

# Document Management Service (BC-SRV-KPR)



HELP.BCSRVKPRDMS

Release 4.6C



## Copyright

© Copyright 2001 SAP AG. Alle Rechte vorbehalten.

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

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

Microsoft<sup>®</sup>, WINDOWS<sup>®</sup>, NT<sup>®</sup>, EXCEL<sup>®</sup>, Word<sup>®</sup>, PowerPoint<sup>®</sup> und SQL Server<sup>®</sup> sind eingetragene Marken der Microsoft Corporation.

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

ORACLE<sup>®</sup> ist eine eingetragene Marke der ORACLE Corporation.

INFORMIX<sup>®</sup>-OnLine for SAP und Informix<sup>®</sup> Dynamic Server<sup>™</sup> sind eingetragene Marken der Informix Software Incorporated.

UNIX<sup>®</sup>, X/Open<sup>®</sup>, OSF/1<sup>®</sup> und Motif<sup>®</sup> sind eingetragene Marken der Open Group.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C<sup>®</sup>, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA<sup>®</sup> ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT<sup>®</sup> ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

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

## Symbole

Symbol	Bedeutung
	Achtung
	Beispiel
	Empfehlung
	Hinweis
	Syntax
	Tip

## Inhalt

<b>Document Management Service (BC-SRV-KPR)</b> .....	<b>5</b>
Konzepte .....	6
Attribute.....	8
Beziehungen .....	10
Versionierung .....	12
Kontextauflösung .....	15
Content-Modelle .....	17
DMS-Programmierschnittstelle.....	19

## Document Management Service (BC-SRV-KPR)

### Einsatzmöglichkeiten

Der Document Management Service (DMS) ist eine Unterkomponente der informationstechnischen Infrastruktur, die von dem SAP Knowledge Provider im Rahmen der R/3-Basis zur Verfügung gestellt wird. Zentraler Bestandteil des DMS ist die Verarbeitung dokumentartiger Objekte auf der Basis von anwendungsspezifischen Content Modellen. Nur über den DMS können Verwaltungsdaten zu Dokumenten verändert und in einem Ablagemedium (z.B. der R/3 Datenbank) gespeichert werden.

### Einführungshinweise

Für die Nutzung des DMS sind einige Customizing-Schritte notwendig: Wählen Sie dazu im Einführungsleitfaden (IMG) *Basis* → *Basis Services* → *SAP Knowledge Provider* → *Document Management Service*. Erläuterungen zu den einzelnen Aktivitäten sind in den IMG-Bausteinen hinterlegt.

### Funktionsumfang

Der DMS stellt die folgende Funktionalität zur Verfügung:

- Verwaltung von [Attributen \[Seite 8\]](#)
- Verwaltung von [Beziehungen \[Seite 10\]](#)
- [Versionierung \[Seite 12\]](#)
- [Kontextauflösung \[Seite 15\]](#)
- [Content Modelle \[Seite 17\]](#)

Zentrale Aufgabe des DMS ist die **Abschirmung** der SAP-Anwendungen gegenüber Einzelheiten der Ablage von Verwaltungsdaten in Tabellen. Administration und Bearbeitung aller zugrundeliegender Verwaltungsdaten und der dazugehörigen Tabellen werden den Anwendungen vom DMS abgenommen.

Außerdem stellt der DMS einen **Sperrmechanismus** bereit. Dadurch wird gewährleistet, daß niemals mehrere Autoren am gleichen Dokument zur gleichen Zeit arbeiten können.

Des weiteren stellt der DMS eine [Programmierschnittstelle \[Seite 19\]](#) zur Verfügung.

## Konzepte

# Konzepte

Im Kontext des SAP Knowledge Providers beschreibt der Begriff "Dokument" umgangssprachlich alle Informationsentitäten, die über dokumentartige Objekte verfügbar sind. Insbesondere sind dies:

- eigentlicher Inhalt, d.h. Content
- Verwaltungsdaten
- Indexdaten
- Metadaten

Jedes anwendungsspezifische [Content-Modell \[Seite 17\]](#) repräsentiert eine Dokumentdefinition.



Logisches Dokument "Dokumentationskapitel zu den Datenstrukturen des SAP Knowledge Providers".

Zentrale Bestandteile des SAP Knowledge Providers sind **logische Dokumente**, **physische Dokumente** und **Komponenten**. Ein logisches Dokument stellt eine logische Klammerung eines Dokuments in all seinen Ausprägungen dar. Jedes logische Dokument kann also mehrere physische Dokumente beinhalten. Physische Dokumente bestehen somit aus Meta-Beschreibungen der real existierenden Dokumente. Diese Meta-Beschreibungen werden durch Attribute realisiert.



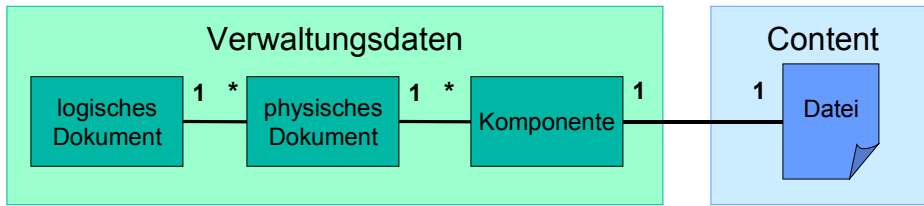
Physische Dokumente für das logische Dokument "Dokumentationskapitel zu den Datenstrukturen des SAP Knowledge Providers":  
"Dokumentationskapitel zu den Datenstrukturen des SAP Knowledge Providers im HTML-Format, in Englisch, dritte Version, vom 1.10.1999" und  
"Dokumentationskapitel zu den Datenstrukturen des SAP Knowledge Providers im WinWord-Format, in Deutsch, zweite Version, vom 2.10.1999".

In physischen Dokumenten sind Referenzen auf eine oder mehrere Komponenten vorhanden. Komponenten beinhalten Verwaltungsdaten zum eigentlichen Content, z.B. die Größe der Datei.



Komponenten für das physische Dokument "Dokumentationskapitel zu den Datenstrukturen des SAP Knowledge Providers im HTML-Format, in Englisch, dritte Version, vom 1.10.1999", wobei dieses physische Dokument aus einer HTML-Datei und einer PowerPoint-Datei auf dem Content Server besteht:  
Administratives Gegenstück zu der HTML-Datei und administratives Gegenstück zu der PowerPoint-Datei auf dem Content Server.

Auf der physischen Ebene ist jeder Komponente genau eine Datei zugeordnet, die den eigentlichen Content enthält.



Die erforderlichen Customizing-Schritte finden Sie im Einführungsleitfaden (IMG) unter *Basis* → *Basis Services* → *SAP Knowledge Provider* → *Document Management Service*.

## Attribute

## Attribute

Attribute kennzeichnen Meta-Beschreibungen von physischen Dokumenten. Beispiele solcher Attribute sind Format oder Sprache.

Attribute können entweder von der jeweiligen SAP-Anwendung vergeben werden, in der die Dokumente erstellt wurden, oder aber vom SAP Knowledge Provider selbst. Die folgenden vordefinierten Attribute werden vom SAP Knowledge Provider zur Verfügung gestellt:

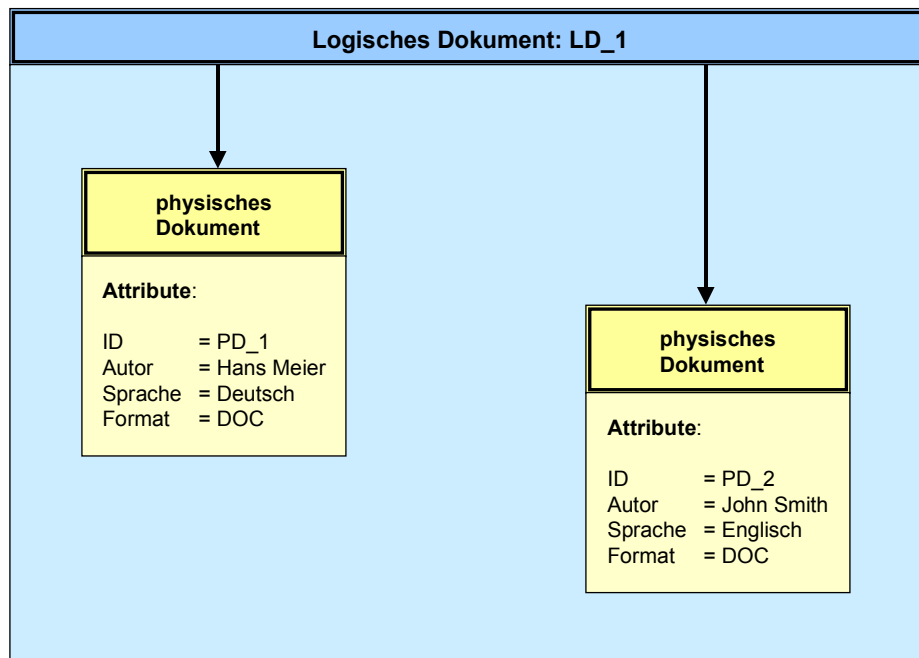
Attributname	Beschreibung	relevant für:		
		logische Dokumente	physische Dokumente	Beziehungen
CREATED_AT	Zeitpunkt des Anlegens	x	x	x
CREATED_BY	Angelegt von	x	x	x
LAST_CHANGED_AT	Zeitpunkt der letzten Änderung	x	x	x
LAST_CHANGED_BY	Letzter Änderer	x	x	x
EXPIRYTIME	Auslaufzeitpunkt	x	x	x
ORIGINAL_LANGUAGE	Originalsprache	x		
CHECKED_OUT	Kennzeichen für ausgecheckte Dokumente		x	
CHECKOUT_USER	Name des Benutzers, der das Dokument ausgecheckt hat		x	
DOCUMENT_FORMAT	Dokumentformat		x	
LANGUAGE	Sprache		x	
RESERVED	Kennzeichen für das Zurückstellen eines Dokuments		x	
STATE	Status		x	
STORAGE_CATEGORY	ID des Ablagetyps		x	

Attribute

DESCRIPTION	Kurzbeschreibung	x	x	
-------------	------------------	---	---	--



Das logische Dokument mit der ID LD\_1 besteht aus 2 physischen Dokumenten mit den IDs PD\_1 und PD\_2. Das erste physische Dokument ist durch die folgenden Attribute gekennzeichnet: ID = PD\_1, Autor = Hans Meier, Sprache = Deutsch und Format = DOC. Das zweite physische Dokument ist durch die folgenden Attribute gekennzeichnet: ID = PD\_2, Autor = Jane Smith, Sprache = Englisch und Format = DOC.



## Beziehungen

# Beziehungen

Der SAP Knowledge Provider benutzt Beziehungen, um Abhängigkeiten zwischen Dokumenten darzustellen. Eine interne Beziehung verknüpft genau zwei KPro-Verwaltungsdokumente nach bestimmten Kriterien. Abhängig von diesen Kriterien werden verschiedene Klassen von Beziehungen unterschieden. Jede Beziehung ist ein eigenständiges Objekt. Neben internen Beziehungen gibt es auch Beziehungen, die KPro-Verwaltungsobjekte mit Dokumenten verknüpfen, die nicht im KPro verwaltet werden. Diese werden als externe Beziehungen bezeichnet.



Wird ein Dokument, das im KPro als physisches Dokument verwaltet wird, von Deutsch nach Englisch übersetzt, so besteht zwischen beiden Dokumenten eine Sprachbeziehung, die es erlaubt, die Entstehung des englischen Dokuments nachzuvollziehen. Das Kriterium zur Bildung der Beziehung ist in diesem Fall die „Sprache“. Das Anlegen und Verwalten einer solchen Sprachbeziehung wird als Sprachversionierung bezeichnet.

Der SAP Knowledge Provider stellt die folgenden vordefinierten Arten von Beziehungen oder Beziehungsklassen zur Verfügung:

- **Collection-Beziehung**

Die Collection-Beziehung (LOGOBJECT) modelliert das Verhalten logischer Dokumente, physische Dokumente zu beinhalten. Ein bestimmtes physisches Dokument kann zu einem einzigen logischen Dokument gehören. Diese Beziehung erlaubt es dem logischen Dokument, zu einem Behälter für physische Dokumente zu fungieren.
- **Template-Beziehung**

Die Template-Beziehung (EXPORT\_MODEL) besteht lediglich zwischen physischen Dokumenten. Dadurch kann ein bestimmtes Dokument als Vorlage für das Anlegen eines weiteren Dokuments oder mehrere weiterer Dokumente dienen. Diese Beziehung wird beim Anlegen neuer Dokumente eingesetzt.
- **Versionsbeziehung**

Die Versionsbeziehung (VERSIONREF, FORMATREF, TRANSLREF) zwischen Dokument A und Dokument B sagt aus, daß beispielsweise Dokument A als Version von Dokument B angelegt wurde. Dabei kann Dokument A folgendes sein:

  - eine Übersetzung
  - eine Aktualisierung des Inhalts
  - eine Sicherung in einem von B unterschiedlichen Format
- **Beschreibungsbeziehung**

Die Beschreibungsbeziehung (DESCOBJECT) besagt, daß ein Informationsobjekt eine Beschreibung eines anderen Informationsobjekts darstellt. Dies ist z.B. besonders für den Fall nützlich, wenn es sich bei dem im KPro abgelegten Content um ein Bild handelt.

- **Strukturbeziehung**  
Die Strukturbeziehung (STRUCTLINK) modelliert eine hierarchische Relation zwischen den Inhalten von Dokumenten. Dies können beispielsweise die Beziehungen zwischen Kapiteln und Absätzen innerhalb eines Buchs sein.
- **Hyperlink-Beziehung**  
Die Hyperlink-Beziehung (HYPERLINK) wird eingesetzt, um Hyperlinks zwischen Dokumenten darzustellen. Zeigt ein Hyperlink auf ein logisches Dokument, wird die vom KPro zur Verfügung gestellte Funktionalität der Kontextauflösung eingesetzt.

## Versionierung

# Versionierung

Der Begriff Versionierung ist im KPro ein Oberbegriff für unterschiedliche Subtypen der Versionierung, die durch verschiedene Beziehungsklassen der Versionierung abgebildet werden. Versionierung kann allgemein verstanden werden als die Bildung einer Ableitungsbeziehung zwischen zwei Dokumenten. Die Beziehung ist gerichtet von einem Dokument 2 zu einem Dokument 1, wobei Dokument 2 unter einem bestimmten Kriterium aus Dokument 1 hervorgegangen ist.

Folgende Beziehungsklassen der Versionierung sind im KPro vordefiniert:

- Inhaltsversionierung  
Kriterium: Inhalt hat sich verändert
- Formatversionierung  
Kriterium: Format hat sich verändert
- Sprachversionierung  
Kriterium: Sprache hat sich verändert

## Multiple Versionierung

Unabhängig von diesen Kriterien unterstützt das KPro das Prinzip der **multiplen Versionierung**: Werden aus einem physischen Dokument mehrere weitere physische Dokumente abgeleitet, so handelt es sich hierbei um eine multiple Versionierung. Die abgeleiteten Dokumente können sich in bezug auf Format, Sprache, etc. unterscheiden.

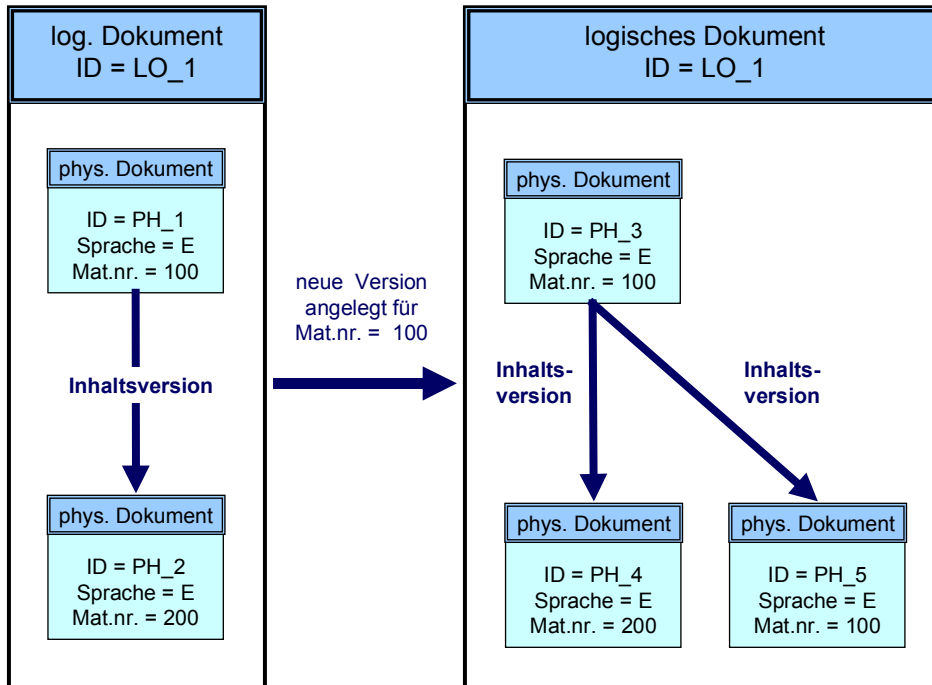


Aus einem physischen Dokument werden drei verschiedene Inhaltsversionen abgeleitet. Diese drei Versionen sind gleichberechtigte Söhne von einem physischen Dokument.

Wird ein deutsches Dokument ins Englische und Spanische übersetzt, so liegt ebenfalls eine multiple Versionierung vor.

Das folgende Bild illustriert das Prinzip der multiplen Versionierung bezüglich des Versionierungskriterium der Inhaltsversionierung.

## Multiple Versionierung



Das logische Dokument auf der linken Seite beinhaltet zwei Handbücher für Waschmaschinen auf Englisch, eins für Modell 100 und eins für Modell 200. Das Handbuch für Modell 200 ist aus dem Handbuch, das ursprünglich für Modell 100 geschrieben wurde, hervorgegangen. Da nun das Modell 100 einen technischen Fehler aufwies, mußte das Handbuch entsprechend geändert werden. Um den Namen des Änderers und den Zeitpunkt der Änderung jederzeit nachvollziehen zu können (z.B. im Rahmen von Qualitätssicherungsmaßnahmen), wurde eine neue Inhaltsversion angelegt, ohne daß das Original überschrieben wurde. Somit wurde ein neues physisches Dokument für das Modell 100 angelegt, das im rechten Bildteil zu sehen ist. Dadurch existieren zwei Inhaltsversionen, die von dem Content abgeleitet wurden, der ursprünglich für das Modell 100 geschrieben wurde.

## Strikte Versionierung

Durch das Prinzip der **strikten Versionierung** wird der Content der Komponenten von physischen Dokumenten automatisch vor Änderungen geschützt. Jede Änderung zieht eine neue Version mit einer neuen ID nach sich. Dies kann bei Anwendungen, die eine Vielzahl von Dokumenten bearbeiten, zu einem großem Overhead an Versionen führen. Wenn nun Referenzen auf physische Dokumente eingesetzt würden, müßte jede einzelne dieser Referenzen aktualisiert werden, sobald eine neue Version angelegt oder eine vorhandene Version gelöscht wird. Diese umständliche und aufwendige Handhabung ist jedoch durch das dreistufige Content-Modell (siehe auch [Content-Modelle \[Seite 17\]](#)) nicht notwendig. Das dreistufige Content-Modell unterstützt sowohl die strikte Versionierung als auch Referenzen zwischen Dokumenten: Alle Referenzen beziehen sich auf logische Dokumente. Dadurch ist es unerheblich, ob für die einzelnen physischen Dokumente neue Versionen mit neuen IDs angelegt werden. Es gibt keinen unnötigen Versions-Overhead.

## Versionierung

### Versionierungsarten

#### Inhaltsversionierung

Wird durch die Veränderung des Contents der Komponenten eines bereits existierenden physischen Dokuments ein neues physisches Dokument erzeugt, so ist es Aufgabe der Inhaltsversionierung, diese Information im KPro zu speichern und zu verwalten. Zur Bildung der Inhaltsversionierung werden Inhaltsbeziehungen genutzt.

#### Formatversionierung

Die Formatversionierung hat die Aufgabe, physische Dokumente, die sich ausschließlich in ihrem technischen Format unterscheiden, geschlossen zu verwalten. Die Formatversionierung verwaltet hierzu Formatbeziehungen. Physische Dokumente, die an einer Formatversionierung teilnehmen, müssen zum selben logischen Dokument gehören.



Wird durch Formatkonvertierung aus einem DOC ein PDF erzeugt, so wird dieser Zusammenhang durch eine Formatbeziehung im KPro abgelegt. Die Formatversionierung verwaltet diese Beziehung und erlaubt zu jeder Zeit die Kontrolle, aus welchem DOC das PDF erzeugt wurde. Somit wird eine Anfrage der Art „Gibt es zu diesem physischen Informationsobjekt ein Format, das meine Applikation versteht?“ möglich.

#### Sprachversionierung

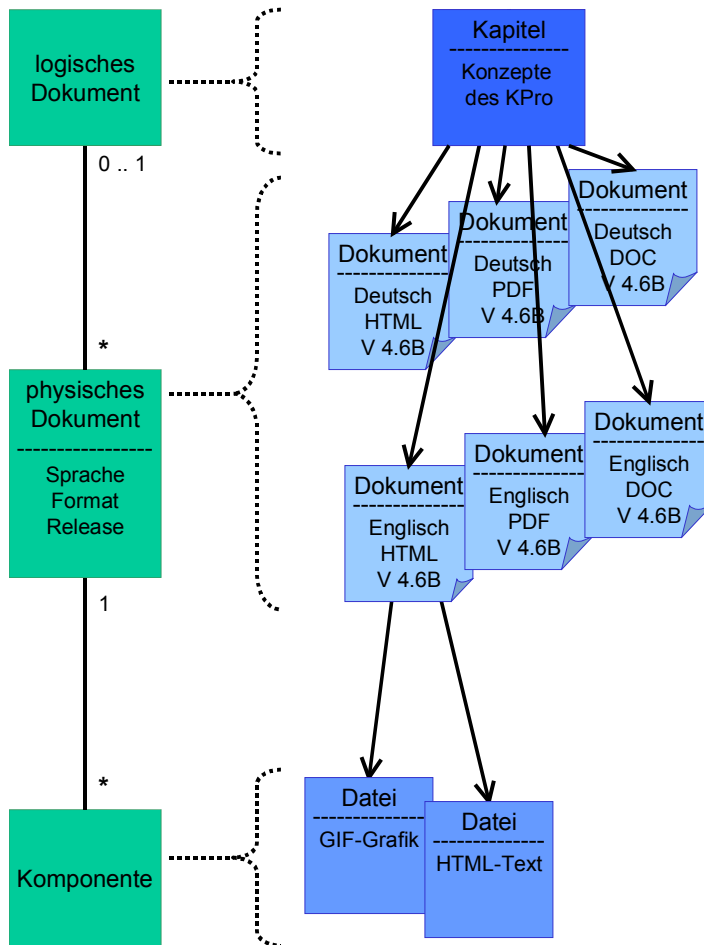
Die Verwaltung der Sprachbeziehungen wird als Sprachversionierung bezeichnet.



Ein Benutzer möchte ein im KPro verwaltetes Dokument von Deutsch nach Englisch übersetzen. Über die Schnittstellen des KPro besteht die Möglichkeit, eine Kopie des deutschen Originals an die Applikation des Benutzers zu übergeben. Zwischen der Kopie und dem Original wird die Sprachbeziehung angelegt. Hat der Benutzer die Übersetzung beendet, so erfolgt die Übergabe der übersetzten Version an das KPro über die Check-In Funktionalität. Die Sprachversionierung hat die Aufgabe, die angelegte Sprachbeziehung zu verwalten und Anfragen der Art „Gibt es Übersetzungen zu dem physischen Dokument 4711 in die Sprachen x, y und z?“ zu beantworten.

## Kontextauflösung

Im KPro existiert das Konzept der logischen Hyperlinks auf logische Dokumente. Der wichtigste Bestandteil eines logischen Hyperlinks auf ein logisches Dokument ist die ID des logischen Dokuments. Zur Laufzeit wird ein dynamisch hinzugefügter Kontextstring (Attributname1 = Attributwert1, Attributname2 = Attributwert2, ...) hinzugefügt, um den logischen Hyperlink aufzulösen und das physische Dokument und dessen Komponenten und den Content zu ermitteln. Dieser Prozeß wird als Kontextauflösung bezeichnet.



Eine Komponente repräsentiert genau eine Datei auf dem Content Server. Dadurch kann ein Dokument aus mehreren Dateien bestehen, was besonders bei HTML-Dokumenten häufig der Fall ist (HTML-Dokumente enthalten oft GIF-Dateien).

Physische Dokumente dagegen repräsentieren individuelle Dokumente und alles, was mit einem individuellen Dokument assoziiert ist (d.h. Komponenten). Da physische Dokumente auch die Eigenschaften von Dokumenten beinhalten, die eine bestimmte Version ausmachen, werden sie für den Prozeß der Kontextauflösung ausgewertet.

## Kontextauflösung

Logische Dokumente dagegen repräsentieren alle zu einer Collection gehörenden Dokumente. Somit beinhalten sie auch alle physischen Dokumente. Logische Dokumente enthalten keinerlei relevanten Informationen für die Kontextauflösung und sind auch nicht dem Versionierungsprozeß unterworfen.

Logische Dokumente stellen eine Klammerung derjenigen physischen Dokumente dar, die logisch zu einer Gruppe gehören. Die Definition einer Gruppe und ihrer Eigenschaften hängt von der jeweiligen Client-Anwendung ab. Referenzen weisen immer auf logische Dokumente, wodurch ihre Unabhängigkeit vom Lebenszyklus der einzelnen Dokumente gewährleistet ist.



Zur gleichen Zeit können mehrere aktuelle Versionen (siehe auch: [Versionierung \[Seite 12\]](#)) eines Dokuments vorliegen. Ein Benutzer, der in einem SAP-System auf Englisch angemeldet ist und einen Web Browser als Viewer für den KPro auf dem Frontend installiert hat, möchte sich die englische HTML-Version der Dokumentation "Konzepte des KPro" anzeigen lassen. Ein deutscher Benutzer hingegen, der auf Deutsch angemeldet ist, möchte sich die deutsche Version dieser Dokumentation anzeigen lassen. Außerdem erwarten beide Benutzer, die jeweils zu ihrem Releasestand passende Dokumentation zu sehen. Nun gleicht der KPro die Attribute des Benutzerkontexts mit den entsprechenden Dokument-Eigenschaften ab:

Attribute des Benutzerkontextes	Eigenschaften des Dokuments
Anmeldesprache	Sprache
Viewer-Anwendung	Format
aktuelles Release des SAP-Systems	Release des SAP-Systems, das für das Dokument definiert ist

Wenn ein Dokument in der Collection existiert, für das alle Eigenschaften mit den Kontext-Attributen übereinstimmen, liefert der KPro dieses Dokument zurück. Sonst wird das Dokument zurückgeliefert, für das die meisten Eigenschaften übereinstimmen.

Eine Client-Anwendung des KPro kann zusätzliche Kontext-Attribute und Dokument-Eigenschaften definieren (siehe auch: [Attribute \[Seite 8\]](#)).

Existieren mehrere Dokumente mit der gleichen Anzahl von übereinstimmenden Eigenschaften, dann wird die applikationsspezifische Kontextauflösung in Form von anwendungsspezifischen Funktionsbausteinen aktiviert. Diese Funktionsbausteine werten die zusätzlichen Attribute aus, um ein passendes Dokument zurückzuliefern, oder sie priorisieren Eigenschaften im Kontext der Anwendung.

## Content-Modelle

Die einzelnen SAP-Anwendungen erstellen Meta-Datenmodelle für die verschiedenen Typen von Dokumenten, die sie benutzen und die in Beziehungen zueinander stehen. Diese Modelle werden als Content-Modelle bezeichnet.

In einem Content-Modell werden auf einer abstrakten Ebene die Charakteristika der einzelnen Content-Entitäten im KPro beschrieben. Die verschiedenen Arten von Content werden auf die Dokumente im Content-Modell abgebildet und verwaltet. Mögliche Arten von Content sind Endbenutzer-Dokumentation, eingescannte Dokumente, etc.

### KPro Content-Modell

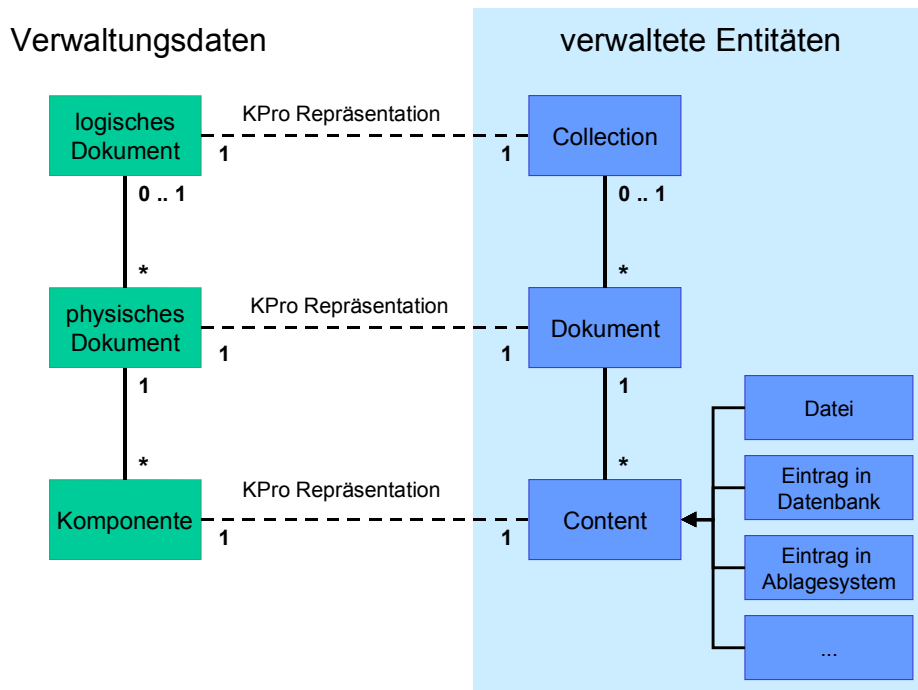
Das KPro Content-Modell beinhaltet drei Ebenen für Verwaltungsdaten. Daher wird das KPro Content-Modell auch als **dreistufiges Content-Modell** bezeichnet:

- Komponenten verwalten Content
- physische Dokumente verwalten Dokumente
- logische Dokumente verwalten Collections von Dokumenten

Mehrere physische Dokumente werden einem logischen Dokument zugeordnet. Ein physisches Dokument repräsentiert ein individuelles Dokument, während ein logisches Dokument eine Collection von Dokumenten repräsentiert.

Ein Dokument kann aus unterschiedlichen Dateien bestehen, d.h. zu jedem physischen Dokument kann eine oder mehrere Komponenten bestehen. Jede Komponente wiederum wird mit genau einer Content-Einheit assoziiert.

### KPro Content-Modell



## Content-Modelle



Der KPro erlaubt ebenfalls die Modellierung von zweistufigen Content-Modellen. In diesem Fall benötigt eine den KPro nutzende Anwendung keinerlei logische Dokumente. Es werden lediglich physische Dokumente und Komponenten eingesetzt.

## Anwendungsspezifisches Content-Modell

Um die Dokumenteninfrastruktur des KPro zu nutzen, können Anwendungen ihre eigenen Content-Modelle entwickeln. Während der Laufzeit greift der KPro auf diese anwendungsspezifischen Content-Modelle zu und verwendet die darin enthaltenen Meta-Informationen, um die benötigten Services auszuführen. Somit können Anwendungen beispielsweise ihre eigenen Attribute oder bestimmte Arten der Versionierung definieren.

Als erstes entscheidet sich die Anwendung, ob ein zwei- oder dreistufiges Content-Modell für ihre Belange einzusetzen ist. Wenn erwartet werden kann, daß die Dokumente im Laufe der Zeit einem Änderungs- oder Versionierungsprozeß unterworfen sein werden, wird das dreistufige Content-Modell eingesetzt. Mögliche Änderungen sind Änderungen des Inhalts, der Sprache und des Formats. Das zweistufige Content-Modell ist nur für Dokumente sinnvoll, die während ihres Lebenszyklus keinerlei Änderungen oder Versionierungen unterworfen sein werden.

Das dreistufige Content-Modell wird auch für Szenarios empfohlen, in denen mehr als eine kontextabhängige Instanz eines bestimmten physischen Dokuments gleichzeitig existiert.

Sobald eine Entscheidung für das zwei- oder dreistufige Content-Modell gefällt wurde, kann das Content-Modell selbst in den folgenden Schritten modelliert werden:

- Definition von anwendungsspezifischen Attributen
- Definition von Klassen für physische Dokumente
- Definition von Klassen für logische Dokumente
- Definition von Klassen von Beziehungen
- Definition der erlaubten Beziehungen

Im letzten Schritt werden Regeln für das Anlegen von Beziehungen bestimmter Klassen zwischen Instanzen logischer und physischer Dokumente festgelegt. Während der Laufzeit greift der KPro auf diese Informationen innerhalb des anwendungsspezifischen Content-Modells zu, um zu klären, ob die geforderte Aktivität zulässig ist. Diese Regeln können beispielsweise eingesetzt werden, um unerlaubte Beziehungen zwischen Inhaltsversionen physischer Dokumente abzufangen.



Sie können Ihr eigenes Content-Modell über die **Document Modeling Workbench** (Transaktion DMWB) anlegen.

## DMS-Programmierschnittstelle

### Verwendung

Sie verwenden diese Programmierschnittstelle, um die Funktionalität des Document Management Service in Ihre SAP-Anwendung zu integrieren.

### Funktionsumfang

Es stehen die folgenden Funktionsbausteine zur Verfügung:

- Funktionsbausteine für Dokument- und Beziehungsklassen
  - SDOK\_CLASS\_ATTRIBUTES\_GET
  - SDOK\_CLASS\_FIND\_BY\_QUERY
  - SDOK\_CLASS\_PARTNERS\_GET
  - SDOK\_CLASS\_PROPERTIES\_DELETE
  - SDOK\_CLASS\_PROPERTIES\_GET
  - SDOK\_CLASS\_PROPERTIES\_SET
- Funktionsbausteine für physische Dokumente
  - SDOK\_PHIO\_CREATE
  - SDOK\_PHIO\_CREATE\_AS\_COPY
  - SDOK\_PHIO\_CREATE\_NEW\_VERSION
  - SDOK\_PHIO\_CREATE\_WITH\_MODEL
  - SDOK\_PHIO\_DELETE
  - SDOK\_PHIO\_ATTRIBUTES\_GET
  - SDOK\_PHIO\_PROPERTIES\_SET
  - SDOK\_PHIO\_PROPERTIES\_GET
  - SDOK\_PHIO\_PROPERTIES\_DELETE
  - SDOK\_PHIO\_FROM\_RELATIONS\_GET
  - SDOK\_PHIO\_TO\_RELATIONS\_GET
  - SDOK\_PHIO\_FIND\_BY\_QUERY
  - SDOK\_PHIO\_STORE\_CONTENT
  - SDOK\_PHIO\_STORE\_AS\_UPDATE
  - SDOK\_PHIO\_STORE\_NEW\_VERSION
  - SDOK\_PHIO\_STORE\_NEW\_CONTENT
  - SDOK\_PHIO\_LOAD\_CONTENT
  - SDOK\_PHIOS\_CHECKIN

**DMS-Programmierschnittstelle**

- SDOK\_PHIOS\_CHECKIN\_AS\_UPDATE
- SDOK\_PHIOS\_CHECKIN\_NEW\_VERSION
- SDOK\_PHIOS\_CHECKOUT
- SDOK\_PHIOS\_CHECKOUT\_CANCEL
- SDOK\_PHIOS\_GET\_COPY
- SDOK\_PHIOS\_FILE\_PROPERTIES\_GET
- SDOK\_PHIO\_GET\_URL\_FOR\_GET
- SDOK\_PHIO\_GET\_URL\_FOR\_UPDATE
- SDOK\_PHIO\_GET\_URL\_FOR\_NEW\_VERS
- SDOK\_PHIO\_GET\_URL\_NEW\_CONTENT
- SDOK\_PHIO\_GET\_URL\_FOR\_PUT
- Funktionsbausteine für logische Dokumente
  - SDOK\_LOIO\_CREATE
  - SDOK\_LOIO\_CREATE\_WITH\_PHIO
  - SDOK\_LOIO\_DELETE
  - SDOK\_LOIO\_ATTRIBUTES\_GET
  - SDOK\_LOIO\_PROPERTIES\_SET
  - SDOK\_LOIO\_PROPERTIES\_GET
  - SDOK\_LOIO\_PROPERTIES\_DELETE
  - SDOK\_LOIO\_DESCRIPTIONS\_SET
  - SDOK\_LOIO\_DESCRIPTIONS\_GET
  - SDOK\_LOIO\_DESCRIPTIONS\_DELETE
  - SDOK\_LOIO\_FROM\_RELATIONS\_GET
  - SDOK\_LOIO\_TO\_RELATIONS\_GET
  - SDOK\_LOIO\_FIND\_BY\_QUERY
  - SDOK\_LOIO\_PHYSICAL\_OBJECT\_GET
- Funktionsbausteine für Beziehungen
  - SDOK\_RELA\_CREATE
  - SDOK\_RELA\_DELETE
  - SDOK\_RELA\_PROPERTIES\_SET
  - SDOK\_RELA\_PROPERTIES\_GET
  - SDOK\_RELA\_PROPERTIES\_DELETE
- Weitere Funktionsbausteine

- SDOK\_ATTRS\_QUALITIES\_GET
- SDOK\_DOCUSPACE\_SEARCH



Bei den hier angegebenen Funktionsbausteinen handelt es sich lediglich um einen Auszug der wichtigsten DMS-Funktionsbausteine.