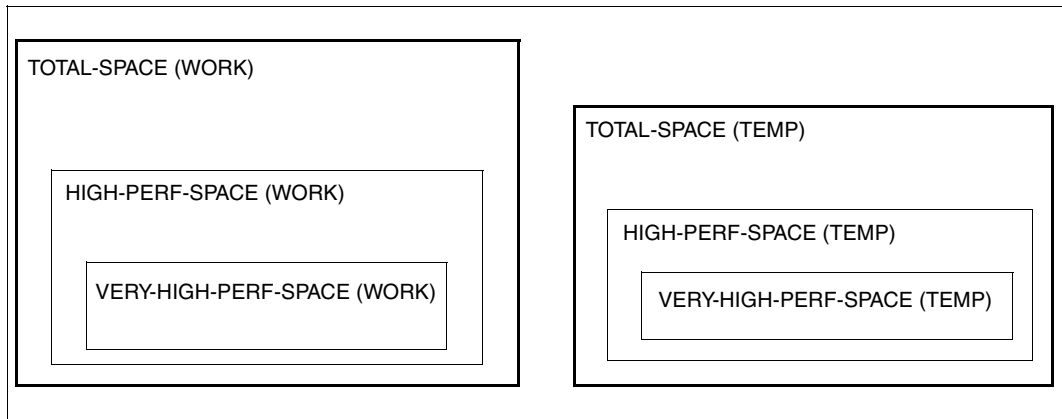


Bild 34: Kontingentstruktur für temporäre Dateien und Work-Dateien

Bei temporären Dateien und Work-Dateien müssen die Maximalwerte für die Kontingente folgenden Ungleichungen genügen:

$$\text{TOTAL-SPACE} \geq \text{HIGH-PERF-SPACE} \geq \text{VERY-HIGH-PERF-SPACE}$$

### 11.1.3 Kommandooberfläche zum Administrieren der Kontingente

Die für SF-Pubsets üblichen Kommandos ADD-USER/MODIFY-USER-ATTRIBUTES zum Festlegen von Space-Limits und Überschreitungsbedingungen sind auch für SM-Pubsets nutzbar, sofern kein Bedarf für eine differenzierte Administration der Benutzerkontingente besteht. Eine weitergehende Differenzierung wird durch das Kommando MODIFY-USER-PUBSET-ATTRIBUTES ermöglicht. Es kann auch für SF-Pubsets benutzt werden, allerdings nur für die dafür definierten Kontingente. Die Kommandos ADD-USER/MODIFY-ATTRIBUTES bzw. MODIFY-USER-PUBSET-ATTRIBUTES bieten damit jeweils sowohl für SF- wie auch für SM-Pubsets alternative Möglichkeiten, die Pubset-Kontingente zu verwalten. Sie können auch wechselweise verwendet wird. Dies ist im Allgemeinen nicht sinnvoll, kann aber beispielsweise während eines Umstiegs von SF-Pubsets auf SM-Pubsets nützlich sein. In diesen Kommandos werden für die einzelnen Kontingente unterschiedliche Begriffe benutzt. Sie lassen sich wie folgt aufeinander abbilden:

<b>S0-LEVEL-SPACE</b> für permanente Dateien	in <b>MODIFY-USER-PUBSET-ATTRIBUTES</b>
bezieht sich auf das gleiche Kontingent wie	
<b>PUBLIC-SPACE</b>	in <b>ADD-USER/MODIFY-USER-ATTRIBUTES</b>

<b>TOTAL-SPACE</b> für temporäre Dateien	in <b>MODIFY-USER-PUBSET-ATTRIBUTES</b>
bezieht sich auf das gleiche Kontingent wie	
<b>TEMP-SPACE</b>	in <b>ADD-USER/MODIFY-USER-ATTRIBUTES</b>

### 11.1.4 Pubset-Space-Kontingentierung auf Benutzergruppenebene

Sowohl auf SM-Pubsets wie auch auf SF-Pubsets können Benutzergruppen definiert werden. Der Verwalter einer auf einem Pubset definierten Gruppe ist berechtigt, die Space-Limits für die Mitglieder seiner Gruppe auf diesem Pubset festzulegen. Die Limitwerte, die einem einzelnen Gruppenmitglied durch den Gruppenverwalter zugewiesen werden können, sind ihrerseits durch Space-Limits auf Gruppenebene nach oben beschränkbar. Die Space-Limits auf Gruppenebene werden durch den Systembetreuer bei dem Definieren der Gruppe durch das Kommando ADD-USER-GROUP festgelegt; sie können später durch das Kommando MODIFY-USER-GROUP modifiziert werden.

Das Kommando SHOW-USER-GROUP zeigt die eingestellten Werte an.

## 11.2 DMS-TUNING-RESOURCES und Benutzerkontingente

Sowohl die Benutzerkontingente als auch die DMS-TUNING-RESOURCES-Berechtigung bieten dem Systembetreuer Möglichkeiten, das Anlegen von Dateien mit erhöhter Performance durch die einzelnen Benutzer zu limitieren, d.h. zwischen den beiden Funktionen bestehen Überlappungen. Allerdings bietet die Funktion der DMS-TUNING-RESOURCES Möglichkeiten (automatische Modifikation der Benutzerangaben bei der Vergabe des statischen Performance-Attributs für Dateien, Einbeziehung in die Ermittlung der dynamischen Performance), welche bei den Benutzerkontingenten keine Entsprechung haben. Andererseits erlauben nur Benutzerkontingente differenzierte Abstufungen für die Space-Limits hochperformanter Dateien. Wegen der unterschiedlichen Funktionalitäten und zur Wahrung der Kompatibilität beim Umstieg von SF-Pubsets auf SM-Pubsets, werden auf SM-Pubsets beide Mechanismen gleichzeitig unterstützt. Damit sie sich sinnvoll ergänzen, sollten jedoch folgende Empfehlungen durch die Systembetreuer berücksichtigt werden:

1. Verwendet ein Systembetreuer bisher für SF-Pubsets DMS-TUNING-RESOURCES als Mittel der Ressourcenüberwachung, und betrachtet er die dadurch gegebenen Möglichkeiten als ausreichend, so kann er die Space-Limits für HIGH-PERF-SPACE und VERY-HIGH-PERF-SPACE vollständig ignorieren, wenn er von SF-Pubsets auf SM-Pubsets übergeht.
2. Sollen die Space-Limits für HIGH-PERF-SPACE und VERY-HIGH-PERF-SPACE genutzt werden, um eine bessere Differenzierung zu erreichen, so ist zu beachten:
  - a) Eine explizite Spezifizierung eines HIGH-PERF-SPACE-Limits für Arbeitsdateien, temporäre oder permanente Dateien ist im Allgemeinen nur dann sinnvoll, wenn gleichzeitig gilt: DMS-TUNING-RESOURCES =\*CONCURRENT-USE oder \*EXCLUSIVE-USE. Andernfalls kommen die Limit-Einstellungen nicht zum Tragen.
  - b) Entsprechend gilt: Eine explizite Spezifizierung eines VERY-HIGH-PERF-SPACE-Limits für Arbeitsdateien, temporäre oder permanente Dateien ist im Allgemeinen nur dann sinnvoll, wenn gleichzeitig DMS-TUNING-RESOURCES=\*EXCLUSIVE-USE gilt.

Die DMS-TUNING-RESOURCES-Berechtigung kann für SF-Pubsets und SM-Pubsets über die Kommandos ADD-USER und MODIFY-USER-ATTRIBUTES oder das Kommando MODIFY-USER-PUBSET-ATTRIBUTES eingestellt werden. Der eingestellte Wert wird durch das Kommando SHOW-USER-ATTRIBUTES angezeigt.

## 11.3 Berechtigung zur physikalischen Allokierung

Die physikalische Allokierung erlaubt eine massive Einflussnahme auf die Allokierungsstrategien innerhalb eines SM-Pubsets. Die Möglichkeit zu ihrer Nutzung muss daher einschränkbar sein. Das TSOS-Privileg berechtigt generell zur physikalischen Allokierung auf allen Benutzerkennungen. Um die physikalische Allokierung auf SM-Pubsets auch für Normalbenutzer zugänglich zu machen, muss der Systembetreuer durch die Kommandos ADD-USER und MODIFY-USER-PUBSET-ATTRIBUTES den entsprechenden Benutzerkennungen das pubset-spezifische Recht einräumen, dass Dateien unter dieser Kennung physikalisch allokiert werden dürfen. Für Privatplatten und Platten von Work-Volume-Sets ist die physikalische Allokierung generell erlaubt. Nach dem Einrichten einer Benutzerkennung auf einem Pubset ist die Berechtigung zur physikalischen Allokierung mit dem Wert „nicht berechtigt“ initialisiert. Die Benutzer können sich über ihre Berechtigung zur physikalischen Allokierung durch das Kommando SHOW-USER-ATTRIBUTES informieren.

Auch physikalisch allokierte Dateien unterliegen der Kontingentüberwachung. Damit die Benutzer diese nicht unterlaufen können, indem sie Dateien physikalisch auf hochwertige Volume-Sets legen, ihnen aber keine entsprechenden Dateiattribute zuweisen (Beispiel: Datei mit AVAILABILITY=\*STD wird physikalisch auf einen Volume-Set mit AVAILABILITY=\*HIGH gelegt und erhält damit implizit hohe Verfügbarkeit), werden die Dateiattribute ggf. angehoben (in unserem Beispiel: AVAILABILITY=\*HIGH). Diese Problematik wird in [Abschnitt „Physikalische Allokierung“ auf Seite 197](#) ausführlicher behandelt.

## 11.4 GUARDS-Profile für Storage-Klassen und HSMS-Management-Klassen

Storage-Klassen und HSMS-Management-Klassen können mit GUARDS-Profilen geschützt werden. Für Storage-Klassen, denen keine Volume-Set-Listen zugeordnet sind, ist dies als Maßnahme für Ressourcenschutz nicht sinnvoll, da ein Benutzer jederzeit Dateien mit entsprechenden Einzelattributen anlegen kann, sofern seine Benutzerkontingente ausreichen.

Das wesentliche Anwendungsgebiet des GUARDS-Schutzes für Storage-Klassen stellen daher Storage-Klassen dar, die mit Volume-Set-Listen verknüpft sind. Sie bieten die Möglichkeit, dass bestimmte Volume-Sets abgesehen von Sondersituationen (z.B. bei Speichereingpässen) nur durch Benutzer mit bestimmten GUARDS-Berechtigungen genutzt werden können. Dazu müssen diese Volume-Sets in Volume-Set-Listen aufgenommen werden, die mit Storage-Klassen verknüpft werden, welche durch GUARDS-Profile geschützt sind. Wir sprechen in diesem Fall vereinfachend von GUARDS-geschützten Volume-Sets.

Allerdings ist zu beachten, dass für die Wirksamkeit des GUARDS-Schutzes für Volume-Sets bestimmte Anforderungen an die Volume-Set-Konfiguration eines SM-Pubsets gestellt werden müssen. Spezifiziert ein Benutzer durch Direktattributierung eine Kombination von Dateiattributen, die auf den allgemein zugänglichen Volume-Sets (d.h. Volume-Sets, die keiner Volume-Set-Liste angehören und nicht mit Allokierungssperren belegt sind) erfüllbar ist, so werden für die Erfüllung der Benutzeranforderung auch die Volume-Sets herangezogen, die Volume-Set-Listen angehören. Damit gelingt auch Benutzern ohne entsprechende GUARDS-Berechtigung das Anlegen von Dateien auf GUARDS-geschützten Volume-Sets. Der GUARDS-Schutz für Volume-Sets ist daher nur dann lückenlos, wenn jede Kombination von ablageortrelevanten Dateiattributen, die auf einem Volume-Set erfüllbar ist, der einer Volume-Set-Liste angehört, auch auf einem allgemein zugänglichen Volume-Set erfüllbar ist. Durch das Kommando SHOW-PUBSET-FILE-SERVICES kann sich der Systembetreuer anzeigen lassen, ob ein SM-Pubset diese Bedingung erfüllt. Ist dies nicht der Fall, muss der Systembetreuer Wohlverhalten vonseiten der Benutzer voraussetzen.





---

# Literatur

Die Handbücher sind online unter <http://manuals.fujitsu-siemens.com> zu finden oder in gedruckter Form gegen gesondertes Entgelt unter <http://FSC-manualshop.com> zu bestellen.

- [1] **BS2000/OSD-BC V6.0**  
**Dienstprogramme**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Das Handbuch beschreibt die Dienstprogramme

DPAGE V15.0A, INIT V15.0A, JMP V2.0A, JMU V14.0A, LMSCONV V3.3B, PAMCONV V12.0A, PASSWORD V15.0A, PVSREN V2.0A, RMS V7.1E, SCDM V6.0A, SMPGEN V15.0A, SPCCTRL V15.0A, TPCOMP2 V15.0A, VOLIN V15.0A.

*Bestellnummer*

U4303-J-Z125-8

- [2] **SECOS V5.0 (BS2000/OSD)**  
**Security Control System**  
Benutzerhandbuch

*Zielgruppe*

- BS2000-Systemverwalter
- BS2000-Anwender, die den erweiterten Zugriffsschutz für Dateien nutzen

*Inhalt*

Leistung und Anwendung der Funktionseinheiten:

- SRPM (Privilegien und Betriebsmittel verwalten)
- SECOS-KRB (Authentisierung mit Kerberos)
- SRPMSSO (Single Sign On)
- GUARDS (Zugriffsbedingungsverwaltung und -auswertung für Objekte)
- GUARDEF (Default Protection, Standardschutz)
- GUARDCOO (Co-owner Protection, Miteigentümerschutz)
- SAT (Protokollierung und Auswertung sicherheitsrelevanter Daten, Ereignisüberwachung mit Alarmfunktion).

*Bestellnummer*

U5605-J-Z125-8

[3] **BS2000/OSD-BC V6.0**

**Systeminstallation**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die BS2000/OSD-Systemverwaltung.

*Inhalt*

Beschrieben wird die Generierung der Hardware-Konfiguration und die Installationsdienste. Letztere beinhalten die Plattenorganisation mit MPVS, die Installation von Datenträgern mit dem Dienstprogramm SIR und das Subsystem IOFCOPY.

*Bestellnummer*

U2505-J-Z125-15

[4] **HSMS / HSMS-SV V7.0A (BS2000/OSD)**

**Hierarchisches Speicher Management System**

**Band 1: Funktionen, Verwaltung und Installation**

Benutzerhandbuch

*Zielgruppe*

- BS2000/OSD-Benutzer
- BS2000/OSD-Systembetreuer
- HSMS-Verwalter

*Inhalt*

- Beschreibung der Funktionen Datensicherung, Langzeitarchivierung, Verdrängung (Migration) und Datentransfer
- Verwaltung, Aufruf, Ablauf und Installation von HSMS
- Meldungen von HSMS

*Bestellnummer*

U6043-J-Z125-11

[5] **BS2000/OSD-BC V6.0**

**Einführung in die Systembetreuung**

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung und das Operating des Betriebssystems BS2000/OSD.

*Inhalt*

Es sind u.a. folgende Themen zur Verwaltung und Überwachung des BS2000/OSD-Grundausbau enthalten: Systemeinführung, Parameterservice, Job- und Tasksteuerung, Speicher-, Geräte-, Benutzer-, Datei-, Pubset- und Systemzeit-Verwaltung, Privilegienvergabe, Accounting und Operatorfunktionen.

*Bestellnummer*

U2417-J-Z125-15

- [6] **DRV V3.0A (BS2000/OSD)**  
**Dual Recording by Volume**  
Benutzerhandbuch

*Zielgruppe*

Systembetreuung, Operating und nichtprivilegierte Anwender

*Inhalt*

Das Manual beschreibt das Daten-Aufzeichnungsverfahren DRV (Dual Recording by Volume), mit dem Daten auf zwei Platten doppelt geführt werden. Der Einsatz von DRV im RZ erhöht die Verfügbarkeit der auf Platten gespeicherten Daten. Im Handbuch sind alle für die Installation, den Einsatz, die Steuerung und Überwachung von DRV notwendigen Vorgehensweisen, Kommandos und Makroerweiterungen beschrieben. Auch der Einsatz von DRV für die Plattenmigration ist ausführlich dargestellt.

*Bestellnummer*

U6515-J-Z125-4

- [7] **BS2000/OSD-BC V5.0**  
**Dateien und Volumes größer 32 GB**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an

- die Programmierer im BS2000/OSD
- die Systembetreuung des BS2000/OSD

*Inhalt*

Das Handbuch erleichtert den Umstieg auf Konfigurationen mit großen Objekten (Dateien und Volumes größer/gleich 32 GB). Es beschreibt:

- die theoretischen Grundlagen für die Einführung großer Objekte
- die Rolle des Systembetreuers bei der Einführung und Pflege großer Objekte (Aufgabenbereiche Systemeinleitung, Parameterservice, Datei- und Pubset-Verwaltung), einschl. der Kommandoschnittstellen
- die Erweiterungen der nicht-privilegierten Kommandoschnittstellen und der Assembler-Makroschnittstellen bzgl. großer Dateien
- den theoretischen Ablauf einer Programmumstellung von Assemblerprogrammen und Programmen in höheren Programmiersprachen

Das Handbuch enthält eine Klassifizierung der Ablauffähigkeit der Programme der SWK BS2000/OSD V5.0 bzgl. großer Dateien.

*Bestellnummer*

U41253-J-Z125-1

- [8] **SHC-OSD V4.0A** (BS2000/OSD)  
**Symmetrix Host Komponente**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung und den Service-Techniker.

*Inhalt*

Das Softwareprodukt SHC-OSD ist die BS2000-Host-Komponente für Symmetrix-Plattensysteme. Es stellt Informationsdienste und Kommandos zur Steuerung der Symmetrix-Funktionen SRDF (Symmetrix Remote Data Facility) und TimeFinder (Symmetrix Multi Mirror Facility) zur Verfügung.

Inhaltlicher Überblick:

- Symmetrix in BS2000/OSD
- Softwareprodukt SHC-OSD
- SRDF: Ausfallszenarien und Maßnahmen zur Weiterführung des Betriebs
- TimeFinder: Arbeiten mit Multi-Mirror-Paaren
- Kommandos in alphabetischer Reihenfolge
- Assembler- und C-Programmschnittstelle

*Bestellnummer*

U41000-J-Z125-4

- [9] **SPACEOPT V3.0** (BS2000/OSD)  
**Optimierung und Reorganisation von Platten**  
Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an die Systembetreuung.

*Inhalt*

Das Subsystem SPACEOPT dient der optimalen Reorganisation der Volumes eines Pubsets. Durch die Reorganisation wird wieder großer zusammenhängender freier Speicherplatz geschaffen und die Anzahl der Datei-Extents verringert. Die Reorganisation kann volume- oder datei-bezogen erfolgen.

SPACEOPT bietet Möglichkeiten zur Bewertung des Belegungs- und Fragmentierungszustands der Volumes eines Pubsets sowie zur Überwachung der SPACEOPT-Aufträge. Für ISAM-Dateien kann außerdem der Anteil freier Seiten, die bei der Bearbeitung entstehen können, ermittelt werden.

Inhaltlicher Überblick:

- SPACEOPT-Aufträge und Auftragsoptionen
- Bewertung des Volume-Zustands
- Installation, Starten, Beenden, Verwalten
- Kommandos in lexikalischer Reihenfolge

*Bestellnummer*

U41073-J-Z125-3

- [10] **BS2000/OSD-BC V6.0**  
**Kommandos Band 1 - 5**  
 Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Die Bände 1 - 5 enthalten die Kommandos ADD-... bis WRITE-... (BS2000/OSD-Grundausbau und ausgewählte Produkte) mit der Funktionalität für alle Privilegien. Die Kommando- und Operandenfunktionen werden ausführlich beschrieben; viele Beispiele unterstützen das Verständnis. Am Anfang jedes Bandes informiert eine Übersicht über alle in den Bänden 1 - 5 beschriebenen Kommandos.

Der Anhang von Band 1 enthält u.a. Informationen zur Kommandoeingabe, zu bedingten Jobvariablenausdrücken, Systemdateien, Auftragschaltern, Geräte- und Volumetypen. Der Anhang der Bände 4 und 5 enthält jeweils eine Übersicht zu den Ausgabespalten der SHOW-Kommandos der Komponente NDM. Der Anhang von Band 5 enthält zusätzlich eine Übersicht aller START-Kommandos.

In jedem Band ist ein umfangreiches Stichwortverzeichnis mit allen Stichwörtern der Bände 1 - 5 enthalten.

*Bestellnummern*

U2338-J-Z125-16 Kommandos Band 1, A – C  
 U41074-J-Z125-3 Kommandos Band 2, D – MOD-JO  
 U21070-J-Z125-6 Kommandos Band 3, MOD-JV – R  
 U41075-J-Z125-3 Kommandos Band 4, S – SH-PR  
 U23164-J-Z125-5 Kommandos Band 5, SH-PU – Z

- [11] **BS2000/OSD-BC V6.0**  
**Kommandos Band 6, Ausgabe in S-Variablen und SDF-P-BASYS**  
 Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich an Programmierer und Anwender, die Prozeduren erstellen.

*Inhalt*

Band 6 enthält die tabellarische Darstellung aller S-Variablen, die von den SHOW-Kommandos bei einer strukturierten Ausgabe mit Werten versorgt werden. Weitere Kapitel:

- Einführung in das Arbeiten mit S-Variablen
- SDF-P-BASYS V2.3A

*Bestellnummer*

U23165-J-Z125-5

- [12] **BS2000/OSD**  
**Softbooks Deutsch**  
CD-ROM

*Zielgruppe*

BS2000/OSD-Anwender

*Inhalt*

Auf der CD-ROM „BS2000/OSD SoftBooks Deutsch“ sind nahezu alle deutschen Handbücher und Readme-Dateien zur BS2000-Systemsoftware der aktuellsten BS2000/OSD-Version und auch von Vorgängerversionen gespeichert, incl. der aufgeführten Handbücher. Diese Softbooks finden Sie auch im Internet auf unserem Manual Server. Sie können in den Handbüchern nachschlagen oder sich vollständige Handbücher herunterladen.

*Bestellnummer der CD-ROM*

U26175-J8-Z125-1

*Internet-Adresse*

<http://manuals.fujitsu-siemens.com>

---

# Stichwörter

## A

ADD-FILE-LINK [223](#)  
ALLOCATION-QUALITY [253](#)  
Allocation-Unit [23](#), [163](#), [168](#)  
Allokierungssperre [203](#)  
Ändern  
  der Cache-Konfiguration [171](#)  
  der Eigenschaftsprofile [168](#)  
  der Nutzungsart [169](#)  
  der Nutzungsrestriktionen [169](#)  
  der Pubset-Space-Voreinstellungen [174](#)  
  der Sättigungsschwellwerte [173](#)  
  von Storage-Klassen [175](#)  
  von Volume-Set-Listen [176](#)  
Arbeitsdateien [217](#), [240](#)  
ARCHIVE [116](#)  
Ausfalleinheit [11](#), [14](#), [29](#)  
Ausführungsmodus (SMPGEN) [78](#)  
automatische Verlagerung von Dateien [205](#)  
Automatisierbarkeit [179](#)  
AVAILABILITY [216](#), [234](#)

## B

Backup-Archiv [15](#), [45](#), [71](#), [86](#), [103](#), [241](#)  
Backup-Konfiguration eines SM-Pubsets [45](#)  
Benutzer [5](#), [179](#)  
Benutzerberechtigungen [82](#)  
Benutzerdatei [85](#)  
Benutzereinträge [62](#), [70](#), [177](#)  
Benutzergruppen [177](#)  
Benutzerkatalog [60](#), [81](#)  
benutzerkontrollierte Domänen [201](#)  
Benutzerverwaltung [11](#)  
BLKCTRL [213](#), [219](#), [223](#)  
BLKSIZE [213](#), [219](#), [223](#)

## C

Cache [25](#)  
Cache-Konfiguration eines Volume-Sets [79](#), [171](#)  
Cache-Zuordnung für Volume-Sets [59](#), [62](#)  
Control-Volume-Set [13](#), [19](#), [155](#)  
CREATE-FILE [211](#), [233](#), [242](#)  
CREATE-MANAGEMENT-CLASS [231](#)  
CREATE-SYSTEM- MANAGED- PUBSET [78](#)  
CREATE-SYSTEM- MANAGED-PUBSET [77](#)

## D

Dateiattribute [83](#), [203](#), [210](#)  
  Voreinstellungen [210](#)  
Dateien für die Systemeinführung [76](#)  
Dateiformat, Voreinstellung [37](#)  
Dateigröße [228](#), [239](#)  
Dateikatalog [61](#)  
  des SM-Pubsets [40](#)  
Dateinamensraum [19](#)  
Datei-Services [5](#), [16](#)  
Dateisicherung [241](#)  
Dateiverwaltung [5](#), [16](#)  
Datenadministration [5](#), [179](#)  
Default-File-Format [220](#)  
Default-Storage-Klasse [175](#), [211](#), [225](#)  
Defect-Garbage-Datei [42](#)  
defective [148](#)  
defined only [147](#), [151](#)  
Differenzierte Ressourcennutzung [192](#)  
DIRCONV [86](#)  
Direktattributierung [16](#), [181](#), [182](#)  
DMS-TUNING-RESOURCES [210](#), [215](#)  
DRV [56](#), [163](#)  
DUAL-COPY [164](#)

## E

- EAM-Dateien 85
- Eigenschaftsprofile
  - der Volume-Sets 58, 192
  - für die Volume-Set-Selektion 23, 80
- Eignung als Home-Pubset 85
- Entfernen
  - von defekten Volume-Sets 159
  - von Volume-Sets 146

## F

- FCBTYPE 223
- FDDRL 78
- File-Format-Voreinstellung 60, 220
- FILE-Schnittstelle 213
- FORCE-IMPORT-Einstellung 52
- Format 210
  - eines Volume-Sets 23
- FREE-FOR-DELETION-Attribut 240
- FREE-FOR-DELETION-Frist 244
- freier Volume-Set 20, 147, 151

## G

- GS-Partition 31
- GUARDS-Katalog 42, 70
- GUARDS-Profile 81
- GUARDS-Schutzprofile 177

## H

- Hintergrundebenen 14, 45, 71
- Hinzunahme
  - eines Volume-Sets 146, 150
  - von Volumes 164
- Home-Pubset 55
- HSMS 14, 86, 229
- HSMS-Administrator 9
- HSMS-Dateien 47
- HSMS-Konfiguration 177
  - eines SM-Pubsets 43
- HSMS-Management-Klassen 177, 229, 231, 242
- HSMS-Management-Klassen-Katalog 47
- HSMS-Management-Services 7, 16

## I

- im Globalspeicher emulierte Volumes 192
- IMPORT-PUBSET 147, 148
- in hold 147
  - Restriktion 35
- In-place-Konvertierung 69, 87, 88

## J

- Jobvariable 235

## K

- K 224
- Katalogdateien 58, 81
- Kennung
  - des Pubsets 79
  - des SM-Pubsets 75
- Konfigurationsdatei 81
- Konfigurationszustand der Volume-Sets 22, 59, 80
- Kontingenzstruktur 261

## L

- Langzeitarchiv 46, 116, 244
- Lebensdauer von Dateien 240
- Lebenszyklus einer Datei 179
- Leeren
  - eines Volumes 167
  - eines Volume-Sets 159
- logisches Entfernen eines Volume-Sets 158
- Löschen eines Volume-Set-Eintrags 158

## M

- Magnetband 14
- Magnetbandkassette 14
- MAREN-Katalog 86
- maximale I/O-Länge 165
  - des SM-Pubsets 37
  - eines Volume-Sets 28
- Merkmale eines SM-Pubsets 19
- Metadatendatei 13
- MIGRATE-FILES 236
- Migrationsarchiv 15, 45, 86, 87
- Migrations-Directory 45
- Migrations Sperre 17, 215, 233, 236



MODIFY-FILE-ATTRIBUTES 184, 212, 233, 242  
MODIFY-HSMS-PARAMETERS 234  
MODIFY-MANAGEMENT-CLASS 231  
MODIFY-PUBSET-DEFINITION-FILE 147, 150,  
158  
MODIFY-PUBSET-PROCESSING 147, 155,  
165, 166  
MODIFY-PUBSET-RESTRICTIONS 147  
MODIFY-SPACE-SATURATION-LEVELS 173,  
248  
MRSCAT-Eintrag 81, 150

## N

Namenskollisionen 75  
Neueinrichten eines SM-Pubsets 55  
Neukonnektieren von Cache-Bereichen 53  
NK2 224  
normal use 148  
Nutzungsarten für Volume-Sets 33, 58, 80  
Nutzungsrestriktionen 169  
für Volumes 80  
für Volume-Sets 34, 59, 61, 80

## O

OPEN 223  
Operator 9  
optische Platte 14

## P

Paging-Datei 76  
Paging-Pubset 55  
Performance-Attribute 210, 214  
Performance-Profil 168  
eines Volume-Sets 23, 25  
Performance-Spektrum 25  
Pfadnamen-Überlängen 75  
Physikalische Allokierung 49, 181, 197  
Physikalische Struktur des SM-Pubsets 79  
Physikalische Volume-Set-Merkmale 28  
Privatplattenablösung 207  
Prüfmodus (SMPGEN) 77  
Pubres 52  
Pubset 9  
Pubset-Benutzerkatalog 39

Pubset-Generierung 55  
Pubset-Kennung 19, 57  
Pubset-Konfigurationsdatei 19, 39  
Pubset-Pflege 245  
Pubset-Rekonfiguration 146  
Pubset-Reorganisation 253  
Pubset-Space-Voreinstellungen 61, 80  
Pubset-Überwachung 245  
PVSREN 143

## Q

Qualität der Storage-Services 251

## R

RAID 164  
Readme-Datei 4  
Recall 238  
RECALL-MIGRATED-FILES 238  
Rekonfigurationseinheit 13  
Rekonnektieren von Cache-Bereichen 52  
Ressourcentyp 13  
RETENTION-PERIOD 240  
Rückkonvertierung 117, 118  
Rücksetzen 72, 118, 119  
Rückstieg 117

## S

S0-Ebene 14, 75  
S0-MIGRATION 227  
S1-Ebene 14, 45, 71, 230  
S1-Pubset 87  
S1-Volume-Set 87  
S2-Ebene 14, 45, 71, 230  
Sättigungsschwellwerte 247  
der Volume-Sets 60, 173  
für Volume-Sets 36, 61, 80  
Sättigungszustand eines Volume-Sets 36  
SAVE-Attribute 242  
Save-File 86  
Save-Version 86  
SECURE-Lock 161  
SECURE-RESOURCE-ALLOCATION 238  
Separierung der Pubset-Benutzer 195

SF-Pubset 10  
  in bestehenden SM-Pubset aufnehmen 162  
SHOW-MASTER-CATALOG-ENTRY 141  
SHOW-PUBSET-CONFIGURATION 141, 161  
SHOW-PUBSET-FILE-SERVICES 141, 182, 186  
SHOW-PUBSET-PARAMETERS 248  
SHOW-PUBSET-SPACE-ALLOCATION 141,  
  248  
SHOW-PUBSET-USAGE 141  
SHOW-STORAGE-CLASS 186  
SHOW-USER-ATTRIBUTES 141  
Single-Feature-Pubset 10  
SIR 56, 146, 153  
SIZE-TOLERANCE 53  
SMPGEN 75  
  Ausführungsmodus 78  
  Prüfmodus 77  
SM-Pubset  
  aus SF-Pubsets erzeugen 65  
  neu einrichten 55  
SNAPSHOT-Dateien 55  
Space-Limit 82  
SPCCNTRL 141  
SPEEDCAT 49  
Spoolaufträge 85  
STAMCE-Programmschnittstelle 141, 225  
statische Pubset-Pflege 146  
Storage-Klasse 16, 181  
Storage-Klassen-Katalog 42  
Storage-Services 7, 16, 37, 175  
SYSEAM-Datei 42  
System 5, 183  
System Managed Storage 5  
System-Backup-Archiv 15, 46  
Systembetreuer 5, 179  
Systemdateien 76, 77  
  auf dem SM-Pubset 39  
System-Managed-Pubset 10

## T

temporäre Dateien 240  
TSOS-Privileg 160

## U

Umbenennen  
  eines Pubsets 143  
  eines Volume-Sets 145  
Umschaltbarkeit eines Pubsets 48  
Umstiegsszenario 122

## V

Verarbeitungsebene 12, 19, 69, 180  
Verfügbarkeit 210, 216  
  eines Volume-Sets 24  
Verfügbarkeitsprofil 168  
Verfügbarkeits-Profil eines Volume-Sets 23  
Verfügbarkeitsprofil eines Volume-Sets 23  
VOLATILITY 31  
Volres 20  
Volume 20  
Volume Serial Number 20  
Volume-Katalog 20  
Volume-Set 12, 19  
Volume-Set-Access-Sperre 160  
Volume-Set-Format 163, 168  
Volume-Set-Konfigurationszustände 146  
Volume-Set-Liste 16, 176, 184, 187  
Volume-Set-Listen-Katalog 42, 187  
Volume-Set-Qualität 254  
Volume-Set-Selektion 183  
Volume-Set-Sperren 160  
Voreinstellungen  
  Dateiattribute 210  
  Dateiformat 37  
  File-Format 60  
  Pubset-Space 37, 60, 61, 80  
VSN 20

## W

Watch-Dog-Datei 42  
WORK-FILE 217, 234

## Z

Zwangsentfernen eines Volume-Sets 156

---

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Zielsetzung des Handbuchs	1
1.2	Zielgruppe des Handbuchs	1
1.3	Aufbau des Handbuchs	2
1.4	Readme-Datei	4
<b>2</b>	<b>System Managed Storage – Idee</b>	<b>5</b>
<b>3</b>	<b>System Managed Storage im BS2000 – Übersicht</b>	<b>9</b>
3.1	Datenadministration und Pubset-Management	9
3.2	Pubset-Konzept und SMS	10
3.3	Aufbau eines SM-Pubsets	12
3.4	Dateiverwaltung in einem SM-Pubset	16
3.4.1	Storage-Services	16
3.4.2	HSMS-Management-Services	17
<b>4</b>	<b>Merkmale eines SM-Pubsets</b>	<b>19</b>
4.1	Pubset-Kennung und Dateinamensraum	19
4.2	Aufbau der Verarbeitungsebene eines SM-Pubsets	19
4.3	Merkmale der einzelnen Volume-Sets eines SM-Pubsets	22
4.3.1	Konfigurationszustand eines Volume-Sets	22
4.3.2	Eigenschaftsprofile für die Volume-Set-Selektion	23
4.3.3	Physikalische Volume-Set-Merkmale	28
4.3.4	Nutzungsarten für Volume-Sets	33
4.3.5	Nutzungsrestriktionen für Volume-Sets	34
4.3.6	Sättigungsschwellwerte für Volume-Sets	36
4.4	Volume-Set-übergreifende Merkmale eines SM-Pubsets	37
4.4.1	Umfang der gebotenen Storage-Services	37
4.4.2	Maximale I/O-Länge des SM-Pubsets	37
4.4.3	Voreinstellungen für Pubset-Space	37
4.4.4	Large-Objects-Eigenschaft	38
4.5	Systemdateien auf dem SM-Pubset	39
4.6	HSMS-Konfiguration eines SM-Pubsets	43
4.6.1	Konfiguration der Hintergrundebenen	45
4.6.2	Backup-Konfiguration eines SM-Pubsets	45
4.6.3	Langzeitarchive	46

4.6.4	HSMS-Management-Klassen-Katalog .....	47
4.6.5	HSMS-Dateien .....	47
4.7	Nutzung von ARCHIVE .....	48
4.8	SF-Pubset-Merkmale ohne Entsprechung bei SM-Pubsets .....	49
4.9	Nutzung eines SM-Pubsets an einem Rechner .....	50
4.9.1	MRSCAT-Einträge für SM-Pubsets .....	50
4.9.2	Festlegungen für die Pubset-Inbetriebnahme .....	52
4.9.3	Sonstige Festlegungen für die Pubset-Nutzung .....	53
<b>5</b>	<b>Neueinrichten eines SM-Pubsets .....</b>	<b>55</b>
5.1	Überblick über die SIR-Funktionalität für SM-Pubsets .....	55
5.2	Generierung eines neuen SM-Pubsets durch SIR .....	56
5.2.1	Ausgangsmaterial .....	56
5.2.2	Vorbereitung der Pubset-Generierung .....	56
5.2.3	Festlegung der Pubset-Merkmale .....	57
5.3	Weitere Installationsmaßnahmen .....	61
5.3.1	Anpassung der von SIR getroffenen Voreinstellungen .....	61
5.4	Klonen eines SM-Pubsets mit PVSREN .....	63
<b>6</b>	<b>Erzeugen eines SM-Pubsets aus bestehenden SF-Pubsets .....</b>	<b>65</b>
6.1	Grundsätzliche Aspekte .....	65
6.2	Übersicht über die Vorgehensweisen .....	69
6.2.1	Anpassung der Verarbeitungsebene .....	69
6.2.2	Anpassung der Hintergrundebenen für Verdrängung .....	71
6.2.3	Anpassung der Backup-Archive .....	71
6.2.4	Maßnahmen, die ein Rücksetzen ermöglichen .....	72
6.2.5	Vorbereitungen .....	73
6.3	Konvertierung der S0-Ebene durch SMPGEN .....	75
6.3.1	Voraussetzungen .....	75
6.3.2	Vorbereitung und Durchführung .....	77
6.3.3	Merkmale eines durch SMPGEN erzeugten SM-Pubsets .....	79
6.3.4	Nachbearbeitung der durch SMPGEN getroffenen Voreinstellungen .....	84
6.3.5	Funktionale Reduzierungen als Folge der Pubset-Konvertierung .....	85
6.4	Anpassung der Migrations- und Backup-Archive .....	86
6.4.1	Voraussetzungen für die in-place-Konvertierung .....	87
6.4.2	Anpassung der Migrationsarchive .....	87
6.4.2.1	In-place-Konvertierung von Migrationsarchiven .....	88
6.4.2.2	Anpassung der Migrationsarchive mithilfe von Dateisicherungen .....	98
6.4.3	Anpassung der Backup-Archive .....	103
6.4.3.1	Anpassung bestehender System-Backup-Archive .....	105
6.4.3.2	Erzeugen eines neuen System-Backup-Archivs für den SM-Pubset .....	112
6.4.3.3	Anpassung privater Backup-Archive .....	116
6.4.4	Anpassung der Langzeitarchive .....	116
6.4.5	Nutzung von ARCHIVE-Funktionen .....	116

6.5	Rückstieg von einem SM-Pubset auf SF-Pubsets	117
6.5.1	Grundsätzliche Aspekte	117
6.5.2	Anpassung der Verarbeitungsebene	117
6.5.3	Anpassung der Backup-Archive	118
6.5.4	Anpassung des Migrationsarchivs	119
6.6	Exemplarische Umstiegsszenarien und Vorgehensweisen	122
6.6.1	Umstiegsszenario 1	122
6.6.2	Umstiegsszenario 2	126
6.6.3	Umstiegsszenario 3	130
6.6.4	Umstiegsszenario 4	134
6.6.5	Umstiegsszenario 5	138
<b>7</b>	<b>Auskunftsfunktionen für SM-Pubsets</b>	<b>141</b>
<b>8</b>	<b>Änderung der Merkmale von SM-Pubsets</b>	<b>143</b>
8.1	Umbenennen von Pubsets und Volume-Sets	143
8.1.1	Pubset-Umbenennung	144
8.1.2	Volume-Set-Umbenennung	145
8.2	Hinzunahme und Entfernen von Volume-Sets	146
8.2.1	Pubset-Rekonfiguration und Volume-Set-Konfigurationszustände	146
8.2.2	Hinzunahme eines Volume-Sets im laufenden Pubset-Betrieb	150
8.2.3	Hinzunahme von Volume-Sets durch SIR	153
8.2.4	Entfernen von Volume-Sets im laufenden Pubset-Betrieb	155
8.2.5	Pubset-Rekonfiguration bei Problemen bei der Pubset-Inbetriebnahme	159
8.2.6	Leeren eines Volume-Sets	159
8.3	Erweitern eines SM-Pubsets mit SF-Pubsets	162
8.4	Hinzunahme und Entfernen von Volumes	163
8.4.1	Hinzunahme von Volumes durch SIR	163
8.4.2	Hinzunahme von Volumes bei der Pubset-Inbetriebnahme	164
8.4.3	Hinzunahme von Volumes durch dynamische Pubset-Rekonfiguration	165
8.4.4	Entfernen eines leeren Volumes aus einem SF-Pubset oder Volume-Set	166
8.4.5	Leeren eines Volumes	167
8.5	Ändern der Eigenschaftsprofile für die Volume-Set-Selektion	168
8.6	Ändern der Nutzungsart und der Nutzungsrestriktionen	169
8.7	Ändern der Cache-Konfiguration von Volume-Sets	171
8.8	Ändern der Sättigungsschwellwerte von Volume-Sets	173
8.9	Ändern der Pubset-Space-Voreinstellungen	174
8.10	Ändern von Storage-Klassen und Volume-Set-Listen	175
8.11	Ändern sonstiger Pubset-Merkmale	177

<b>9</b>	<b>Management des Lebenszyklus einer Datei</b>	<b>179</b>
9.1	Nutzung der Verarbeitungsebene	180
9.1.1	Bestimmung des Ablageorts einer Datei bei SM-Pubsets – Übersicht	181
9.1.2	Direktattributierung	182
9.1.3	Storage-Klassen ohne Volume-Set-Listen	184
9.1.4	Storage-Klassen mit Volume-Set-Listen	187
9.1.5	Physikalische Allokierung	197
9.1.6	Privatplattenablösung durch SM-Pubsets	207
9.1.7	Ablageortrelevante Dateiattribute und zugewiesene Voreinstellungen	210
9.2	Nutzung der Hintergrundebenen für Dateiverdrängung	229
9.2.1	Schnittstelle Benutzer/Systembetreuer	229
9.2.2	Durchführung der Dateimigration auf die Hintergrundebenen	235
9.2.3	Recall von Dateien in die Verarbeitungsebene	238
9.2.4	Einfluss von Migrate und Recall auf Dateieigenschaften	239
9.3	Lebensdauer von Dateien	240
9.4	Sichern und Wiedereinspielen von Dateien	241
9.4.1	Anwendungsszenarien für Dateisicherung	241
9.4.2	Benutzer-Systembetreuer-Schnittstelle	242
9.4.3	Durchführung der Dateisicherung durch den Systembetreuer	243
9.4.4	Durchführung des Wiedereinspielens	243
9.5	Langzeitarchivierung	244
<b>10</b>	<b>Pubset-Überwachung und Pubset-Pflege</b>	<b>245</b>
10.1	Übersicht	245
10.2	Speicherengpässe in der S0-Ebene	246
10.3	Qualität der Storage-Services	251
10.3.1	Optimaler Ablageort von Dateien	252
10.3.2	Mangelhafte Volume-Set-Qualität	254
10.4	Vorkehrungen für die Wiederherstellbarkeit eines SM-Pubsets	255
<b>11</b>	<b>Ressourcenkontrolle für SM-Pubsets</b>	<b>259</b>
11.1	Benutzerkontingente für belegten Platz	259
11.1.1	Kontingentsstruktur für permanente Dateien auf SM-Pubsets	261
11.1.2	Kontingente für temporäre Dateien und Work-Dateien auf SM-Pubsets	267
11.1.3	Kommandooberfläche zum Administrieren der Kontingente	268
11.1.4	Pubset-Space-Kontingentierung auf Benutzergruppenebene	268
11.2	DMS-TUNING-RESOURCES und Benutzerkontingente	269
11.3	Berechtigung zur physikalischen Allokierung	270
11.4	GUARDS-Profile für Storage-Klassen und HSMS-Management-Klassen	271
	<b>Literatur</b>	<b>273</b>
	<b>Stichwörter</b>	<b>279</b>

---

# BS2000/OSD-BC V6.0

## System Managed Storage

### Benutzerhandbuch

#### *Zielgruppe*

Systembetreuung

#### *Inhalt*

Das Handbuch gibt eine Übersicht über die Funktionen, die in BS2000/OSD und den Produkten des Grundausbaus zur Unterstützung von System Managed Storage vorhanden sind. Dabei ist der SM-Pubset von zentraler Bedeutung, dessen Aufbau, Eigenschaften und Nutzungsmöglichkeiten dargestellt werden. Anhand von verschiedenen Szenarien wird aufgezeigt, wie die Funktionalität von SMS genutzt werden kann und wie der Übergang von SF-Pubsets auf die SM-Pubsets erfolgt.

**Ausgabe: Dezember 2004**

**Datei: sms.pdf**

Copyright © Fujitsu Siemens Computers GmbH, 2004.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Dieses Handbuch wurde erstellt von  
cognitas. Gesellschaft für Technik-Dokumentation mbH  
[www.cognitas.de](http://www.cognitas.de)

Fujitsu Siemens Computers GmbH  
Handbuchredaktion  
81730 München

# Kritik Anregungen Korrekturen

**Fax: 0 700 / 372 00000**

e-mail: [manuals@fujitsu-siemens.com](mailto:manuals@fujitsu-siemens.com)  
<http://manuals.fujitsu-siemens.com>

---

Absender

---

Kommentar zu BS2000/OSD-BC V6.0  
System Managed Storage





## Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

The Internet pages of Fujitsu Technology Solutions are available at [http://ts.fujitsu.com/...](http://ts.fujitsu.com/) and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

## Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter [http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009