



Infopark CMS Fiona

▶ **Administration /
Layout**

Infopark CMS Fiona

Administration / Layout

Die Informationen in allen technischen Dokumenten der Infopark AG wurden mit größter Sorgfalt erarbeitet. Dennoch können Fehler nicht vollständig ausgeschlossen werden. Wir übernehmen keine juristische Verantwortung oder Haftung für eventuell verbliebene fehlerhafte Angaben und deren Folgen. Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen. Wir richten uns im Wesentlichen nach den Schreibweisen der Hersteller. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Alle Rechte vorbehalten, einschließlich der Vervielfältigung, Übersetzung, Mikroverfilmung sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Inhalt

1 Vorbemerkungen	11
2 Einführung	12
3 Konzepte	13
3.1 Benutzerverwaltung	13
3.1.1 Globale Rechte	13
3.1.2 Dateispezifische Rechte	14
3.2 Dateien und Dateivorlagen	15
3.2.1 Dateitypen und der Export	15
3.2.2 Felder, Feldergruppen und Feldvorbelegung	16
3.2.3 Workflows	18
3.2.4 Checks und Funktionen	18
3.3 Jobs	19
3.4 Channels	19
4 Benutzer und Benutzergruppen verwalten	21
4.1 Die Sektion Benutzer	22
4.1.1 Benutzer einrichten	23
4.1.2 Persönliche Einstellungen des Benutzers bearbeiten	27
4.1.3 Benutzerdaten ändern und Benutzer löschen	27
4.2 Die Sektion Gruppen	28
4.2.1 Gruppen einrichten	29
4.2.2 Gruppeneinstellungen ändern und Gruppen löschen	32
4.3 Die Sektion Benutzerfelder	32
4.3.1 Benutzerfelder anlegen	33
4.3.2 Benutzerfelder ändern oder löschen	34
5 Den Content Manager konfigurieren	36
5.1 Die Sektion Vorlagen	36
5.1.1 Vorlagen definieren	38
5.1.2 Feldeigenschaften bearbeiten	42
5.1.3 Anlegeeinschränkungen	44
5.1.4 Bearbeitung und Export	45
5.1.5 Checks und Funktionen	47
5.1.6 Feldergruppen konfigurieren	50

5.1.7	Vorlagen ändern und löschen	54
5.2	Die Sektion Felder	55
5.2.1	Felder definieren	56
5.2.2	Feldeigenschaften ändern	60
5.2.3	Allgemeine Eigenschaften	60
5.2.4	Spezielle Eigenschaften	60
5.2.5	Funktionen	61
5.2.6	62
5.2.7	Felder löschen	63
5.3	Die Sektion Workflows	64
5.3.1	Workflows definieren	65
5.3.2	Allgemeine Eigenschaften	66
5.3.3	Bearbeitungsschritte	66
5.3.4	Prüfungsschritte	67
5.3.5	Workflows bearbeiten oder löschen	68
5.4	Die Sektion Jobs	69
5.4.1	Jobs definieren	71
5.4.2	Allgemeine Eigenschaften	71
5.4.3	Ausführung	72
5.4.4	Skript	73
5.4.5	Jobs ausführen oder abbrechen	73
5.4.6	Jobs bearbeiten oder löschen	74
5.4.7	Den Ausführungsplan eines Jobs bearbeiten	75
5.4.8	Job-Protokoll und Job-Ausgabe anzeigen	77
5.5	Die Sektion Channels	78
5.5.1	Channels definieren	79
5.5.2	Channels bearbeiten oder löschen	80
6	Layouts	81
6.1	Funktion der Layouts	81
6.2	Ein Layoutkonzept entwerfen	82
6.3	Layouts bearbeiten	83
6.4	Aufbau einer Layoutdatei	84
6.4.1	XML- und XHTML-Kennzeichnung	84
6.4.2	HTML-Header	84

6.4.3	Style sheets	85
6.4.4	HTML-Hauptinhalt und andere Feldwerte	85
6.4.5	Automatische Navigationseinträge	87
6.4.6	Mikronavigation	88
7	Automatisch generierte Inhaltsübersichten	90
7.1	Anforderungen an Startseiten	90
7.2	Bestandteile automatisch generierter Inhaltsübersichten	90
7.2.1	Aufbau einer automatisch generierten Inhaltsübersicht	90
7.2.2	NPSOBJ-Parameter für automatisch generierte Inhaltsübersichten	91
7.2.3	Anzahl der Einträge in Inhaltsübersichten festlegen	91
7.2.4	Andere Felder verwenden	92
7.2.5	Sortierung der Einträge	92
7.3	Einträge steuern	93
7.3.1	Den Dateityp abfragen	93
7.3.2	Feldwerte abfragen	94
7.3.3	Kombinierte Vergleiche	95
7.4	Beispiele für automatisch generierte Inhaltsübersichten	96
7.4.1	Formatierung als Folge von Absätzen	96
7.4.2	Weitere Feldwerte aufnehmen	96
7.4.3	Formatierung als zweispaltige Tabelle	97
7.4.4	Trennung nach Dateityp	97
7.4.5	Geschwisterdateien verlinken	97
7.5	Inhaltsübersichten per Feldwert auswählen	98
8	Linksammlungen	100
8.1	NPSOBJ-Tags zur Erzeugung von Linksammlungen	100
8.2	Inhalt und Formatierung von Linksammlungen	100
8.2.1	Links auf Dateien auswerten	101
8.2.2	Elemente in Linksammlungen selektieren und sortieren	101
9	RSS-News-Feeds erzeugen	103
9.1	Konzept der RSS-Feed-Erzeugung	103
9.2	Die npsobj-newslist-Anweisung verwenden	103
9.3	Beispiele zum Gebrauch von News-Listen	104
9.3.1	Eine Liste mit den 10 neuesten News erzeugen	104
9.3.2	Einen RSS-Feed für Politik-News einrichten	105

9.3.3	Einen Newsletter versenden	106
10	Export-Variablen	107
10.1	Texte speichern und abrufen	107
10.2	Kontextlisten speichern und abrufen	108
10.3	Lokale Exportvariablen	110
11	Weitere NPSOBJ-Anweisungen	111
11.1	Den Hauptinhalt einer Datei verwenden	111
11.2	Systemkommandos ausführen	111
11.3	Absolute Pfade in Skripten ausgeben	112
12	Beispiele für Layouts	113
12.1	Verschachtelte Layouts	113
12.1.1	Eine Sitemap als Inhaltsübersicht erstellen	113
12.1.2	Sitemap mit ausgewählten Dateien in Ordnern	114
12.1.3	Unendliche Rekursion	114
12.1.4	Tabellenlayouts definieren	114
12.2	Framelayouts	115
12.2.1	Framesets mit dem NPSOBJ-Frame-Kommando automatisch erzeugen	116
12.2.2	Frames mit Ordnerhierarchien realisieren	118
13	Die NPSOBJ-Syntax	120
13.1	Konventionen	120
13.2	Wertbestimmung im Kontext	121
13.2.1	Was ist ein Kontext?	121
13.2.2	Der Aufbau von Namen	122
13.2.3	Namen in Datei-Kontexten	123
13.2.4	Namen in Link-Kontexten	125
13.2.5	Namen in allen Kontexten	125
13.2.6	@-Referenzen	126
13.2.7	Tcl-Prozeduren zur Formatierung von Werten	127
13.3	npsobj break	130
13.3.1	Syntax	130
13.3.2	Aufgabe	130
13.3.3	Beispiel	131
13.4	npsobj condition	131

13.4.1	Syntax	131
13.4.2	Aufgabe	131
13.4.3	Beispiel	133
13.5	npsobj context	134
13.5.1	Syntax	134
13.5.2	Aufgabe	134
13.5.3	Beispiel	134
13.6	npsobj frame	134
13.6.1	Syntax	134
13.6.2	Aufgabe	135
13.6.3	Beispiel	135
13.6.4	Verwendung von target="_top" in der Vorschau	135
13.7	npsobj includetext	136
13.7.1	Syntax	136
13.7.2	Aufgabe	136
13.7.3	Beispiel	136
13.8	npsobj insertvalue anchor	136
13.8.1	Syntax	136
13.8.2	Aufgabe	137
13.8.3	Beispiel	137
13.9	npsobj insertvalue dynamiclink	137
13.9.1	Syntax	137
13.9.2	Aufgabe	137
13.9.3	Beispiel	138
13.10	npsobj insertvalue image	138
13.10.1	Syntax	138
13.10.2	Aufgabe	139
13.10.3	Beispiel	139
13.11	npsobj insertvalue link	139
13.11.1	Syntax	139
13.11.2	Aufgabe	139
13.11.3	Beispiel	140
13.12	npsobj insertvalue meta	140
13.12.1	Syntax	140
13.12.2	Aufgabe	140

13.12.3	Beispiel	140
13.13	npsobj insertvalue systemexecute	141
13.13.1	Syntax	141
13.13.2	Aufgabe	141
13.13.3	Beispiel	141
13.14	npsobj insertvalue template	141
13.14.1	Syntax	141
13.14.2	Aufgabe	142
13.14.3	Beispiele	143
13.15	npsobj insertvalue var	143
13.15.1	Syntax	143
13.15.2	Aufgabe	143
13.15.3	Beispiel	144
13.16	npsobj list	144
13.16.1	Syntax	144
13.16.2	Aufgabe	145
13.16.3	Beispiel	146
13.17	npsobj micronavigation	146
13.17.1	Syntax	146
13.17.2	Aufgabe	146
13.17.3	Beispiel	147
13.18	npsobj modifyvar append	147
13.18.1	Syntax	147
13.18.2	Aufgabe	147
13.18.3	Beispiel	147
13.19	npsobj modifyvar clear	148
13.19.1	Syntax	148
13.19.2	Aufgabe	148
13.19.3	Beispiel	148
13.20	npsobj modifyvar range	148
13.20.1	Syntax	148
13.20.2	Aufgabe	148
13.20.3	Beispiel	148
13.21	npsobj modifyvar set	149
13.21.1	Syntax	149

13.21.2	Aufgabe	149
13.21.3	Beispiel	150
13.22	npsobj modifyvar sort	151
13.22.1	Syntax	151
13.22.2	Aufgabe	151
13.22.3	Beispiel	151
13.23	npsobj newslst all	151
13.23.1	Syntax	151
13.23.2	Aufgabe	151
13.23.3	Beispiel	152
13.24	npsobj newslst selected	152
13.24.1	Syntax	152
13.24.2	Aufgabe	152
13.24.3	Beispiel	152
13.25	npsobj switch	152
13.25.1	Syntax	152
13.25.2	Aufgabe	153
13.25.3	Beispiel	153
13.26	npsobj table	153
13.26.1	Syntax	153
13.26.2	Aufgabe	154
13.26.3	Beispiel	154
14	Die Bearbeitungsmöglichkeiten in der Vorschau erweitern	155
14.1	Bearbeitungshinweise in der Vorschau anzeigen	155
14.2	Aktionen anbieten	156
14.2.1	Syntax	156
14.2.2	Assistenten parametrisieren	157
14.2.3	Bearbeitungselemente dynamisch positionieren	157
14.3	Anzeige automatisch erzeugter Marker steuern	160



1 Vorbemerkungen

Dieses Handbuch wendet sich an Administratoren, die im Content Management Server Benutzer und Benutzergruppen administrieren sowie Felder, Dateivorlagen, Workflows und Jobs einrichten möchten. Es ist auch an Designer gerichtet, die Layouts für den Content Management Server erstellen und pflegen möchten.

Als Leser dieses Handbuchs sollten Sie über allgemeine Computerkenntnisse sowie Erfahrungen beim Umgang mit dem Internet und WWW-Browsern verfügen. Als Administrator sollten Sie das Benutzer- und Gruppenkonzept kennen und mit den redaktionellen Konzepten von Infopark CMS Fiona vertraut sein. Als Designer benötigen Sie ein solides Grundwissen über Dateien und Versionen im Content Management Server sowie gute HTML-Kenntnisse.

Die Erläuterungen und Handlungsbeschreibungen in dieser Dokumentation beziehen sich auf die deutschsprachige Benutzeroberfläche – den Content Navigator – mit den Standardvorgaben des Content Management Servers. Die Erläuterungen sind weitgehend neutral bezüglich des von Ihnen verwendeten Betriebssystems und Ihres Browsers. Bitte konsultieren Sie bei Fragen oder Problemen das Handbuch des jeweiligen Produkts.

Die Funktionen des Content Navigators sind nur dann nutzbar, wenn in Ihrem Browser Java und Javascript eingeschaltet sind. Die Vorschaufunktion setzt voraus, dass Ihr Browser Cookies annimmt.

2

2 Einführung

Infopark CMS Fiona ist eine flexible und leistungsfähige Server-Software, mit der Sie umfangreiche Online-Systeme wie Websites erstellen, pflegen und publizieren können. Der Content Management Server ist die Hauptkomponente dieses Systems. Er verwaltet Ihre Dokumente und gibt Ihnen die Möglichkeit, sie kontrolliert zu gestalten und zu bearbeiten.

Eine Website ist ein komplex strukturiertes Gebilde. Sie besteht oft aus vielen Bereichen mit unterschiedlichem Charakter, in Bezug auf Themen, Layout und Funktion. Sie wird gemeinsam von Redakteuren, Designern und Administratoren, vielleicht auch Programmierern hergestellt, gepflegt und erweitert. Das Webangebot Ihres Unternehmens soll stets aktuell und attraktiv sein, das Corporate Design konsequent umgesetzt werden. Der Content Management Server kann Ihnen dabei helfen, dies zu verwirklichen.

Mit der Benutzerverwaltung des Content Management Servers können Sie Benutzer und Benutzergruppen anlegen. Sie können Benutzer zu Gruppen hinzufügen und deren Befugnisse definieren, indem Sie Rechte erteilen. Mit Benutzergruppen steuern Sie den Zugriff der Gruppen auf jede einzelne Datei. Sie können Workflows anlegen und jeden Arbeitsschritt an eine Benutzergruppe binden; mit globalen Rechten können Sie schließlich festlegen, welche Gruppen welche Dateivorlagen verwenden dürfen.

Dateivorlagen und Felder geben Ihnen die Möglichkeit, die Eigenschaften aller Dateien festzulegen und von ihrer Position in der Ordnerhierarchie abhängig zu machen. Die Workflows, die Sie definiert haben, können Sie in Vorlagen angeben, um den Weg festzuschreiben, den Dateien in Ihrem Unternehmen gehen müssen. Mit Jobs steht Ihnen schließlich ein Mittel zur Verfügung, das Sie verwenden können, um Tcl-Skripte über die HTML-Benutzerschnittstelle zu definieren und zeitgesteuert ablaufen zu lassen.

So mächtig diese Mittel auch sind - sie greifen erst, wenn die Zuständigkeiten aller Beteiligten detailliert erfasst sind und die Struktur Ihrer Website feststeht. Dann allerdings tragen sie dazu bei, Ihre Arbeit effizienter zu machen.

3 Konzepte

3.1 Benutzerverwaltung

Der Content Management Server hat eine einfache, aber flexible Benutzerverwaltung.

Die Benutzerverwaltung ist intern vom Content Management Server getrennt und wird von ihm über eine Schnittstelle angesprochen. Aufgrund dieser Architektur ist es möglich, den Content Management Server mit einem anderen Benutzermanager, beispielsweise einem auf LDAP basierenden zu verbinden.

Jeder Benutzer wird vom Content Management Server mittels eines Anmeldenamens verwaltet, der als eindeutige Bezeichnung den Benutzer "verkörpert". Jedem Login ist ein Kennwort zugeordnet, das dem Benutzer den Zugriff auf den Content Management Server ermöglicht.

Die in den Content Management Server integrierte Benutzerverwaltung ist auch für Benutzergruppen und die Zuordnung von Benutzern zu Gruppen zuständig. Ferner verwaltet sie die Zuordnung globaler Rechte zu Benutzern und Gruppen.

Mit globalen Rechten legen Sie die Befugnisse einzelner Benutzer oder der Mitglieder einer Benutzergruppe in Bezug auf administrative Aufgaben fest. Zum Zwecke einer rationellen Benutzerverwaltung sollten erst die Benutzergruppen eingerichtet und ihnen Rechte erteilt werden. Anschließend können die Benutzer in die Gruppen aufgenommen werden, um ihnen die Rechte der jeweiligen Gruppe zu erteilen. Ein Benutzer ist automatisch Mitglied einer Standardgruppe. Er kann Mitglied weiterer Benutzergruppen sein.

Benutzergruppen erfüllen im Content Management Server auch in Bezug auf Dateien eine wichtige Aufgabe, weil ihnen dateispezifische Rechte erteilt werden können. Mit diesen Rechten wird festgelegt, welche Benutzergruppen in welcher Weise auf Dateien zugreifen können.

Neben Benutzern und Benutzergruppen verwaltet der Content Management Server sogenannte Benutzerfelder, die bei der Benutzerverwaltung hilfreich sein können. In den Feldwerten können Sie benutzerspezifische Zusatzinformationen wie die Anschrift oder eine private Telefonnummer des jeweiligen Benutzers speichern, um bei Bedarf darauf zurückzugreifen. Allen Logins sind alle definierten Benutzerfelder zugeordnet.

3.1.1 Globale Rechte

Mit der Erteilung globaler Rechte werden Benutzern Befugnisse eingeräumt, die sich nicht auf die Arbeit mit einzelnen Dateien, sondern auf administrative Tätigkeiten beziehen. Ein Anwender, der beispielsweise über das Benutzerverwaltungsrecht verfügt, darf Anmeldenamen anlegen, ändern oder löschen. Im Einzelnen können folgende globale Rechte erteilt werden:

- **Administrator** (`permissionGlobalRoot`): Benutzer, denen dieses Recht erteilt wurde, besitzen alle Befugnisse. Ein Administrator wird auch als *Superuser* bezeichnet. Dieses Recht ist immer dann erforderlich, wenn kein anderes Recht ausreicht, um eine Aktion auszuführen (beispielsweise Jobs anlegen und bearbeiten).
- **Benutzer und Gruppen bearbeiten** (`permissionGlobalUserEdit`): Benutzer und Benutzergruppen anlegen und löschen. Änderungen können nur vom jeweiligen Verwalter vorgenommen werden.
- **Benutzerfelder bearbeiten** (`permissionGlobalUserAttributeEdit`): Benutzerfelder anlegen, ändern und löschen.
- **Globale Einstellungen bearbeiten** (`permissionGlobalRTCEdit`): Dateivorlagen, Felder, Workflows und Channels anlegen, ändern und löschen.
- **Export durchführen** (`permissionGlobalExport`): Kommandos ausführen, mit denen Ordner exportiert werden können.
- **Spiegeldateien anlegen** (`permissionGlobalMirrorHandling`): Dieses Recht ist erforderlich, um [Spiegeldateien](#) anlegen zu können. Zusätzlich gelten die dateispezifischen Einschränkungen, d.h. der Benutzer benötigt das Recht `permissionCreateChildren` im betreffenden Ordner.

Zusätzlich zu den genannten Rechten können in der Konfigurationsdatei des Content Management Servers `userManagement.xml` im Eintrag `globalPermissions` weitere globale Rechte frei definiert werden, die beispielsweise bei der Verwendung von [Dateivorlagen](#) eingesetzt werden können: in jeder Definition einer Vorlage kann eines der vordefinierten oder zusätzlichen Rechte angegeben werden, sodass ein Benutzer nur dann Dateien auf der Basis dieser Vorlage anlegen kann, wenn er dieses globale Recht hat.

3.1.2 Dateispezifische Rechte

Neben globalen Rechten gibt es im Content Management Server dateispezifische Rechte. Diese Rechte können je Datei vergeben werden und legen die Befugnisse fest, die Mitglieder von Benutzergruppen in Bezug auf die Datei haben. Beispielsweise muss ein Anwender bei einer Datei das Administrationsrecht haben, um sie löschen zu können; um die Feldwerte der Arbeitsversion der Datei ändern zu können, benötigt er das Schreibrecht.

Dateispezifische Rechte können nicht einzelnen Benutzern, sondern nur Benutzergruppen erteilt werden. Je Datei können folgende Rechte erteilt werden:

- **Datei administrieren** (`permissionRoot`)
Benutzer, denen dieses Recht erteilt wurde, besitzen alle Befugnisse. Das Administrationsrecht ist erforderlich, um Dateien löschen, umbenennen oder verschieben sowie ihre Freigabe zurücknehmen zu können. Als Administrator einer Datei haben Sie das Recht, es unabhängig von seinem Workflow freizugeben oder während der Prüfungsphase auch dann abzuzeichnen, wenn Sie nicht Mitglied in der dafür vorgesehenen Benutzergruppe sind.
- **Datei lesen** (`permissionRead`)
Jeder Zugriff auf die Datei, der sie nicht verändert.
- **Arbeitsversion bearbeiten** (`permissionWrite`)
Eine neue Arbeitsversion erzeugen sowie eine Arbeitsversion bearbeiten (dazu gehört auch die Änderung von Feldwerten).
- **Dateien anlegen** (`permissionCreateChildren`)
Dieses Recht kann nur für Ordner erteilt werden. Es erlaubt dem Inhaber, Dateien in einem Ordner anzulegen.
- **Veröffentlichte Datei lesen** (`permissionLiveServerRead`)

Dieses Recht kann in Verbindung mit dem Portal Manager verwendet werden. Diese CMS-Komponente erweitert den Live-Server unter anderem um dokumentspezifische Zugriffsrechte für die Besucher Ihrer Website. Die Gruppen, denen dieses Recht erteilt werden kann, können andere als bei den anderen Rechten sein, wenn dies vom Administrator so eingerichtet wurde. Wenn diesem Recht einer Datei wenigstens eine Benutzergruppe zugewiesen wurde, so haben nur die entsprechenden Gruppenmitglieder das Recht, auf das betreffende Dokument zuzugreifen. Andernfalls können alle Besucher darauf zugreifen.

Ob ein Benutzer in einem Ordner eine Datei anlegen kann, hängt sowohl von seinen dateispezifischen Rechten, die er in dem Ordner hat, als auch von seinen globalen Rechten ab. Zunächst benötigt der Benutzer in dem Ordner das Recht, Dateien anzulegen, das er automatisch hat, wenn er das Administrationsrecht in dem Ordner hat. Ist zur Verwendung der Dateivorlage, auf dem die neue Datei beruhen soll, ein spezielles globales Recht erforderlich, so benötigt der Benutzer auch dieses Recht. Hat er das globale Administrationsrecht, so darf er jede Vorlage verwenden.

Soll ein Benutzer eine neue Datei anlegen können, so gelten für diese Datei die gleichen Zugriffsrechte wie für den darüber liegenden Ordner. Die Datei erbt die Rechte des darüber liegenden Ordners. Sollten diese Rechte nicht das Lese- und Schreibrecht umfassen, so erteilt der Content Management Server der Standardgruppe des Benutzers die fehlenden dateispezifischen Rechte.

Rechte werden nicht nachträglich vererbt. Wird einem Benutzer beispielsweise das Administrationsrecht für den über einer Datei liegenden Ordner erteilt, so bleiben die Rechte, die der Benutzer in Bezug auf diese Datei hat, von dieser Maßnahme unberührt.

3.2 Dateien und Dateivorlagen

Die Ordnerhierarchie, aus der Ihre Website entsteht, setzt sich aus vielen Dateien zusammen. Diese Dateien werden im CMS auf der Grundlage von Dateivorlagen (kurz: "Vorlagen") erzeugt. Mit Vorlagen werden die Eigenschaften von Dateien konfiguriert und damit Art und Zweck der Dateien festgelegt. Eine Vorlage ist gewissermaßen eine Sammlung von Regeln, die vom Content Management Server angewendet werden, wenn ein Benutzer eine neue Datei anlegt.

3.2.1 Dateitypen und der Export

Ob und in welcher Form eine Datei exportiert wird, wird bereits durch den Dateityp festgelegt, der wiederum Bestandteil der Vorlage ist, die für die Datei verwendet wird. Bei Dateien vom Typ *Ordner*, *Dokument*, *Bild* oder *Ressource* entsteht beim Export zu jeder Datei eine korrespondierende Ausgabedatei. Dateien vom Typ *Layout* werden nicht exportiert.

Dateien vom Typ *Bild* oder *Ressource* werden ohne Layout exportiert, d.h. so wie sie sind, unmodifiziert. Die Hauptinhalte von Bilddateien sind importierte Grafikdateien. Ressourcen können beliebige Inhalte haben (zum Beispiel PDF-Dateien) und werden meist dem Besucher Ihrer Website zum Download angeboten.

Der Unterschied zwischen Ordnern und Dokumenten

Im Gegensatz zu Dokumenten können Ordner Dateien enthalten und dadurch Ihre Website strukturieren. Ein Ordner in Ihrer Website kann als Ausgangsseite (Indexseite) mit Verweisen auf darunter liegende Seiten dienen. In Bezug auf ihren Inhalt besteht jedoch kein Unterschied zwischen Dateien dieser beiden Typen.

Die Ausgabedatei von Ordnern und Dokumenten entspricht normalerweise einer Webseite. Beim Export von Dateien dieses Typs erzeugt der Content Management Server jedoch nicht einfach eine

Datei mit dem Haupttext der freigegebenen Version der Datei, sondern verwendet zum Export ein Basislayout namens `mastertemplate`.

Die Aufgabe des Basislayouts

Ein Basislayout ist ein spezielles Exemplar des Dateityps *Layout*. Es besteht üblicherweise aus HTML-Text und Exportanweisungen. Das Basislayout wird beim Export eines Ordners oder eines Dokuments vom Content Management Server wie ein Programm abgearbeitet, um die korrespondierende Ausgabedatei zu erzeugen. Der in der Layoutdatei enthaltene HTML-Text wird dabei unverändert in die Ausgabedatei geschrieben – mit einem speziellen Tag, der NPSOBJ-Anweisung, können Sie den Content Management Server jedoch anweisen, Feldwerte beispielsweise als `meta`-Tags oder den Haupttext der freigegebenen Version zu exportieren. Fehlen im Basislayout entsprechende Anweisungen, so wird die Datei unvollständig exportiert.

Gibt es mehrere Basislayouts, beispielsweise eines für den reinen Text, eines für das komplette aktuelle Layout und eines für den Relaunch, so kann jeder Benutzer seine individuelle Basislayoutdatei in seinen persönlichen Einstellungen festlegen. Dadurch ist es möglich, gleichzeitig an unterschiedlichen Layouts der gleichen Website zu arbeiten. In der Standardinstallation des Content Management Servers gibt es nur eine Basislayoutdatei, die schlicht `mastertemplate` heißt. Basislayoutdateien sind häufig Dateien im Basisordner.

Layouts können auf andere Layouts zurückgreifen

Typischerweise referenziert ein Basislayout weitere Layouts. Beim Export einer Datei sucht der Content Management Server ein referenziertes Layout zuerst in dem Ordner, in dem sich die Datei befindet. Ist die Layoutdatei dort nicht vorhanden, so wird die Ordnerhierarchie aufwärts bis zum Basisordner nach der Layoutdatei durchsucht. Dieser Mechanismus gibt Ihnen die Möglichkeit, lokale Layouts einzusetzen, die nur in Teilbereichen Ihrer Website wirksam sind.

Layouts sind der ideale Ort für Layout- und Navigationselemente, die den Haupttext umgeben sollen. Sie sind folglich ein Mittel, mit dem eine Website mit homogenem Layout effizient erstellt werden kann. Das Verfahren, mit dem das CMS, d. h. der Content Management Server und die [Template Engine](#), beim Export nach Layoutdateien sucht, unterstützt Sie zusätzlich dabei, Website-Bereiche mit individuellem Layout zu erstellen.

3.2.2 Felder, Feldergruppen und Feldvorbelegung

Dateien bestehen immer aus einem Hauptinhalt und einem Satz Felder. Es gibt zahlreiche vom CMS vorgegebene Felder, beispielsweise für den Namen einer Datei, den Titel seines Inhalts, etc. Ferner können Administratoren beliebige viele zusätzliche Felder konfigurieren.

Mit Feldern können Sie im Content Management Server Informationen zu einer Datei oder einer Version speichern oder die Versionen einer Datei mit Zusatzinformationen (Metadaten) versehen. Jedes Feld hat einen Namen, einen Titel, einen Typ und einen Wert. Der Content Management Server unterscheidet zwischen Dateifeldern und Versionsfeldern.

Dateifelder

Dateifelder sind vom Content Management Server fest vorgegeben und werden für jede Datei unabhängig von ihrem Typ erzeugt, wenn die Datei angelegt wird. Die Werte dieser Felder sind zur Verwaltung der Dateien und ihrer Beziehungen untereinander sowie zur Steuerung des Exports unverzichtbar.

Die Felder *ID* und *Pfad* werden automatisch vom System mit Werten belegt. Die ID ist die eindeutige Identifikationsnummer einer Datei im Content Management Server. Sie ist nicht änderbar. Dagegen ändert sich der Wert des Feldes *Pfad*, wenn eine Datei innerhalb der Ordnerhierarchie verschoben wird. In diesem Fall wird der Feldwert vom Content Management Server angepasst.

Jede Datei verfügt über das Feld *Name*, dessen Wert vom Bearbeiter der Datei festgelegt werden kann. Der Name einer Datei dient dazu, sie innerhalb eines Ordners eindeutig zu identifizieren. Wird kein Name angegeben, so wird vom Content Management Server ein vordefinierter Name, in der Standardinstallation *new*, vergeben. Werden weitere Namen automatisch erzeugt, so sorgt der Content Management Server für Eindeutigkeit, indem er einen Zähler an den vordefinierten Namen anhängt.

Neben *ID*, *Pfad* und *Name* haben Dateien unter anderem die Dateifelder *Dateityp*, *Workflow*, *Vorlage*, *Verantwortlicher* und *Von Export ausschließen*.

Der *Dateityp* ergibt sich aus der Vorlage und kann nicht geändert werden. Der Wert des Feldes *Workflow* ergibt sich ebenfalls aus der Vorlage, kann jedoch von einem Dateiaministrator nachträglich geändert werden. Das Feld *Vorlage* enthält bei jeder Datei den Namen der Vorlage, auf der sie basiert. Sein Wert kann geändert werden. Dies ist jedoch nur in Ausnahmefällen erforderlich und sollte vermieden werden (die Datei kann ungültig werden). Auch der Wert des Dateifeldes *Vom Export ausschließen* kann geändert werden.

Versionsfelder

Neben den Dateifeldern gibt es im Content Management Server die Versionsfelder, die nicht für eine Datei selbst sondern für die Arbeitsversion einer Datei erzeugt werden. Ein Versionsfeld kann in jeder Version einer Datei mit einem anderen Wert belegt sein. Es gibt fest vorgegebene und kundenspezifische Versionsfelder.

Analog zur ID von Dateien haben auch Versionen eine *Versions-ID*, die dazu dient, die Version eindeutig zu identifizieren. Die fest vorgegebenen Versionsfelder *Titel*, *Dateiendung*, *Gültig ab* und *Gültig bis* können Sie - wie einige Dateifelder - in der Vorlage der Datei, zu der die Version gehört, vorab mit Werten belegen.

Bei kundenspezifischen Versionsfeldern stehen Ihnen mehrere Feldtypen zur Verfügung (siehe auch [Typ](#)). Legen Sie beispielsweise ein Feld vom Typ *Auswahl* an, so kann der Bearbeiter einer Version aus mehreren von Ihnen vorgegebenen Feldwerten einen Wert auswählen.

Kundenspezifische Versionsfelder sind im Content Management Server als Felddefinitionen zu verstehen, die Sie in Dateivorlagen angeben können. Dadurch legen Sie fest, welche Felder die Arbeitsversionen der Dateien haben sollen, die auf den jeweiligen Dateivorlagen beruhen. Erst wenn eine Arbeitsversion angelegt wird, wird sie mit den Feldern ausgestattet, die in der Vorlage, auf der die entsprechende Datei beruht, angegeben sind. Versionen können je nach Vorlage ihrer Datei über unterschiedliche und beliebig viele kundenspezifische Felder verfügen.

Die Versionsfelder einer Dateivorlage können in der Definition der Dateivorlage zu Feldergruppen zusammengefasst werden. Die Feldergruppen werden in der Detailansicht einer Datei dargestellt. So können die Werte der Felder bequem bearbeitet und strukturiert erfasst werden.

Ferner können Sie in Vorlagen festlegen, welche Versionsfelder obligatorische Felder sein sollen, d. h. welche Felder einer Arbeitsversion einen Wert enthalten müssen, damit die entsprechende Datei freigegeben werden kann. Außerdem können alle Felder mit Werten vorbelegt werden, sodass die Felder der Arbeitsversionen neuer Dateien, die auf dieser Dateivorlage basieren, voreingestellte Werte enthalten.

3.2.3 Workflows

In vielen Fällen durchlaufen Dateien im Redaktionssystem einen Workflow, bevor sie exportiert werden. Mit Workflows können Sie festlegen, welche Mitarbeitergruppen nacheinander damit befasst sind, eine Datei zu bearbeiten oder zu prüfen. Dateien können sowohl minimale als auch umfangreiche Workflows durchlaufen.

In Dateivorlagen können Administratoren das Workflow-Feld der Dateien mit dem Namen eines Workflows vorbelegen. Wird die Arbeitsversion einer Datei mit einer bestimmten Vorlage erzeugt, so bestimmt der Content Management Server den zu verwendenden Workflow, indem er zunächst den Workflownamen aus der Vorlage ausliest. Der Workflow mit diesem Namen wird anschließend in der Arbeitsversion der Datei gespeichert.

Ein neuer Workflow beginnt immer dann, wenn eine Arbeitsversion erzeugt, eine Version abgelehnt oder eine freigegebene Version zurückgezogen wird. Da die Workflowdefinition in der Version gespeichert ist, kann ein aktiver Workflow nicht dadurch neu begonnen oder modifiziert werden, dass dem Feld *Workflow* der Name eines anderen Workflows zugewiesen oder die Workflowdefinition selbst geändert wird. Dies ist zwar möglich, die Änderungen werden jedoch erst dann wirksam, wenn eine neue Arbeitsversion der betreffenden Datei erzeugt wird.

3.2.4 Checks und Funktionen

Mit Checks und Funktionen können Sie sicherstellen, dass bei der Bearbeitung von Dateien Ihre inhaltlichen und formalen Kriterien eingehalten werden. Die Checks und Funktionen sind Tcl-Routinen, die immer dann abgearbeitet werden, wenn im Content Management Server applikationsspezifische Ereignisse eintreten, etwa wenn ein Redakteur Content geändert hat. Diese Routinen sind Bestandteil der Definitionen von Dateivorlagen und Feldern und können direkt in der Konfigurationsoberfläche des Redaktionssystems eingegeben werden, ein Shell-Zugang zum System ist also nicht erforderlich.

Checks

Checks sind Tcl-Routinen, die bei Prüfungen durch das System als Zusatzprüfungen ausgeführt werden. Sie geben Ihnen die Möglichkeit, den Abschluss einer Operation daran zu knüpfen, dass von Ihnen festgelegte Zusatzbedingungen erfüllt sind. So lässt Sie das System eine Datei nur freigeben, wenn sie vollständig ist. Dazu gehört auch, dass alle internen Links der Arbeitsversion ein Ziel haben. Die Liste der Vollständigkeitskriterien können Sie mit einer zusätzlichen Prüfroutine erweitern, so dass nicht nur die Bedingungen des Systems, sondern auch Ihre eigenen erfüllt sein müssen, damit eine Datei freigegeben werden kann. Es gibt zwei Checks, die Sie je Dateivorlage mit einer Tcl-Routine belegen können:

- Mit dem Anlegecheck für darunter liegende Dateien kann bei der Dateierzeugung in einem Ordner beispielsweise geprüft werden, ob die neue Datei auf einer Vorlage basiert, die in dem betreffenden Ordner verwendet werden darf.
- Mit einem Vollständigkeitscheck können Sie Versionen nach Ihren Kriterien auf Vollständigkeit prüfen. Der Tcl-Code für den Anlegecheck für darunter liegende Dateien oder den Vollständigkeitscheck kann in Dateivorlagen definiert werden.

Funktionen

Funktionen sind ebenfalls kundenspezifische Tcl-Routinen. Sie werden aufgerufen, nachdem ein Redakteur den Inhalt eines Feldes geändert hat, jedoch bevor diese Änderung in der Arbeitsversion gespeichert wird. Funktionen sind also dazu geeignet, Inhalte zu untersuchen und gegebenenfalls zu modifizieren.

- Die [Wertzuweisungsfunktion](#) eines Feldes wird immer dann ausgeführt, wenn der Wert des Feldes geändert wird. Dieses Feature können Sie nutzen, um zu prüfen, ob ein Feld einen gültigen Wert hat und um den Wert gegebenenfalls zu ändern.
- Die [Wertanzeigefunktion](#) für Felder können Sie verwenden, um im Content Navigator den anzuzeigenden Wert eines Feldes berechnen zu lassen. Der Tcl-Code einer Wertzuweisungsfunktion oder einer Wertanzeigefunktion ist Bestandteil einer Felddefinition.
- Eine Versionszuweisungsfunktion oder eine Workflowzuweisungsfunktion können Sie in der Definition der Vorlage eintragen. Die [Versionszuweisungsfunktion](#) wird genutzt, um die Konsistenz der Feldwerte einer Version zu prüfen und wird vom Content Management Server aufgerufen, bevor er die Feldwerte einer Version setzt.
- Mit der [Workflowzuweisungsfunktion](#) schließlich können Sie auf den Workflow einer Arbeitsversion Einfluss nehmen, bevor die Version erzeugt wird.

3.3 Jobs

Mit Jobs lassen sich kundenspezifische Tcl-Skripte einmalig oder wiederholt ausführen. Jobs stehen sowohl im Content Manager als auch in der Template Engine zur Verfügung, können bei der Template Engine jedoch nur über die [Tcl-Schnittstelle](#) konfiguriert werden.

Es gibt System-Jobs und Benutzer-Jobs. System-Jobs sind vordefinierte Routinen, etwa zur regelmäßigen Übertragung von Update-Records zur Template Engine im Rahmen des inkrementellen Live-Server-Updates. Sie können weder gelöscht noch geändert werden. Benutzer-Jobs dagegen können bedarfsweise angelegt, modifiziert und auch wieder gelöscht werden. Beide Job-Kategorien teilen sich einen Namensraum, d. h. zwei Jobs können auch dann nicht denselben Namen haben, wenn sie zu unterschiedlichen Kategorien gehören.

Die Jobs einer Kategorie werden zum Ausführungszeitpunkt in die Warteschlange dieser Kategorie aufgenommen, sofern sie nicht bereits aufgenommen wurden. Dadurch ist sicher gestellt, dass ein häufiger auszuführender Job nicht mehrmals in die Warteschlange gestellt wird, wenn ein anderer oder auch dieser Job gerade abgearbeitet wird.

Alle Jobs einer Kategorie werden sequenziell abgearbeitet, da sie sich eine Warteschlange teilen. Dies hat den Vorteil, dass mehrere Jobs auf einmal angestoßen werden können und sich dennoch nicht überschneiden (Initiierung und Ausführung sind asynchron). Dadurch werden Performance-Einbußen vermieden, die durch die gleichzeitige Abarbeitung vieler zeitintensiver Tcl-Skripte entstehen können.

Die Ausführung von Jobs wird protokolliert. Je Job legt das System maximal so viele Protokolleinträge an, wie es der Wert des Systemkonfigurationseintrags `jobMaxLogLength` festgelegt.

Wie Jobs in der HTML-Benutzeroberfläche definiert werden, ist im Abschnitt [Die Sektion Jobs](#) beschrieben.

3.4 Channels

Channels dienen dazu, Inhalte thematisch zu kategorisieren. Insbesondere auf Portal-Sites geben sie Besuchern die Möglichkeit, schnell auf die Nachrichten (News) zu den von ihnen bevorzugten Themen zugreifen.

Der Portal Manager und der Content Management Server unterstützen Channels. Im Content Manager lassen sich [Channels definieren](#), denen die Redakteure anschließend mit Hilfe des vorgegebenen Versionsfeldes `channels` Dateien zuordnen können. Die Dateien und die Channeldefinitionen werden im Rahmen des inkrementellen Exports zum Live-Server übertragen.

Auf dem Live-Server können automatisch News Feeds erzeugt werden, die dann beispielsweise vom Portal Manager ausgeliefert werden können. Ob eine Datei als News-Artikel betrachtet werden soll, kann mit einer Option in der Vorlage der News-Datei, also im Redaktionssystem, gesteuert werden.

Zum Zwecke der Personalisierung von Webseiten kann für jede Datei das Live-Server-Leserecht bestimmten Benutzergruppen zugewiesen werden, so dass es erforderlich ist, sich anzumelden, um einen Artikel lesen zu können.

4

4 Benutzer und Benutzergruppen verwalten



Der Content Management Server verwaltet die Logins (Anmeldenamen) mit den dazu gehörenden Passwörtern, die Benutzerfelder und Benutzergruppen. Mit ihm können Sie Benutzern und Benutzergruppen globale Rechte sowie Gruppen dateispezifische Rechte erteilen. Wie die Benutzerverwaltung des Content Management Servers organisiert ist, wird im Abschnitt Benutzerverwaltung erläutert.

Als Benutzer des Content Management Servers, der Logins oder Benutzergruppen anlegen, löschen und ändern können soll, benötigen Sie das globale Recht *Benutzer und Gruppen bearbeiten*. Um Benutzerfelder definieren zu können, ist das Recht *Benutzerfelder bearbeiten* erforderlich.

Bitte beachten Sie, dass mehrere Administratoren die gleichen Konfigurationswerte (beispielsweise die Eigenschaften einer Benutzergruppe) parallel bearbeiten und auch abspeichern können. Bei diesem Vorgang werden stets die unmittelbar vorher abgespeicherten Werte überschrieben.

Um Benutzer, Gruppen oder Benutzerfelder anzulegen, zu bearbeiten oder zu löschen, wählen Sie bitte den Punkt *Extras > Benutzerverwaltung* aus dem Menü des Content Navigators. Die Seite *Benutzerverwaltung* mit den drei Sektionen *Benutzer*, *Gruppen* und *Benutzerfelder* wird angezeigt. Sie können zu einer Sektion wechseln, indem Sie den entsprechenden Reiter anklicken:

Benutzerverwaltung - Benutzer
Benutzer anlegen, suchen und bearbeiten

Benutzer Gruppen Benutzerfelder

Suche nach Benutzern, deren Login oder vollständiger Name Folgendes enthält:

Suchen Zurücksetzen Neu

11 Benutzer, Anzeige aller Ergebnisse

Login	Vollständiger Name	E-mail	Gruppenzuordnung
chefredakteur	Chefredakteur	postmaster@localhost	chiefeditors, all, live_standard
designer	Designer	postmaster@localhost	designers, all, live_standard
redakteur	Redakteur fuer Pressemitteilung	joerg.borm@infopark.de	editors, all, live_standard
root	Superuser	postmaster@localhost	admins, all, live_premium, live_standard

In allen Sektionen der Benutzerverwaltung gibt es eine Suchfunktion, die Sie nutzen können, um Benutzer, Benutzergruppen oder Benutzerfelder im Hauptbereich der jeweiligen Sektion anzeigen zu lassen.

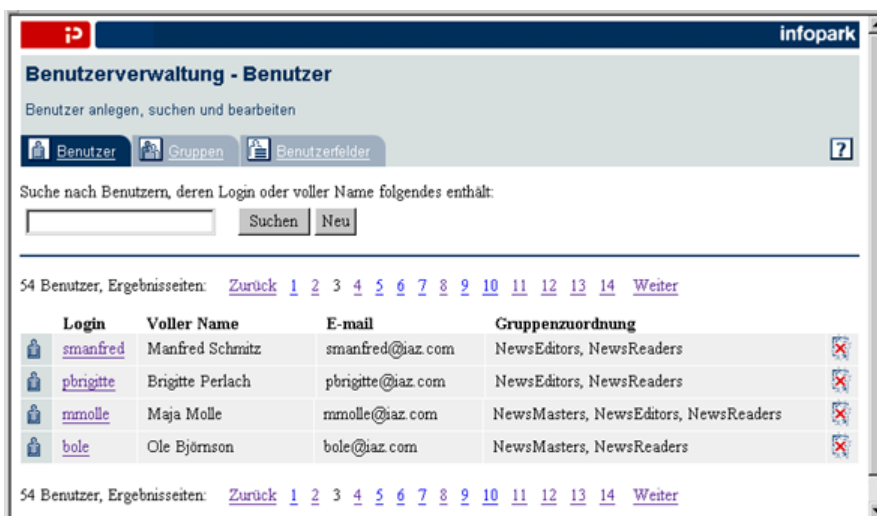
Möchten Sie eine Liste der Benutzer, Gruppen oder Benutzerfelder anzeigen lassen, deren Name oder voller Name einen Suchbegriff enthält, so tragen Sie diesen Suchbegriff bitte in das Eingabefeld der jeweiligen Sektion ein und klicken anschließend auf den Button *Suche*. Im Hauptbereich der Sektion wird daraufhin eine Liste der entsprechenden Benutzer, Gruppen oder Benutzerfelder angezeigt. Den Suchbegriff können Sie ändern, indem Sie einen anderen Suchbegriff in das Eingabefeld der jeweiligen Sektion eintragen und ein weiteres Mal auf den *Suche*-Button klicken.

Sie können die Suchfunktion auch dazu nutzen, alle im Content Management Server eingetragenen Benutzer, Gruppen oder Benutzerfelder anzeigen zu lassen. Zu diesem Zweck lassen Sie das Eingabefeld bitte leer und klicken auf den Button *Suche*.

Wie viele Benutzer, Gruppen oder Benutzerfelder die jeweilige Liste im Hauptbereich einer Sektion beinhaltet, ist von Ihren persönlichen Einstellungen abhängig. Enthält die Liste mehr Einträge als Sie eingestellt haben, so wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie die Zahl anklicken. Ferner können Sie mit *Weiter* und *Zurück* zur nächsten bzw. vorigen Ergebnisseite blättern.

4.1 Die Sektion Benutzer

Um einen Benutzer einzurichten, ihn Gruppen zuzuordnen, ihm Rechte zu erteilen und Benutzerfeldern Werte zuzuweisen, klicken Sie auf der Seite *Benutzerverwaltung* auf den Reiter *Benutzer*. Der Content Management Server zeigt daraufhin die Sektion *Benutzer* an:



Diese Seite wird auch angezeigt, wenn Sie in einem Dialog auf den *Durchsuchen* - oder *Hinzufügen*-Button klicken, um einen Benutzer auszuwählen. In diesem Fall ist der Button *Neu* nicht vorhanden, und im oberen Bereich der Seite befindet sich ein *Abbrechen*-Button, mit dem Sie die Auswahlaktion bei Bedarf abbrechen können.

Verwenden Sie bitte die Suchfunktion, um Benutzer anzeigen zu lassen. Gehen Sie dabei so vor, wie im Abschnitt *Benutzerverwaltung* beschrieben.

Im Hauptbereich der Sektion wird als Suchergebnis eine Liste derjenigen Benutzer angezeigt, deren Login oder voller Name Ihren Suchbegriff enthält. Jeder Listeneintrag besteht aus dem Login, dem vollen Namen und der E-Mail-Adresse eines Benutzers. Ferner wird angezeigt, in welchen Gruppen der Benutzer Mitglied ist.

Die Anzahl der Listeneinträge je Ergebnisseite hängt von Ihren persönlichen Einstellungen ab. Enthält das Suchergebnis mehr Einträge als auf einer Ergebnisseite aufgeführt werden können, so können Sie mit *Weiter* und *Zurück* oder durch einen Klick auf eine der Zahlen auf andere Ergebnisseiten wechseln.

Wenn diese Seite angezeigt wurde, weil Sie in einem Dialog auf einen *Durchsuchen* - oder *Hinzufügen*-Button geklickt haben, so können Sie nun einen Benutzer auswählen, indem Sie einfach auf dessen Anmeldenamen klicken.

Klicken Sie auf das Login eines aufgeführten Benutzers, so zeigt der Content Manager sämtliche über ihn verfügbaren Informationen an:

The screenshot shows a web interface for user management. At the top, there's a header with the 'infopark' logo. Below it, the title 'Benutzerverwaltung - Benutzer - Anzeigen' is displayed, followed by the subtitle 'Eigenschaften des Benutzers "Redakteur"'. There are three tabs: 'Benutzer' (selected), 'Gruppen', and 'Benutzerfelder'. Below the tabs are three buttons: 'Neu', 'Bearbeiten', and 'Löschen'. Further down, there are four buttons: 'Kennwort ändern', 'Persönliche Einstellungen bearbeiten', 'Lokale Anwendungen festlegen', and 'Ansicht und Layout festlegen'. The main content area displays the following user information:

- Login:** redakteur
- Vollständiger Name:** Redakteur
- E-mail:** redakteur@mail-example.net password demo
- Standardgruppe:** editors
- Gruppenzuordnung:** editors, all, live_standard
- Verwalter:** root
- Gesperrt:** Nein
- Globale Rechte:**

Auf der Ansichtsseite des Benutzers sind untereinander die vorgegebenen Felder, die Gruppenzuordnung, die globalen Rechte und – falls vorhanden – die kundenspezifischen Felder zu sehen.

Sie können von der Ansichtsseite aus auch die persönlichen Einstellungen des angezeigten Benutzers ändern (das Kennwort, die Grundeinstellungen, die lokalen Anwendungen sowie die Content-Navigator-Einstellungen *Ansicht und Layout*), indem Sie auf die entsprechenden Buttons klicken. Die Aufgaben der Buttons *Neu*, *Bearbeiten* und *Löschen* werden in den folgenden Abschnitten beschrieben.

4.1.1 Benutzer einrichten

Um einen neuen Benutzer einzurichten, klicken Sie bitte entweder auf der Ansichtsseite eines Benutzers oder in der Sektion *Benutzer* auf den Button *Neu*. Das Formular *Benutzerverwaltung - Benutzer - Einrichten* wird geöffnet:

Das Formular ist in die Bereiche *Allgemeine Eigenschaften*, *Gruppenzuordnung*, *Globale Rechte* und *Benutzerfelder* unterteilt, die im Folgenden beschrieben werden.

Allgemeine Eigenschaften

Im obersten Bereich des Formulars legen Sie die allgemeinen Eigenschaften des neuen Benutzers fest:

Login

Geben Sie bitte in das Feld *Login* den künftigen Anmeldenamen des Benutzers ein. Anmeldenamen und die Namen der Benutzergruppen gehören zum gleichen Namensraum; ein Name, der als Anmeldenamen verwendet wird, kann also später nicht mehr als Gruppenname vergeben werden (und umgekehrt).

Voller Name

Tragen Sie in dieses Feld bitte den vollen Namen des Benutzers ein.

E-Mail

Hier können Sie die E-Mail-Adresse des Benutzers eintragen.

Verwalter

Wenn Sie einen neuen Benutzer anlegen, sind Sie zunächst sein Verwalter. Das bedeutet, nur Sie können die Benutzerdaten ändern, also beispielsweise die Werte von Benutzerfeldern setzen oder den Zugang des Benutzers zum Content Management Server sperren. Ferner können Sie den Benutzer in Benutzergruppen aufnehmen, die Sie selbst angelegt haben.

Sowohl einzelne Benutzer als auch Benutzergruppen können der Verwalter eines Benutzers sein. Sie können den Verwalter des Benutzers ändern, indem Sie das Login oder den Gruppennamen in das Feld *Verwalter* eintragen. Alternativ können Sie auf den Button *Durchsuchen* klicken, um auf die [Auswahlseite für Benutzer](#) und [Gruppen](#) zu wechseln. Sie können die Suche von vornherein beschränken, indem Sie einen Teil eines Benutzernamens in das Eingabefeld eingeben und anschließend auf den *Durchsuchen*-Button klicken. Sie können jederzeit auf der Auswahlseite eine neue Suche auslösen, indem Sie den Suchbegriff in das Eingabefeld eingeben und auf *Suchen* klicken. Wählen Sie anschließend aus der erscheinenden Liste einen Benutzer oder eine Gruppe aus, indem Sie auf sein Login bzw. ihren Namen klicken. Der neue Verwalter wird in das Eingabefeld übernommen. Sie können die Suche auch abbrechen, indem Sie auf den *Abbrechen*-Button klicken.

Beachten Sie bitte, dass der neue Verwalter - ob einzelner Benutzer oder Benutzergruppe - das globale Recht zur Benutzerverwaltung benötigt.

Gesperrt

Es ist möglich, einem Benutzer den Zugriff auf den Content Management Server zu verwehren, wenn dies erforderlich sein sollte. Schalten Sie in diesem Fall die Option *Gesperrt* ein.

Gruppenzuordnung

In diesem Bereich des Formulars nehmen Sie den Benutzer in Benutzergruppen auf:

The screenshot shows a web form titled "Gruppenzuordnung". It has two main sections. The first section is labeled "Standardgruppe:" and contains a text input field followed by a "Durchsuchen" button. The second section is labeled "Gruppen, in denen der Benutzer Mitglied ist" and contains a list box with a scroll bar, and two buttons: "Hinzufügen" and "Entfernen".

Standardgruppe

Legen Sie hier bitte die Standardgruppe des neuen Benutzers fest, indem Sie den Namen der Gruppe in das Feld *Standardgruppe* eintragen.

Alternativ können Sie einen Suchbegriff in das Eingabefeld eintragen und auf den *Durchsuchen*-Button klicken, um nach der gewünschten Gruppe zu suchen. Anschließend werden alle Benutzergruppen, deren Name oder voller Name den Suchbegriff enthält, auf der Auswahlseite für Gruppen angezeigt. Klicken Sie auf den Button *Durchsuchen*, ohne etwas in das Eingabefeld einzutragen, so wird die Auswahlseite für Gruppen geöffnet, in der Sie die Suchfunktion (siehe [Benutzerverwaltung](#)) verwenden können, um die gewünschte Gruppe zu ermitteln. Klicken Sie abschließend im entsprechenden Listeneintrag auf den Namen der Gruppe, um ihn in das Feld *Standardgruppe* zu übernehmen. Falls Sie den Vorgang abbrechen möchten, so klicken Sie bitte auf den Button *Abbrechen*.

Nachdem Sie die Einstellungen mit *OK* bestätigt haben, wird die Standardgruppe des Benutzers automatisch zur Liste der Gruppen hinzugefügt, in denen der Benutzer Mitglied ist (falls sie dort nicht bereits enthalten ist).

Über die Standardgruppe erteilt der Content Management Server einem Benutzer dateispezifische Rechte, wenn er eine Datei angelegt hat, jedoch nicht lesend oder schreibend darauf zugreifen darf. Dies ist der Fall, wenn der Benutzer aufgrund seiner Gruppenzugehörigkeit zwar in dem darüber liegenden Ordner Dateien anlegen darf, er jedoch in keiner Gruppe mit Lese- oder Schreibrecht Mitglied ist (die neue Datei erbt die Rechte ihres darüber liegenden Ordners).

Benutzer zu Gruppen zuordnen

Ein neuer Benutzer ist zunächst in keiner Gruppe Mitglied. Um den neuen Benutzer in Benutzergruppen aufzunehmen, klicken Sie bitte auf den *Hinzufügen*-Button neben der Liste *Gruppen, in denen der Benutzer Mitglied* ist. Dies öffnet eine Suchseite, auf der Sie die gewünschte Gruppe suchen und auswählen können, indem Sie auf *Suchen* und anschließend im angezeigten Suchergebnis auf den Namen der Gruppe klicken. Die Gruppe wird daraufhin in die Liste *Gruppen, in denen der Benutzer Mitglied* aufgenommen. Wiederholen Sie diesen Vorgang bitte für alle Gruppen, in die der Benutzer aufgenommen werden soll.

Um Benutzer aus Gruppen zu entfernen, markieren Sie bitte die Gruppen in der Liste und klicken auf den *Entfernen*-Button. Die markierten Gruppen erscheinen anschließend nicht mehr in der Liste. Der Content Management Server erlaubt Ihnen jedoch nicht, den Benutzer aus der Benutzergruppe zu entfernen, die als seine Standardgruppe festgelegt wurde.

Globale Rechte

In diesem Formularbereich weisen Sie dem Benutzer globale Rechte zu:

Sie können dem neuen Benutzer Rechte erteilen, indem Sie die entsprechenden Rechte aus der Liste *Verfügbare Rechte* auswählen und anschließend auf den nach rechts weisenden Pfeil klicken. Die Bedeutung der in der Liste aufgeführten globalen Rechte wird im Abschnitt [Globale Rechte](#) erläutert. Alle Rechte, die dem Benutzer zugewiesen wurden, werden in der Liste *Benutzerrechte* angezeigt.

Um dem Benutzer Rechte wieder zu entziehen, wählen Sie die entsprechenden Rechte bitte in der Liste *Benutzerrechte* aus und klicken auf den nach links weisenden Pfeil. Die aus der Liste entfernten Rechte werden wieder in die Liste *Verfügbare Rechte* aufgenommen.

Zusätzlich zu den Rechten, die im Formularbereich *Globale Rechte* aufgeführt sind, können weitere Rechte in der Konfigurationsdatei des Content Management Servers *cm.xml* definiert sein. Diese Rechte erscheinen ebenfalls im Auswahlfeld *Verfügbare Rechte*. Sie spielen insbesondere bei Dateivorlagen eine Rolle: in jeder Dateivorlage kann ein globales Recht festgelegt werden, das ein Benutzer haben muss, um Dateien auf der Basis dieser [Vorlage anlegen](#) zu dürfen.

Wenn Sie einzelnen Benutzern globale Rechte erteilen, so bedenken Sie bitte, dass es dadurch aufwändiger wird, die Benutzer zu verwalten. Es ist empfehlenswert, Benutzern die Rechte über Benutzergruppen zu erteilen.

Benutzerfelder

In diesem Formularbereich können Sie die Benutzerfelder mit Werten belegen:

Bei Bedarf können in der Sektion *Benutzerfelder* weitere Felder definiert werden.

Bestätigen Sie bitte abschließend die Angaben für den neuen Benutzer mit *OK*. Der Content Management Server zeigt daraufhin die Ansichtsseite des neuen Benutzers an.

4.1.2 Persönliche Einstellungen des Benutzers bearbeiten

Bei Bedarf können Sie die [persönlichen Einstellungen](#) eines Benutzers, beispielsweise das Kennwort, konfigurieren. Dies ist jedoch nur dann möglich, wenn Sie der Verwalter des Benutzers oder ein Superuser sind.

Um die persönlichen Einstellungen eines Benutzers zu ändern, ermitteln Sie den gewünschten Benutzer bitte zunächst in der Sektion *Benutzer* mit Hilfe der Suchfunktion. Klicken Sie anschließend auf das Login im entsprechenden Listeneintrag, um zur Ansichtsseite des Benutzers zu wechseln. Hier haben Sie die Möglichkeit, das Kennwort, die Grundeinstellungen, die lokalen Anwendungen sowie die Content-Navigator-Einstellungen des Benutzers zu bearbeiten, indem Sie auf den entsprechenden Button klicken.

4.1.3 Benutzerdaten ändern und Benutzer löschen

Um Benutzerdaten zu ändern, machen Sie bitte zunächst den Benutzer in der Sektion *Benutzer* mit Hilfe der Suchfunktion ausfindig. Klicken Sie nun in dem Listeneintrag des Benutzers auf das Login, so zeigt der Content Management Server zunächst die Ansichtsseite des gewünschten Benutzers an. Hier klicken Sie bitte auf den Button *Bearbeiten*, um das Formular *Benutzerverwaltung - Benutzer - Bearbeiten* zu öffnen:

In diesem Formular erhalten Sie sämtliche über diesen Benutzer verfügbaren Informationen. Das Formular enthält die gleichen Daten wie das Formular, mit dem ein neuer Benutzer angelegt wird. Das Login kann jedoch nicht geändert werden.

Klicken Sie auf *OK*, um die geänderten Einstellungen für den Benutzer zu übernehmen. Der Content Management Server zeigt nun wieder die Ansichtsseite des Benutzers an.

Möchten Sie einen Benutzer endgültig aus dem CMS entfernen, so lokalisieren Sie ihn bitte mit Hilfe der Suchfunktion in der Sektion *Benutzer* und klicken anschließend auf den Löschen-Button am rechten Rand des Listeneintrags. Der Benutzer kann ebenfalls auf seiner Ansichtsseite gelöscht werden. Zu diesem Zweck klicken Sie im Suchergebnis auf das *Login* des gewünschten Benutzers und auf der Ansichtsseite wiederum auf den Button *Löschen*. Das CMS bittet Sie darum, den Löschvorgang zu bestätigen:



Klicken Sie auf *OK*, so wird der Benutzer gelöscht. Der Löschvorgang kann nicht rückgängig gemacht werden.

4.2 Die Sektion Gruppen

Die Benutzerverwaltung von Infopark CMS Fiona erlaubt es, Benutzergruppen einrichten, ihnen Rechte erteilen und Benutzer in die Gruppen aufnehmen. Benutzer können Mitglied mehrerer Gruppen sein.

Um Benutzergruppen einzurichten, zu ändern oder zu löschen, wechseln Sie bitte zur Sektion *Gruppen*, indem Sie auf den entsprechenden Reiter klicken.



Diese Seite wird auch angezeigt, wenn Sie in einem Dialog auf den Durchsuchen- oder Hinzufügen-Button klicken, um eine Benutzergruppe auszuwählen. In diesem Fall ist der Button *Neu* nicht

vorhanden, und im oberen Bereich der Seite befindet sich ein Abbrechen-Button, mit dem Sie die Auswahlaktion bei Bedarf abbrechen können.

Um Gruppen anzeigen zu lassen, können Sie die Suchfunktion dieser Sektion nutzen. Gehen Sie dabei bitte so vor, wie im Abschnitt [Benutzerverwaltung](#) beschrieben.

Im Hauptbereich der Sektion wird als Suchergebnis eine Liste der Benutzergruppen, deren Name oder voller Name Ihren Suchtext enthält, oder eine Liste sämtlicher Benutzergruppen angezeigt. Die Listeneinträge enthalten jeweils den Namen und den vollen Namen einer Gruppe.

Enthält die Liste der Gruppen mehr Einträge als Sie eingestellt haben, so wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie die Zahl anklicken. Ferner können Sie mit *Weiter* und *Zurück* blättern.

Wenn diese Seite angezeigt wurde, weil Sie in einem Dialog auf einen *Durchsuchen*- oder *Hinzufügen*-Button geklickt haben, so können Sie nun eine Benutzergruppe auswählen, indem Sie einfach auf ihren Namen klicken.

Klicken Sie auf den Namen einer aufgeführten Gruppe, so zeigt der Content Manager sämtliche über sie verfügbaren Informationen an:



Auf der Ansichtsseite der Gruppe sind untereinander der Name und der vollständige Name sowie die Gruppenmitglieder, der Verwalter und die zugewiesenen Rechte der Gruppe zu sehen.

4.2.1 Gruppen einrichten

Klicken Sie bitte in der Sektion *Gruppen* oder auf der Ansichtsseite einer Gruppe auf den Button *Neu*, um eine Benutzergruppe anzulegen. Es erscheint das Formular *Benutzerverwaltung - Gruppen - Einrichten* :

Das Formular besteht aus mehreren Bereichen, die im Folgenden beschrieben werden.

Allgemeine Eigenschaften

In diesem Formularbereich legen Sie die allgemeinen Eigenschaften der Benutzergruppe fest:

Name

Geben Sie in das Feld *Name* bitte einen eindeutigen Bezeichner für die neue Gruppe ein. Der Name einer Gruppe kann nachträglich nicht geändert werden.

Bitte beachten Sie, dass Gruppennamen und die Anmeldenamen der Benutzer zum selben Namensraum gehören. Ein Name, der als Name einer Gruppe verwendet wird, kann also nicht mehr als Login eines Benutzers vergeben werden (und umgekehrt).

Voller Name

Tragen Sie in das Feld *Voller Name* bitte eine aussagekräftige Beschreibung der Gruppe ein.

Verwalter

Wie auch Benutzer haben Gruppen einen Verwalter, der die Gruppe administrieren darf. Zunächst sind Sie (als derjenige, der die Gruppe anlegt) der Verwalter der Gruppe. Sie können jedoch im Feld *Verwalter* sowohl einen Benutzer als auch eine Benutzergruppe als zukünftigen Verwalter der Gruppe einsetzen. Tragen Sie zu diesem Zweck in das Eingabefeld bitte den Anmelde- oder Gruppennamen des neuen Verwalters ein.

Alternativ können Sie auf den Button *Durchsuchen* klicken, um auf die Auswahlseite für Benutzer und Gruppen zu wechseln. Sie können die Suche von vornherein beschränken, indem Sie einen Teil eines Benutzernamens in das Eingabefeld eingeben und anschließend auf den *Durchsuchen*-Button klicken.

Sie können jederzeit auf der [Auswahlseite für Benutzer](#) oder [Gruppen](#) eine neue Suche auslösen, um den gewünschten Verwalter zu ermitteln. Klicken Sie abschließend im entsprechenden Listeneintrag des Suchergebnisses auf das Login oder den Namen, um den neuen Verwalter in das Eingabefeld zu übernehmen.

Möchten Sie einen Verwalter angeben, so sollten Sie prüfen, ob dieser das globale Recht hat, Benutzer und Gruppen zu verwalten, und ihm gegebenenfalls dieses Recht erteilen.

Gruppenmitglieder

In diesem Bereich des Formulars können Sie festlegen, welche Benutzer in die neue Gruppe aufgenommen werden sollen:

Eine neue Benutzergruppe hat zunächst keine Mitglieder. Sie können einen Benutzer in die neue Gruppe aufnehmen, indem Sie auf den Button *Hinzufügen* klicken. Dies öffnet die Benutzerauswahlseite, in der Sie die Suchfunktion verwenden können, um den gewünschten Benutzer zu ermitteln. Klicken Sie im Suchergebnis auf das Login eines Benutzers, so übernimmt der Content Manager das Login in die Liste *Gruppenmitglieder*. Wiederholen Sie diesen Vorgang bitte für alle Benutzer, die in die neue Gruppe aufgenommen werden sollen.

Um Benutzer wieder aus der Gruppe zu entfernen, wählen Sie sie aus der Liste *Gruppenmitglieder* aus und klicken auf den *Entfernen*-Button. Die ausgewählten Benutzer erscheinen anschließend nicht mehr in der Liste.

Globale Rechte

Im untersten Formularbereich weisen Sie der Gruppe globale Rechte zu:

Sie können einer Gruppe Rechte erteilen, indem Sie die entsprechenden Rechte aus der Liste *Verfügbare globale Rechte* auswählen und anschließend auf den nach rechts weisenden Pfeil klicken. Die Rechte, die der Gruppe zugewiesen wurden, werden in der Liste *Gruppenrechte* angezeigt.

Um der Gruppe Rechte wieder zu entziehen, wählen Sie die entsprechenden Rechte bitte aus der Liste *Gruppenrechte* aus und klicken auf den nach links weisenden Pfeil. Die aus der Liste entfernten Rechte werden wieder in die Liste *Verfügbare globale Rechte* aufgenommen.

Unter Umständen sind in der Liste *Verfügbare globale Rechte* weitere globale Rechte aufgeführt, die Sie auswählen können. Wie zusätzliche Rechte im Content Management Server eingesetzt werden können, ist im Abschnitt [Globale Rechte](#) beschrieben.

Bestätigen Sie bitte abschließend die Angaben für die neue Gruppe mit *OK*. Der Content Manager zeigt daraufhin die Ansichtsseite der neuen Gruppe an.

4.2.2 Gruppeneinstellungen ändern und Gruppen löschen

Möchten Sie Gruppeneinstellungen ändern, so machen Sie bitte zunächst die gewünschte Gruppe in der Sektion *Gruppen* mit Hilfe der Suchfunktion ausfindig. Klicken Sie nun in dem Listeneintrag der Gruppe auf den Namen, so zeigt der Content Management Server zunächst die Ansichtsseite der gewünschten Gruppe an. Klicken Sie hier bitte auf den Button *Bearbeiten*, um das Formular *Benutzerverwaltung - Gruppen - Bearbeiten* zu öffnen:

Benutzerverwaltung - Gruppen - Bearbeiten

Eigenschaften der Gruppe "NewsEditors" bearbeiten

Allgemeine Eigenschaften

Name: NewsEditors

Voller Name: Redakteure Nachrichtenressort

Verwalter:

In diesem Formular können Sie alle Eigenschaften der gewählten Gruppe (bis auf den Gruppennamen) ändern. Das Formular entspricht dem Formular, mit dem eine Gruppe angelegt werden kann.

Klicken Sie auf *OK*, um die geänderten Einstellungen für die Gruppe zu speichern. Der Content Management Server zeigt daraufhin die Ansichtsseite der Gruppe an.

Um eine Gruppe zu löschen, machen Sie bitte die entsprechende Benutzergruppe in der Sektion *Gruppen* mit Hilfe der Suchfunktion ausfindig und klicken auf den Löschen-Button am rechten Rand des Listeneintrags. Die Gruppe kann auch auf ihrer Ansichtsseite gelöscht werden. Zu diesem Zweck klicken Sie bitte im Suchergebnis auf den Namen der gewünschten Gruppe und auf ihrer Ansichtsseite wiederum auf den Button *Löschen*. Der Content Management Server bittet Sie nun, den Löschvorgang zu bestätigen:

Benutzerverwaltung - Gruppen - Löschen

Gruppe "Redakteure Nachrichtenressort" löschen

Wollen Sie diese Gruppe wirklich löschen?

Name: NewsEditors

Voller Name: Redakteure Nachrichtenressort

Ok Abbrechen

Klicken Sie auf *OK*, so wird die Gruppe endgültig gelöscht, sofern sie nicht die Standardgruppe eines Benutzers oder mehrerer Benutzer ist. In diesem Fall können Sie die Gruppe erst löschen, nachdem Sie dem Benutzer oder den Benutzern eine andere Standardgruppe zugewiesen haben.

4.3 Die Sektion Benutzerfelder

Benutzerfelder dienen dazu, die vom Content Management Server gespeicherten Benutzerdaten um zusätzliche Informationen wie eine Telefonnummer oder die Bezeichnung der Abteilung, in der der Benutzer tätig ist, zu ergänzen.

Klicken Sie auf der Seite *Benutzerverwaltung* auf den Reiter *Benutzerfelder*, wenn Sie Benutzerfelder anlegen, ändern oder löschen möchten. Der Content Management Server zeigt daraufhin die Sektion *Benutzerfelder* an:



Um Benutzerfelder anzeigen zu lassen, nutzen Sie bitte die Suchfunktion in der Sektion *Benutzerfelder*. Ihre Funktionsweise wird im Abschnitt [Benutzerverwaltung](#) beschrieben.

Im Hauptbereich der Sektion wird als Suchergebnis eine Ihrem Suchkriterium entsprechende Liste angezeigt, die je nach Suchtext alle Benutzerfelder oder nur diejenigen umfasst, deren Name den Suchtext enthält. Die Listeneinträge enthalten jeweils den Namen und den Typ eines Benutzerfeldes.

Es ist von Ihren persönlichen Einstellungen (siehe Abschnitt [Persönliche Einstellungen des Benutzers bearbeiten](#)) abhängig, wie viele Benutzerfelder die Liste im Hauptbereich der Sektion enthält. Sie können mit *Weiter* und *Zurück* blättern, sofern die Liste mehr Einträge beinhaltet, als Sie eingestellt haben. Ferner können Sie auf andere Ergebnisseiten wechseln, indem Sie auf eine der Zahlen in der aufsteigenden Zahlenfolge ober- und unterhalb der Liste klicken.

Klicken Sie auf den Namen eines aufgeführten Benutzerfeldes, so zeigt der Content Manager sämtliche über das Benutzerfeld verfügbaren Informationen an:



Auf der Ansichtseite des Benutzerfeldes sind untereinander der Name und der Typ zu sehen. Bei Benutzerfeldern vom Typ *Auswahl* oder *Mehrfachauswahl* werden ferner die erlaubten Werte des Benutzerfeldes angezeigt.

4.3.1 Benutzerfelder anlegen

Um ein Benutzerfeld anzulegen, klicken Sie bitte entweder auf der Ansichtseite eines Benutzerfeldes oder in der Sektion *Benutzerfelder* auf den Button *Neu*. Das Formular *Benutzerverwaltung - Benutzerfelder - Anlegen* wird geöffnet:

Name

Geben Sie bitte in das Eingabefeld *Name* den Namen des Benutzerfeldes ein. Verwenden Sie als Name nicht login.

Typ

Wählen Sie aus dem Auswahlménü *Typ* bitte den Typ des Feldes aus. Als Feldtypen stehen Ihnen *Auswahl*, *Datum*, *Mehrfachauswahl* und *Zeichenkette* zur Verfügung.

Klicken Sie auf *OK*, um das Feld anzulegen. Der Content Manager zeigt daraufhin die Ansichtsseite des Benutzerfeldes an.

Um die erlaubten Werte eines Feldes vom Typ *Auswahl* oder *Mehrfachauswahl* festzulegen, öffnen Sie bitte das Formular *Benutzerverwaltung - Benutzerfelder - Bearbeiten*, indem Sie auf den Button *Bearbeiten* klicken:

Tragen Sie in das mehrzeilige Eingabefeld *Werte* bitte alle Werte ein, die das Feld annehmen kann und klicken Sie auf *OK*, um die Eingabe abzuschließen. Beachten Sie bitte, dass Sie in jeder Zeile des Eingabefeldes nur einen Wert angeben können.

Ein Benutzerfeld vom Typ *Auswahl* oder *Mehrfachauswahl* ist nun vollständig definiert, und der Content Management Server zeigt wieder die Ansichtsseite des neuen Benutzerfeldes an.

4.3.2 Benutzerfelder ändern oder löschen

Sie können bei einem Benutzerfeld vom Typ *Auswahl* oder *Mehrfachauswahl* nachträglich die verfügbaren Feldwerte ändern. Machen Sie zu diesem Zweck das gewünschte Feld in der Sektion

Benutzerfelder mit Hilfe der Suchfunktion ausfindig und klicken Sie bitte im entsprechenden Listeneintrag auf seinen Namen. Der Content Management Server zeigt zunächst die Ansichtsseite des Benutzerfeldes an. Klicken Sie bitte auf den Button *Bearbeiten*, um das Formular *Benutzerverwaltung - Benutzerfelder - Bearbeiten* zu öffnen:

Hier können Sie die möglichen Werte des Feldes ändern, indem Sie die neuen Werte nacheinander in das Eingabefeld *Werte* eintragen. Bei Bedarf können Sie natürlich auch Werte löschen.

Klicken Sie auf *OK*, um die geänderten Feldwerte zu übernehmen und zu der Ansichtsseite des Benutzerfeldes zurückzukehren.

Möchten Sie ein Benutzerfeld löschen, so verwenden Sie bitte die Suchfunktion, um es in der Liste der Suchergebnisse im Hauptbereich der Sektion *Benutzerfelder* anzeigen zu lassen. Klicken Sie anschließend auf den Löschen-Button am rechten Rand des Listeneintrags oder wechseln Sie mit einem Klick auf den Namen auf die Ansichtsseite des gewünschten Benutzerfeldes und klicken dort auf den Button *Löschen*. Der Content Management Server bittet Sie darum, den Löschvorgang zu bestätigen:

Das Benutzerfeld wird endgültig gelöscht, wenn Sie auf *OK* klicken.

5

5 Den Content Manager konfigurieren

Um Dateivorlagen, Felder, Workflows, Jobs oder Channels anzulegen, wählen Sie bitte im Content-Navigator-Menü den Punkt *Extras > Systemeinstellungen*. Die Seite *Konfiguration* mit den Sektionen *Vorlagen*, *Felder*, *Workflows*, *Jobs* und *Channels* wird geöffnet. Zu einer Sektion wechseln Sie, indem Sie den entsprechenden Reiter anklicken.



Vorlagen, Felder, Workflows und Channels sind Bestandteil der sogenannten *Runtime Configuration*, deren Werte ein Benutzer nur bearbeiten kann, wenn er das globale Recht *Globale Einstellungen bearbeiten* hat. Jobs dagegen kann man nur als Benutzer mit globalem Administrationsrecht anlegen, bearbeiten und löschen.

5.1 Die Sektion Vorlagen

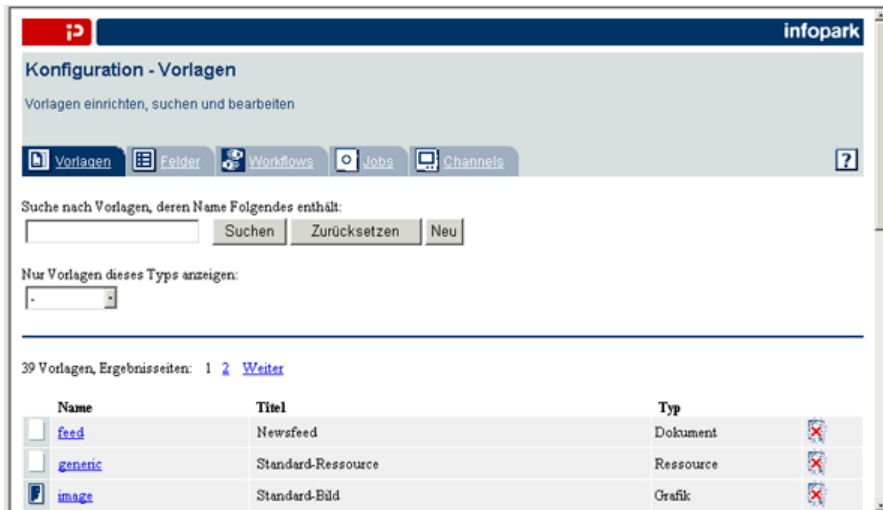
Dateien werden auf der Grundlage von Dateivorlagen (kurz *Vorlagen*) erzeugt. Eine Vorlage definiert unter anderem die folgenden Eigenschaften der Dateien, die auf ihr basieren:

- Dateityp und Dateiendung;
- Zusätzliche (kundenspezifische) Felder sowie Feldergruppen, mit denen Felder zusammengefasst werden können;
- Ein globales Anlegerecht - nur Benutzer, die dieses Recht haben, dürfen Dateien anlegen, die auf dieser Vorlage beruhen;
- Erlaubte Vorlagen für Dateien in Ordnern;
- Den Namen eines Haupttextlayouts - eine Layoutdatei, die immer dann abgearbeitet wird, wenn der Hauptinhalt einer Version einer auf der Vorlage beruhenden Datei exportiert oder in der Vorschau angezeigt wird;
- Mehrere Funktionen; eine Funktion ist Tcl-Code, der vom Content Management Server ausgeführt wird, wenn ein applikationsspezifisches Ereignis eintritt, beispielsweise wenn eine Version bearbeitet wurde;

Vorlagen definieren also gewissermaßen Vorschriften, die vom Content Management Server beachtet werden müssen, wenn Sie eine neue Datei anlegen oder Inhalte einer Datei (eine Version) modifizieren oder exportieren.

Um Vorlagen definieren zu können, benötigen Sie das Recht, globale Einstellungen zu bearbeiten.

Möchten Sie eine Vorlage anlegen, ändern oder löschen, so wechseln Sie auf der Seite *Konfiguration* in die Sektion *Vorlagen*, indem Sie auf den entsprechenden Reiter klicken.



Diese Seite wird auch angezeigt, wenn Sie in einem Dialog auf den *Durchsuchen* - oder *Hinzufügen*-Button klicken, um eine Vorlage auszuwählen. In diesem Fall ist der Button *Neu* nicht vorhanden und im oberen Bereich der Seite befindet sich ein *Abbrechen*-Button, mit dem Sie die Auswahlaktion bei Bedarf abbrechen können.

Um im Hauptbereich der Sektion Vorlagen anzeigen zu lassen, verwenden Sie bitte die Suchfunktion.

Möchten Sie nur die Vorlagen anzeigen lassen, deren Name einen bestimmten Suchtext enthält, so tragen Sie den Suchtext bitte in das Feld *Suche nach Vorlagen, deren Name folgendes enthält* ein und klicken anschließend auf den Button *Suche*. Im Hauptbereich der Sektion wird daraufhin eine Liste der Vorlagen angezeigt, deren Name den Suchtext enthält. Um den Suchtext zu ändern, tragen Sie den neuen Suchtext in das Feld *Suche nach Vorlagen, deren Name folgendes enthält* ein und klicken ein weiteres Mal auf den *Suche*-Button. Möchten Sie alle vorhandenen Vorlagen anzeigen lassen, so lassen Sie das Eingabefeld bitte leer und klicken auf den Button *Suche*.

Sie können die Suche ferner verfeinern, indem Sie den Dateityptyp als zusätzliches Suchkriterium angeben. Wählen Sie zu diesem Zweck einen der möglichen Dateitypen in dem Aufklappenmenü *Nur Dateitypen dieses Typs anzeigen* aus.

Die Listeneinträge des Suchergebnisses enthalten jeweils den Namen, den Titel und den Dateityp einer Vorlage.

Es ist von Ihren persönlichen Einstellungen abhängig, wie viele Vorlagen im Hauptbereich der Sektion angezeigt werden. Enthält die Liste mehr Einträge als Sie eingestellt haben, so wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie die Zahl anklicken. Ferner können Sie mit *Weiter* und *Zurück* blättern.

Wenn diese Seite angezeigt wurde, weil Sie in einem Dialog auf einen *Durchsuchen* - oder *Hinzufügen*-Button geklickt haben, so können Sie nun eine Vorlage auswählen, indem Sie einfach auf ihren Namen klicken.

Klicken Sie auf den Namen einer aufgeführten Vorlage, so zeigt der Content Management Server auf der Ansichtsseite der Vorlage sämtliche darüber verfügbaren Informationen an:

5.1.1 Vorlagen definieren

Klicken Sie bitte entweder auf der Ansichtsseite einer Dateivorlage oder in der Sektion *Vorlagen* auf den Button *Neu*, um eine neue Vorlage anzulegen. Das Formular *Konfiguration - Vorlagen - Anlegen* erscheint:

Hier können Sie die Eigenschaften der neuen Vorlage festlegen. Ob die Dateien mit dieser Vorlage voreingestellt als News auf dem Live-Server bereitgestellt werden sollen, wenn sie eingereicht oder freigegeben werden, können Sie nur einstellen, indem Sie die Vorlage bearbeiten (siehe auch [Vorlagen ändern und löschen](#)), nachdem es angelegt wurde.

Das Formular ist in mehrere Bereiche unterteilt, die im Folgenden beschrieben werden.

Allgemeine Eigenschaften

Im obersten Bereich des Formulars legen Sie die allgemeinen Eigenschaften der Vorlage fest.

Allgemeine Eigenschaften

☐ Aktiv

Name:

Der Name kann später nicht mehr geändert werden!

Titel:

Andere Sprachen

Aktiv

Sie bestimmen mit dieser Option, ob die Benutzer Dateien anlegen dürfen, die auf dieser Vorlage basieren. Nutzen Sie diese Option, wenn eine Dateivorlage verfügbar sein soll und deaktivieren Sie sie, wenn die Vorlage nicht mehr verwendet werden soll, jedoch nicht gelöscht werden darf, weil es noch Dateien gibt, die auf ihr basieren.

Name

Tragen Sie in dieses Eingabefeld bitte den Namen der neuen Vorlage ein. Der Name kann nachträglich nicht geändert werden.

Titel

Geben Sie hier den Titel der Vorlage in der Sprache an, die Sie in Ihren persönlichen Einstellungen konfiguriert haben. Sie können den Titel der Vorlage auch in weiteren Sprachen angeben, indem Sie über den Button *Andere Sprachen* zum Formular *Konfiguration - Vorlagen - Mehrsprachige Titel* wechseln.

infopark

Konfiguration - Vorlagen - Mehrsprachige Titel

[Titel in anderen Sprachen definieren](#) ?

Deutscher Titel:

Englischer Titel:

Spanischer Titel:

Französischer Titel:

Italienischer Titel:

Ok Abbrechen

Tragen Sie dort in die entsprechenden Felder die Titel in der jeweiligen Landessprache ein, und bestätigen Sie die Eingabe abschließend mit *OK*.

Ist kein Titel in der Sprache verfügbar, die der Benutzer in seinen persönlichen Einstellungen festgelegt hat, so zeigt der Content Management Server den Namen der Vorlage an. Die Sprachen (und damit die festlegbaren sprachspezifischen Titel) sind installationsabhängig.

Dateityp

In diesem Bereich des Formulars legen Sie den Typ der Dateien fest, die auf der Grundlage dieser Vorlage angelegt werden.

Den Dateityp legen Sie fest, indem Sie die entsprechende Option wählen. Der gewählte Dateityp bestimmt den Typ aller Dateien, die auf dieser Vorlage basieren, und kann nachträglich nicht geändert werden.

Felder

In diesem Formularbereich können Sie die Felder der neuen Vorlage definieren und Feldeigenschaften bearbeiten, d. h. beispielsweise Felder vorab mit Werten belegen:

Felder der Vorlage

Hier wählen Sie die kundenspezifischen Versionsfelder aus, die den Arbeitsversionen aller Dateien zugewiesen werden, die auf dieser Vorlage basieren (siehe auch [Felder](#), [Feldergruppen](#) und [Feldvorbelegung](#)).

Einer neuen Vorlage sind zunächst keine Versionsfelder zugewiesen. Sie können einer Vorlage nacheinander Felder zuweisen, indem Sie auf den Button *Hinzufügen* klicken, um das Feld in der daraufhin geöffneten Auswahlseite für Felder mit Hilfe der Suchfunktion (siehe [Die Sektion Felder](#)) zu ermitteln. Der Content Manager übernimmt den Namen des gewünschten Feldes in die Felderliste, wenn Sie im entsprechenden Listeneintrag des Suchergebnisses auf den Namen des Feldes klicken. Wiederholen Sie diesen Vorgang so oft, bis Sie alle gewünschten Versionsfelder in die Liste *Felder der Vorlage* aufgenommen haben.

Möchten Sie Felder wieder aus der Liste *Felder der Vorlage* entfernen, so wählen Sie sie bitte aus der Liste aus und klicken anschließend auf den *Entfernen*-Button. Die ausgewählten Versionsfelder werden aus der Liste *Felder der Vorlage* entfernt.

Feldeigenschaften bearbeiten

Die Felder, die Sie ausgewählt haben, können Sie vorab mit Werten belegen, sodass bereits bei der Erzeugung einer Datei auf der Basis dieser Vorlage die Felder ihrer Arbeitsversion mit diesen Werten belegt sind. Ferner können Sie ein Haupttextlayout für die neue Dateivorlage angeben. Klicken Sie zu diesem Zweck auf den Button [Feldeigenschaften bearbeiten](#).

Dateianlegeoptionen und Workflow

In diesem Bereich des Formulars können Sie die Verwendung der Vorlage an ein globales Recht binden. Ferner können Sie hier für Vorlagen vom Typ *Ordner* erlaubte Vorlagen für Dateien in diesem Ordner sowie den Anlegecheck für solche Dateien definieren und einen Workflow für die späteren Dateien mit dieser Vorlage angeben:

Dateianlegeoptionen und Workflow

Globales Anlegerecht:

-

Anlegeeinschränkungen

Workflow:

☒ nicht vorbelegt

☐ vorbelegt aus übergeordnetem Ordner

☐ vorbelegt mit folgendem Wert:

Durchsuchen

Globales Anlegerecht

Hier können Sie festlegen, welches globale Recht ein Benutzer benötigt, damit er Dateien auf der Grundlage dieser Vorlage anlegen darf, indem Sie die entsprechende Option auswählen. Ist kein globales Anlegerecht angegeben, so dürfen alle Benutzer diese Vorlage verwenden, vorausgesetzt sie haben im darüber liegenden Ordner das dateispezifische Recht, Dateien anzulegen.

Anlegeeinschränkungen

Klicken Sie auf den Button [Anlegeeinschränkungen](#), um für Vorlagen mit dem Dateityp *Ordner* diejenigen Vorlagen anzugeben, die für Dateien in dem jeweiligen Ordner verwendet werden dürfen. Ferner können Sie Tcl-Code angeben, der ausgeführt werden soll, wenn eine neue Datei in dem Ordner angelegt wird.

Workflow

Hier können Sie einer Vorlage einen Workflow zuweisen. Wählen Sie in dem Auswahlfeld die Option *nicht vorbelegt* aus, so haben die Dateien, die auf dieser Vorlage beruhen, keinen Workflow. Indem Sie die Option *vorbelegt aus darüber liegendem Ordner* wählen, definieren Sie, dass Dateien, die auf der Basis dieser Vorlage erzeugt werden, denselben Workflow durchlaufen sollen, wie der jeweils darüber liegende Ordner.

Möchten Sie der Vorlage einen eigenen Workflow zuweisen, so wählen Sie bitte die Option *vorbelegt mit folgendem Wert* aus. Tragen Sie anschließend den Namen des Workflows in das Eingabefeld ein. Alternativ können Sie auf den Button *Durchsuchen* klicken, um auf die Auswahlseite für Workflows zu wechseln. Sie können die Suche von vornherein beschränken, indem Sie einen Teil eines Workflownamens in das Eingabefeld eingeben und anschließend auf den *Durchsuchen*-Button klicken. Sie können jederzeit auf der Auswahlseite eine neue Suche auslösen, indem Sie den Suchbegriff in das Eingabefeld eingeben und auf *Suchen* klicken, um den gewünschten Workflow zu ermitteln. Klicken Sie abschließend auf den Namen des Workflows im entsprechenden Listeneintrag, um ihn in das Eingabefeld zu übernehmen.

Workflows können Sie im Bereich *Konfiguration* in der Sektion [Workflows](#) anlegen und bearbeiten.

Weitere Optionen

In diesem Formularbereich sind weitere Optionen für Vorlagen zusammengefasst. Hier haben Sie die Möglichkeit, mehrere Checks und Funktionen anzugeben sowie die erlaubten Dateierweiterungen und die verfügbaren Editoren für den Hauptinhalt von Dateien mit dieser Vorlage festlegen:

Weitere Optionen

Bearbeitung und Export

Checks und Funktionen

Bearbeitung und Export

Hier können Sie die erlaubten Dateiendungen für Dateien mit der betreffenden Vorlage definieren. Ferner haben Sie an dieser Stelle die Möglichkeit, die Editoren festzulegen, mit denen der Hauptinhalt der Arbeitsversion einer Datei mit dieser Vorlage bearbeitet werden kann. Klicken Sie zu diesem Zweck auf den Button *Bearbeitung und Export*.

Checks und Funktionen

Möchten Sie für die Vorlage Checks und Funktionen definieren, so klicken Sie bitte auf den entsprechenden Button, um zu dem Formular *Konfiguration - Vorlagen - Checks und Funktionen* zu wechseln.

Bestätigen Sie abschließend Ihre Angaben für die neue Vorlage mit *OK*, um zu ihrer Ansichtsseite zu gelangen.

5.1.2 Feldeigenschaften bearbeiten

Um das Formular zur Bearbeitung der Feldeigenschaften einer Vorlage zu öffnen, haben Sie zwei Möglichkeiten: Legen Sie gerade eine Vorlage an, so klicken Sie im entsprechenden Formular auf den Button *Feldeigenschaften bearbeiten* im Bereich *Felder*. Möchten Sie dagegen die Felder einer bestehenden Vorlage vorbelegen, so machen Sie die gewünschte Vorlage in der Sektion *Vorlagen* mit Hilfe der Suchfunktion ausfindig. Klicken Sie nun auf den Namen der Vorlage im Listeneintrag, um auf die Ansichtsseite der Vorlage zu wechseln. Um das Bearbeitungsformular für Vorlagen zu öffnen, klicken Sie hier bitte auf den Button *Bearbeiten*. Wechseln Sie anschließend in den Formularbereich *Felder* und klicken auf den Button *Feldeigenschaften bearbeiten*, um in das entsprechende Formular zu gelangen:

infopark

Konfiguration - Vorlagen - Feldeigenschaften

Feldeigenschaften von "Article" bearbeiten

Name: Article

Titel: Article

Standardfelder

Titel

☐ obligatorisch

☒ nicht vorbelegt

☐ Felder, die aus dem darüber liegenden Ordner vorbelegt sind

☐ vorbelegt mit folgendem Wert:

Vom Export ausschließen

☒ nicht vorbelegt

☐ Felder, die aus dem darüber liegenden Ordner vorbelegt sind

☐ vorbelegt mit folgendem Wert:

Dateiendung

Obligatorische Felder

Sie können für jedes Feld definieren, ob es auf einen gültigen Wert gesetzt sein muss, damit eine Datei eingereicht oder freigegeben werden kann, indem Sie die Option *obligatorisch* auswählen. Sollen die Bearbeiter der Dateien, die auf diesen Vorlagen basieren, beispielsweise dem Feld *Titel* einen Wert zuweisen müssen, so wählen Sie für dieses Feld die Option *obligatorisch* aus.

Felder vom Typ *Zeichenkette*, *Text* oder *HTML* gelten als gültig, wenn vom Benutzer eine beliebige Zeichenkette als Wert eingetragen wurde. Datumsfelder gelten als gültig, wenn die eingegebenen Zahlen vom Content Management Server interpretierbar sind und sich in zulässigen Bereichen bewegen. Bei Feldern vom Typ *Auswahl* oder *Mehrfachauswahl* muss mindestens einer der vorgegebenen Werte ausgewählt worden sein, damit sie als belegt gelten.

Um die Vollständigkeit einer Version auch von anderen Kriterien abhängig zu machen, können Sie die Vorlageneigenschaft *Vollständigkeitscheck* verwenden.

Optionen für die Vorbelegung

Sie können die Felder, die in einer Vorlagendefinition angegeben werden, vorab mit Werten belegen. Dadurch sind bereits bei der Erzeugung einer Datei auf der Basis dieser Vorlage die Felder seiner Arbeitsversion mit diesen Werten belegt.

Alle kundenspezifischen Felder sowie die in der folgenden Tabelle aufgeführten vom CMS bereitgestellten Felder können vorbelegt werden. Bitte beachten Sie, dass *Bei Export ignorieren* sowie die Sortierkriterien, der -modus und die -richtung erst ab Version 6.7.1 vorbelegbar sind. Der Workflow kann nicht auf dieser sondern auf der übergeordneten Seite vorbelegt werden.

Vorbelegbares Feld	Ordner	Dokument	Bild	Ressource	Layout
Titel	•	•	•	•	•
Vom Export ausschließen / Bei Export ignorieren	•	•	•	•	
Dateiendung	•	•	•	•	
Hauptinhalt	•	•	•	•	•
Sortierkriterien, -Modi, -Richtung	•				
Channels	•	•	•	•	
Workflow	•	•	•	•	•

Mit Hilfe eines Auswahlfeldes können Sie festlegen, ob und in welcher Weise ein Feld vorab mit Werten belegt wird. Wählen Sie in dem Auswahlfeld die Option *Wert nicht vorbelegt* aus, so hat das Feld in den Arbeitsversionen neuer Dateien, die auf dieser Vorlage beruhen, keinen Wert. Das Feld ist also nicht vorbelegt.

Indem Sie bei einem Feld die Option *vorbelegt aus darüber liegendem Ordner* wählen, definieren Sie, dass bei Arbeitsversionen von Dateien, die auf der Basis dieser Vorlage werden, die Felder mit den Feldwerten aus dem darüber liegenden Ordner belegt werden sollen (die Felder erben die Werte des darüber liegenden Ordners).

Möchten Sie Felder mit konstanten Werten vorbelegen, so markieren Sie im Auswahlfeld die Option *vorbelegt mit folgendem Wert*. Tragen Sie hier bitte ein oder wählen Sie aus, welche Werte die Felder haben sollen, wenn die Arbeitsversion einer Datei, die auf dieser Vorlage basiert, erzeugt wird. Je nach Typ des Feldes stehen zur Bearbeitung des Vorbelegungswertes unterschiedliche Eingabeelemente zur Verfügung. Diese Eingabeelemente entsprechen nicht unbedingt den Eingabefeldern, die Sie bei den jeweiligen Feldern festgelegt haben und mit denen die Benutzer die Feldwerte bearbeiten können.

Besonderheiten beim Hauptinhalt

Bei dem Feld *Hauptinhalt* ist zusätzlich die Option *Inhalt mit diesem Layout formatieren* verfügbar. Wählen Sie diese Option, so geben Sie bitte in dem Eingabefeld den Namen der Layoutdatei für den Hauptinhalt an. Eine Hauptinhaltslayout ("body template") wird immer dann angewendet, wenn der Hauptinhalt einer Version (beispielsweise mit einer NPSOBJ-Anweisung bei der Vorschau oder beim Export) abgerufen wird. Ein Hauptinhaltslayout dient dazu, das Aussehen von Dokumenten zu vereinheitlichen. Bei Vorlagen, die Layouts definieren, kann kein Hauptinhaltslayout angegeben werden.

Alternativ dazu können Sie den Hauptinhalt der Version einer Datei mit Hilfe der Option *vorbelegt mit folgendem Wert* vorab belegen. Diese Option wird Ihnen jedoch erst dann angezeigt, wenn Sie eine Dateieindung angegeben haben. Die Dateieindung können Sie aus den von Ihnen definierten erlaubten Dateieindungen für die Vorlage wählen.

Nutzen Sie dieses Feature, um beispielsweise neuen Ordnern oder Dokumenten einen immer gleichen Hauptinhalt zuzuweisen. Selbstverständlich kann der Bearbeiter der Datei den Hauptinhalt wie den Wert jedes anderen Feldes jederzeit ändern, sofern in der Vorlage kein Hauptinhaltslayout angegeben wurde.

Klicken Sie abschließend auf *OK*, um die Feldvorbelegung zu übernehmen.

5.1.3 Anlegeeinschränkungen

Anlegeeinschränkungen sind vorgegebene Vorlagen für Dateien in Ordnern sowie ein Tcl-Skript, das ausgeführt wird, wenn eine Datei in einem Ordner angelegt wird. Diese Einschränkungen können nur in Vorlagen für Ordner definiert werden. Sie können beispielsweise vorgeben, dass in Ordnern, die auf der Vorlage *Presse* beruhen, nur Dateien angelegt werden dürfen, die auf den Vorlagen *Pressemitteilung* und *Kurzmitteilung* beruhen. Mit dem Anlegecheck für Dateien im *Presse*-Ordner können Sie ferner sicherstellen, dass eine Datei nur unter bestimmten Bedingungen angelegt wird, etwa wenn die Höchstzahl von Dateien in dem Ordner noch nicht erreicht wurde.

Es gibt zwei Möglichkeiten, zu dem Formular zu gelangen, in dem Sie die Anlegeeinschränkungen definieren können. Wenn Sie gerade dabei sind, eine Vorlage anzulegen, so klicken Sie im entsprechenden Formular auf den Button *Anlegeeinschränkungen* im Bereich *Dateianlegeoptionen und Workflow*. Wenn Sie dagegen in einer bestehenden Dateivorlage die Anlegeeinschränkungen definieren oder bearbeiten möchten, so lokalisieren Sie die Vorlage bitte zunächst in der Sektion *Vorlagen* mit Hilfe der Suchfunktion. Klicken Sie anschließend auf den Vorlagennamen im Listeneintrag, um die Ansichtsseite der Vorlage zu öffnen und dort wiederum auf den Button *Bearbeiten*, um zu dem Bearbeitungsformular für Vorlagen zu gelangen. Hier klicken Sie bitte im Formularbereich *Dateianlegeoptionen und Workflow* auf den Button *Anlegeeinschränkungen*, um das entsprechende Formular zu öffnen.

Erlaubte Vorlagen für darunter liegende Dateien

Bei einer Vorlage, die Dateien vom Typ *Ordner* definiert, können Sie festlegen, auf welchen Vorlagen die Dateien in diesem Ordner basieren dürfen. Sie können mehrere erlaubte Vorlagen für Dateien festlegen. Um eine Vorlage zur Liste *Erlaubte Vorlagen für darunter liegende Dateien* hinzuzufügen, klicken Sie bitte auf den *Hinzufügen*-Button. In der daraufhin geöffneten Auswahlseite für Vorlagen können Sie die [Suchfunktion](#) verwenden, um die gewünschte Vorlage zu ermitteln. Der Content Manager übernimmt den Namen der gewünschten Vorlage in die Liste, wenn Sie im entsprechenden Listeneintrag auf den Vorlagennamen klicken. Wiederholen Sie diesen Vorgang bitte für alle Vorlagen, die in die Liste aufgenommen werden sollen.

Möchten Sie Vorlagen wieder aus der Liste entfernen, so markieren Sie die gewünschten Vorlagen und klicken auf den *Entfernen*-Button.

Geben Sie keine Vorlage an, so dürfen in dem betreffenden Ordner enthaltene Dateien auf beliebigen Vorlagen basieren.

Anlegecheck für darunter liegende Dateien

Wenn Sie eine Vorlage vom Typ *Ordner* definieren, so können Sie im Feld *Anlegecheck für darunter liegende Dateien* Tcl-Code eingeben, der vom Content Management Server immer dann abgearbeitet wird, wenn zu einem Ordner eine Datei hinzugefügt werden soll. Der Content Management Server legt die Datei nur dann an, wenn der Code die Variable `result` auf einen Wert ungleich 0 (null) gesetzt hat. Wenn der Code vom Content Management Server aufgerufen wird, ist die Variable `objId` auf die Datei-ID des darüber liegenden Ordners gesetzt, und in der Variablen `subObjClassName` ist der Name der Vorlage gespeichert, auf der die neue Datei basieren soll. `result` hat voreingestellt den Wert 1. Schreiboperationen sind in dieser und anderen Funktionen nicht erlaubt.

Bestätigen Sie die Änderungen bitte mit *OK*.

5.1.4 Bearbeitung und Export

Die Optionen für die Bearbeitung und den Export umfassen die Namenserverweiterungen, die die Dateien haben dürfen, nachdem sie exportiert wurden, sowie die Editoren, die für die Bearbeitung des Hauptinhalts der Arbeitsversion zur Verfügung stehen.

Möchten Sie Änderungen an diesen Einstellungen vornehmen, so haben Sie zwei Möglichkeiten, um das entsprechende Formular zu öffnen: Legen Sie gerade eine Vorlage an, so klicken Sie im

entsprechenden Formular auf den Button *Bearbeitung und Export* im Bereich *Weitere Optionen*. Möchten Sie dagegen diese Einstellungen bei einer bestehenden Vorlage ändern, so ermitteln Sie die Vorlage bitte zunächst in der Sektion *Vorlagen* mit Hilfe der Suchfunktion. Klicken Sie anschließend auf den Namen der Vorlage im entsprechenden Listeneintrag, um auf die Ansichtsseite der Vorlage zu wechseln. Hier klicken Sie bitte auf den Button *Bearbeiten*, um zu dem Bearbeitungsformular für Vorlagen zu gelangen. Im Formularbereich *Weitere Optionen* klicken Sie abschließend auf den Button *Bearbeitung und Export*, um das entsprechende Formular zu öffnen.

Konfiguration - Vorlagen - Bearbeitung und Export
 Bearbeitungs- und Exportoptionen für "Standard-Ordner" bearbeiten

Name: publication
Titel: Standard-Ordner

Erlaubte Dateieindungen:

Nicht ausgewählte Dateieindungen
 asp
 css
 htm
 inc
 java

Ausgewählte Dateieindungen
 html
 shtml

Verfügbare Editoren:
 HTML-Editor
 Internet Editor
 Lokale Anwendung
 TinyMCE

Ok Abbrechen

Copyright © 1997-2011 [infopark AG](#) · All rights reserved. Aktueller Benutzer: Superuser

Erlaubte Dateieindungen

In der Liste *Erlaubte Dateieindungen* können Sie auswählen, welche Dateieindungen (eigentlich Dateinamenserweiterungen) die beim Export erzeugten Ausgabedateien haben dürfen. Gleichzeitig legen die von Ihnen ausgewählten Dateieindungen die zulässigen Namenserverweiterungen von Dateien fest, die als Hauptinhalt importiert werden können.

Wenn Sie eine neue Vorlage anlegen, so werden Ihnen in diesem Formular die Dateieindungen angeboten, die für den von Ihnen gewählten Dateityp voreingestellt zulässig sind. Für den Fall, dass Sie eine Vorlage anlegen und nach der Bestätigung Ihrer Angaben zu den erlaubten Dateieindungen einen anderen Dateityp auswählen, gestattet es Ihnen der Content Management Server nicht, die Vorlage anzulegen.

In der Standardinstallation sind die gängigen Dateieindungen vordefiniert. Systemadministratoren können die verfügbaren Dateieindungen mit dem Eintrag `mimeType` im Abschnitt [content](#) der Systemkonfiguration konfigurieren.

Um die erlaubten Dateieindungen der Vorlage festzulegen, wählen Sie die gewünschten Dateieindungen in der Liste *Verfügbare Dateieindungen* aus und klicken auf den nach rechts weisenden Pfeil. Die gewählten Dateieindungen werden in die Liste *Ausgewählte Dateieindungen* aufgenommen. Möchten Sie Dateieindungen wieder aus dieser Liste entfernen, so markieren Sie sie und klicken auf den nach links weisenden Pfeil, um sie wieder zu der Liste *Verfügbare Dateieindungen* hinzuzufügen.

Wenn Sie keine Dateieindungen angeben, so sind beim Import einer Datei alle Dateieindungen zulässig, die der Dateityp unterstützt.

Editoren für den Hauptinhalt

Im Content Navigator sind mehrere Editoren verfügbar: *Interner Editor*, *Externer Editor*, *HTML-Editor* und der *TinyMCE* (bis Version 6.7.3 ist anstelle des *TinyMCE* der *Microsoft-HTML-Editor* verfügbar). Welche Editoren für den Hauptinhalt in der Vorlagendefinition zur Auswahl stehen, ist vom Dateityp der Vorlage abhängig. Der Hauptinhalt einer Arbeitsversion, die zu einer Datei vom Typ *Ordner*, *Dokument* oder *Layout* gehört, kann mit allen verfügbaren Editoren bearbeitet werden. Dateien vom Typ *Bild* können nur mit dem Externen Editor bearbeitet werden. Für *Ressourcen* stehen der Externe Editor und der Interne Editor zur Auswahl.

Verfügbare Editoren

Redakteure können den Hauptinhalt von Arbeitsversionen mit den hier ausgewählten Editoren bearbeiten. Beachten Sie bitte, dass binäre Daten nur mit dem externen Editor bearbeitet werden können.

gestartet, wenn der Hauptinhalt bearbeitet wird. Die Editoren sind:

- **Interner Editor**
Der interne Editor ist ein mehrzeiliges Eingabefeld, in dem der HTML-Text des Hauptinhalts bearbeitet werden kann.
- **HTML-Editor**
Dieser Editor ist ein Java-Applet, mit dem der Hauptinhalt in einem HTML-Text-Modus oder WYSIWYG-Modus bearbeitet werden kann.
- **Tiny MCE** (ab CMS Fiona 6.8.0)
Ein JavaScript-basierter WYSIWYG-Editor für den Hauptinhalt.
- **Microsoft-HTML-Editor** (bis CMS Fiona 6.7.3)
Mit diesem Editor können die Redakteure den Haupttext von Versionen mit dem Microsoft Internet Explorer 6 bearbeiten. Im Bearbeitungsmodus des Internet Explorers wird der HTML-Text weitgehend so dargestellt, wie er nach dem Export aussieht.
- **Lokale Anwendung**
Dieser Editor ist ein Java-Applet, das dem Redakteur die Möglichkeit gibt, den Hauptinhalt mit einer beliebigen, auf dem lokalen Rechner installierten Anwendung zu bearbeiten. Jeder Benutzer kann in seinen persönlichen Einstellungen je Dateieindung den Pfad der lokalen Anwendung angeben, die als Editor fungieren soll. Wählt man den Editor zur Bearbeitung aus, so wird das Java-Applet ausgeführt, das dann die zur Dateieindung passende lokale Applikation startet.

Klicken Sie bitte auf *OK*, um die Angaben zu übernehmen.

5.1.5 Checks und Funktionen

Mit Checks und Funktionen können Sie in bestimmte Aktionen, die mit Versionen durchgeführt werden, eingreifen. Dies geschieht mit Hilfe von Tcl-Code.

Es gibt drei Checks und Funktionen, die in Dateivorlagen angegeben werden können, den Vollständigkeitscheck, die Versionszuweisungsfunktion und die Workflowzuweisungsfunktion. Das Formular, in dem der Tcl-Code eingegeben werden kann, erreichen Sie auf zwei Arten.

- Wenn Sie gerade eine Vorlage anlegen, so klicken Sie bitte im entsprechenden Formular auf den Button *Checks und Funktionen* im Bereich *Weitere Optionen*.
- Wenn Sie dagegen den Tcl-Code in einer bestehenden Vorlage bearbeiten möchten, so lokalisieren Sie die Vorlage bitte zunächst in der Sektion *Vorlagen* mit Hilfe der Suchfunktion. Klicken Sie anschließend auf den Vorlagennamen im entsprechenden Listeneintrag, um auf

die Ansichtsseite der Vorlage zu wechseln und dort wiederum den Button *Bearbeiten*, um zu dem Bearbeitungsformular für Vorlagen zu gelangen. Hier klicken Sie bitte abschließend im Formularbereich *Weitere Optionen* auf den Button *Checks und Funktionen*, um das entsprechende Formular zu öffnen.

Versionszuweisungsfunktion

Die Versionszuweisungsfunktion wird vom Content Management Server aufgerufen, bevor er die Feldwerte einer Version setzt. Dies geschieht, nachdem Sie die Felder in einer Feldergruppe oder den Haupttext einer Version bearbeitet haben. Die Funktion wird aufgerufen, nachdem die Wertzuweisungsfunktionen der einzelnen Felder ausgeführt wurden. Ihr Tcl-Code hat nur lesenden Zugriff auf die Content-Management-Server-Daten. Der Code der Funktion kann die folgenden Variablen verwenden:

- **modifiedAttributes:** Die Liste der Versionsfelder und ihrer Werte, die durch die Schreiboperation geändert werden sollen, kodiert als Name-Wert-Paare. Der Hauptinhalt der Version ist in dieser Liste nicht enthalten.
- **contentId:** Die ID der Version, dessen Felder geändert werden sollen.
- **inBlobFile:** Der Name der Datei, in der die Funktion den Hauptinhalt findet, falls dieser geändert wurde. Wurde der Hauptinhalt nicht geändert, so ist diese Variable nicht definiert.
- **outBlobFile:** Der Name der Datei, in die die Funktion den Hauptinhalt schreiben muss, wenn er ihn ändert. Beim Eintritt in den Code der Funktion existiert diese Datei nicht. Legt die Funktion sie an, so setzt der Content Management Server den Hauptinhalt auf ihren Inhalt.
- **result:** Diese Variable sollte von der Funktion bei erfolgreicher Ausführung auf 1, andernfalls auf 0 gesetzt werden, um dem Content Manager den Ausführungsstatus zu signalisieren.
- **messages:** Ist result 0, so interpretiert der Content Manager den Wert von messages als auszugebende Fehlermeldung.

Sie können mit Ihrem Code die Konsistenz der Feldwerte prüfen und gegebenenfalls die Liste **modifiedAttributes** ändern. Bitte beachten Sie, dass der Hauptinhalt einer Version separat über die Variablen **inBlobFile** und **outBlobFile** referenziert wird. Dies zeigt das folgende Beispiel für einen Hauptinhalt, der Text enthält:


```

if {[info exists inBlobFile]} {
    set fh [open $inBlobFile r]
    fconfigure $fh -encoding utf-8
    set blob [read $fh]
    close $fh

    ### Code, der den Blob und andere Felder modifiziert, kommt
    ### hierher. Jetzt den Blob in der vorgesehenen Datei speichern:

    set fh [open $outBlobFile w]
    fconfigure $fh -encoding utf-8
    puts -nonewline $fh $blob
    close $fh
    set result 1
}

```

Hat Ihr Code `result` auf 1 gesetzt, so interpretiert der Content Management Server `modifiedAttributes` als Feldname-Wert-Paare und speichert die Feldwerte ab. Bei diesem Vorgang werden Eingabewerte, die in einem internen Format gespeichert werden (beispielsweise Datumswerte), konvertiert. Setzen Sie dagegen `result` auf 0 (null), so speichert der Content Management Server die Feldwerte nicht. In diesem Fall nimmt der Content Management Server an, dass ein Fehler aufgetreten ist und dass Ihr Code eine oder mehrere Fehlermeldungen in der Liste `messages` gespeichert hat. Diese Fehlermeldungen werden vom Content Management Server angezeigt.

Workflowzuweisungsfunktion

Mit diesem Code können Sie auf den Workflow einer Arbeitsversion Einfluss nehmen, bevor die Version angelegt wird. Er wird aufgerufen, wenn eine Datei angelegt oder eine der Workflow-Aktionen *Bearbeiten*, *Zurückweisen* oder *Zurückziehen* ausgeführt wird. Bevor der Content Management Server den Code abarbeitet, setzt er die Variable `contentId` auf die ID der betreffenden Arbeitsversion und die Variable `objId` auf die ID der Datei, zu der die Version gehört. Ferner sind in der Variablen `workflow` die Workflow-Parameter gespeichert. Diese Parameter ergeben sich aus dem Workflow, der der Vorlage der Datei zugeordnet ist. Die Parameter sind in der Variablen `workflow` als Tcl-Liste abgelegt. Die Elemente dieser Liste sind:

- `editGroups` *editGroups*
editGroups ist die Liste der Benutzergruppen, die die Version bearbeiten müssen.
- `signatureDefs` *signatureDefs*
signatureDefs ist die Liste der Signaturdefinitionen. Jede Definition ist eine Liste aus zwei Elementen, deren erstes Element der Name eines Signaturfeldes und deren zweites Element der Name der Benutzergruppe ist, deren Mitglieder die Unterschrift leisten dürfen.
- `allowsMultipleSignatures` *multiSig*
multiSig enthält 0, wenn es nicht zulässig sein soll, dass die gleiche Benutzergruppe eine Version mehrfach unterzeichnet. Andernfalls enthält der Parameter den Wert 1.

Mit Ihrem Tcl-Code können Sie die Workflow-Definition ändern, indem Sie sie in der Variablen `workflow` speichern. Das folgende Beispiel zeigt, wie je nach Pfad unterschiedliche Workflows verwendet werden können:

```

set pfa [obj withId $objId get path]
switch -regexp $pfa {
    /intranet {set workflow {allowsMultipleSignatures 1 editGroups Intranet_Editors
        signatureDefs {{Sig_Chiefs1 Chiefs1}}}}
    /internet {set workflow {allowsMultipleSignatures 1 editGroups Internet_Editors
        signatureDefs {{Sig_Chiefs2 Chiefs2}}}}
}

```

Vollständigkeitscheck

Hier können Sie Tcl-Code eintragen, der vom Content Management Server immer abgearbeitet wird, nachdem eine Arbeitsversion geändert wurde. Unter anderem lassen sich hier Konsistenzprüfungen der Werte mehrerer Felder der Version realisieren.

Der Content Management Server ruft den Code nur auf, wenn alle anderen Vollständigkeitsprüfungen erfolgreich abgeschlossen wurden oder die Gründe für die Unvollständigkeit ermittelt werden. Der Content Management Server lässt keine schreibenden Operationen (wie die Änderung eines Feldwertes) zu. Variablen können jedoch gesetzt werden.

Die folgende (globale) Variable wird vom Content Management Server vorgelegt:

- `contentId`: Die ID der Version, die überprüft werden soll.

Mit folgenden (globalen) Variablen wird das Ergebnis zurückgegeben:

- `result`: 0 bedeutet, dass die Version unvollständig ist. Andere Rückgabewerte werden als *vollständig* interpretiert.
- `messages`: Liste der Meldungen der Gründe für die Unvollständigkeit (String-Array).

Nachdem Sie die Checks und Funktionen der Vorlage festgelegt haben, klicken Sie bitte auf **OK**, um die Änderungen zu speichern.

5.1.6 Feldergruppen konfigurieren

Die Felder, die Sie zu einer Vorlage hinzugefügt haben, können Sie zu Feldergruppen zusammenfassen. Sie können die Felder so gruppieren, dass Redakteure die Feldwerte bei der Bearbeitung einer Version in sinnvollen Zusammenstellungen eintragen können. Feldergruppen werden im Content Navigator im Detailbereich angezeigt (dies kann von Administratoren [konfiguriert](#) werden).

Um eine Feldergruppe anzulegen oder die Reihenfolge der Feldergruppen zu ändern, klicken Sie bitte in der Anzeige der Vorlageneigenschaften auf den Button *Feldergruppen anzeigen*. Der Content Manager zeigt nun die Liste der Feldergruppen an, die in der Vorlage definiert wurden:



Bei einer neuen Vorlage ist nur die Basisgruppe vorhanden, die alle Felder enthält. Wenn Sie weitere Feldergruppen erzeugen, so können Sie diesen Gruppen stets die Felder zuordnen, die in der Basisgruppe enthalten sind. Ein Feld wird automatisch aus der Basisgruppe entfernt, wenn es einer anderen Gruppe zugeordnet wird. Folglich enthält die Basisgruppe stets die Felder, die keiner anderen

Gruppe zugewiesen wurden. Die Basisgruppe kann weder gelöscht, noch können Felder manuell aus ihr entfernt werden.

Die Basisgruppe ist im Content Navigator die Felderguppe, deren Felder den Redakteuren in einem Dialog zur Bearbeitung angeboten werden, nachdem sie eine neue Datei angelegt haben. Fügen Sie Felder zu dieser Gruppe hinzu oder entfernen Sie sie daraus, um die wichtigsten Felder unmittelbar nach der Dateierzeugung von den Redakteuren bearbeiten zu lassen.

Um die Reihenfolge der Felderguppen zu ändern, können Sie auf das entsprechende Bild in der Spalte *Reihenfolge der Felderguppen ändern* klicken. Mit dem nach oben oder dem nach unten weisenden Doppelpfeil können Sie eine Felderguppe an den Anfang bzw. an das Ende der Felderguppenliste verschieben. Klicken Sie dagegen auf den nach oben oder den nach unten weisenden einfachen Pfeil, so wird die entsprechende Felderguppe um eine Position nach oben bzw. nach unten verschoben.

Klicken Sie auf den Namen einer Felderguppe, so wird die Ansichtssseite dieser Felderguppe angezeigt:



Hier werden die in dieser Gruppe zusammengefassten Felder angezeigt. Die Reihenfolge, in der sie angezeigt werden, bestimmt die Reihenfolge, in der sie in der Felderguppe angeordnet sind, wenn Sie in der Detailansicht angezeigt werden.

Sie können die Reihenfolge der Felder ändern, indem Sie auf das entsprechende Bild in der Spalte *Reihenfolge der Felder ändern* klicken. Mit dem nach unten oder dem nach oben weisenden Doppelpfeil können Sie ein Feld an das Ende bzw. an den Anfang der Felderliste verschieben. Klicken Sie dagegen auf den nach oben oder den nach unten weisenden einfachen Pfeil, so wird das entsprechende Feld um eine Position nach oben bzw. nach unten verschoben.

Felderguppen anlegen

Klicken Sie in der Felderguppenliste oder auf der Ansichtssseite einer Felderguppe auf den Button *Neu*, um eine neue Felderguppe anzulegen. Der Content Management Server zeigt das Formular *Konfiguration - Vorlagen - Felderguppen - Anlegen* an:

Geben Sie in das Eingabefeld *Name* den Namen der neuen Gruppe ein. Sie können ebenfalls den Titel der Gruppe in der in Ihren persönlichen Einstellungen konfigurierten Sprache festlegen, optional auch in den Sprachen, für die Ihr Content Manager eingerichtet ist.

Klicken Sie zu diesem Zweck bitte auf den Button *Andere Sprachen*, um das Formular *Konfiguration - Vorlagen - Feldgruppen - Mehrsprachige Titel* zu öffnen:

Geben Sie hier bitte die mehrsprachigen Titel der Feldgruppe ein und bestätigen Sie die Eingaben abschließend mit **OK**. Welchen Titel der Content Management Server in der HTML-Benutzeroberfläche anzeigt, hängt von der Sprache ab, die ein Benutzer in seinen persönlichen Einstellungen konfiguriert hat. Ist kein Titel in der Sprache angegeben, die ein Benutzer in seinen persönlichen Einstellungen konfiguriert hat, so zeigt der Content Management Server stattdessen den Namen der Feldgruppe an.

Auswahl der Felder

Einer neuen Gruppe sind zunächst keine Felder zugeordnet. Wählen Sie die gewünschten Felder aus der Liste *Verfügbare Felder* aus und klicken Sie auf den nach rechts weisenden Pfeil.

Die gewählten Felder werden in die Liste *Felder der Feldgruppe* aufgenommen. Möchten Sie Felder wieder aus dieser Liste entfernen, so markieren Sie sie und klicken auf den nach links weisenden Pfeil, um sie wieder zu der Liste *Verfügbare Felder* hinzuzufügen.

Klicken Sie auf **OK**, um die Feldgruppe anzulegen. Der Content Manager zeigt nun die Ansichtssseite der neuen Feldgruppe an.

Feldergruppen bearbeiten und löschen

Um eine Feldergruppe zu bearbeiten oder zu löschen, lokalisieren Sie bitte zunächst die Vorlage, zu der die Feldergruppe gehört, mit Hilfe der Suchfunktion in der Sektion *Vorlagen* und klicken Sie anschließend im Sucheintrag auf den Namen der gewünschten Vorlage, um auf dessen Ansichtsseite zu wechseln. Klicken Sie hier auf den Button *Feldergruppen anzeigen*, so zeigt der Content Manager die Liste der konfigurierten Feldergruppen an. Wechseln Sie nun bitte zur Ansichtsseite der gewünschten Feldergruppe, indem Sie sie durch einen Klick auf den Feldergruppennamen im entsprechenden Listeneintrag der Feldergruppenseite auswählen.

Um die Eigenschaften einer Feldergruppe zu bearbeiten, klicken Sie bitte auf den Button *Bearbeiten*. Der Content Manager zeigt das Formular *Konfiguration - Vorlagen - Feldergruppen - Bearbeiten* an:

In diesem Formular können Sie alle Eigenschaften der Feldergruppe bis auf den Namen ändern. Das Formular enthält die gleichen Daten wie das Formular, mit dem eine neue Feldergruppe angelegt wird.

Klicken Sie bitte auf *OK*, um die geänderten Einstellungen zu übernehmen. Der Content Management Server zeigt anschließend wieder die Ansichtsseite der Feldergruppe an.

Eine Feldergruppe können Sie löschen, indem Sie auf den Löschen-Button am rechten Rand des Listeneintrags der gewünschten Gruppe in der Feldergruppenliste klicken oder indem Sie zur Ansichtsseite der Feldergruppe wechseln und dort auf den Button *Löschen* klicken.

Bestätigen Sie die Sicherheitsabfrage mit *OK*, so wird die Feldergruppe endgültig gelöscht. Die ihr zugewiesenen Felder werden wieder zur Basisgruppe hinzugefügt.

5.1.7 Vorlagen ändern und löschen

Um eine Dateivorlage zu ändern oder zu löschen, wechseln Sie bitte zunächst in die Sektion *Vorlagen* im Bereich *Konfiguration*. Machen Sie hier die Vorlage mit Hilfe der Suchfunktion ausfindig und klicken Sie anschließend in der Liste auf ihren Namen, um ihre Ansichtsseite öffnen. Klicken Sie hier auf den Button *Bearbeiten*, um das Formular *Konfiguration - Vorlagen - Bearbeiten* zu öffnen:

In diesem Formular können Sie die Vorlage bearbeiten, d. h. beispielsweise die Auswahl der ihm zugeordneten Felder ändern. Das Formular enthält weitgehend die gleichen Daten wie das Formular, mit dem eine [neue Vorlage](#) angelegt wird. Der Name einer Vorlage und der Dateityp können hier jedoch nicht mehr geändert werden.

Zusätzlich zu den Optionen, die verfügbar sind, wenn eine Vorlage angelegt wird, können Sie hier bei allen Dateien außer Layouts festlegen, ob sie auf dem Live-Server als News bereitgestellt werden sollen, wenn sie eingereicht oder freigegeben werden. Im Zusammenhang mit dem Versionsfeld [Channels](#) können dadurch Inhalte in News-Feeds aufgenommen werden, die der Portal Manager ausliefern kann.

Bitte entnehmen Sie die Bedeutung der anderen Eingabeelemente den Abschnitten [Vorlagen definieren](#), [Feldeigenschaften bearbeiten](#), und [Anlegeeinschränkungen](#) sowie den Abschnitten [Checks und Funktionen](#) und [Bearbeitung und Export](#).

Beachten Sie bitte, dass sich Änderungen an einer Dateivorlage auf sämtliche Dateien und ihre Arbeitsversionen auswirken, denen diese Vorlage zugrunde liegt. Wird zu der Vorlage beispielsweise ein obligatorisches Feld hinzugefügt, so sind alle Arbeitsversionen, in denen dieses Feld nicht mit einem Wert belegt wurde, anschließend unvollständig und können nicht eingereicht oder freigegeben werden.

Klicken Sie bitte auf *OK*, um die geänderten Einstellungen zu übernehmen. Der Content Management Server zeigt anschließend wieder die Ansichtsseite der Vorlage an.

Möchten Sie eine Vorlage löschen, so lokalisieren Sie es zunächst mit Hilfe der Suchfunktion in der Sektion *Vorlagen*. Klicken Sie anschließend bitte auf den Löschen-Button am rechten Rand des Listeneintrags oder wechseln Sie in die Ansichtsseite der Vorlage, indem Sie auf ihren Namen klicken, und klicken dort wiederum auf den Button *Löschen*.

Der Content Management Server gestattet es Ihnen jedoch nicht, eine Vorlage zu löschen, die noch von einer anderen Vorlagendefinition referenziert wird, bei der also die betreffenden Dateien auf der zu löschenden Vorlage basieren. Es ist jedoch durchaus möglich, eine Vorlage zu löschen, auf der Dateien basieren. Da diese Dateien und ihre Versionen anschließend erst wieder bearbeitet werden können, nachdem ihnen eine andere Vorlage zugewiesen wurde, müssen Sie den Löschvorgang in einer Sicherheitsabfrage bestätigen:

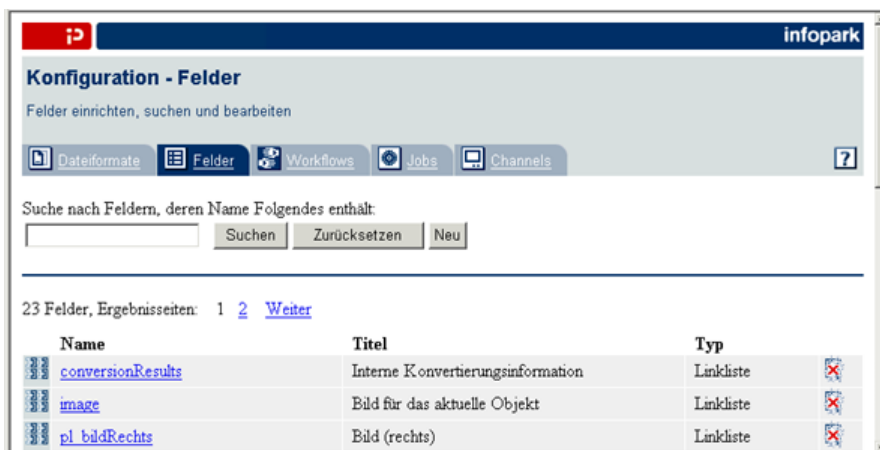


Die Vorlage wird endgültig gelöscht, wenn Sie auf **OK** klicken.

5.2 Die Sektion Felder

Infopark CMS Fiona bietet Ihnen an, kundenspezifische Felder anzulegen, in Dateivorlagen aufzunehmen und so Ihre Inhalte mit Daten anzureichern. Immer wenn Sie eine Datei anlegen, wird ihre Arbeitsversion mit den Feldern ausgestattet, die in der Vorlagendefinition vorgegeben sind. Die Felder einer Arbeitsversion können Sie mit Werten belegen. Später, beim Export, können Sie die Feldwerte vom Content Management Server auslesen und in die erzeugten Webseiten einfügen lassen.

Um ein Feld zu definieren, zu ändern oder zu löschen, wechseln Sie zu der Sektion *Felder*, indem Sie auf der Seite *Konfiguration* den entsprechenden Reiter anklicken. Der Content Management Server zeigt daraufhin die Sektion *Felder* an:



Diese Seite wird auch angezeigt, wenn Sie in einem Dialog auf den *Durchsuchen* – oder *Hinzufügen*-Button klicken, um ein Feld auszuwählen. In diesem Fall ist der Button *Neu* nicht vorhanden und im oberen Bereich der Seite befindet sich ein *Abbrechen*-Button, mit dem Sie die Auswahlaktion bei Bedarf abbrechen können.

Um Felder anzeigen zu lassen, verwenden Sie bitte die Suchfunktion in der entsprechenden Sektion.

Möchten Sie nur die Felder anzeigen lassen, deren Name einen Suchtext enthält, so tragen Sie bitte den Suchtext in das Feld *Suche nach Feldern, deren Name folgendes enthält* ein und klicken anschließend auf den Button *Suche*. Daraufhin wird im Hauptbereich der Sektion eine Liste derjenigen Felder angezeigt, deren Name den Suchtext enthält. Die Listeneinträge enthalten jeweils den Namen, den Titel und den Typ eines Feldes. Möchten Sie einen anderen Suchtext verwenden, so tragen Sie ihn

bitte ein und klicken nochmals auf den Suche-Button. Sie können alle vorhandenen Felder anzeigen lassen, indem Sie das Eingabefeld leer lassen und anschließend auf den Button *Suche* klicken.

Die Suche kann ferner verfeinert werden, indem Sie den Feldtyp als zusätzliches Suchkriterium angeben. Wählen Sie zu diesem Zweck einen der verfügbaren Feldtypen in dem Aufklappmenü *Nur Felder dieses Typs anzeigen* aus.

Wie viele Felder die Liste beinhaltet, ist von Ihren persönlichen Einstellungen abhängig. Enthält die Liste mehr Felder als Sie eingestellt haben, so können Sie mit *Weiter* und *Zurück* blättern. Ferner wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie auf die Zahl klicken.

Wenn diese Seite angezeigt wurde, weil Sie in einem Dialog auf einen *Durchsuchen* – oder *Hinzufügen*-Button geklickt haben, so können Sie nun ein Feld auswählen, indem Sie einfach auf dessen Namen klicken.

Klicken Sie auf den Namen eines Feldes, so zeigt der Content Manager sämtliche über dieses Feld verfügbaren Informationen an:

5.2.1 Felder definieren

Sie können ein neues Feld anlegen, indem Sie entweder auf der Ansichtsseite eines Feldes oder in der Sektion *Felder* auf den Button *Neu* klicken. Das Formular *Konfiguration – Felder – Anlegen* wird geöffnet:


Name

Geben Sie hier bitte den Namen des Feldes an. Der Name kann nachträglich nicht geändert werden. Beachten Sie bitte, dass Feldnamen nicht mit einer Ziffer beginnen dürfen und dass es Namen gibt, die intern vom Content Manager verwendet und daher nicht vergeben werden können. Der Content Management Server erzeugt eine Fehlermeldung, wenn Sie versuchen, ein Feld mit einem solchen Namen anzulegen. Die folgenden Namen sind reserviert:

anchors, archivedContentIds, blob, blobLength, body, bodyTemplateName, channels, children, codeForActionPreview, codeForPreview, codeForSourceView, codeForThumbnail, collection, contentIds, contentType, createNewItem, displayTitle, editor, encodedBlob, encodedExportBlob, exportBlob, exportCharset, exportFiles, exportMimeType, externalAttributes, externalAttrNames, externalAttrTypeDict, frameNames, freeLinks, getKeys, hasChildren, hasSuperLinks, height, id, isCommitted, isComplete, isEdited, isReleased, isRoot, lastChanged, linkListAttributes, mimeType, name, next, nextEditGroup, nextSignGroup, noPermissionLiveServerRead, objClass, objectId, objectsToRoot, objType, parent, path, permissionCreateChildren, permissionLiveServerRead, permissionRead, permissionRoot, permissionWrite, prefixPath, previous, reasonsForIncompleteState, releasedVersions, setKeys, signatureAttrNames, sortValue, subLinks, superLinks, superObjects, suppressExport, textLinks, thumbnail, title, toclist, validControlActionKeys, validCreateObjClasses, validFrom, validObjClasses, validPermissions, validSortKeys, validSortOrders, validSortTypes, validUntil, version, visibleExportTemplates, visibleName, visiblePath, width, workflow, workflowComment, workflowName, xmlBlob.

Typ

Wählen Sie aus dem Aufklappmenü *Typ* den Typ des Feldes aus. Der Typ bestimmt, welche Werte einem Feld zugewiesen werden können und kann nicht nachträglich geändert werden. Der Content Management Server kennt folgende Feldtypen:

Feldtyp	Beschreibung
 Datum	Felder vom Typ <i>Datum</i> nehmen Datumswerte auf. Der Content Management Server speichert alle Datumsangaben in der sogenannten kanonischen Form. Bei diesem Format ist ein Datum als 14-stellige Zeichenkette repräsentiert, die von links beginnend

aus der Jahreszahl, dem Monat, dem Tag, der Stunde, der Minute und der Sekunde zusammengesetzt ist, wobei bis auf die vierstellige Jahreszahl alle Angaben zweistellig sind. Beispielsweise wird das Datum *18.8.2001* als *20010818000000* gespeichert.



Zeichenkette

Felder des Typs *Zeichenkette* nehmen beliebigen Text auf. Der Content Management Server wandelt alle Leerraumzeichen (wie Tabulatoren und Zeilenumbrüche) in Leerzeichen um und entfernt führende und abschließende Leerzeichen, wenn er Zeichenketten speichert. Die Länge einer Zeichenkette ist auf 250 Zeichen beschränkt. Zeichenketten werden nicht umkodiert.



Text

Der Wert eines Feldes vom Typ *Text* kann beliebig langen Text enthalten. Wird der Wert bei der Vorschau oder beim Export ausgelesen, so werden alle HTML-spezifischen Zeichen umkodiert, damit sie lesbar bleiben. So wird beispielsweise die spitze Klammer *<* zu *<*.



HTML

Die Werte der Felder vom Typ *HTML* können beliebig langen HTML-Text enthalten. Im Gegensatz zum Typ *Text* kodiert der Content Management Server beim Typ *HTML* den Feldwert nicht HTML-konform um. Das bedeutet, dass die Zeichen so verwendet werden, wie sie eingegeben wurden. Ferner werden Links in den Feldwerten ins Link-Management des Content Management Servers übernommen.



Markdown

Wie HTML, jedoch können die Feldwerte zusätzlich Text in der [Markdown](#)-Syntax enthalten. Dieser Feldtyp ist ab Version 6.6 verfügbar. In der Vorschau und beim Export wird Markdown-Text nur bei Einsatz des [Rails Connectors](#) ausgewertet.



Auswahl

Bei einem Feld vom Typ *Auswahl* sind die möglichen Werte, die der Benutzer des Content Management Servers dem Versionsfeld zuweisen kann, von vornherein festgelegt.



Mehrfachauswahl

Bei Aufzählungsfeldern mit Mehrfachauswahl sind die möglichen Werte ebenfalls festgelegt. Wenn der Benutzer des Content Management Servers den Feldwert festlegt, kann er jedoch nicht nur einen, sondern mehrere der vordefinierten Werte auswählen.



Linkliste

Felder des Typs *Linkliste* enthalten freie Links, also Links, die eine Version mit einer beliebigen Datei oder einem beliebigen externen Dokument assoziieren. Die Anzahl der freien Links, die in einem Feld vom Typ *Linkliste* gespeichert werden können, ist nicht begrenzt.



Signatur

Signaturfelder sind spezielle Felder, deren Werte vom Content Management Server gesetzt werden, wenn ein Benutzer eine Datei abzeichnet. Der Wert eines Signaturfeldes besteht aus zwei Teilen, dem Zeitpunkt, an dem die Unterschrift geleistet wurde, sowie dem Anmeldenamen des Benutzers, der die Unterschrift geleistet hat. Sie benötigen Signaturfelder, um Workflows definieren zu können, bei denen Dateien unterschrieben werden müssen.

Titel

Vergeben Sie hier bitte einen aussagekräftigen Titel in der Sprache, die sie in Ihren persönlichen Einstellungen konfiguriert haben. So ist die Bedeutung eines Feldwerts in der Regel leichter zu verstehen.

Wenn Ihr Content Management Server für mehrsprachigen Einsatz eingerichtet ist, so können Sie über den Button *Andere Sprachen* zum Formular *Konfiguration – Felder – Mehrsprachige Titel* wechseln und dort in die entsprechenden Eingabefelder Titel in den jeweiligen Landessprachen eingeben.

Konfiguration - Felder - Bearbeiten - Mehrsprachige Titel

Titel in anderen Sprachen definieren

Deutscher Titel:

Englischer Titel:

Spanischer Titel:

Französischer Titel:

Italienischer Titel:

Ok Abbrechen

Bestätigen Sie Ihre Eingaben bitte mit *OK*. Welchen Titel der Content Management Server in der HTML-Benutzeroberfläche anzeigt, hängt von der Sprache ab, die ein Benutzer in den persönlichen Einstellungen konfiguriert hat.

Beschreibung

Nutzen Sie das Eingabefeld *Beschreibung*, wenn Sie anderen Benutzern zusätzliche Informationen über das Feld und seine Funktion geben möchten. Die Beschreibung ist als Hilfetext verfügbar, wenn der Wert des Feldes bearbeitet wird. Ist Ihr Content Management Server für mehrsprachigen Einsatz eingerichtet, so können Sie auf den Button *Andere Sprachen* klicken, um in das Formular *Konfiguration – Felder – Mehrsprachige Beschreibungen* zu wechseln und hier Beschreibungen in den jeweiligen Landessprachen anzugeben:

Konfiguration - Felder - Mehrsprachige Beschreibungen

Beschreibung in anderen Sprachen angeben

Deutsche Beschreibung:

Englische Beschreibung:

Spanische Beschreibung:

Französische Beschreibung:

Italienische Beschreibung:

Ok Abbrechen

Bestätigen Sie bitte auch diese Eingaben mit *OK*.

Durchsuchbar im Content Manager

Wählen Sie diese Option, so werden die Werte dieses Feldes mit dem optionalen Infopark Search Server indiziert und bei einer erweiterten Suche im Content Management Server durchsucht.

Durchsuchbar im Live-Server

Wählen Sie diese Option, so werden die Werte dieses Feldes mit dem optionalen Infopark Search Server auf der Seite des Live-Servers indiziert und bei einer Suche auf dem Live-Server durchsucht.

Klicken Sie abschließend auf *OK*, um die Felddefinition zu speichern. Der Content Manager zeigt daraufhin die Ansichtseite des neuen Feldes an.

5.2.2 Feldeigenschaften ändern

Um die Eigenschaften eines Feldes – beispielsweise seine Beschreibung oder den Typ seines Eingabefeldes – zu ändern, machen Sie bitte zunächst das Feld mit Hilfe der Suchfunktion in der Sektion *Felder* ausfindig. Wechseln Sie auf die Ansichtsseite der Feldeigenschaften, indem Sie anschließend im Listeneintrag des Feldes auf seinen Namen klicken. Klicken Sie auf der Ansichtsseite des Feldes bitte auf den Button *Bearbeiten*. Der Content Management Server zeigt das Formular *Konfiguration - Felder - Bearbeiten* an:

Konfiguration - Felder - Bearbeiten
Das Feld "Empfänger des Newsletters" bearbeiten

Name: ip_newsletterRecipients
Typ: Mehrfachauswahl

Titel:
Empfänger des Newsletters Andere Sprachen

Beschreibung:
Angabe der Empfängergruppen für den Newsletter Andere Sprachen

☒ Durchsuchbar im Content Manager
☐ Durchsuchbar im Live-Server

Werte:
Geben Sie bitte einen Wert je Zeile ein.
Kunden
Interessenten
Mitarbeiter

Funktionen:
Funktionen

Eingabefelder:
Auswahlliste Details

Ok Abbrechen

Sie können Felddefinitionen jederzeit ändern (sofern Sie das Recht haben, globale Einstellungen zu ändern). So ist es beispielsweise möglich, die Liste der Aufzählungswerte bei einem Feld vom Typ *Auswahl* oder *Mehrfachauswahl* zu verkürzen oder zu erweitern. Solche Änderungen können jedoch zur Folge haben, dass Arbeitsversionen unvollständig werden, weil sich der Wert eines ihrer Felder nicht mehr in der Liste der zulässigen Werte befindet.

5.2.3 Allgemeine Eigenschaften

Der obere Formularbereich enthält die gleichen Daten wie das Formular, mit dem ein neues Feld angelegt wird. Der Name und der Typ des Feldes können jedoch nicht geändert werden.

5.2.4 Spezielle Eigenschaften

Die änderbaren Eigenschaften eines Feldes hängen von seinem Typ ab. Die wichtigsten Feldeigenschaften sind:

- **Auswahl/Mehrfachauswahl: Werte**

Bei Feldern vom Typ *Auswahl* oder *Mehrfachauswahl* können Sie hier die wählbaren Werte eintragen. Beachten Sie dabei bitte, dass nur ein Wert je Zeile angegeben werden kann.

- **HTML: Erlaubte Tags**

Bei Feldern vom Typ *HTML* können Sie an dieser Stelle die erlaubten Tags definieren, indem Sie sie (ohne die umgebenden spitzen Klammern) in das mehrzeilige Eingabefeld *Erlaubte Tags* eingeben. Beachten Sie bitte, dass immer nur ein Wert je Zeile angegeben werden darf.

- **Linkliste:**
Minimale und maximale Anzahl Links (ab Version 6.7.0)

Bei Feldern vom Typ *Linkliste* können Sie hier die Mindest- bzw. Höchstanzahl der einzutragenden Links angeben. Die Einhaltung der Vorgabe wird durch die interne Vollständigkeitsprüfung überwacht. Enthält die Liste zuwenige oder zuviele Links, kann die betreffende Datei nicht freigegeben werden.

Ab Version 6.7.2 können Linklisten leer sein, ohne dass dies zu einem Fehler führt, wenn die geforderte Mindestanzahl Links ungleich 0 ist. Um zu erzwingen, dass die Linkliste nicht leer ist, markieren Sie das Feld in der Vorlage als obligatorisch.

5.2.5 Funktionen

Das Verhalten von Feldern kann durch frei definierbare Funktionen ergänzt werden. Um eine der Funktionen für Felder zu definieren, so klicken Sie bitte auf den Button *Funktionen*, um das Formular *Konfiguration – Felder – Bearbeiten – Funktionen* zu öffnen:

Wertanzeigefunktion

Die Wertanzeigefunktion ist eine Tcl-Routine, mit der Sie den in der HTML-Benutzerschnittstelle anzuzeigenden Wert eines Feldes berechnen können. Die Funktion wird aufgerufen, wenn der Wert eines Feldes gesetzt wird. Der Tcl-Routine werden zwei Werte übergeben:

- **value:** Der Feldwert. Er kann von der Tcl-Routine ausgelesen und bei Bedarf neu gesetzt werden.
- **contentId:** Die ID der Version, in der sich der Feldwert befindet. Über den Versionsparameter `objectId(content withId contentId get objectId)` kann beispielsweise die zu der Datei gehörende Vorlage ermittelt werden, um je nach Vorlage unterschiedliche Werte anzeigen lassen zu können.

Die Tcl-Routine kann Fehlermeldungen mit dem Kommando `error` zurück geben. Wird bei einem Feld kein Tcl-Code für die Wertanzeigefunktion angegeben, so wird der ursprüngliche Wert des Feldes

angezeigt. Die Routine hat nur lesenden Zugriff auf die Daten im Content Manager. Insbesondere wird der Feldwert dadurch nicht geändert, dass `value` gesetzt wird.

Wertzuweisungsfunktion

Die Wertzuweisungsfunktion ist eine Tcl-Routine, die der Content Management Server immer dann ausführt, wenn der Wert des Feldes geändert wird. Nutzen Sie dieses Feature, um beispielsweise zu prüfen, ob ein Feld einen gültigen Wert hat und um den Wert gegebenenfalls zu ändern. Der Tcl-Routine werden zwei Werte übergeben:

- `value`: Der Feldwert. Er kann von der Tcl-Routine ausgelesen und bei Bedarf neu gesetzt werden.
- `contentId`: Die ID der Version, in der sich der Feldwert befindet. Über den Versionsparameter `objectId` (`content withId contentId get objectId`) können bei Bedarf Feldwerte der zur Version gehörenden Datei ermittelt werden.

Fehlermeldungen können mit dem Kommando *error* zurückgegeben werden. Die Routine hat keinen schreibenden Zugriff auf die Daten im Content Management Server.

5.2.6 Eingabefelder

Bei allen Feldern – außer solchen vom Typ *Signatur* (und bis Version 6.0.x auch *Linkliste*) – können Sie definieren, welche Art von Eingabefeld der Benutzer verwenden soll, um den Feldwert festzulegen. Wählen Sie zu diesem Zweck bitte den Typ des Eingabefeldes aus dem Aufklappmenü aus.

Die Eingabefeldtypen, die Ihnen der Content Management Server im Aufklappmenü *Eingabefelder* anbietet, hängen vom Typ des Feldes ab, das Sie definiert haben. Folgende Typen sind im Content Management Server definiert:

Eingabefeldtyp	Beschreibung
Einzeiliges Texteingabefeld	Der Typ steht Ihnen bei Feldern vom Typ <i>Zeichenkette</i> , <i>Text</i> , <i>HTML</i> und <i>Datum</i> zur Verfügung.
Mehrzeiliges Texteingabefeld	Dieser Eingabetyp ist bei Feldern vom Typ <i>Zeichenkette</i> , <i>Text</i> und <i>HTML</i> verfügbar.
Kennwortfeld	Dieser Eingabetyp ist nur bei Zeichenketten verfügbar. Zeichen, die in ein solches Feld eingegeben werden, werden durch ein anderes, stets gleiches Zeichen dargestellt. Das Feld dient zur Eingabe von Passwörtern.
Auswahlmenü	Das Auswahlmenü ist ein aufklappbares Menü, das angezeigt wird, wenn der Benutzer den Wert eines Feldes vom Typ <i>Auswahl</i> festlegen kann.
Auswahlschalter	Bei Feldern vom Typ <i>Auswahl</i> zeigt der Content Management Server eine Liste mit den möglichen Werten an, wobei vor jedem Listenelement ein Schalter erscheint, mit dem das Element ausgewählt werden kann. Es kann nur ein Element der Liste ausgewählt werden.
Auswahlliste	Bei Feldern vom Typ <i>Mehrfachauswahl</i> zeigt der Content Management Server eine Liste mit den zulässigen Werten an, aus der die Redakteure beliebig viele auswählen können.
Auswahlkästchen	Bei Feldern vom Typ <i>Mehrfachauswahl</i> zeigt der Content Management Server vor jedem zulässigen Wert ein Kästchen an, das die Redakteure aktivieren können. Sie können beliebig viele Auswahlkästchen aktivieren, um die entsprechenden Werte auszuwählen.

HTML-Editor	Mit dem optional für den Content Manager erhältlichen HTML-Editor können Benutzer die Werte von HTML-Feldern bequem ändern.
Microsoft-HTML-Editor	Wie beim HTML-Editor können Benutzer mit diesem Editor die Werte von HTML-Feldern bequem ändern. Der Microsoft-HTML-Editor macht vom Bearbeitungsmodus des Microsoft Internet Explorers 6 Gebrauch.
Lokale Anwendung	Mit diesem Editor können Benutzer die Werte von HTML-Feldern mit einer lokal (auf dem Client) installierten Applikation ändern. Diese Applikation lässt sich in den persönlichen Einstellungen festlegen. Es wird die Applikation für die Dateierendung <i>html</i> verwendet.
Schreibgeschützt	Wählen Sie diesen Typ aus, wenn der Benutzer nicht die Möglichkeit haben soll, den Feldwert in der HTML-Benutzerschnittstelle einzugeben. Der Typ <i>Schreibgeschützt</i> steht bei allen Feldtypen zur Verfügung.
Linklistenfeld (konfigurierbar ab Version 6.5.0)	Felder des Typs <i>Linkliste</i> können mit dem Dialog Linklisten bearbeiten bearbeitet werden. Ab Version 6.7.0 kann bei Linklistenfeldern ein Startordner für die Linkzielauswahl angegeben werden. Des Weiteren können Sie die zur Auswahl anzubietenden Dateien einschränken, indem Sie deren Vorlagen angeben. Tragen Sie die Namen der Vorlagen durch Leerzeichen getrennt ein.
Assistent	Bis auf Felder des Typs <i>Signatur</i> können alle Felder mit Hilfe eines Assistenten bearbeitet werden. Linklisten können erst ab Version 6.5 mit einem Assistenten bearbeitet werden. Dem Assistenten werden die Standard-Argumente sowie die zusätzlichen Argumente <code>wizard.attributeName</code> (Name des zu bearbeitenden Feldes) und <code>wizard.contentId</code> (ID der Arbeitsversion) übergeben. Der Assistent muss geänderte Feldwerte selbst speichern.

Um die Details für das gewählte Eingabefeld festzulegen, klicken Sie bitte auf den Button *Details*. Es erscheint ein Formular, in dem Sie abhängig vom Eingabefeldtyp die Details – bei Texteingabefeldern beispielsweise die Breite – festlegen können:

The screenshot shows a dialog box titled 'Konfiguration - Felder - Bearbeiten - Eingabefeld' with the subtitle 'Eingabefeld Eigenschaften bearbeiten'. It contains the following fields and controls:

- Name:** ip_newsletterRecipients
- Titel:** Empfänger des Newsletters
- Anzahl der Zeilen:** A text input field.
- ☒ **Leerer Wert erlaubt**
- Buttons:** Ok, Abbrechen

Geben Sie die gewünschten Details des Eingabefeldes an und klicken Sie abschließend auf *OK*, um zum Formular *Konfiguration – Felder – Bearbeiten* zurückzukehren.

Klicken Sie bitte auf *OK*, um die geänderten Feldeinstellungen zu speichern. Der Content Management Server zeigt anschließend wieder die Ansichtseite des Feldes an.

5.2.7 Felder löschen

Möchten Sie ein Feld löschen, so machen Sie es bitte zunächst mit Hilfe der Suchfunktion in der Sektion *Felder* ausfindig. Klicken Sie anschließend auf den Löschen-Button am rechten Rand des

entsprechenden Listeneintrags oder auf den Namen des Feldes, um auf seine Ansichtsseite zu wechseln, und dort wiederum auf den Button *Löschen*:



Das Feld wird endgültig gelöscht, wenn Sie die Sicherheitsabfrage mit *OK* bestätigen.

The Content Manager erlaubt es nicht, ein Feld zu löschen, das in Dateivorlagen verwendet wird. Ist dies der Fall, entfernen Sie zuerst das Feld von den betreffenden Vorlagen und löschen Sie es dann.

5.3 Die Sektion Workflows

Workflows legen fest, welchen Bearbeitungs- und Prüfungsschritten eine Datei auf ihrem Weg zur Freigabe unterworfen ist. Jeder Arbeitsversion einer neuen Datei weist der Content Management Server automatisch den Workflow zu, der in der Vorlage definiert ist, auf der die Datei beruht.

Möchten Sie einen Workflow anlegen, ändern oder löschen, so wechseln Sie bitte zur Sektion *Workflows*, indem Sie auf der Seite *Konfiguration* den Reiter *Workflows* anklicken.



Diese Seite wird auch angezeigt, wenn Sie in einem Dialog auf den *Durchsuchen*- oder *Hinzufügen*-Button klicken, um einen Workflow auszuwählen. In diesem Fall ist der Button *Neu* nicht vorhanden, und im oberen Bereich der Seite befindet sich ein *Abbrechen*-Button, mit dem Sie die Auswahlaktion bei Bedarf abbrechen können.

Verwenden Sie bitte die Suchfunktion der Sektion *Workflows*, um Workflows anzeigen zu lassen. Möchten Sie nur die Workflows anzeigen lassen, deren Name einen bestimmten Suchtext enthält, so tragen Sie den Suchtext in das Feld *Suche nach Workflows, deren Name folgendes enthält* ein und klicken anschließend auf den Button *Suche*. Daraufhin wird im Hauptbereich der Sektion eine Liste derjenigen Workflows angezeigt, deren Name den Suchtext enthält. Möchten Sie nach Workflows

suchen, die einen anderen Suchtext enthalten, so tragen Sie diesen Suchtext bitte in das Eingabefeld ein und klicken anschließend nochmals auf den Suche-Button. Möchten Sie alle vorhandenen Workflows anzeigen lassen, so lassen Sie das Eingabefeld bitte leer und klicken auf den Button *Suche*.

Es ist von Ihren persönlichen Einstellungen abhängig, wie viele Workflows die Liste enthält. Gegebenenfalls wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie auf die Zahl klicken. Ferner können Sie mit *Weiter* und *Zurück* blättern.

Wenn diese Seite angezeigt wurde, weil Sie in einem Dialog auf einen *Durchsuchen*- oder *Hinzufügen*-Button geklickt haben, so können Sie nun einen Workflow auswählen, indem Sie einfach auf dessen Namen klicken.

Klicken Sie auf den Namen eines aufgeführten Workflows, so zeigt der Content Manager auf der Ansichtsseite des Workflows sämtliche über ihn verfügbaren Informationen an:

Konfiguration - Workflows - Anzeigen
Eigenschaften des Workflows "Workflow für die Pressemitteilungen"

Dateiformate Felder Workflows Jobs Channels

Neu Bearbeiten Löschen

Name: pressemitteilungWorkflow
Titel: Workflow für die Pressemitteilungen
Aktiv: Ja

Bearbeitungsschritte:	Schritt	Gruppe
	1	Redakteure
	2	Rechtsabteilung

Prüfungsschritte:	Schritt	Gruppe	Feld
	1	Chefredakteure	Signatur

Mehrfaches Zeichnungsrecht: Nein

5.3.1 Workflows definieren

Sie können einen neuen Workflow anlegen, indem Sie auf der Ansichtsseite eines Workflows oder in der Sektion *Workflows* auf den Button *Neu* klicken. Das Formular *Konfiguration – Workflows – Anlegen* wird geöffnet:

Konfiguration - Workflows - Anlegen
Anlegen eines neuen Workflows

Allgemeine Eigenschaften

☒ Aktiv

Name:

Der Name kann später nicht mehr geändert werden!

Titel:

Bearbeitungsschritte:

Das Formular ist in die Bereiche *Allgemeine Eigenschaften*, *Bearbeitungsschritte* und *Prüfungsschritte* unterteilt, die im Folgenden beschrieben werden.

5.3.2 Allgemeine Eigenschaften

Im ersten Bereich des Formulars legen Sie die allgemeinen Eigenschaften des Workflows fest:



Aktiv

Mit dieser Option können Sie festlegen, ob ein Workflow verwendbar sein soll. Soll der Workflow verwendbar sein, so wählen Sie die Option bitte aus. Wenn Sie eine Workflowdefinition momentan nicht benötigen, sich jedoch nicht sicher sind, ob sie später eingesetzt werden soll, so deaktivieren Sie die Option bitte.

Name

Geben Sie in dieses Eingabefeld bitte den Namen des neuen Workflows ein. Der Name kann nachträglich nicht geändert werden.

Titel

Geben Sie hier den Titel des Workflows in der Sprache an, die Sie in Ihren persönlichen Einstellungen konfiguriert haben. Wenn Ihr Content Management Server für mehrsprachigen Einsatz eingerichtet ist, so können Sie zum Formular *Konfiguration – Workflows – Bearbeiten – Mehrsprachige Titel* wechseln, indem Sie auf den Button *Andere Sprachen* klicken:



Tragen Sie hier in die entsprechenden Felder die Titel in den jeweiligen Landessprachen ein. Bestätigen Sie Ihre Eingaben bitte mit *OK*. Welchen Titel der Content Management Server in der HTML-Benutzeroberfläche anzeigt, hängt von der Sprache ab, die ein Benutzer in seinen persönlichen Einstellungen konfiguriert hat. Gibt es keinen Titel in der Sprache, die sich der Benutzer eingestellt hat, so wird der Name des Workflows als Workflowtitel angezeigt.

5.3.3 Bearbeitungsschritte

Im zweiten Bereich des Formulars legen Sie fest, welche Benutzergruppen an der Bearbeitung von Dateien beteiligt sind:

Bearbeitungsschritte:

Schritt	Gruppe	
1	<input type="text"/>	Durchsuchen
2	<input type="text"/>	Durchsuchen
3	<input type="text"/>	Durchsuchen

Schritt einfügen nach 3 Schritt entfernen Letzter Schritt

Prüfungsschritte: vor 1
nach 1
nach 2
nach 3

Hier können Sie definieren, welche Benutzergruppen eine Datei, die diesen Workflow durchläuft, bearbeiten sollen. Klicken Sie zu diesem Zweck auf den Button *Schritt einfügen*. Haben Sie einen Bearbeitungsschritt definiert, so können Sie mit dem Button *Schritt einfügen* und dem Aufklappmenü rechts daneben an beliebiger Stelle einen Workflowschritt einfügen. Tragen Sie bitte anschließend jeweils den Namen der Bearbeitergruppe in das Eingabefeld ein. Alternativ können Sie auf den Button *Durchsuchen* klicken, um auf die Auswahlseite für Gruppen zu wechseln. Sie können die Suche von vornherein beschränken, indem Sie einen Teil eines Gruppennamens in das Eingabefeld eingeben und anschließend auf den *Durchsuchen*-Button klicken. Sie können jederzeit auf der [Auswahlseite](#) eine neue Suche auslösen, indem Sie den Suchbegriff in das Eingabefeld eingeben und auf *Suchen* klicken. Klicken Sie bitte abschließend auf den Namen der gewünschten Benutzergruppe im entsprechenden Listeneintrag, um den Gruppennamen in das Eingabefeld zu übernehmen. Wiederholen Sie diesen Vorgang bitte so oft, bis Sie alle für Ihren Workflow notwendigen Bearbeitungsschritte festgelegt haben.

Haben Sie versehentlich zu viele Bearbeitungsschritte eingefügt oder soll eine Gruppe aus dem Bearbeitungsworkflow entfernt werden, so können Sie die überzähligen Schritte entfernen, indem Sie den zu löschenden Bearbeitungsschritt aus dem Aufklappmenü rechts neben dem Button *Schritt entfernen* auswählen und anschließend auf diesen Button klicken.

5.3.4 Prüfungsschritte

Im dritten Bereich des Formulars legen Sie fest, welche Benutzergruppen an der Prüfung von Dateien beteiligt sind:

Prüfungsschritte:

Schritt	Gruppe	Attribut	
1	<input type="text"/>	<input type="text"/>	Durchsuchen
			Durchsuchen

Schritt einfügen nach 1 Schritt entfernen Letzter Schritt

☐ Mehrfaches Zeichnungsrecht

Legen Sie hier analog zu dem oben beschriebenen Verfahren die Namen der Benutzergruppen fest, deren Mitglieder die Unterschrift leisten, d. h. eine Version abzeichnen dürfen. Der Content Management Server legt die Unterschrift in einem Signaturfeld ab. Tragen Sie für jeden Prüfungsschritt den Namen des Feldes bitte in das Eingabefeld ein. Alternativ können Sie auf den Button *Durchsuchen* klicken, um auf die Auswahlseite für Felder zu wechseln. Sie können die Suche von vornherein beschränken, indem Sie einen Teil eines Feldnamens in das Eingabefeld eingeben und anschließend auf den *Durchsuchen*-Button klicken. Sie können jederzeit auf der [Auswahlseite](#) eine neue Suche auslösen, um das gewünschte Feld zu ermitteln. Mit einem Klick auf den Feldnamen im entsprechenden Listeneintrag übernehmen Sie den Namen des Feldes in das Eingabefeld.

Immer wenn ein Benutzer eine Datei mit diesem Workflow abzeichnet, speichert der Content Management Server Datum und Uhrzeit sowie den Anmeldenamen des abzeichnenden Benutzers in dem Signaturfeld. Signaturfelder werden auf die gleiche Weise angelegt wie andere kundenspezifische Versionsfelder auch.

Mehrfaches Zeichnungsrecht

Aktivieren Sie diese Option, wenn für diesen Workflow gelten soll, dass die Mitarbeiter der zu Unterschriften berechtigten Gruppen eine Version auch mehrfach abzeichnen dürfen. Ist beispielsweise ein Prüfer Mitglied in zwei Prüfungsgruppen des Workflows, so darf er bei beiden Prüfungsschritten die Unterschrift leisten, wenn diese Option aktiviert ist. Dateiadministratoren dürfen Versionen ihrer Dateien immer abzeichnen.

Klicken Sie auf *OK*, um die Workflowdefinition abzuschließen. Der Content Manager zeigt Ihnen die Ansichtsseite des neuen Workflows an.

5.3.5 Workflows bearbeiten oder löschen

Möchten Sie die Eigenschaften eines Workflows bearbeiten, so machen Sie den Workflow bitte zunächst mit Hilfe der Suchfunktion in der Sektion *Workflows* ausfindig und klicken anschließend auf seinen Namen im entsprechenden Listeneintrag, um auf seine Ansichtsseite zu wechseln. Klicken Sie hier auf den Button *Bearbeiten*. Der Content Management Server zeigt nun die Seite *Konfiguration - Workflows - Bearbeiten* an:

Hier können Sie Änderungen an den Workfloweinstellungen vornehmen. Das Formular enthält die gleichen Daten und Optionen wie das Formular, mit dem ein neuer Workflow angelegt wird. Der Name des Workflows kann jedoch nicht geändert werden. Bitte entnehmen Sie die Bedeutung der Eingabeelemente den Abschnitten *Allgemeine Eigenschaften* und *Bearbeitungs-* sowie *Prüfungsschritte* unter Workflows definieren.

Klicken Sie auf *OK*, um die geänderten Einstellungen des Workflows zu übernehmen. Der Content Management Server zeigt wieder die Ansichtsseite des Workflows an.

Haben Sie einen Workflow geändert, so werden die Änderungen erst wirksam, wenn eine neue Arbeitsversion angelegt wird. Dies ist immer dann der Fall, wenn eine neue Datei erzeugt, eine freigegebene Version bearbeitet oder eine eingereichte Version abgelehnt wird.

Möchten Sie einen Workflow löschen, so lokalisieren Sie ihn bitte zunächst über die Suchfunktion der Sektion *Workflows* . Klicken Sie anschließend auf den Löschen-Button am rechten Rand des betreffenden Listeneintrags oder auf den Namen des Workflows, um zu seiner Ansichtseite zu wechseln, und dort wiederum auf den Button *Löschen* .



Der Workflow wird endgültig gelöscht, wenn Sie auf *OK* klicken.

5.4 Die Sektion Jobs

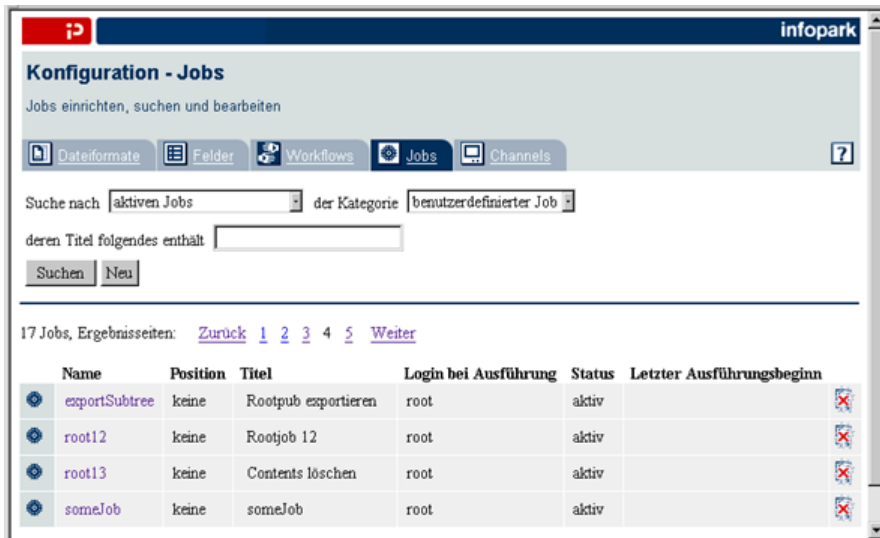
Bei Infopark CMS Fiona gibt es vordefinierte und benutzerdefinierte (kundenspezifische) *Jobs*. Sie können benutzerdefinierte Jobs nach Belieben einrichten, ändern und löschen. Vordefinierte Jobs dagegen können nicht angelegt oder gelöscht werden, und es ist auch nicht möglich, deren Code zu ändern. Selbstverständlich kann bei allen Jobs der jeweilige Ausführungsplan beliebig gesetzt werden. Es gibt die folgenden vordefinierten Jobs:

Jobname	Aufgabe
systemPublish	Überträgt Datei-Aktualisierungsinformationen zur Template Engine und stößt auf dem Live-Server den Export und die Veröffentlichung an.
systemTransferUpdates	Überträgt Datei-Aktualisierungsinformationen zur Template Engine .
systemSendReminderNotif	Versendet E-Mails bei fälligen Wiedervorlagen .

In der Konfigurationssektion *Jobs* können Sie Jobs anlegen und bearbeiten, sofern Sie das globale Administrationsrecht haben. Bei Bedarf können Sie Jobs ausführen, d. h. in die Warteschlange stellen. Sie können Jobs auch abbrechen, also sie aus der Warteschlange entfernen. Es ist jedoch nicht möglich, einen Job abzubrechen, dessen Tcl-Skript gerade abgearbeitet wird. In diesem Fall hat der Job die Position 0 in der Warteschlange und kann nicht daraus entfernt werden.

Bitte beachten Sie, dass Jobs nur im Server-Modus ausgeführt werden. Jobs werden nicht ausgeführt, wenn der Content Management Server mit dem Kommandozeilenparameter -single aufgerufen wurde.

Möchten Sie einen Job anlegen, bearbeiten, löschen, ausführen oder abbrechen, oder dessen Protokoll anzeigen lassen, so wechseln Sie bitte zur Sektion *Jobs*, indem Sie auf der Seite *Konfiguration* den Reiter *Jobs* anklicken.



Nutzen Sie bitte die Suchfunktion der Sektion *Jobs*, um Jobs anzeigen zu lassen. Möchten Sie die Jobs anzeigen lassen, deren Titel einen bestimmten Suchtext enthält, so wählen Sie bitte zunächst eines der Jobkriterien (*allen Jobs*, *wartenden Jobs*, *aktiven Jobs* oder *Jobs, die unter meinem Login ausgeführt werden*) sowie die Jobkategorie aus den Aufklappmenüs aus. Tragen Sie anschließend bitte den Suchtext in das Eingabefeld ein und klicken auf den Button *Suche*. Im Hauptbereich der Sektion wird daraufhin die Liste der Jobs angezeigt, die auf die Suchkriterien passen. Sie können jederzeit eine neue Suche auslösen, indem Sie einen neuen Text in das Eingabefeld eintragen und anschließend ein weiteres Mal auf den Button *Suche* klicken. Möchten Sie alle vorhandenen Jobs einer Kategorie anzeigen lassen, so wählen Sie bitte das Kriterium *alle Jobs* aus, lassen das Eingabefeld leer und klicken auf den Button *Suche*.

Die Listeneinträge des Suchergebnisses enthalten jeweils den Namen des Jobs, seine Position in der Warteschlange und seinen Titel, das Login unter dem der Job ausgeführt werden darf und den Status des Jobs sowie das Datum des letzten Ausführungsbeginns des Jobs.

Enthält die Liste mehr Einträge als auf eine Seite passen, so wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie die Zahl anklicken. Ferner können Sie mit *Weiter* und *Zurück* blättern.

Klicken Sie auf den Namen eines aufgeführten Jobs, so zeigt der Content Manager auf der Ansichtseite des Jobs sämtliche über ihn verfügbaren Informationen an:

Konfiguration - Jobs - Anzeigen
Eigenschaften des Jobs "Teilhierarchie exportieren"

Neu Bearbeiten Löschen Job starten Protokoll anzeigen Letzte Ausgabe anzeigen

Name: exportSubtree
 Titel: Teilhierarchie exportieren
 Kategorie: benutzerdefinierter Job
 Beschreibung: Exportiert Dateien unterhalb von 'intranet'
 Aktiv: Ja
 Erforderliches Jobausführungsrecht: permissionGlobalExport
 Login, unter dem das Skript ausgeführt wird: root
 Beginn der letzten Ausführung: 01.07.2002 13:55 (Server-Zeit: 01.07.2002 11:55)
 Ende der letzten Ausführung: 01.07.2002 13:55 (Server-Zeit: 01.07.2002 11:55)
 Ergebnis der letzten Ausführung:
 Position in der Warteschlange: keine
 Nächster Ausführungstermin:
 Ausführungsplan: Kein Ausführungsplan gesetzt.
 Skript: obj root exportSubtree filePrefix /tmp

5.4.1 Jobs definieren

Wenn Sie ein Superuser sind, können Sie einen neuen Job anlegen, indem Sie auf der Ansichtsseite eines Jobs oder in der Sektion *Jobs* auf den Button *Neu* klicken. Der Content Manager zeigt das Formular *Konfiguration - Jobs - Anlegen* an:

Konfiguration - Jobs - Anlegen
Einen neuen Job anlegen

Allgemeine Eigenschaften

Name:

Der Name kann später nicht mehr geändert werden!
☒ Aktiv
 Titel:

 Beschreibung:

Ausführung

Das Formular ist in die Bereiche *Allgemeine Eigenschaften*, *Ausführung* und *Skript* unterteilt, die im Folgenden beschrieben werden.

5.4.2 Allgemeine Eigenschaften

Im ersten Bereich des Formulars legen Sie die allgemeinen Eigenschaften des Jobs fest:

Name

Geben Sie hier bitte den Namen des Jobs ein. Damit der Name nicht mit dem Namen eines System-Jobs kollidiert, darf er weder mit `system` noch mit einem Unterstrich beginnen. Ferner gelten die üblichen Regeln für Namen (keine Sonderzeichen, nur die Zeichen 0 bis 9, A bis Z, a bis z sowie der Unterstrich sind erlaubt).

Aktiv

Mit dieser Option können Sie festlegen, ob ein Job verwendbar sein soll, d. h. in die Warteschlange gestellt werden kann. Wenn Sie den Job nicht verwenden möchten, so wählen Sie diese Option ab, um den Job zu deaktivieren.

Titel

Geben Sie in dieses Eingabefeld bitte den Titel des neuen Jobs ein. Beachten Sie bitte, dass der Titel eines Jobs einschließlich aller Zeichenumwandlungen (beispielsweise von ANSI nach UTF-8) eine maximale Länge von 250 Zeichen haben kann.

Beschreibung

Geben Sie in dieses Eingabefeld bitte eine möglichst sinnfällige Beschreibung des neuen Jobs ein. Auch die Beschreibung eines Jobs kann einschließlich aller Zeichenumwandlungen (beispielsweise von ANSI nach UTF-8) eine maximale Länge von 250 Zeichen haben.

5.4.3 Ausführung

Im zweiten Bereich des Formulars legen Sie fest, unter welchem Login das Job-Skript ausgeführt wird und welches globale Recht erforderlich ist, um den Job ausführen oder abbrechen zu können. Ferner können Sie hier den Ausführungsplan des Jobs definieren:

Login, unter dem das Skript ausgeführt wird

Geben Sie in dieses Eingabefeld den Anmeldenamen des Benutzers ein, unter dessen Login das Skript des Jobs ausgeführt werden soll. Lassen Sie das Eingabefeld leer, so übernimmt der Content Manager als Ausführungs-Login den Anmeldenamen des aktuellen Benutzers, wenn der Job gespeichert wird.

Ist Ihnen der Anmelde-name des Benutzers, unter dessen Login das Skript des Jobs ausgeführt werden soll, nicht bekannt, so klicken Sie bitte auf den Button *Durchsuchen*. Daraufhin wird die Auswahlseite für Benutzer geöffnet. Sie können die Suche von vornherein beschränken, indem Sie einen Teil eines Benutzernamens in das Eingabefeld eingeben und anschließend auf den *Durchsuchen*-Button klicken. Sie können auf der [Auswahlseite](#) jederzeit eine neue Suche auslösen, um den gewünschten Benutzer zu ermitteln. Klicken Sie im entsprechenden Listeneintrag auf das Login des gewünschten Benutzers, so wird der Anmelde-name in das Feld *Login, unter dem das Skript ausgeführt wird* übernommen.

Erforderliches Jobausführungsrecht

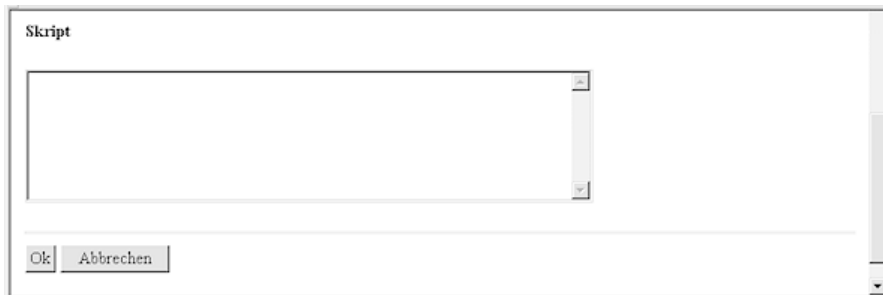
Wählen Sie aus dem Aufklappmenü *Erforderliches Jobausführungsrecht* bitte das globale Recht aus, das ein Benutzer benötigt, um den Job zu starten, also in die Warteschlange zu stellen, oder abzubrechen. Geben Sie kein globales Recht an, so kann der Job von allen Benutzern ausgeführt werden.

Ausführungsplan

Wann ein Job ausgeführt wird, ist durch seinen Ausführungsplan festgelegt. Neue Jobs haben zunächst keinen Ausführungsplan. Sie können einen solchen Plan jedoch anlegen, indem Sie auf den Button *Ausführungsplan* erzeugen klicken.

5.4.4 Skript

Im dritten Bereich des Formulars legen Sie fest, welches Tcl-Skript abgearbeitet werden soll, wenn der Job ausgeführt wird. Geben Sie den Tcl-Code bitte in das Eingabefeld ein:



Beachten Sie bitte, dass Sie das Ergebnis der Ausführung setzen können, indem Sie der Variablen `result` einen Wert zuweisen. Um Daten auszugeben, darf das Tcl-Skript nicht in den Standard-Ausgabekanal schreiben. Stattdessen kann die Prozedur `writeToJobLog` mit der auszugebenden Zeichenkette als einzigem Argument aufgerufen werden.

Klicken Sie auf **OK**, um die Jobdefinition abzuschließen. Der Content Management Server zeigt nun die Ansicht der neuen Jobs an.

5.4.5 Jobs ausführen oder abbrechen

Möchten Sie einen Job ausführen oder abbrechen, so machen Sie ihn bitte zunächst mit Hilfe der Suchfunktion in der Sektion *Jobs* ausfindig und klicken anschließend auf den Namen des Jobs im entsprechenden Listeneintrag. Der Content Management Server zeigt die Definition des Jobs auf der Seite *Konfiguration – Jobs – Anzeigen* an:



Konfiguration - Jobs - Anzeigen
Eigenschaften des Jobs "Teilhierarchie exportieren"

Dateiformate | Felder | Workflows | **Jobs** | Channels

Neu | Bearbeiten | Löschen | Job starten | Protokoll anzeigen | Letzte Ausgabe anzeigen

Name: exportSubtree
 Titel: Teilhierarchie exportieren
 Kategorie: benutzerdefinierter Job
 Beschreibung: Exportiert Dateien unterhalb von 'intranet'
 Aktiv: Ja
 Erforderliches Jobausführungsrecht: permissionGlobalExport
 Login, unter dem das Skript ausgeführt wird: root
 Beginn der letzten Ausführung: 01.07.2002 13:55 (Server-Zeit: 01.07.2002 11:55)
 Ende der letzten Ausführung: 01.07.2002 13:55 (Server-Zeit: 01.07.2002 11:55)
 Ergebnis der letzten Ausführung:
 Position in der Warteschlange: keine
 Nächster Ausführungstermin:
 Ausführungsplan: Kein Ausführungsplan gesetzt.
 Skript: obj root exportSubtree filePrefix /tmp

Um einen Job auszuführen, klicken Sie bitte auf den Button *Job starten*. Dieser Button wird nur angezeigt, wenn Sie das erforderliche Recht besitzen, um den Job auszuführen, oder wenn für den Job kein Ausführungsrecht angegeben wurde.

Der Content Manager bittet Sie nun, die Ausführung des Jobs in einer Sicherheitsabfrage zu bestätigen:

Bestätigen Sie die Sicherheitsabfrage mit *OK*, wenn Sie den Job starten wollen. Andernfalls klicken Sie bitte auf den Button *Abbrechen*, um zur Jobdefinition zurückzugelangen.

Befindet sich ein Job gerade in der Ausführungswarteschlange, so erscheint auf der Ansichtsseite des Jobs der Button *Job abbrechen*, auf den Sie klicken können, um den Job aus der Warteschlange zu entfernen:



Nur wenn Sie das erforderliche Recht besitzen, um den Job auszuführen, oder wenn für den Job kein Ausführungsrecht angegeben wurde, wird dieser Button angezeigt.

Der Content Management Server bittet Sie, den Abbruch des Jobs in einer Sicherheitsabfrage zu bestätigen:



Möchten Sie den Job abbrechen, so bestätigen Sie die Sicherheitsabfrage mit *OK*. Andernfalls klicken Sie bitte auf den Button *Abbrechen*, um zur Jobdefinition zurückzugelangen.

5.4.6 Jobs bearbeiten oder löschen

Um die Eigenschaften eines Jobs zu bearbeiten, machen Sie den gewünschten Job bitte zunächst mit Hilfe der Suchfunktion in der Sektion *Jobs* ausfindig und klicken anschließend auf seinen Namen im entsprechenden Listeneintrag. Sofern Sie ein Superuser sind, können Sie auf der daraufhin angezeigten Ansichtsseite des Jobs auf den Button *Bearbeiten* klicken, um das Formular *Konfiguration – Jobs – Bearbeiten* zu öffnen:

Bitte entnehmen Sie die Bedeutung der Eingabeelemente den Abschnitten *Allgemeine Eigenschaften*, *Ausführung* und *Skript* unter [Jobs definieren](#).

Haben Sie die Jobdefinition bearbeitet, so werden die Änderungen erst wirksam, wenn der Job das nächste Mal ausgeführt wird.

Klicken Sie auf **OK**, um die geänderten Einstellungen des Jobs zu übernehmen. Der Content Management Server zeigt anschließend wieder die Ansichtsseite des Jobs an.

Um einen Job zu löschen, machen Sie bitte den entsprechenden Job in der Sektion *Jobs* mit Hilfe der Suchfunktion ausfindig und klicken auf den Löschen-Button am rechten Rand des Listeneintrags. Der Job kann auch auf seiner Ansichtseite gelöscht werden. Zu diesem Zweck klicken Sie im Suchergebnis auf den Namen des gewünschten Jobs und auf seiner Ansichtseite wiederum auf den Button *Löschen*. Der Content Management Server bittet Sie nun, den Löschvorgang zu bestätigen:

Der Job wird endgültig gelöscht, wenn Sie auf **OK** klicken. Der Content Manager gestattet Ihnen jedoch nicht, einen Job zu löschen, der sich in der Warteschlange befindet, also gerade ausgeführt wird.

5.4.7 Den Ausführungsplan eines Jobs bearbeiten

Wenn Sie einen Job definieren oder bearbeiten, wie dies in Abschnitt [Jobs definieren](#) beschrieben ist, so haben Sie ebenfalls die Möglichkeit, den Ausführungsplan des Jobs anzulegen oder zu bearbeiten. Klicken Sie zu diesem Zweck bitte auf den Button *Ausführungsplan erzeugen* bzw. *Bearbeiten* neben einem Eintrag des Plans. Daraufhin erscheint das folgende Formular:

Konfiguration - Jobs - Bearbeiten - Ausführungsplan

Ausführungsplan-Eintrag des Jobs "Teilhierarchie exportieren" bearbeiten

Diese Angaben beziehen sich auf die Zeitzone des Servers: GMT
Die aktuelle Zeit ist: 01.07.2002 16:56 (Server-Zeit: 01.07.2002 14:56)

Minute	Stunde	Wochentag	Tag	Monat	Jahr
0	0	Montag	1	Januar	2002
1	1	Dienstag	2	Februar	2003
2	2	Mittwoch	3	März	2004
3	3	Donnerstag	4	April	2005
4	4	Freitag	5	Mai	2006
5	5	Samstag	6	Juni	2007
6	6	Sonntag	7	Juli	2008
7	7		8	August	2009
8	8		9	September	2010
9	9		10	Oktober	2011
10	10		11	November	
11	11		12	Dezember	
12	12		13		
13	13		14		
14	14		15		

Ok Abbrechen

Ein Ausführungsplan kann mehrere Einträge umfassen, die jeweils auf der obigen Seite bearbeitet werden können. Alle Einträge beziehen sich auf die Server-Zeit mit ihrer spezifischen Zeitzone, die im oberen Bereich der Seite angegeben ist. Bei sich überschneidenden Angaben in mehreren Einträgen, hat der in der Liste weiter oben liegende Eintrag Vorrang.

Im Hauptbereich des Formulars können Sie mit Hilfe der Listen *Minute*, *Stunde*, *Wochentag*, *Tag*, *Monat* und *Jahr* einzelne oder auch wiederkehrende Ausführungszeitpunkte festlegen. Ein einzelner Ausführungszeitpunkt wird angegeben, indem aus jeder Liste außer *Wochentag* genau ein Wert ausgewählt wird.

Wiederkehrende Ausführungszeitpunkte können definiert werden, indem in einer Liste mehrere Werte oder gar kein Wert markiert wird. Ist kein Wert markiert, so werden implizit alle Werte der betreffenden Liste als ausgewählt interpretiert. Führt man beispielsweise nicht die Tage oder Wochentage auf, in denen der Job ausgeführt werden soll, so wird er täglich ausgeführt. Ein Eintrag, bei dem in keiner Liste etwas ausgewählt wurde, bewirkt also, dass der Job jede Minute ausgeführt wird, und zwar so lange, bis der Eintrag gelöscht oder geändert wird. Sind sowohl Tage als auch Wochentage ausgewählt, so wird der Job ausgeführt, wenn das aktuelle Datum auf einen der ausgewählten Tage oder Wochentage fällt. Das System hindert Sie nicht daran, niemals eintreffende Ausführungszeitpunkte zu definieren (beispielsweise den 31. Juni).

Klicken Sie abschließend auf *OK*, um Ihre Definition zu speichern oder auf *Abbrechen*, um ohne Änderungen am Ausführungsplan zur Job-Bearbeitungsseite zurückzukehren. Dort werden die Einträge des Ausführungsplans folgendermaßen dargestellt:

Ausführungsplan:

Minute	Stunde	Wochentag	Tag	Monat	Jahr		
0	0	Mo	1 - 10, 20	alle	alle	Bearbeiten	Löschen
alle	10	alle	alle	Okt	2002	Bearbeiten	Löschen

Eintrag hinzufügen

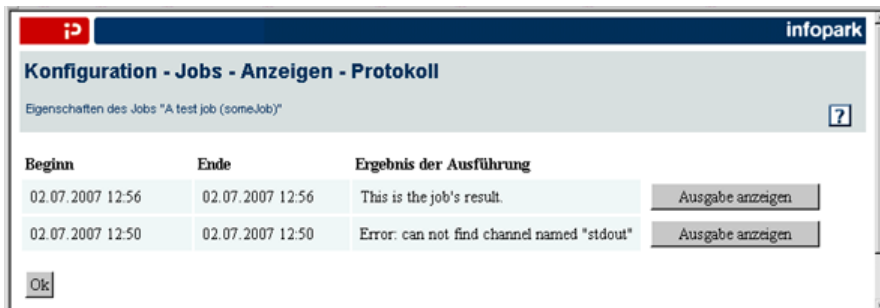
Jede Zeile stellt einen Eintrag dar und ist mit den Buttons *Bearbeiten* und *Löschen* versehen. Sollten Sie einen Eintrag versehentlich gelöscht haben, so brechen Sie die Bearbeitung bitte ab und bearbeiten Sie den Job erneut.

Bitte beachten Sie auch die Informationen im Abschnitt [Jobs](#).

5.4.8 Job-Protokoll und Job-Ausgabe anzeigen

Jedes Mal, wenn ein Job ausgeführt wird, erzeugt der Content Management Server (und auch die Template Engine) einen Protokolleintrag für diesen Job. In diesem Eintrag werden beispielsweise die Startzeit und das Ergebnis der Ausführung aufgezeichnet. Die maximale Anzahl der Protokolleinträge je Job wird durch den Systemkonfigurationseintrag `jobMaxLogLength` bestimmt. Bei Bedarf werden ältere Einträge automatisch gelöscht.

Das Protokoll eines Jobs können Sie öffnen, indem Sie auf dessen Ansichtsseite (siehe [Die Sektion Jobs](#)) auf den Button *Protokoll anzeigen* klicken. Daraufhin erscheint die Liste der Protokolleinträge:



Das Protokoll gibt Aufschluss darüber, wann ein Job ausgeführt wurde und welches Ergebnis der jeweilige Lauf hatte. Das Ergebnis der Ausführung kann innerhalb des Job-Skripts gesetzt werden, indem man der Variablen `result` einen Wert zuweist.

Beachten Sie bitte, dass es zu einem Fehler führt, wenn ein Job-Skript Daten in den Standard-Ausgabekanal schreibt. Stattdessen kann man in dem Skript die Prozedur `writeToJobLog` mit dem Argument `message` verwenden, um Text auszugeben, der auf der Ausgabeseite zu sehen sein soll.

Um die Ausgabe eines Jobs zu sehen, klicken Sie bitte auf den Button *Ausgabe anzeigen* neben dem betreffenden Eintrag. Daraufhin wird die Job-Ausgabeseite angezeigt:



Die Job-Ausgabeseite können Sie auch von der Ansichtsseite eines Jobs öffnen, indem Sie dort auf den Button *Letzte Ausgabe anzeigen* klicken. In diesem Fall liefert die Job-Ausgabeseite stets die aktuelle Ausgabe des Jobs und wird in regelmäßigen Abständen neu geladen. Darüber hinaus befindet sich neben dem *OK*-Button der Button *Aktualisieren*, auf den Sie klicken können, um die Seite manuell neu zu laden und damit den aktuellen Stand der Ausgabe anzeigen zu lassen.

Klicken Sie auf *OK*, um zur Liste der Log-Einträge zurückzukehren. Von dort aus können Sie ebenfalls mit *OK* zur Ansichtsseite des Jobs zurückkehren.

5.5 Die Sektion Channels

In der Sektion *Channels* können Sie Channels definieren, denen die Redakteure später Dateien (beispielsweise News-Artikel) zuordnen können. Unter Verwendung des Portal Managers lassen sich die Channels und die ihnen zugeordneten Nachrichten auf dem Live-Server als RSS-Feeds abrufen.

Um einen Channel anzulegen, zu bearbeiten oder zu löschen, wechseln Sie bitte zur Sektion *Channels*, indem Sie auf der Seite *Konfiguration* den Reiter *Channels* anklicken.



Nutzen Sie bitte die Suchfunktion der Sektion *Channels*, um bereits angelegte Channels anzeigen zu lassen. Tragen Sie bitte einen Namensbestandteil des gesuchten Channels in das Eingabefeld ein und klicken auf den Button *Suche*. Im Hauptbereich der Sektion werden daraufhin die Channels angezeigt, deren Name den Suchtext enthält. Sie können jederzeit eine neue Suche auslösen, indem Sie einen neuen Text in das Eingabefeld eintragen und anschließend ein weiteres Mal auf den Button *Suche* klicken. Wenn Sie möchten, dass alle Channels aufgeführt werden, so lassen Sie bitte das Eingabefeld leer und klicken auf den Button *Suche*.

Die Listeneinträge des Suchergebnisses enthalten jeweils den Namen und den Titel des Channels. Wie viele Channels die Liste beinhaltet, ist von Ihren persönlichen Einstellungen abhängig. Enthält die Liste mehr Einträge als Sie eingestellt haben, so wird ober- und unterhalb der Liste eine aufsteigende Zahlenfolge angezeigt. Jede Zahl verweist auf eine weitere Ergebnisseite, die Sie anzeigen lassen können, indem Sie die Zahl anklicken. Ferner können Sie mit *Weiter* und *Zurück* blättern.

Klicken Sie auf den Namen eines aufgeführten Channels, so zeigt der Content Manager auf der Ansichtsseite des Channels die über ihn verfügbaren Informationen an:



5.5.1 Channels definieren

Sie können einen neuen Channel anlegen, indem Sie auf der Ansichtsseite eines Channels oder in der Sektion *Channels* auf den Button *Neu* klicken. Der Content Manager zeigt das Formular *Konfiguration – Channels – Anlegen* an:

Mit dem Formular können die folgenden Channel-Eigenschaften festgelegt werden.

Name

Geben Sie hier bitte den Namen des Channels ein. Mit Hilfe von Punkten im Namen können Channel-Hierarchien gebildet werden. So ist beispielsweise der Channel *wetter.europa.deutschland* im Channel *wetter.europa* und dieser wiederum im Channel *wetter* enthalten.

Titel

Geben Sie in dieses Eingabefeld bitte den Titel des Channels in der Sprache an, die Sie in Ihren persönlichen Einstellungen konfiguriert haben. Beachten Sie bitte, dass der Titel eines Channels einschließlich aller Zeichenumwandlungen (beispielsweise von ANSI nach UTF-8) eine maximale Länge von 250 Zeichen haben kann. Wenn Ihr Content Management Server für mehrsprachigen Einsatz eingerichtet ist, so können Sie zum Formular *Konfiguration – Channels – Bearbeiten – Mehrsprachige Titel* wechseln, indem Sie auf den Button *Andere Sprachen* klicken:

Tragen Sie hier in die entsprechenden Felder die Titel in den jeweiligen Landessprachen ein. Bestätigen Sie Ihre Eingaben bitte mit *OK*. Welchen Titel der Content Management Server in der HTML-Benutzeroberfläche anzeigt, hängt von der Sprache ab, die ein Benutzer in seinen persönlichen Einstellungen konfiguriert hat. Gibt es keinen Titel in der Sprache, die sich der Benutzer eingestellt hat, so wird der Name des Channels als Channeltitel angezeigt.

Klicken Sie auf **OK**, um die Channeldefinition abzuschließen. Der Content Manager zeigt anschließend die Ansichtseite des Channels an.

5.5.2 Channels bearbeiten oder löschen

Um die Eigenschaften eines Channels zu bearbeiten, machen Sie den gewünschten Channel bitte zunächst mit Hilfe der Suchfunktion in der Sektion *Channels* ausfindig und klicken anschließend auf den Namen des Channels im entsprechenden Listeneintrag. Auf der daraufhin angezeigten Ansichtseite des Channels klicken Sie bitte auf den Button *Bearbeiten*, um das Formular *Konfiguration – Channels – Bearbeiten* zu öffnen:



Konfiguration - Channels - Bearbeiten
Den Channel "economy" bearbeiten

Name: economy
Titel: Wirtschaft
Andere Sprachen

Ok Abbrechen

Bitte entnehmen Sie die Bedeutung der Eingabebelemente dem Abschnitt [Channels definieren](#).

Klicken Sie auf **OK**, um die geänderten Einstellungen des Channels zu übernehmen. Der Content Management Server zeigt anschließend wieder die Ansichtseite des Channels an.

Bitte beachten Sie, dass Dateien nicht mehr eingereicht oder freigegeben werden können, wenn Sie einen Channel löschen, denen die Dateien zugeordnet sind.

Um einen Channel zu löschen, machen Sie ihn bitte in der Sektion *Channels* mit Hilfe der Suchfunktion ausfindig und klicken auf den Löschen-Button am rechten Rand des Listeneintrags. Der Channel kann auch auf seiner Ansichtseite gelöscht werden. Zu diesem Zweck klicken Sie im Suchergebnis auf den Namen des gewünschten Channels und auf seiner Ansichtseite wiederum auf den Button *Löschen*. Der Content Management Server bittet Sie nun, den Löschvorgang zu bestätigen:



Konfiguration - Channels - Löschen
Channel "economy" löschen

Wollen Sie diesen Channel wirklich löschen?

Name: economy
Titel: Wirtschaft

Ok Abbrechen

Der Channel wird endgültig gelöscht, wenn Sie auf **OK** klicken. Nicht freigegebene Dateien, die diesem Channel zugeordnet sind, werden dadurch unvollständig.

6

6 Layouts

6.1 Funktion der Layouts

Layouts dienen dazu, eine Website mit homogenem Design effizient zu realisieren. Häufig verwendete Elemente wie Logos und Navigationselemente müssen nur einmal in ein Layout aufgenommen werden und können anschließend bei allen Dateien verwendet werden, die sich in Ihrer Ordnerhierarchie auf oder unterhalb der gleichen Ebene befinden wie die Layoutdatei.

Für den Export jeder Datei vom Typ *Ordner* und *Dokument* (deren Dateiendung der MIME-Typ `text/html` zugeordnet ist), verwenden der Content Manager und die Template Engine ein Layout als Basislayout. Ferner wird beim Export einer Datei nur dessen freigegebene Version berücksichtigt. Welches Basislayout verwendet wird, hängt beim Content Manager vom Benutzerkonfigurationseintrag `defaultTemplate` ab, während bei der Template Engine der Systemkonfigurationseintrag `defaultTemplate` den Namen des Basislayouts bestimmt. Dadurch, dass die beim Export zu verwendende Basislayoutdatei konfiguriert werden kann, können mehrere Layouts parallel verwaltet werden. Indem das gewünschte Layout vor dem Export als Basislayout festgelegt wird, lässt sich ein Layout spontan und bedarfsweise aktivieren.

Die Basislayoutdatei wird wie ein Programm abgearbeitet. Die resultierende Website wird erzeugt, indem das Programm für jede freigegebene Datei genau einmal abgearbeitet wird. Dabei beziehen sich spezielle Anweisungen im Basislayout auf die Datei, die gerade exportiert wird. Selbstverständlich können Sie nicht nur diese Anweisungen, sondern auch Standard-HTML-Tags in Layouts verwenden.

Mit den Anweisungen in einem Layout wie dem Basislayout können Sie auf die Feldwerte der gerade exportierten Datei zugreifen, um daraus `meta`-Tags, Links und andere HTML-Elemente zu erzeugen. Ferner stehen Ihnen Anweisungen zur Verfügung, mit denen Sie automatisch Inhaltsübersichten (Linklisten), eine Mikronavigation, eine Sitemap oder andere Link-Strukturen generieren lassen können.

Die Anweisungen werden mit dem Schlüsselwort `NPSOBJ` eingeleitet, das in der gleichen Weise verwendet wird wie ein HTML-Tag. Während eine Layoutdatei abgearbeitet wird, werden die darin enthaltenen `NPSOBJ`-Tags vom Content Management Server und von der Template Engine zu einer Zeichenkette ausgewertet, die anstelle des gesamten `NPSOBJ`-Tags in die Ausgabedatei geschrieben wird. So wird die folgende Anweisung durch den Wert des kundenspezifischen Feldes *abstract* ersetzt. Dieser Wert wird also in die erzeugte Webseite eingefügt:

```
<NPSOBJ insertvalue="var" name="abstract">
```

Das Basislayout ist im Grunde genommen lediglich ein spezielles Layout, das aufgrund der Anweisungen, die es enthält, beim Export als initiales Layout fungieren kann. Neben dem Basislayout

können Sie beliebig viele weitere Layouts verwenden, um Ihre Website oder untergeordnete Webseiten homogen zu gestalten.

Mit einer NPSOBJ-Anweisung ist es möglich, in einer Layoutdatei weitere oder dieselbe Layoutdatei zu verwenden. Beispielsweise kann in einem Basislayout jeweils eine Layoutdatei zur Erzeugung der Navigationselemente am Kopf und am Fuß der Seite aufgerufen werden, um das Basislayout möglichst allgemein zu halten und damit universell verwendbar zu machen. Sämtliche Layoutdateien, die im Laufe eines Exports verwendet werden, nennt man einen Layoutsatz.

Die Wirkungsweise Ihrer Layouts können Sie jederzeit mit der [Vorschaufunktion](#) überprüfen.

Hauptinhaltslayouts

Ein Hauptinhaltslayout ist ein Layout, das in einer Vorlage referenziert wird (siehe [Vorlagen definieren](#)). Es erzeugt bei der Vorschau und beim Export den Hauptinhalt jeder Datei mit dieser Vorlage. Hauptinhaltslayouts dienen dazu, das Aussehen von Dokumenten festzuschreiben und konsequent zu vereinheitlichen. Dem entsprechend kann ein Hauptinhalt weder mit Tcl-Befehlen noch mit einem Editor in der HTML-Schnittstelle modifiziert werden, wenn in der Vorlage der betreffenden Datei ein Hauptinhaltslayout angegeben wurde. Es ist in diesem Fall ebenfalls nicht möglich, den Hauptinhalt aus einer Datei zu importieren.

Layouts für den Hauptinhalt sind insbesondere dazu geeignet, uniforme Dokumente wie Datenblätter oder Produktbeschreibungen zu erzeugen. Solche Dokumente bestehen hauptsächlich aus Feldwerten, die Sie mit Hilfe eines Hauptinhaltslayouts in der gewünschten Weise anordnen und formatieren können.

Das Hauptinhaltslayout, mit dem freigegebene Versionen exportiert oder in der Vorschau angezeigt werden sollen, wird zum Zeitpunkt der Freigabe direkt in den Versionen gespeichert. Die betreffenden Dateien müssen daher erneut freigegeben werden, nachdem ein (anderes) Hauptinhaltslayout in der Vorlage eingetragen oder das Hauptinhaltslayout aus ihr entfernt wurde. Dies gilt auch, wenn das Hauptinhaltslayout selbst modifiziert wurde. Auf Arbeitsversionen dagegen wirken sich Änderungen an Hauptinhaltslayouts unmittelbar aus.

Wenn Hauptinhaltslayouts auf einem inkrementell exportierten Live-Server verwendet werden, so beachten Sie bitte folgendes: Eine Layoutdatei für einen Hauptinhalt muss sich immer im gleichen Ordner oder in einem Ordner befinden, der über den Dateien liegt, die mit der Layoutdatei exportiert werden. Andernfalls werden die referenzierenden Dateien nicht neu exportiert, wenn die Layoutdatei aktualisiert wird.

6.2 Ein Layoutkonzept entwerfen

Bevor Sie im Redaktionssystem von Infopark CMS Fiona Layouts erstellen, sollten Sie immer ein Konzept erarbeiten, in dem die Struktur der Website, das Layout sowie eventuelle weitere Verwendungsmöglichkeiten der Inhalte festgelegt werden. Dabei sollten insbesondere folgende Aspekte berücksichtigt werden:

- Welche Benutzer sind für die Erstellung und Pflege der Layouts verantwortlich?
- Werden mehrere Layoutsätze benötigt, weil die Website beispielsweise für unterschiedliche Zielgruppen individuell erzeugt werden soll?
- Gibt es ein globales Layout oder haben einzelne Bereiche der Ordnerhierarchie individuelle Layouts?
- An welchen Stellen des Layouts müssen weitere Layouts für wechselnde Elemente der Seite vorgesehen werden (Titelgrafiken, Navigationseinträge, automatisch eingefügte Feldwerte u.ä.)?

- An welchen Stellen des Layouts sollen automatisch generierte Navigationseinträge erscheinen?
- Welche Informationen benötigen die Redakteure, damit die in den Layouts vorgesehenen Mechanismen funktionieren (z.B. zwingend anzugebende Feldwerte)?

Möglichst großen Wert sollten Sie auf die Flexibilität des Layoutkonzepts legen, um den Pflegeaufwand und den Aufwand bei nachträglichen Anpassungen zu minimieren. Das heißt insbesondere, Layoutdateien konsequent zu nutzen: an allen Stellen, bei denen künftig Änderungen zu erwarten sind, sollte eine Layoutdatei vorgesehen werden.

Wird bereits in der Basislayoutdatei der Hauptinhalt einer Datei mit

```
<npsobj name="body" insertvalue="var"/>
```

eingefügt, so gibt es in Unterordnern keine Möglichkeit mehr, diesen Bereich der Seite mittels lokaler Layoutdateien anders zu gestalten. Deshalb ruft man zunächst nur eine Layoutdatei als Platzhalter für den Hauptinhalt auf, etwa in der Form

```
<npsobj name="localtemplate" insertvalue="template"/>
```

und legt in der gleichen Hierarchieebene eine Layoutdatei mit dem Namen *localtemplate* und dem Inhalt

```
<npsobj name="body" insertvalue="var"/>
```

an. Dadurch können bei Bedarf je Ordner in einem dedizierten *localtemplate* weitere Formatierungen definiert werden.

6.3 Layouts bearbeiten

Bevor Sie eine bereits freigegebene Layoutdatei bearbeiten, sollten Sie eine Kopie von ihr anlegen, auf die Sie zurückgreifen können, wenn die bearbeitete Layoutdatei nach der Freigabe nicht wunschgemäß funktioniert. Diese Sicherheitsmaßnahme erübrigt sich, wenn bei Ihrem Content Management Server die Archivierung eingeschaltet ist.

Sie können Layoutdateien in der gleichen Weise anlegen und bearbeiten wie Ordner oder Dokumente. Auch werden Links auf andere Dateien in Layouts nicht anders behandelt als Links in Ordnern und Dokumenten. Anders als bei Dateien dieser beiden Typen verwendet der Content Management Server jedoch immer den gesamten Hauptinhalt einer Layoutdatei, also nicht nur den HTML-Code im *body*-Tag.

Der Content Management Server und die Template Engine unterstützen sogenannte *empty element tags* entsprechend der XML-Spezifikation. Solche Tags sind gleichzeitig Elemente ohne Inhalt. Sie weisen einen Schrägstrich vor dem Tag-Ende-Zeichen auf:

```
<empty-element-tag />
```

Sie sind äquivalent zu Elementen, die ein öffnendes und ein schließendes Tag und keinen Inhalt haben, also:

```
<element-tag></element-tag>
```

Empty-element-tags bleiben beim Export als solche erhalten, d. h. sie werden nicht in Elemente mit öffnendem und schließendem Tag umgewandelt.

6.4 Aufbau einer Layoutdatei

6.4.1 XML- und XHTML-Kennzeichnung

Wenn Inhalte als XML- oder XHTML-Dokument exportiert werden sollen, muss das Dokument zunächst entsprechend gekennzeichnet werden. Zu diesem Zweck exportiert man als erstes eine entsprechende XML-Kennung wie typischerweise

```
<?xml version="1.0"?>
```

am Anfang einer Zeile bzw. eine XHTML-Kennung, beispielsweise

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Beide Tags müssen am Anfang des exportierten Dokuments stehen. Sie werden nicht erkannt, wenn sie in einer *processing instruction* (`<? ... >`) oder in einem SGML-Kommentar (`<!-- ... -->`) enthalten sind.

Die Kennung dient nicht nur dem Client (meist der Browser) dazu, die Art des Dokuments zu erkennen, sondern beeinflusst auch den Export so genannter *empty element tags*. Diese Tags werden als solche exportiert, wenn mindestens eine der beiden folgenden Bedingungen erfüllt ist:

- Das *empty element tag* ist in dem zu exportierenden Versionsbestandteil (in einem Feld) enthalten;
- das *empty element tag* wird von einer NPSOBJ-Anweisung generiert (beispielsweise `<npsobj insertvalue="image" name="..." />` und vorher wurde eine XML oder XHTML-Kennung exportiert.

In allen anderen Fällen werden keine *empty element tags*, sondern einfache Tags ohne abschließenden Schrägstrich exportiert.

6.4.2 HTML-Header

Im HTML-Header einer Layoutdatei werden die gewünschten *meta*-Elemente in die zu erzeugende Webseite eingefügt. Wichtig sind hier insbesondere die Dateieindung und der Zeichensatz.

Üblicherweise werden auch der Titel der exportierten Datei in den HTML-Titel der Webseite sowie weitere Versionsfelder in die korrespondierenden *Meta*-Elemente umgesetzt:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title><npsobj name="title" insertvalue="var"/></title>
<npsobj name="attribute-name" insertvalue="meta"/>
</head>
```

Für jedes in ein *meta*-Element umzusetzende Feld muss eine dedizierte NPSOBJ-Anweisung nach obigem Muster in das Layout aufgenommen werden. Bitte beachten Sie, dass in obigem Beispiel der Name des Feldes zum Namen des *meta*-Elements wird; um den Namen des *meta*-Elements frei (d.

h. unabhängig vom Feldnamen) bestimmen zu können, verwenden Sie bitte ein NPSOBJ-Tag nach folgendem Muster:

```
<npsobj insertvalue="meta" name="meta-name" content="attribute-name" />
```

Damit Sie Ihre Layouts universell verwenden können, sollten Sie die als meta -Elemente vorgesehenen Felder in einer zentralen Layoutdatei auslesen. Diese Layoutdatei können Sie mit der Anweisung

```
<npsobj name="template-name" insertvalue="template"/>
```

in das Basislayout oder andere Layouts aufnehmen.

6.4.3 Style sheets

Wenn Sie in Ihre Website Style sheets einbinden möchten, so verwenden Sie bitte die Form `<link rel="stylesheet" type="text/css" href="styles.css">`, damit die betreffenden Links in die Linkverwaltung aufgenommen werden. Ferner ist es wichtig, das Tag-Attribut `title` anzugeben und für alle zusammen gehörenden Style sheets den gleichen Titel zu verwenden. Dadurch kann der Browser Ihnen verschiedene Stile unter jeweils genau einem Titel anbieten. Beispiel:

```
<link rel="stylesheet" type="text/css" href="typo.css" media="screen" title="Standard" />
<link rel="stylesheet" type="text/css" href="navi.css" media="screen" title="Standard" />
```

6.4.4 HTML-Hauptinhalt und andere Feldwerte

Auf den Hauptinhalt der exportierten Datei greifen Sie wie auf jedes andere Feld zu. Feldwerte werden mit einem NPSOBJ-Tag nach folgendem Muster ausgelesen:

```
<npsobj name="attribute-name" insertvalue="var"/>
```

Dabei ist es unerheblich, ob Sie auf Dateifelder (wie `name`), vorgegebene Versionsfelder (wie `lastChanged` oder `title`) oder kundenspezifische Versionsfelder zugreifen.

Den Hauptinhalt fügen Sie – üblicherweise in einem vom Basislayout referenzierten Layout – mittels folgender Anweisung in die resultierende Ausgabedatei ein:

```
<body>
  <npsobj name="body" insertvalue="var"/>
</body>
```

Bitte beachten Sie, dass es nicht möglich ist, den Wert eines HTML-Tag-Attributs mit einer NPSOBJ-Anweisung, also beispielsweise folgendermaßen zu setzen:

```
<!-- Dies ist nicht zulässig-->
<A href="/" title="<npsobj insertvalue="var" name="myAttribute"/>">
Click here</A>
```

Verwenden Sie zu diesem Zweck bitte eine [@-Referenz](#) oder einen Formatter, mit dem das vollständige HTML-Element erzeugt wird (siehe [npsobj insertvalue var](#)).

Bitte beachten Sie auch, dass NPSOBJ-Anweisungen nur ausgewertet werden, wenn sie sich im Hauptinhalt von Layouts, Dokumenten oder Ordnern befinden und diese Dateien den MIME-Typ `text/html` haben.

Ferner dürfen die exportierten Seiten (d.h. die Layouts und die darin eingefügten Feldwerte), die über den Portal Manager ausgeliefert werden, keinen Velocity-Code enthalten, denn der Portal Manager wertet diesen Code aus, wodurch die Seite zerstört werden kann oder gar nicht angezeigt wird. Wenn Sie Velocity-Code in eine Seite einfügen möchten, so ersetzen Sie alle #-Zeichen durch `#` und alle \$-Zeichen durch `$`.

Die folgende Tabelle enthält eine Auswahl der verfügbaren Datei- und Versionsfelder. Der volle Sprachumfang ist im [Syntax-Teil](#) beschrieben. Bitte beachten Sie, dass die Schreibweise eines Feldnamens in einem NPSOBJ-Tag exakt der tatsächlichen Schreibweise des Feldnamens entsprechen muss – Groß- und Kleinschreibung wird also berücksichtigt.

Feldname	Feldwert
body	Der Hauptinhalt der freigegebenen Version der exportierten Datei.
contentType	Die Dateieindung.
id	Die Datei-ID.
name	Der Name der Datei.
path	Der Pfad der aktuellen Datei.
prefixPath	Wenn die aktuelle Datei ein Ordner ist, wird der Pfad, gefolgt von dem Zeichen „/“ geliefert, andernfalls wird der Pfad geliefert.
title	Titel der aktuellen Datei.
visibleName	Wenn die aktuelle Datei ein Dokument ist, wird der Dateiname, gefolgt von der Dateieindung geliefert, ist es ein Ordner, wird nur der Dateiname geliefert.
visiblePath	Bei Dokumenten der komplette Pfad einschließlich Dateiname, mit dem die Datei exportiert wird; bei Ordnern, der komplette Pfad einschließlich Dateiname, gefolgt von <code>/index</code> , gefolgt von der Dateieindung des Ordners.
lastChanged	Das Datum der letzten Änderung der aktuellen Datei. Änderungen im Inhalt der Datei werden hierbei nicht berücksichtigt. Beim Export einer freigegebenen Datei ist die letzte Änderung die Freigabe.
validFrom	Das Datum, ab dem der Inhalt der Datei gültig ist. Dieses Feld wird bei Layouts ignoriert. Der Wert von <code>validUntil</code> muss leer oder größer als der Wert von <code>validFrom</code> sein.
validUntil	Das Datum, bis zu dem der Inhalt der Datei gültig ist. Wenn dieser Wert leer ist, ist das Dokument unbegrenzt gültig. Dieses Feld wird bei Layouts ignoriert.

Das Format, in dem Datumswerte beim Export und in der Vorschau geliefert werden, wird normalerweise durch den ersten Wert im Systemkonfigurationseintrag [export.validDateOutputFormats](#) bestimmt. Mit dem zusätzlichen Tag-Attribut `format="formatName"` im NPSOBJ-Tag können Sie jedoch die Formatierung des Datums beeinflussen, indem Sie als `formatName` einen der anderen Formatnamen angeben, die im genannten Systemkonfigurationseintrag definiert sind.

Bei Mehrfachauswahl-Feldern, deren Wert aus mehreren Elementen besteht, muss mit `separator="string"` eine Zeichenkette als Trenner im NPSOBJ-Tag angegeben werden. Der Content

Management Server gibt den Trenner zwischen den einzelnen Elementen des Feldwerts aus. Als Separator können Sie auch eine leere Zeichenkette angeben.

Das folgende Beispiel zeigt, wie der Titel eines Dokuments (festes Versionsfeld `title`) und eine Schlüsselwörterliste (kundenspezifisches Versionsfeld `keywords`) als HTML-meta-Elemente verwendet sowie das Datum der letzten Änderung (festes Versionsfeld `lastChanged`) und der Wert des kundenspezifischen Feldes `autor` automatisch am Ende der erzeugten HTML-Seite eingefügt werden.

```
<head>
  <title><npsobj name="title" insertvalue="var"/></title>
  <npsobj name="keywords" insertvalue="meta" separator=","/>
</head>
<body>
  <npsobj name="body" insertvalue="var"/>
  <hr>
  <font SIZE="-2">
    Zuletzt geändert am
    <npsobj name="lastChanged" insertvalue="var">
  </npsobj>.
  Autor: <npsobj name="autor" insertvalue="var"/>
  </font>
</body>
```

Es ist möglich, HTML-Text nur unter bestimmten Bedingungen zu generieren, beispielsweise nur beim Export von Ordnern oder abhängig vom Wert eines Feldes.

6.4.5 Automatische Navigationseinträge

Mit NPSOBJ-Tags in Layouts können Sie automatisch Links zu anderen exportierten Webseiten erzeugen lassen:

Link auf die vorherige Seite

```
<npsobj name="previous" insertvalue="anchor">Zur&uuml;ck</npsobj>
```

Link auf die nächste Seite

```
<npsobj name="next" insertvalue="anchor">Vor</npsobj>
```

Link auf die übergeordnete Seite

```
<npsobj name="parent" insertvalue="anchor">Hoch</npsobj>
```

Welche Datei die vorherige oder nächste ist, wird vom Content Management Server anhand des Sortierschlüssels ermittelt, der in dem darüber liegenden Ordner verwendet wird (siehe Sortierung der Einträge). Die darüber liegende Seite entspricht dem Ordner, der in Ihrer Ordnerhierarchie unmittelbar über dem exportierten Datei liegt, für die das Navigationselement erzeugt wird.

Anstatt der oben in den Beispielen verwendeten Navigationstexte *Zurück*, *Vor* und *Hoch* können Sie auch Bilder oder Felder der Zieldateien verwenden, indem Sie die Texte durch einen Verweis auf ein Bild bzw. durch ein NPSOBJ-Tag ersetzen, mit dem der Wert des Feldes ausgelesen wird. Beispielsweise verwenden Sie den Titel der Link-Ziel-Datei als Verweistext, indem Sie NPSOBJ-Tags nach folgendem Muster verwenden:

```
<npsobj name="previous" insertvalue="anchor">
  <npsobj name="previous.title" insertvalue="var"/>
</npsobj>
```

6.4.6 Mikronavigation

Eine Mikronavigation ist ein Pfad als HTML-Text, in dem jedes Element bis auf das letzte mit der entsprechenden Webseite verlinkt ist. Mit einem NPSOBJ-Tag können Sie Mikronavigationen automatisch erzeugen lassen, wobei Sie die Anzahl der erzeugten Elemente festlegen können.

```
<npsobj micronavigation="attribute-name" levels="3">delimiter</npsobj>
```

Wenn der Content Management Server die `micronavigation`-Anweisung abarbeitet, bildet er den Pfad vom Basisordner bis zu dem Ordner, die unmittelbar über der aktuellen Datei liegt, wobei für jedes Element ein `a-href`-Link erzeugt wird. Als `href` wird der Link auf die entsprechende Ordner-Indexseite erzeugt, und als Linktext wird der Wert des Feldes `attribute-name` im Kontext des jeweiligen Zielordners verwendet.

Die einzelnen Elemente werden aneinandergereiht und durch die angegebene Zeichenfolge `delimiter` getrennt. Sie können beliebige Trennzeichen verwenden oder auch HTML-Code wie Zeilenumbrüche, Links auf Grafiken etc. einsetzen.

Mit dem Tag-Attribut `levels` können Sie festlegen, wieviele Elemente des Pfades in die Mikronavigation aufgenommen werden sollen. Ist `levels` positiv, so werden die Elemente beginnend beim Basisordner verwendet; bei einem negativen Wert werden die Elemente verwendet, die sich im Pfad vor der aktuellen Datei befinden. So bewirkt `levels="1"`, dass nur der Basisordner verwendet wird; bei `levels="-1"` besteht die Mikronavigation nur aus dem jeweils darüber liegenden Ordner. `levels="0"` entspricht einem nicht angegebenen `levels`-Attribut und bewirkt, dass der gesamte Pfad in die Mikronavigation aufgenommen wird.

Das folgende Beispiel bildet eine auf drei Elemente beschränkte Mikronavigation aus den Titeln der übergeordneten Ordner. Die Einträge werden dabei durch einen Schrägstrich voneinander getrennt:

```
<npsobj micronavigation="title" levels="3"> / </npsobj>
```

Dieses NPSOBJ-Tag erzeugt auf den Webseiten, die aus der Ordnerhierarchie `/de/company/press/archive` exportiert wurden, die folgenden Mikronavigationen:

Angezeigte Datei	Erzeugte Mikronavigation
/de	Basisordner-Titel
/de/company	Basisordner-Titel / <i>Deutsch</i>
/de/company/press	Basisordner-Titel / <i>Deutsch</i> / <i>Das Unternehmen</i>
/de/company/press/archive	Basisordner-Titel / <i>Deutsch</i> / <i>Das Unternehmen</i>

Alle Einträge in der erzeugten Mikronavigation sind mit den Seiten verlinkt, die aus den jeweiligen Ordnern erzeugt wurden.

Wenn Sie möchten, dass die Mikronavigation stets nur die letzten der aktuellen Datei übergeordneten Ordner enthält, so muss das Attribut `levels` mit einem negativen Wert belegt werden:

```
<npsobj micronavigation="title" levels="-2"> / </npsobj>
```

Auf den Webseiten, die aus der Ordnerhierarchie */de/company/press/archive* exportiert wurden, erzeugt dieses NPSOBJ-Tag die folgenden Mikronavigationen (eine Mikronavigation kann an beliebiger Stelle innerhalb des `body`-Elements einer Layoutdatei eingesetzt werden):

Angezeigte Datei	Erzeugte Mikronavigation
/de	Titel des Basisordners
/de/company	Titel des Basisordners / <i>Deutsch</i>
/de/company/press	<i>Deutsch / Das Unternehmen</i>
/de/company/press/archive	<i>Das Unternehmen / Presse</i>

7

7 Automatisch generierte Inhaltsübersichten

7.1 Anforderungen an Startseiten

Die Startseiten einer Website spielen bei der Benutzerführung eine besondere Rolle. Bei der Konzeption und Gestaltung von Startseiten sollten Sie deshalb folgende Anforderungen berücksichtigen:

- Auf jeder Startseite sollte der Besucher einen möglichst vollständigen Überblick über Inhalt und Struktur der untergeordneten Bereiche erhalten.
- Dem Besucher sollten direkte Links zu den untergeordneten Dokumenten mit ausreichender Inhaltsbeschreibung angeboten werden.
- Die Auflistung der untergeordneten Dokumente muss aktuell und vollständig sein.

Diese Kriterien lassen sich mit vollständig manuell gepflegten Inhaltsübersichten meistens nur unzureichend erfüllen. Insbesondere in Bereichen mit häufig wechselnden Inhalten (News, Pressemitteilungen, Veranstaltungskalender, Messevorschauen u.ä.) ist die Pflege einer Inhaltsliste arbeitsaufwändig und fehleranfällig. Deshalb stellt der Content Management Server für solche Aufgaben NPSOBJ-Anweisungen zur Verfügung, mit denen Inhaltsübersichten automatisch erzeugt werden können. Ihre wesentlichen Vorteile:

- Nach der einmaligen Erstellung entfällt der Pflegeaufwand für die Inhaltsübersicht komplett.
- Die Einträge der Inhaltsübersicht erscheinen automatisch auf der Startseite, nachdem eine untergeordnete Datei erstellt und freigegeben wurde. Nachdem eine Datei gelöscht wurde, erscheint sie nicht mehr in der Übersicht.
- Bestimmte Informationen, wie Erstellungsdatum, Autor usw., können automatisiert veröffentlicht werden, indem die entsprechenden Felder ausgelesen werden.
- Wenn die Felder *Gültig ab* und *Gültig bis* mit Werten versehen wurden, erscheint die Datei nur während dieses Veröffentlichungszeitraums in der Inhaltsübersicht.

7.2 Bestandteile automatisch generierter Inhaltsübersichten

7.2.1 Aufbau einer automatisch generierten Inhaltsübersicht

Eine automatisch generierte Inhaltsübersicht wird durch eine Kombination von Standard-HTML-Elementen (zur Formatierung der Liste) und NPSOBJ-Anweisungen erzeugt:

```
<ul>
  <npsobj list="toclist">
    <li>
```

```

    <npsobj name="self" insertvalue="anchor">
    <npsobj name="title" insertvalue="var"/>
    </npsobj>
  </li>
</npsobj>
</ul>

```

Dieser Code bewirkt folgendes: Die ganze Inhaltsübersicht wird als ungeordnete HTML-Liste definiert:

```

<ul>
  ...
</ul>

```

Alle untergeordneten Ordner, Dokumente und Ressourcen des exportierten Ordners sollen berücksichtigt werden. Alle Einträge werden zusammengefasst als automatisch generierte Inhaltsübersicht auf der Seite platziert:

```

<npsobj list="toclist">
  ...
</npsobj>

```

Jeder Eintrag wird als Listenelement formatiert:

```

<li>
  ...
</li>

```

Für jeden Eintrag wird ein Link auf die jeweilige Datei erzeugt:

```

<npsobj name="self" insertvalue="anchor">
  ...
</npsobj>

```

Als verlinkte Texte werden die Titel der Dateien verwendet:

```

<npsobj name="title" insertvalue="var"/>

```

Wie die automatisch generierte Inhaltsübersicht aussieht, können Sie jederzeit in der Vorschau prüfen. Bitte beachten Sie, dass in Inhaltsübersichten, die mit dem Schlüsselwort `toclist` erzeugt wurden, nur freigegebene und gültige Dokumente, Ordner und Ressourcen erscheinen. Um auch Bilder einzubeziehen, verwenden Sie `children`.

7.2.2 NPSOBJ-Parameter für automatisch generierte Inhaltsübersichten

7.2.3 Anzahl der Einträge in Inhaltsübersichten festlegen

Mit welcher Datei eine automatisch generierte Inhaltsübersicht beginnt und wieviele Elemente sie enthält, können Sie optional durch Parameter im NPSOBJ-Tag festlegen. Sie können damit beispielsweise eine Liste aktueller Pressemitteilungen auf eine maximale Anzahl beschränken.

Möchten Sie die Übersicht beispielsweise erst mit der fünften Datei beginnen lassen und die Liste auf drei Einträge beschränken, so verwenden Sie folgenden Code:

```
<npsobj list="toclist" start="5" length="3">
...
</npsobj>
```

Bei negativem Parameter `start` wird die Startposition vom Ende her bestimmt. `start="-10"` und `length="3"` bewirkt also, dass von den letzten zehn Dateien die ersten drei in die Liste aufgenommen werden. Anstelle von `length` können Sie auch `end` verwenden, um den Index des letzten Elements anzugeben.

7.2.4 Andere Felder verwenden

Bei den Einträgen automatisch generierter Inhaltsübersichten sind Sie nicht auf den Titel der Dateien in einem Ordner beschränkt. Sie können jeden beliebigen Wert [vorgegebener Felder](#) und kundenspezifischer Felder in Ihre Inhaltsübersichten aufnehmen. Beispielsweise können Sie bereits auf der Indexseite Informationen über Name und Typ des Dokuments, den Autor (kundenspezifisches Feld) und das Erstellungsdatum anzeigen lassen:

```
<ul>
  <npsobj list="toclist">
    <li>
      <npsobj insertvalue="var" name="name"/>.
      <npsobj insertvalue="var" name="contentType"/><br>
      <npsobj insertvalue="var" name="Autor"/><br>
      <npsobj insertvalue="var" name="lastChanged"
        format="europeanDate"/><br><br>
    </li>
  </npsobj>
</ul>
```

Sie können statt `europeanDate` ein beliebiges Datumsformat angeben (siehe [validDateTimeOutputFormats](#)). Als weitere Felder stehen Ihnen beispielsweise `id` (Datei-ID), `objClass` (Dateivorlage), `objType` (Dateityp), `title` (Titel), `validFrom` (Gültig ab), `validUntil` (Gültig bis) und `version` (Version) zur Verfügung. Bitte beachten Sie, dass der Content Management Server bei Feldnamen zwischen Groß- und Kleinschreibung unterscheidet.

Sie können Feldwerte, die Sie in Anweisungen mit `insertvalue="var"` auslesen möchten, bei Bedarf mit einer eigenen Tcl-Prozedur formatieren. Dies ist insbesondere dann hilfreich, wenn Sie Skripte in Ihre Webseiten aufnehmen möchten. Verwenden Sie dazu das Tag-Attribut in der folgenden Weise:

```
<npsobj insertvalue="var" name="name" formatter="procedureAlias"/>
```

Setzen Sie als Wert von `formatter` den Alias-Namen einer Prozedur ein, die in der Systemkonfiguration im Eintrag [tclFormatterCommands](#) definiert ist. Beim Export und wenn Vorschauseiten erzeugt werden, wird die gesamte NPSOBJ-Anweisung durch den Rückgabewert der Prozedur ersetzt. Weitere Informationen zu `formatter` finden Sie im Abschnitt [npsobj insertvalue var](#).

7.2.5 Sortierung der Einträge

Die Einträge in automatisch generierten Inhaltsübersichten werden normalerweise aufsteigend nach Dateinamen sortiert. Sie können jedoch bei jedem Ordner dediziert bestimmen, wie die darin enthaltenen Dateien sortiert werden sollen.

Diese Einstellungen zur Sortierung sind jedoch nur wirksam, wenn in der betreffenden NPSOBJ-Anweisung kein Sortierschlüssel angegeben wurde. Sortierschlüssel werden in NPSOBJ-Tags mit den Parametern `sortkey1`, `sortkey2` und `sortkey3` (primäres, sekundäres und tertiäres Kriterium) sowie den Modifikatoren `sortmodifier1`, `sortmodifier2` und `sortmodifier3` nach folgendem Muster festgelegt:

```
<npsobj list="toclist" sortkey1="attribute-name"
  sortmodifier1="alpha ascending">
  ...
</npsobj>
```

Geben Sie statt `attribute-name` den Namen eines beliebigen Datei- oder Versionsfeldes an. Die Dateien werden im obigen Beispiel aufsteigend sortiert (`alpha ascending` ist die Voreinstellung, hätte also gar nicht angegeben werden müssen). Verwenden Sie Code nach folgendem Muster, um zu bewirken, dass die Dateien nicht alphabetisch aufsteigend, sondern numerisch absteigend sortiert werden:

```
<npsobj list="toclist" sortkey1="attribute-name"
  sortmodifier1="numeric descending">
  ...
</npsobj>
```

Die Modifikatoren `alpha` und `numeric` können wie in den obigen Beispielen jeweils mit `ascending` oder `descending` kombiniert werden. Sollte die Dateisortierung weder im NPSOBJ-Tag noch durch die Sortierschlüssel des darüber liegenden Ordners festgelegt werden, so versucht der Content Management Server das Feld zu verwenden, das mit dem Systemkonfigurationseintrag [content.sortKey](#) festgelegt wurde. Wenn auch dort kein Sortierschlüssel ermittelt werden kann, so sortiert der Content Management Server die Dateien nach ihren Namen.

Die Dateisortierung wirkt sich auch auf [automatisch erzeugte Navigationseinträge](#) aus.

7.3 Einträge steuern

In einer Inhaltsübersicht, die mit `list=toclist` erzeugt wird, erscheinen normalerweise alle untergeordneten freigegebenen Ordner, Dokumente und Ressourcen. Eine mit `list=children` erzeugte Übersicht enthält dagegen alle Dateien eines Ordners, also auch Layouts.

Mitunter möchte man aber nicht alle Dateien in die Übersicht aufnehmen, sondern nur Dateien eines bestimmten Typs oder mit einem bestimmten Feldwert. Vielleicht möchten Sie auch, dass die Dateien mit einer bestimmten Vorlage anders angezeigt werden.

Dies ist mit konditionalen Exportanweisungen möglich.

7.3.1 Den Dateityp abfragen

Die Einträge in einer automatisch generierten Inhaltsübersicht können mit Hilfe folgender Abfragen auf einen bestimmten Dateityp beschränkt werden (z.B. alle Ressourcen zum Download). Es ist ebenfalls möglich, Dateien eines bestimmten Typs aus der Inhaltsübersicht auszuschließen.

Verwenden Sie folgenden Code, um nur Dateien vom Typ *Dokument* in die Inhaltsübersicht aufzunehmen:

```
<npsobj name1="objType" value2="document" condition="isEqual">
```

```
...
</npsobj>
```

In diesem Beispiel wird mit `name1` der Name des Feldes angegeben, dessen Wert mit dem Wert der Konstanten `value2` verglichen werden soll. Die Indizes 1 und 2 sind bei binären Vergleichsoperationen notwendig, weil bei einigen dieser Operationen (`hasPrefix`, `hasSuffix` und `matches`) die Reihenfolge der Operanden bedeutungstragend ist. Da hingegen in HTML die Reihenfolge der Attribute eines Tags keine Bedeutung hat, muss sie auf andere Weise definiert werden können. Diesen Zweck erfüllen die Indizes.

Analog zum obigen Beispiel werden mit `value2="publication"` nur Ordner und mit `value2="generic"` nur Ressourcen in der Inhaltsübersicht erscheinen. Um nur Ordner und Dokumente anzeigen zu lassen, schließen Sie Dateien vom Typ *Ressource* (`generic`) aus:

```
<npsobj name1="objType" value2="generic" condition="isNotEqual">
...
</npsobj>
```

Alternativ können Sie mit dem Parameter `negate` eine Bedingung umkehren. Der folgende Code hat das gleiche Ergebnis wie der Code im vorausgehenden Beispiel:

```
<npsobj name1="objType" value2="generic" condition="isEqual" negate>
...
</npsobj>
```

Wenn Sie zwei Feldwerte miteinander vergleichen möchten, so verwenden Sie bitte als NPSOBJ-Parameter `name1` und `name2`.

7.3.2 Feldwerte abfragen

Dateien können auch abhängig von Feldwerten in automatisch generierte Inhaltsübersichten aufgenommen oder formatiert werden. Der folgende Code prüft, ob das kundenspezifische Feld `ShowInTocList` einen Wert hat.

```
<npsobj name="ShowInTocList" condition="isNotEmpty">
...
</npsobj>
```

Mit `condition="isEmpty"` prüfen Sie, ob ein Feldwert leer ist. Wenn Sie jedoch den Wert eines Feldes wie beispielsweise `title` mit einer Zeichenkette vergleichen möchten, so verwenden Sie Code nach folgendem Muster:

```
<npsobj name1="title" value2="Ein Datei-Titel" condition="isEqual">
...
</npsobj>
```

Stimmt der Wert des Feldes exakt mit der angegebenen Zeichenkette überein, so arbeitet der Content Management Server den Code vor `</npsobj>` ab. Umgekehrt können Sie mit `condition="isNotEqual"` auf Nicht-Übereinstimmung prüfen.

Ob der Wert eines Feldes mit einer bestimmten Zeichenkette beginnt oder endet, können Sie mit `hasPrefix` bzw. `hasSuffix` testen. Im folgenden Beispiel wird geprüft, ob der Titel der Datei mit den Zeichen „Datenblatt“ beginnt:

```
<npsobj name1="title" value2="Datenblatt" condition="hasPrefix">
...
</npsobj>
```

Datumsfelder abfragen

Wenn Sie Datumsfelder in konditionalen NPSOBJ-Anweisungen abfragen, so sollten Sie berücksichtigen, dass der Content Management Server den Vergleich auf die kanonische Form des Datums anwendet. Bei diesem Format ist ein Datum als 14-stellige Zeichenkette repräsentiert, die von links beginnend aus der Jahreszahl, dem Monat, dem Tag, der Stunde und der Sekunde zusammengesetzt ist, wobei bis auf die vierstellige Jahreszahl alle Angaben zweistellig sind. Beispielsweise wird das Datum 18.12.2010 als 20101218000000 gespeichert.

Verwenden Sie als Vergleichsoperator `hasPrefix`, um nur die relevanten Zeichen des gespeicherten Datums mit einer Zeichenkette zu vergleichen. Bitte beachten Sie, dass die Parameter bei den Vergleichsoperationen `hasPrefix` und `hasSuffix` nicht kommutativ sind: der Index, mit dem `name` und `value` jeweils versehen ist, bestimmt die Position des Parameters. Der Index ist obligatorisch.

Im folgenden Beispiel werden die Dateien ermittelt, die zuletzt am 08.09.2010 geändert wurden:

```
<npsobj condition="hasPrefix" name1="lastChanged" value2="20100908">...</npsobj>
```

Alle Dateien, die zuletzt im Mai 2011 geändert wurden, erhält man mit folgendem Code.

```
<npsobj condition="hasPrefix" name1="lastChanged" value2="201105">...</npsobj>
```

Bitte beachten Sie, dass jeder Datumswert im Content Management Server auf die Zeitzone GMT bezogen ist. Möchten Sie also prüfen, ob eine Datei zeitlich gültig ist, so müssen Sie als `value2` den in GMT umgerechneten Wert des Vergleichszeitpunkts verwenden.

7.3.3 Kombinierte Vergleiche

Wenn je nach Wert eines Feldes viele unterschiedliche Texte ausgegeben werden sollen, bietet es sich an, eine `switch`-Anweisung anstelle vieler `condition`-Anweisungen zu verwenden. Mit `switch` lassen sich Feldwerte effizienter vergleichen und *if-then-else*-Strukturen in Layouts verwenden. Dies zeigt das folgende Beispiel:

```
<npsobj switch="objClass">
  <npsobj casecond="isEqual" value="newspub">
    <npsobj insertvalue="template" name="newsnav" />
  </npsobj>
  <npsobj casecond="isEqual" value="archivepub">
    <npsobj insertvalue="template" name="archivenav" />
  </npsobj>
  <npsobj casecond="isEqual" value="productpub">
    <npsobj insertvalue="template" name="productnav" />
  </npsobj>
  <npsobj casecond="default">
    <npsobj insertvalue="template" name="standardnav" />
  </npsobj>
</npsobj>
```

Hier wird das Feld `objClass` nacheinander mit den Zeichenketten `newspub`, `archivepub` und `productpub` verglichen. Entspricht der Wert von `objClass` einer dieser Zeichenketten, so wird ein entsprechendes Navigationslayout inkludiert, und der Rest der `switch`-Anweisung wird übersprungen.

Hat objClass keinen der drei Werte, so wird der Inhalt der casecond-default-Anweisung abgearbeitet, weil diese Bedingung immer wahr ist.

Mit dem Feld proceed im NPSOBJ-casecond-Tag und seinen erlaubten Werten yes und no kann man steuern, ob die nächste casecond-Anweisungen in der switch-Anweisung abgearbeitet werden soll, wenn die Bedingung zutrifft. Dies ist voreingestellt nicht der Fall, d. h. proceed hat den Wert no, wenn dieses Tag-Attribut nicht angegeben wurde. Dadurch wird der Rest der switch-Anweisung ignoriert.

7.4 Beispiele für automatisch generierte Inhaltsübersichten

7.4.1 Formatierung als Folge von Absätzen

Eine automatisch generierte Inhaltsübersicht muss nicht als Liste formatiert werden. Der folgende HTML-Code erzeugt eine einfache Aneinanderreihung von Absätzen:

```
<npsobj list="toclist">
  <p>
    <npsobj name="self" insertvalue="anchor">
      <npsobj name="title" insertvalue="var"/>
    </npsobj>
  </p>
</npsobj>
```

7.4.2 Weitere Feldwerte aufnehmen

In diesem Beispiel werden automatisch das Erstellungsdatum der Datei sowie der Wert eines kundenspezifischen Feldes Abstract mit ausgelesen, durch einen Zeilenumbruch vom Titel getrennt und unterschiedlich formatiert:

```
<npsobj list="toclist">
  <p>
    <!-- Erstellungsdatum einfügen-->
    <font size="-1">
      Erstellt am: <npsobj name="validFrom" insertvalue="var"/>
    </font>
    <br>

    <!-- Verlinkten Titel einfügen-->
    <b><npsobj name="self" insertvalue="anchor">
      <npsobj name="title" insertvalue="var"/>
    </npsobj></b>
    <br>

    <!-- Abstract einfügen-->
    <i>
      <npsobj name="Abstract" insertvalue="var"/>
    </i>
  </p>
</npsobj>
```


7.4.3 Formatierung als zweispaltige Tabelle

Der folgende Code erzeugt eine zweispaltige Tabelle, bei der in der ersten Spalte die Dateititel und in der zweiten Spalte der Wert des kundenspezifischen Feldes `Autor` aufgeführt wird:

```
<table>
  <npsobj list="toclist">
    <tr><td>
      <!-- Verlinkten Titel einfügen-->
      <b><npsobj name="self" insertvalue="anchor">
        <npsobj name="title" insertvalue="var"/>
      </npsobj></b>
    </td>
    <td>
      <!-- Autor einfügen-->
      <npsobj name="Autor" insertvalue="var"/>
    </td></tr>
  </npsobj>
</table>
```

Natürlich können Sie die Tabelle auch mit weiteren HTML-Elementen formatieren.

7.4.4 Trennung nach Dateityp

Eine nach weiteren Dokumenten und Ressourcen zum Download unterteilte Übersicht erzeugen Sie am besten als zwei Übersichten, die jeweils nur Dateien eines Typs enthalten:

```
<p>Weitere Dokumente:
<npsobj list="toclist">
  <npsobj name1="objType" value2="document" condition="isEqual">
    <npsobj name="self" insertvalue="anchor">
      <npsobj name="title" insertvalue="var"/>
    </npsobj>
  </npsobj>
</npsobj>

<p>Dateien zum Download:
<npsobj list="toclist">
  <npsobj name1="objType" value2="generic" condition="isEqual">
    <npsobj name="self" insertvalue="anchor">
      <npsobj name="title" insertvalue="var"/>
    </npsobj>
  </npsobj>
</npsobj>
```

7.4.5 Geschwisterdateien verlinken

Manchmal ist es nützlich, zusätzlich zum Inhalt einer Datei noch die Liste aller anderen Dateien verfügbar zu haben, die sich im gleichen Ordner befinden, also die Liste der Geschwisterdateien.

Dies können Sie mit dem folgenden NPSOBJ-Code in einer Layout-Datei erreichen. Der Code setzt zunächst mit `list="parent"` als Kontext den Ordner, in dem sich die aktuelle Datei befindet. In diesem Kontext wird eine `toclist` erzeugt, die alle Dateien in dem Ordner, also auch das aktuelle Dokument enthält. Mit Hilfe eines Vergleichs wird das aktuelle Dokument bei der Ausgabe übersprungen, so dass nur die Geschwisterdateien übrig bleiben. Verglichen werden der Name der ursprünglich exportierten Datei und der Name der in der `toclist` gerade verarbeiteten Datei. Die ursprüngliche Datei wird dabei mit `context.context` adressiert. `context` springt aus einer Liste eine Ebene höher, also zu dem Ordner, aus dem die Liste erzeugt wurde.

```

<!-- In den Ordner über dem aktuellen Dokument bewegen -->
<npsobj list="parent">
  <ul>
    <!-- Hier eine toclist erstellen -->
    <npsobj list="toclist">
      <!-- Nur Dokumente berücksichtigen -->
      <npsobj name1="objClass" value2="doc" condition="isEqual">
        <!-- Mit einem Namensvergleich das aktuelle Dokument überspringen. -->
        <npsobj name1="name" name2="context.context.name" condition="isNotEqual">
          <!-- Wenn der Name der ursprünglich exportierten Datei ungleich dem -->
          <!-- Namen des aktuellen toclist-Eintrags ist, wird ein Link erstellt. -->
          <li>
            <npsobj insertvalue="anchor" name="self">
              <npsobj name="title" insertvalue="var" />
            </npsobj>
          </li>
        </npsobj>
      </npsobj>
    </ul>
  </npsobj>

```

7.5 Inhaltsübersichten per Feldwert auswählen

Im Interesse einer möglichst einheitlichen Gestaltung der in Ihrem System verwendeten Inhaltsübersichten und der Vereinfachung der Arbeit für die Redakteure empfiehlt es sich, in einem Layout den Code verschiedener Inhaltsübersichten vorab zu schreiben und die Auswahl, welche dieser Inhaltsübersichten in einem Ordner genutzt wird, über ein kundenspezifisches Feld zu steuern. Gehen Sie dazu folgendermaßen vor:

1. Erstellen Sie ein Feld, mit dem die Art der einzufügenden Inhaltsliste festgelegt wird (siehe [Die Sektion Felder](#)). Eigenschaften dieses Feldes:

Typ: Enumeration

Name z.B.: TocListTyp

Aufzählungswerte z.B.: Liste, Absatzfolge, Tabelle ...

2. Erstellen Sie in der obersten Ordner Ebene eine Layoutdatei mit dem Namen *list* (o.ä.), das den Code für alle Inhaltsübersichten enthält. Jede Inhaltsübersicht wird von einer Bedingung umschlossen, die den Wert des Feldes TocListTyp abfragt. Beispiel:

```

<npsobj condition="isEqual" name1="TocListTyp" value2="Absatzfolge">
  <npsobj list="toclist">
    <p>
      <npsobj name="self" insertvalue="anchor">
        <npsobj name="title" insertvalue="var"/>
      </npsobj>
    </p>
  </npsobj>
</npsobj>

```

3. Ergänzen Sie alle Vorlagen, bei denen als Dateityp *Ordner* festgelegt ist, um das Feld TocListTyp und diesen vorab definierten Hauptinhalt:

```
<npsobj insertvalue="template" name="list"/>
```

Durch diese Maßnahmen wird in jeden Ordner das List-Layout als Standardinhalt eingefügt. Die Redakteure müssen nun lediglich durch Auswahl eines Wertes für das Feld TocListTyp bestimmen, welche der vorgegebenen Formatierungen und Inhalte für die Inhaltsübersicht in dem Ordner

herangezogen werden sollen. Bei Bedarf kann selbstverständlich der voreingestellte Hauptinhalt jedes Ordners geändert werden.

Dieser Mechanismus hat den erwünschten Nebeneffekt, dass alle Änderungen am Layout der Inhaltsübersichten global durch Änderung des Codes in der Layoutdatei vorgenommen werden können. Die zeitaufwändige Bearbeitung jedes einzelnen Ordners entfällt dadurch.

8

8 Linksammlungen

8.1 NPSOBJ-Tags zur Erzeugung von Linksammlungen

Außer bei Layouts können Sie zu jeder Arbeitsversion einer Datei freie Links hinzufügen – üblicherweise werden jedoch nur Ordner und Dokumente mit freien Links ausgestattet. Ziel eines freien Links kann sowohl eine andere Datei (interner Link) als auch ein externes Dokument sein.

Freie Links sind Versionsfeldern vom Typ *Linkliste* zugeordnet. Mit einer NPSOBJ-Anweisung können Sie eine Liste aller freien Links erzeugen, die einem Linklist-Feld zugeordnet sind, um daraus eine formatierte Linksammlung zu erzeugen.

Eine Linksammlung kann auf zwei Arten generiert werden, entweder als Liste oder als Tabelle. Mit dem folgenden Code fügen Sie die Links als Liste ein (`relatedLinks` wird exemplarisch als Name eines beliebigen Feldes vom Typ *Linkliste* verwendet):

```
<npsobj list="relatedLinks">
...
</npsobj>
```

Eine Tabelle erhalten Sie mit diesem Code:

```
<npsobj table="relatedLinks" columns="3" direction="vertical">
...
</npsobj>
```

Mit dem Parameter `columns` legen Sie die Anzahl der Spalten fest, `direction` bestimmt, ob die Zellen von links nach rechts (`horizontal`) oder von oben nach unten (`vertical`) gefüllt werden. In leer bleibende Zellen fügt der Content Management Server ein umbruchgeschütztes Leerzeichen (` `) ein. Zur Formatierung der Tabelle können Sie weitere Parameter wie `border`, `width`, `height` usw. angeben.

8.2 Inhalt und Formatierung von Linksammlungen

Linksammlungen bestehen üblicherweise aus verlinkten Texten. Innerhalb einer NPSOBJ-`list`- oder `table`-Anweisung können Sie mit der NPSOBJ-`insertvalue`-Anweisung für jeden freien Link in der Linkliste einen Hyperlink erzeugen. Das Ziel des Links steht Ihnen als Wert des Link-Parameters `destinationUrl` zur Verfügung; als verlinkten Text können Sie wie im folgenden Beispiel den Titel des Links (`displayTitle`) verwenden:

```
<ul>
```

```
<npsobj list="relatedLinks">
  <li>
    <npsobj name="destinationUrl" insertvalue="anchor">
      <npsobj name="displayTitle" insertvalue="var"/>
    </npsobj>
  </li>
</npsobj>
</ul>
```

Den Titel eines freien Links können Sie festlegen, wenn Sie den Link erstellen oder bearbeiten.

Im obigen Beispiel wird die Linksammlung als ungeordnete Liste formatiert, Sie können jedoch alle erlaubten HTML-Elemente zur Gestaltung der Liste verwenden.

8.2.1 Links auf Dateien auswerten

In Linksammlungen können Sie nicht nur auf die URL des Linkziels zugreifen. Ist das Linkziel eine Datei, die im Content Management Server verwaltet wird, so haben Sie zusätzlich vollen Zugriff auf diese Datei und können beispielsweise seinen Titel als verlinkten Text verwenden. Zugriff auf ein Feld der Zieldatei erhalten Sie, indem Sie den Link-Parameter `destination` verwenden und dahinter den Namen des gewünschten Feldes, getrennt durch einen Punkt, angeben. Im folgenden Beispiel wird der Wert des kundenspezifischen Feldes `abstract` abgefragt:

```
<npsobj name="destination.abstract" insertvalue="var"/>
```

So können Sie in Ihre Linksammlungen jede gewünschte Information über die Zieldatei aufnehmen. Der folgende Beispielcode erzeugt eine Linksammlung, bestehend aus den Titeln und den Erstellungsdaten der Zieldateien. Die Titel sind mit den Zieldateien verlinkt:

```
<ul>
<npsobj list="relatedLinks">
  <li>
    <npsobj name="destinationUrl" insertvalue="anchor">
      <npsobj name="destination.title" insertvalue="var"/>
    </npsobj><br>
    <npsobj name="destination.lastChanged" insertvalue="var"/>
  </li>
</npsobj>
</ul>
```

8.2.2 Elemente in Linksammlungen selektieren und sortieren

Sie können in Linksammlungen die gleichen Mechanismen wie bei Inhaltsübersichten einsetzen, um sie auf bestimmte Einträge zu beschränken (siehe auch [Einträge steuern](#)). Sollen beispielsweise nur Einträge für Dateien erzeugt werden, die sich in oder unterhalb des Ordners `/documents/public/` befinden, so können Sie dies mit einer NPSOBJ-condition-Anweisung erreichen:

```
<ul>
<npsobj list="relatedLinks">
  <npsobj name1="destination.prefixPath" value2="/documents/public/"
    condition="hasPrefix">
    <li>
      <npsobj name="destinationUrl" insertvalue="anchor">
        <npsobj name="destination.title" insertvalue="var"/>
      </npsobj>
    </li>
  </li>
</npsobj>
</ul>
```

```
</npsobj>  
</npsobj>  
</ul>
```

Es ist ebenfalls möglich, die Elemente der Linksammlung nach einem Feldwert der Zielfile zu sortieren, indem Sie im NPSOBJ-`list`- oder `table`-Tag das Tag-Attribut `sortkey1` angeben, um das primäre Sortierkriterium festzulegen. Um beispielsweise die Elemente nach dem Datum der letzten Änderung zu sortieren, verwenden Sie folgenden Code:

```
<npsobj list="relatedLinks" sortkey1="destination.lastChanged">  
...  
</npsobj>
```

Analog zum ersten Sortierkriterium kann das zweite und dritte mit `sortkey2` bzw. `sortkey3` festgelegt werden.

9

9 RSS-News-Feeds erzeugen

9.1 Konzept der RSS-Feed-Erzeugung

RSS-Feeds können in Infopark CMS Fiona generiert werden, indem News-Listen mit einem Layout exportiert werden, das das gewünschte RSS-Format erzeugt. Die News-Listen selbst entstehen dadurch, dass Sie CMS-Dateien mit Hilfe des Versionsfeldes `channels` einem Channel oder mehreren Channels zuweisen und diese CMS-Dateien dann freigeben.

Mit der NPSOBJ-Anweisung `newslst` können in Layouts alle News-Dateien oder nur solche abgefragt werden, die bestimmten Kriterien genügen – die beispielsweise zu mindestens einem von mehreren angegebenen Channels gehören. Ferner kann die maximale Anzahl der zu ermittelnden News beschränkt werden.

Zur Erzeugung von RSS-Feeds (abonnierbare News-Listen) dienen in Infopark CMS Fiona die folgenden Features:

- In den Systemeinstellungen können Channels definiert werden. Channels dienen dazu, Nachrichten thematisch zu gruppieren, also Dateien in Bezug auf deren Inhalte zu kategorisieren – ähnlich wie ein benutzerdefiniertes Feld `Warengruppe` eine Datei bezüglich der in den Inhalten beschriebenen Produkte kategorisieren würde. Die Channel-Einstellungen sind mit dem Content Navigator über das Menü *Extras > Systemeinstellungen > Konfiguration* zu erreichen.
- Die Versionen von Dateien des Typs *Ordner* und *Dokument* haben das fest eingebaute Feld `channels`. Das Feld `channels` ist vom Typ *Mehrfachauswahl*, und man kann es daher mit beliebig vielen der konfigurierten Channels belegen. Auf diese Weise werden Inhalte den Channels zugeordnet.
- Dateiformate haben das Feld `canCreateNewsItem` (als News auf dem Live-Server bereitstellen). Ist dieses Feld bei einem Format aktiviert, so werden Dateien mit diesem Format bei der Freigabe in eine interne Newsliste aufgenommen, sofern das Channels-Feld der freigegebenen Version nicht leer ist (dieses Feld also mindestens einen Channel enthält).
- Es können Listen von News erzeugt werden, die sich in beliebig festlegbaren Channels befinden. Hierfür stehen eine NPSOBJ-Anweisung und ein Tcl-Befehl zur Verfügung. Die NPSOBJ-Anweisung liefert analog zur `toclist`-Anweisung eine Kontextliste, so dass die Felder der Version ausgelesen werden können. Der Tcl-Befehl lautet [news](#).

9.2 Die `npsobj-newslst`-Anweisung verwenden

Zur Erzeugung von Newslisten in Layouts kann die NPSOBJ-`newslst`-Anweisung auf die folgenden drei Arten verwendet werden:

- Liste aller News in allen Channels:

```
<npsobj newslst="all" length="20">
  Für jede News evaluierter Text
</npsobj>
```

- Liste der News, die mindestens einem Channel zugeordnet sind, die als Wert in einem Feld des aktuellen Dokuments enthalten sind:

```
<npsobj newslst="selected" name="Channel-Feld" length="20">
  Für jede News der indirekt angegebenen Channels evaluierter Text
</npsobj>
```

Das Feld muss einen der Typen *Zeichenkette*, *Text*, *Auswahl* oder *Mehrfachauswahl* haben. Bei den Typen *Zeichenkette* und *Text* müssen die Werte kommasepariert angegeben sein. Bei Feldern vom Typ *Mehrfachauswahl* werden die Werte des Feldes als Channels verwendet.

- Liste der News, die direkt angegebenen Channels zugeordnet sind:

```
<npsobj newslst="selected" value="ch1, ch2, ..." length="20">
  Für jede News der direkt angegebenen Channels evaluierter Text
</npsobj>
```

Auch wenn eine News mehr als einem der direkt oder indirekt angegebenen Channels zugeordnet ist, ist sie höchstens einmal in der generierten Newsliste enthalten.

Erzeugte Newslisten enthalten nur News, deren Erscheinungsdatum zum Zeitpunkt des Exports in der Vergangenheit liegt. Intern enthält die Newsliste natürlich alle Dateien, denen ein Channel zugewiesen wurde und deren Vorlage die Bereitstellung der Datei als News vorsieht. Das Erscheinungsdatum entspricht dem Anlegezeitpunkt der Newsdatei, der über das Gültig-ab-Feld der Arbeitsversion beeinflusst werden kann. Eine versehentlich freigegebene News kann daher aus der generierten Newsliste entfernt werden, indem ihr Erscheinungsdatum in die Zukunft verlegt wird. In der internen Newsliste bleibt sie jedoch erhalten, sofern die News-Datei nicht gelöscht oder ihr Channels-Feld nicht geleert wird. Eine Datei wird auch aus der internen Newsliste entfernt, wenn die Channels gelöscht werden, denen die News zugeordnet ist. Der Wert des Channels-Feldes der Versionen wird dadurch jedoch nicht beeinflusst. Die News wird nicht wieder in die Newsliste eingetragen, wenn ein gelöschter Channel, dem die News zugeordnet ist, wieder angelegt wird.

Wenn die Template Engine eingesetzt wird, werden alle Dateien, die eine `NPSOBJ-newslst-`Anweisung enthalten, bei jeder Änderung an Dateien und an der Channel-Konfiguration neu evaluiert, damit die erzeugten Newslisten stets aktuell sind. (Die Dateien erhalten eine `usesAll-`[Abhängigkeit](#).)

9.3 Beispiele zum Gebrauch von News-Listen

9.3.1 Eine Liste mit den 10 neuesten News erzeugen

- Legen Sie den Channel `sitenews` an.

- Legen Sie die Dateivorlage `newsitem` an und aktivieren Sie deren Option *Auf dem Live-Server als neu markieren* (`canCreateNewsItems`).
- Legen Sie an beliebiger Stelle in der Ordnerhierarchie eine Datei `news1` mit der Vorlage `newsitem` an.
- Fügen Sie in das Layout der Startseite etwa Folgendes ein:

```
<ul>
  <npsobj newslst="all" length="10">
    <li>
      <npsobj insertvalue="anchor" destination="self">
        <npsobj insertvalue="var" name="title" />
      </npsobj>
    </li>
  </npsobj>
</ul>
```

9.3.2 Einen RSS-Feed für Politik-News einrichten

- Legen Sie den Channel `politics` an.
- Legen Sie die Dateivorlage `newsitem` an und aktivieren Sie bei ihr die Option *Auf dem Live-Server als News bereitstellen* (`canCreateNewsItems`).
- Legen Sie das Feld `description` und die Dateivorlage `feed` an und fügen Sie `description` zu `feed` hinzu. Bei `feed` nicht `canCreateNewsItems` aktivieren.
- Legen Sie den Ordner `rssfeeds` an.
- Legen Sie in `rssfeeds` ein Basislayout an, das einen RSS-Feed ausgibt (s.u.).
- Legen Sie in dem gleichen Ordner die Datei `politicsfeed` mit der Vorlage `feed` an und geben Sie in die `description` "Neues aus der Firmenpolitik" ein. Wählen Sie als Channels `politics` aus.
- Geben Sie dem Basislayout den folgenden Haupttext:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE rss PUBLIC "-//Netscape Communications//DTD RSS 0.92//EN"
  "http://my.netscape.com/publish/formats/rss-0.92.dtd">
<rss version="0.92">
<channel>
  <description>
    <npsobj insertvalue="var" name="description" />
  </description>
  <language>de</language>
  <title><npsobj insertvalue="var" name="title" /></title>
  <link>http://www.mysite.com/</link>
  <copyright>Copyright 1994-2008 Infopark AG</copyright>
  <generator>Infopark CMS Fiona 6.7.0</generator>
  <ttl>60</ttl>
  <npsobj newslst="selected" name="channels" length="20">
    <item>
      <title><npsobj insertvalue="var" name="title"/></title>
```

```

    <link>http://www.infopark.de<npsobj insertvalue="var" name="visiblePath" /></
link>
    <description>
        <npsobj insertvalue="var" name="description"/>
    </description>
</item>
</npsobj>
</channel>
</rss>

```

Für eine RSS-Datei in einer anderen Version (beispielsweise 2.0) kann die Layoutdatei entsprechend angepasst werden.

Um einen weiteren Feed mit Sportnachrichten mit der obigen Layoutdatei zu generieren:

- Richten Sie den Channel `sports` ein.
- Legen Sie im Ordner `rssfeeds` die Datei `politicsfeed` mit der Vorlage `feed` an und geben Sie in dessen Feld `description` "Neues aus dem Sport" ein. Wählen Sie als Channels `sports` aus.
- Legen Sie jetzt Dateien mit dem Format `newsitem` an und wählen Sie als Channel `sports` aus. Die Dateien erscheinen im Sports-Newsfeed.

9.3.3 Einen Newsletter versenden

- Legen Sie den Ordner `newsletters` an.
- Schreiben Sie ein Basislayout, das den Inhalt des Newsletters erzeugt, indem es beispielsweise Felder aus entsprechenden Dateien ausliest.
- Legen Sie das Dateiformat `newsletter` an.
- Legen Sie im Ordner `newsletters` die Datei `politics` mit dem Format `newsletters` an.
- Tragen Sie in den Hauptinhalt von `politics` die Email-Adressen mit einer Adresse pro Zeile ein.
- Schreiben Sie einen [Assistenten](#), der das Feld `exportBlob` von `politics` an jede Email-Adresse im Haupttext von `politics` verschickt.

10

10 Export-Variablen

In Layouts haben Sie die Möglichkeit, Export-Variablen zu verwenden, um Texte und Kontextlisten temporär zu speichern. Die gespeicherten Werte können innerhalb einer Layoutdatei beliebig oft verwendet und auch modifiziert werden. Dadurch brauchen mehrfach zu exportierende Texte, auch wenn sie umfangreicher sind, nicht mehrmals in die Layoutdatei aufgenommen oder in ein anderes Layout ausgelagert zu werden. Ihre Layouts werden dadurch übersichtlicher und sind leichter zu warten.

Export-Variablen werden mit der NPSOBJ-modifyvar-Anweisung gesetzt und modifiziert und können wie Felder ausgelesen oder in NPSOBJ-list- und table-Anweisungen angegeben werden.

10.1 Texte speichern und abrufen

Ein Text kann mit Anweisungen nach dem folgenden Muster in einer Export-Variablen abgelegt und an anderer Stelle in der Layoutdatei abgerufen werden:

```
<npsobj modifyvar="set" varname="myvar">
  Autor des Dokuments: <npsobj insertvalue="var" name="author" />
</npsobj>
...
<npsobj insertvalue="var" name="export.myvar" />
```

Dieser Code weist der Variablen `myvar` die Zeichenkette *Autor des Dokuments*; gefolgt vom Wert des Feldes `author` zu. Anschließend wird der Wert dieser Variablen ausgegeben. Es ist auch möglich, mit

```
<npsobj insertvalue="var" name="myvar" />
```

auf die Variable zuzugreifen. Existiert die Variable nicht, so wird beim Export wegen des fehlenden Präfixes `export.` jedoch eine (nicht erwünschte) Content-[Abhängigkeit](#) für den aktuellen Dateikontext erzeugt.

Während die Datei exportiert wird, lässt sich `myvar` beliebig oft auslesen, sofern diese Variable nicht mit der Anweisung

```
<npsobj modifyvar="clear" varname="myvar" />
```

gelöscht wird. Auch in inkludierten Layouts (`insertvalue="template"`) kann man auf die Variable zugreifen. Dies führt allerdings schnell zu nicht nachvollziehbaren Layouts, weil in dem inkludierten Layout nicht spontan erkennbar ist, welchen Wert die Variable hat. Aus diesem Grund sollten Variablen nur in den Layouts verwendet werden, in denen ihnen ein Wert zugewiesen wird.

Möchte man einer Variablen nur den Wert eines Feldes, d. h. keinen literalen Text zuweisen, so ist dies direkt im Tag der Anweisung möglich:

```
<npsobj modifyvar="set" varname="myvar" name="parent.title" />
```

Die Namen von Export-Variablen haben Vorrang vor den Namen von Feldern. So können Sie die Variable `toclist` anlegen und ihr eine Zeichenkette zuweisen, später jedoch nicht mit `<npsobj list="toclist">` ein Inhaltsverzeichnis generieren lassen. Mit dem Präfix `self` ist es jedoch möglich, wieder auf das gleichnamige Versions- oder Dateifeld zu verweisen (`"self.toclist"`).

10.2 Kontextlisten speichern und abrufen

Eine Kontextliste ist eine Liste von Dateien oder Links. Eine solche Liste wird normalerweise mit der `NPSOBJ-list`-Anweisung erzeugt, in der der Name eines Feldes wie `children` oder `toclist` oder auch der Name eines Linklist-Feldes angegeben wird.

Solche Listen können in Export-Variablen gespeichert werden. Dies allein hätte keine Vorteile gegenüber der Verwendung einer eingebauten Kontextliste wie `toclist` in einer `NPSOBJ-list`-Anweisung. Man kann jedoch die in einer Export-Variablen gespeicherte Kontextliste erweitern, d. h. weitere Kontextlisten an sie anhängen. Die in der Variablen gespeicherte Liste lässt sich darüber hinaus sortieren und beschneiden. Der Code im folgenden Beispiel nimmt an, dass die aktuelle Datei ein Ordner mit Unterordnern ist. In der Variablen `mylist` sollen die Dateien sämtlicher Unterordner abgelegt werden, sofern deren kundenspezifisches Feld `showInList` den Wert `YES` hat:

```
<npsobj list="toclist">
  <npsobj condition="isEqual" name1="objType" value2="publication">
    <npsobj list="children">
      <npsobj condition="isEqual" name1="showInList" value2="YES">
        <npsobj modifyvar="append" varname="mylist" name="self" />
      </npsobj>
    </npsobj>
  </npsobj>
</npsobj>
```

In der inneren Liste hängt die `modifyvar-append`-Anweisung die aktuell in der Liste bearbeitete Datei, die mit `self` referenziert wird, an `mylist` an. `self` ergibt – wie auch `parent`, `previous` und `next` – eine Kontextliste mit genau einem Element. Anstelle der inneren Liste hätte auch

```
<npsobj modifyvar="append" varname="mylist" name="children" />
```

verwendet werden können, wenn nicht die `condition`-Anweisung erforderlich gewesen wäre, um die Dateien nach `showInList` zu selektieren.

Der Code zeigt nebenbei, dass man eine `modifyvar-append`-Anweisung auch dann verwenden kann, wenn die Export-Variable noch nicht existiert. Dadurch brauchen Variablen nicht initialisiert zu werden.

Einer Export-Variablen kann man mit der folgenden Anweisung eine Kontextliste zuweisen:

```
<npsobj modifyvar="set" varname="mylist" name="toclist" />
```

Neben `toclist` können alle Namen angegeben werden, die einen einzelnen Kontext oder eine Kontextliste ergeben. Dies sind unter anderem: `self`, `parent`, `children`, die Namen von Variablen, denen eine Kontextliste zugewiesen wurde und die Namen von Linklist-Feldern.

Die Kontextlisten in Export-Variablen lassen sich mit einer `modifyvar-range`-Anweisung kürzen. So entfernt der folgende Code das erste Element (d. h. den ersten Kontext) aus der Liste `mylist`:

```
<npsobj modifyvar="range" varname="mylist" start="2" />
```

Wie bei der NPSOBJ-list-Anweisung kann zusätzlich zu `start` auch `length` oder `end` verwendet werden, um den gewünschten Ausschnitt der Liste festzulegen. Während mit `length` die Anzahl der gewünschten Elemente angegeben wird, kann mit `end` der Index des letzten Elements festgelegt werden. Der Index des ersten Elements ist 1.

Um die in einer Export-Variablen gespeicherte Kontextliste zu sortieren, verwendet man eine `modifyvar-sort`-Anweisung, die im Abschnitt npsobj modifyvar sort beschrieben ist.

Sie können die Variablen, in denen Sie Kontextlisten gespeichert haben, in `NPSOBJ-list`- und `-table`-Anweisungen angeben, um beispielsweise eine Linkliste zu erzeugen:

```
<npsobj list="export.mylist">
  <npsobj insertvalue="anchor" name="self">
    <npsobj insertvalue="var" name="title" /><br>
  </npsobj>
</npsobj>
```

Kontextlisten sind nicht auf Dateikontexte beschränkt, sondern können auch freie Links enthalten. Bei Bedarf lassen sich Datei- und Linkkontexte auch mischen. Der Code, der eine solche gemischte Liste verarbeitet, muss den Typ jedes Kontexts prüfen. Zu diesem Zweck kann man beispielsweise den Kontext in einer `switch`-Anweisung nach dem Wert eines Feldes (etwa `objType`) fragen, der bei Dateien niemals und bei Links immer leer ist.

```
<npsobj list="export.mymixedlist">
  <npsobj switch="objType">
    <npsobj casecond="isEmpty">
      Muss Link-Kontext sein
    </npsobj>
    <npsobj casecond="default">
      Muss Datei-Kontext sein
    </npsobj>
  </npsobj>
</npsobj>
```

Ab Version 6.5.0 ist es möglich, auf Kontextlisten, die nur einen einzelnen Kontext enthalten, so zuzugreifen wie auf den Kontext selbst. Es ist also nicht erforderlich, mit `npsobj list` über die Liste zu iterieren. Beispiel:

```
<npsobj modifyvar="set" varname="savedContext" name="self" />
...
<npsobj condition="isNotEmpty" name="export.savedContext">
  <npsobj insertvalue="var" name="savedContext.path" />
</npsobj>
```

10.3 Lokale Exportvariablen

Ab Version 6.7.1 können Exportvariablen als lokal definiert werden. Dadurch kann in mehreren Sublayouts derselbe Variablenname verwendet werden, ohne dass es während des Exports zu Namenskonflikten kommt. In Verbindung mit der Möglichkeit, Sublayouts Argumente zu übergeben, die dann ebenfalls als lokale Variablen vorliegen, lassen sich Layouts besser modularisieren und leichter pflegen.

Der Geltungsbereich einer lokalen Variablen ist auf das Layout beschränkt, in dem die Variable definiert wird. Weder in den aufrufenden Layouts noch in Sublayouts auf tieferen Ebenen ist die lokale Variable verfügbar.

Eine Variable kann in der definierenden Anweisung modifyvar set als lokal markiert werden, indem das Attribut `local` mit dem Wert `true` angegeben wird:

```
<npsobj modifyvar="set" varname="myVar" local="true">  
  Der Inhalt der Variablen  
</npsobj>
```

Eine existierende gleichnamige globale oder lokale Variable auf einer höheren Layout-Ebene wird immer durch eine lokale Variable verdeckt, die mit modifyvar set erzeugt wird.

Lokale Variablen, die mit modifyvar-append-Anweisungen erzeugt werden, funktionieren analog. Jedoch wird die angegebene Kontextliste an eine existierende gleichnamige globale oder lokale Variable angehängt. Bei der Anweisung `modifyvar append` werden als lokal markierte Variablen also nur dann lokal sein, wenn die Variable neu angelegt werden muss.

11

11 Weitere NPSOBJ-Anweisungen

11.1 Den Hauptinhalt einer Datei verwenden

In Layoutdateien können Sie den Hauptinhalt eines Dokuments oder eines Ordners mit einer NPSOBJ-Anweisung einfügen. Ersetzen Sie in Ihrem Code `object-path` durch den vollständigen Pfad zur einzufügenden Datei:

```
<npsobj includetext="object-path"/>
```

Die eingefügte Datei wird vom Content Management Server vollständig als Referenz verwaltet. Zwischen beiden Dateien entsteht ein Link, der genau wie andere interne Links bidirektional verwaltet wird und bearbeitet werden kann:

Im Content Navigator wird ein solcher Link im Detailbereich angezeigt.

11.2 Systemkommandos ausführen

In Layouts kann folgende NPSOBJ-Anweisung verwendet werden, um Tcl-Prozeduren auszuführen (ersetzen Sie `procedureAlias` durch den Alias-Namen der Prozedur und `content` durch HTML-Code, Text oder eine NPSOBJ-Anweisung):

```
<npsobj name="procedureAlias" insertvalue="systemExecute">  
  content  
</npsobj>
```

Der auszuführenden Prozedur werden immer zwei Parameter übergeben: als erster Parameter die ID der Datei oder des Links, in dessen Kontext die Anweisung ausgeführt wird, sowie der ausgewertete `content` des Elements als zweiter Parameter.

Nachdem der Content Management Server die Prozedur abgearbeitet hat, wird die NPSOBJ-Anweisung durch das Ergebnis der Prozedur ersetzt.

Um auf diese Weise Tcl-Prozeduren ausführen zu können, muss ein Administrator die Aliase in der Systemkonfiguration (im Dictionary `tclSystemExecuteCommands`) registriert haben.

Bitte beachten Sie, dass der aufgerufene Tcl-Code keinen Schreibzugriff auf die CMS-Daten hat und dass die beim Export auszuführenden Prozeduren auch auf dem Live-Server verfügbar sein müssen, wenn Sie dort die Template Engine einsetzen.

Das mitgelieferte Systemkommando `editLink`

Infopark CMS Fiona wird mit dem vordefinierten Systemkommando `editLink` geliefert. Der Befehl erlaubt es, Links auf die Content-Navigator-Seite der exportierten Datei zu erzeugen:

```
<npsobj insertvalue="systemExecute" name="editLink">
  Diese Datei bearbeiten
</NPSOBJ>
```

Damit beim Export funktionierende Links generiert werden, muss in der Systemkonfiguration des Content Managers der Wert des Eintrags `guiUrl` auf die URL der Web-Applikation des GUIs gesetzt werden.

11.3 Absolute Pfade in Skripten ausgeben

Mit den Werten der Variablen `fsPathPrefix` und `urlPrefix` ist es möglich, in Skripten absolute Pfade und URLs zu erzeugen, die sowohl in der Vorschau als auch auf dem Live-Server funktionieren.

Zu diesem Zweck fügt man am Anfang des Basislayouts (`mastertemplate`) eine NPSOBJ-Anweisung ein, die eine Variablenzuweisung in der gewünschten Skriptsprache erzeugt, beispielsweise:

```
<npsobj insertvalue="var" name="urlPrefix" formatter="phpVardef" />
```

Ein Pfad kann anschließend folgendermaßen in die Datei eingefügt werden, die beim Export erzeugt wird:

```
<npsobj insertvalue="var" name="visiblePath" formatter="phpVardef" />
<? $absUrlPath = $urlPrefix.$visiblePath; ?>
```

Diese Art und Weise, den Wert von `$absUrlPath` zu berechnen, funktioniert sowohl auf dem Live-Server als auch in der Vorschau.

12

12 Beispiele für Layouts

12.1 Verschachtelte Layouts

12.1.1 Eine Sitemap als Inhaltsübersicht erstellen

Unter einer Sitemap versteht man die Darstellung der Hierarchie einer Website oder von Teilen dieser Hierarchie, wobei normalerweise die Elemente dieser Darstellung mit den entsprechenden Webseiten verlinkt sind. Zur Erzeugung einer Sitemap mit Infopark CMS Fiona lassen sich die Mechanismen einer automatisch generierten Inhaltsübersicht nutzen. Im Gegensatz zu den im Abschnitt [Beispiele für automatisch generierte Inhaltsübersichten](#) aufgeführten Beispielen muss bei einer Sitemap die Inhaltsübersicht aber bis zur letzten Hierarchieebene generiert werden.

Dies lässt sich dadurch erreichen, dass die Inhaltsübersicht in einer Layoutdatei erzeugt wird, die sich immer dann selbst aufruft, wenn sie bei der Erzeugung der Inhaltsübersicht auf einen Unterordner trifft. Der Vorgang des Sich-Selbst-Aufrufens wird Rekursion genannt.

Im folgenden Beispiel werden die Einträge durch ineinandergeschachtelte `ul`-Elemente eingerückt. Auf die zusätzliche Kennzeichnung der Einträge (z.B. durch Einsetzen eines Icons in Abhängigkeit vom Typ der angezeigten Datei) wird verzichtet.

```
<ul>
  <npsobj list="toclist">
    <npsobj name1="objType" value2="generic" condition="isNotEqual">
      <li>
        <npsobj name="self" insertvalue="anchor">
          <npsobj name="title" insertvalue="var"/>
        </npsobj>
        <npsobj name="toclist" condition="isNotEmpty">
          <npsobj name="sitemaptemplate" insertvalue="template"/>
        </npsobj>
      </li>
    </npsobj>
  </npsobj>
</ul>
```

Um eine Sitemap mit dem obigen Beispielcode zu erzeugen, erstellen Sie bitte zunächst im Basisordner eine Layoutdatei mit dem Namen `sitemaptemplate`. Bearbeiten Sie nun den Haupttext der Arbeitsversion mit dem Texteditor und setzen Sie den obigen Code ein. Geben Sie die Layoutdatei frei.

Wechseln Sie anschließend zum Basisordner, öffnen Sie den Hauptinhalt ihrer Arbeitsversion mit dem Texteditor und fügen Sie die folgende Zeile ein:

```
<npsobj name="sitemaptemplate" insertvalue="template"/>
```

Wenn Sie sich nun die Vorschau des Ordners anzeigen lassen, wird die Sitemap dargestellt.

Bitte beachten Sie, dass NPSOBJ-Anweisungen normalerweise nicht in den Hauptinhalt von Ordnern oder Dokumenten, sondern nur in Layouts eingefügt werden sollten, um die Wartbarkeit der Website zu gewährleisten.

12.1.2 Sitemap mit ausgewählten Dateien in Ordnern

Inhaltsübersichten, die mit `<npsobj list="toclist">` erzeugt wurden, enthalten nur Dateien des Typs *Dokument*, *Ressource* und *Ordner* (siehe [Bestandteile automatisch generierter Inhaltsübersichten](#)). Soll die Sitemap auch Bilder aufführen, so muss das Feld `children` ausgelesen werden.

Im folgenden Beispiel wird eine Sitemap erzeugt, in der ausschließlich Bilder in einem Ordnerbaum dargestellt werden:

```
<npsobj list="children">
  <npsobj name1="objType" value2="image" condition="isEqual">
    <p>
      <npsobj name="title" insertvalue="var"/>
      <br>
      <npsobj name="self" insertvalue="image"/>
    </p>
  </npsobj>
  <npsobj name1="objType" value2="publication" condition="isEqual">
    <npsobj name="sitemaptemplate" insertvalue="template"/>
  </npsobj>
</npsobj>
```

Sie können die Sitemap verwenden, indem Sie den Code in eine Layoutdatei einfügen und die Vorschau aufrufen.

12.1.3 Unendliche Rekursion

Der Content Management Server und die Template Engine überwachen die rekursive Verwendung von Layouts. "Rekursion" bedeutet in diesem Zusammenhang, dass ein Layout sich selbst inkludiert:

```
<!-- Layoutdatei "myTemplate" -->
<npsobj insertvalue="template" name="myTemplate" />
<!-- ... -->
```

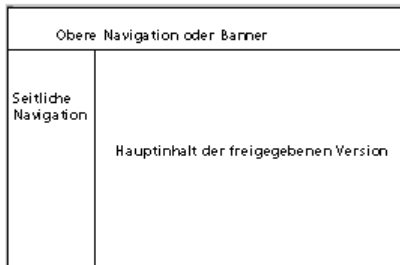
Gibt es in einer solchen Konstruktion keine Abbruchbedingung oder tritt die Abbruchbedingung nie ein, ist die Rekursion unendlich, wodurch der Exportvorgang der betreffenden Datei nie abgeschlossen wird. Um sicherzustellen, dass dieser Fall nicht eintritt, brechen der Content Management Server und die Template Engine beim Export und bei der Vorschau eine Rekursion ab, nachdem die im Systemkonfigurationseintrag [maxRecursionLevel](#) festgelegte Anzahl Durchläufe stattgefunden hat, und geben eine Fehlermeldung aus.

Bitte überprüfen Sie im Falle einer unendlichen Rekursion, ob beispielsweise der Hauptinhalt einer Datei eine Layoutdatei aufruft, die wiederum den Hauptinhalt der Datei in die Ausgabedatei einfügt.

12.1.4 Tabellenlayouts definieren

Typischerweise enthalten die Seiten einer Website Navigationsbereiche und den eigentlichen Inhaltsbereich. Wenn Sie für ein solches Layout Tabellen (anstelle von *Cascading Style Sheets*, CSS)

verwenden, ist es empfehlenswert, im Basislayout für wechselnde Navigationsleisten Platzhalter vorzusehen und anschließend je Ordner eine Layoutdatei anzulegen, mit der die Navigationsleiste erzeugt wird.



Ein solches Layout kann mit Code nach folgendem Muster in einer Layoutdatei realisiert werden:

```
<head>
  <title><npsobj name="title" insertvalue="var"/></title>
  <npsobj name="attribute-name" insertvalue="meta"/>
  ...
</head>

<body>
  <!-- Obere Navigation-->
  <center>
    <a href="/deutsch"></a>
    <a href="/englisch"></a>
    <a href="/email"></a>
  </center>

  <table>
    <tr><td>
      <!-- Platzhalter für seitliche Navigation-->
      <npsobj name="nav_seitl" insertvalue="template"/>
    </td>

    <td>
      <!-- Den Hauptinhalt einfügen-->
      <npsobj name="body" insertvalue="var"/>
    </td></tr>
  </table>
</body>
```

Diese Layoutdatei referenziert eine weitere Layoutdatei, `nav_seitl`, die in allen Unterordnern, die eine seitliche Navigation enthalten sollen, angelegt und mit den jeweils erforderlichen Links und Bildern versehen wird.

Layoutdateien sollten immer dann weitere Layoutdateien aufrufen, wenn dadurch die Struktur einer Website verallgemeinert und ihre Erstellung vereinfacht werden kann.

12.2 Framelayouts

Mit dem Content Management Server haben Sie die Möglichkeit, Framelayouts auf zwei Arten zu erzeugen. Zum einen stellt Ihnen der Content Manager eine NPSOBJ-Anweisung zur Verfügung, die Sie in Layoutdateien verwenden können, um Framesets einschließlich der erforderlichen Inhaltsdateien automatisch erzeugen zu lassen.

Zum anderen können Sie die Frameset-Struktur auf die Ordnerhierarchie abbilden, indem Sie für das Frameset einen Ordner und für das Inhaltsverzeichnis-Frame einen Unterordner anlegen. In diesem Unterordner können die Hauptframes als Dokumente abgelegt werden.

Die erste Methode hat den Vorteil, dass die Redakteure sich bei der Erstellung oder Bearbeitung der Dokumente nicht um die zugrundeliegende Frame-Struktur kümmern müssen: für jedes Dokument erzeugt der Content Management Server beim Export und bei der Vorschau immer ein komplettes Frameset. Das Dokument ist an sein Frameset gebunden, denn jeder Verweis auf ein so exportiertes Dokument ist im Ergebnis ein Verweis auf das Frameset, in dem es angezeigt werden soll. Folglich ist es möglich, für das Dokument ein Lesezeichen zu setzen, ohne auf das umgebende Frameset verzichten zu müssen.

Bei der zweiten Methode ist dies nicht der Fall, denn das Frameset wird nur für jede Indexseite einer Ordnerhierarchie erzeugt, nicht jedoch für die Dokumente, die sich in den Teilhierarchien selbst befinden. Folgt man einem Link auf ein solches Dokument, so wird es nur dann in seinem Frame angezeigt, wenn das Frameset vorher geladen wurde und das Zielframe voreingestellt oder im Link-Tag als Target angegeben ist. Da die Dokumente nicht an ihr Frameset gebunden sind, können für sie keine Lesezeichen gesetzt werden.

Die zuerst genannte Methode zur Erzeugung eines auf Frames basierenden Webservers sollten Sie anwenden, wenn Sie den Aufwand zur Verwaltung der Frame-Struktur minimieren wollen und jedes Dokument bei der Vorschau und im Exportergebnis stets in seinem Frameset angezeigt werden soll. Die zweite Methode bietet sich an, wenn Sie Wert darauf legen, einen schlankeren Webserver zu haben. Es ist nicht sinnvoll, die beiden Methoden zu kombinieren.

12.2.1 Framesets mit dem NPSOBJ-Frame-Kommando automatisch erzeugen

Um für eine exportierte Datei ein Frameset und die erforderlichen Frame-Inhalte zu erzeugen, benötigen Sie eine Layoutdatei. In dieser Layoutdatei wird zunächst das Frameset und anschließend jedes Frame mit einer NPSOBJ-Anweisung nach folgendem Muster definiert:

```
<npsobj frame="frameName" defaulttarget="defaultTarget">
  frameContent
</npsobj>
```

Beim Export und bei der Vorschau bewirkt eine solche NPSOBJ-Anweisung, dass neben der Ausgabedatei für die Datei eine weitere Ausgabedatei erzeugt wird, die den ausgewerteten Inhalt des NPSOBJ-frame-Tags enthält. Der Name dieser Frame-Inhaltsdatei setzt sich aus dem Namen der exportierten Datei, einem Punkt, dem im NPSOBJ-Tag angegebenen Framennamen *frameName* und der Dateinamenserweiterung zusammen. Beim Export des Dokuments *sample* wird also neben der Hauptdatei *sample.html* die Datei *sample.frameName.html* generiert.

In einer Layoutdatei zur Erzeugung von Framesets und Frames wird der Hauptinhalt der zu exportierenden Datei in den Inhalt eines NPSOBJ-frame-Tags aufgenommen, sodass er als Frame-Inhaltsdatei vorliegt. Die erzeugte Hauptdatei dagegen enthält nur den Code, der die NPSOBJ-frame-Tags umgibt, also vor allem den Code, der das Frameset definiert.

Das folgende Beispiel geht davon aus, dass sich im gleichen Ordner die Dateien *sampledoc* und die Layoutdatei *frametemplate* befinden. In *frametemplate* werden ein zweiteiliges Frameset und die beiden dazu gehörenden Frames definiert:

```
<html>
```

```

<head>
<title><npsobj insertvalue="var" name="title"/></title>
</head>
<frameset rows="100,*">
  <npsobj frame="banner">
    <html>
      <head>
        <title>Banner</title>
      </head>
      <body>
        
        <!-- Put navigation code here -->
      </body>
    </html>
  </npsobj>
  <npsobj frame="content">
    <html>
      <head>
        <title><npsobj insertvalue="var" name="title"/></title>
      </head>
      <body>
        <npsobj insertvalue="var" name="body"/>
      </body>
    </html>
  </npsobj>
</frameset>
</html>

```

Die Datei `sampledoc` hat den Titel `Sampledoc-Titel` und enthält den folgenden Code:

```

<h1>Sampledoc</h1>
<p>Ein Beispieldokument mit einem <a href="sample2.html">
internen Link</a> und einem <a href="http://www.infopark.de"
target="_blank">externen Link</a>.</p>

```

Beim Export von `sampledoc` mit Hilfe der Layoutdatei `frametemplate` entstehen die drei Ausgabedateien `sampledoc.html`, `sampledoc.banner.html` und `sampledoc.content.html`. Folgt man einem Link auf das Dokument `sampledoc`, so wird die Hauptdatei `sampledoc.html` angezeigt.

`Sampledoc.html` enthält im Wesentlichen die Frameset-Definition:

```

<html>
  <head>
    <title>Sampledoc-Titel</title>
  </head>
  <frameset rows="100,*">
    <frame name="banner" src="sampledoc.banner.html">
    <frame name="content" src="sampledoc.content.html">
  </frameset>
</html>

```

Die Datei `sampledoc.banner.html`:

```

<html>
  <head>
    <title>Banner</title>
  </head>
  <body>
    
    <!-- Put navigation code here -->
  </body>
</html>

```

Die Datei *sampledoc.content.html* :</p>

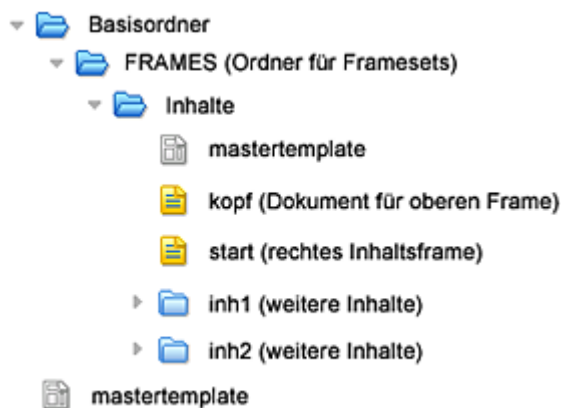
```
<html>
<head>
<title>Sampledoc-Titel</title>
</head>
<body>
<h1>Sampledoc</h1>
<p>Ein Beispieldokument mit einem <a href="sample2.html" target="_top">
internen Link</a> und einem <a href="http://www.infopark.de"
target="_blank">externen Link</a>.</p>
</body>
</html>
```

Bitte beachten Sie, dass Sie in einer NPSOBJ-frame-Anweisung mit dem Tag-Attribut `defaulttarget` festlegen können, in welchem Frame ein verlinktes Dokument angezeigt werden soll. Der voreingestellte Wert für `defaulttarget` ist `_top`, Sie brauchen dieses Tag-Attribut also nicht anzugeben, wenn die Links in einem Frame auf Dokumente verweisen, die ebenfalls als Frameset exportiert wurden. Andere Tag-Attribute, die Sie in NPSOBJ-frame-Tags verwenden, werden unverändert in das resultierende `frame`-Tag übernommen.

NPSOBJ-frame-Tags dürfen nicht verschachtelt werden, d. h. sie dürfen keine NPSOBJ-frame-Tags enthalten.

12.2.2 Frames mit Ordnerhierarchien realisieren

Das im Abschnitt [Tabellenlayouts definieren](#) beschriebene Layout lässt sich anstatt mit einer Tabelle auch mit Frames realisieren:



Im Basislayout (`mastertemplate`) im Basisordner wird zunächst das Frameset definiert:

```
<html>
<head>
<title>
<npsobj name="title" insertvalue="var"/>
</title>
</head>

<frameset marginheight="0" framespacing="0" border="0" scrolling="no"
frameborder="0" rows="50, *">
<frame src="/FRAMES/Inhalte/kopf.html" name="oben">
<frameset cols="100,450" framespacing="0" border="0" scrolling="no"
frameborder="0" marginwidth="0">
<frame src="/FRAMES/Inhalte/index.html" name="links">
<frame src="/FRAMES/Inhalte/start.html" name="rechts">
```

```

</frameset>
</frameset>
<noframes>
  <p>Ihr Browser unterst tzt keine Frames.</p>
</noframes>
</html>

```

Das Dokument `/FRAMES/Inhalte/kopf` enth lt die Navigationsleiste f r den oberen Frame, das Dokument `/FRAMES/Inhalte/start` wird als Startseite f r den rechten Frame definiert.

Der Hauptinhalt des Ordners `/FRAMES/Inhalte` wird im linken Navigationsframe angezeigt. Hier kann beispielsweise eine automatisch generierte Inhalts bersicht eingef gt werden. Dabei muss ber cksichtigt werden, dass die Links in der Inhalts bersicht und die manuell gesetzten Links mit den entsprechenden Targets versehen werden m ssen:

```

<npsobj list="toclist">
  <npsobj target="rechts" name="self" insertvalue="anchor">
    <npsobj name="title" insertvalue="var"/>
  </npsobj>
</npsobj>

```

Die Layoutdatei `/FRAMES/Inhalte/mastertemplate` enth lt die Layoutdefinition f r die Hauptinhalte der im Inhaltsframe angezeigten Dateien:

```

<html>
  <head>
    <title>
      <npsobj name="title" insertvalue="var"/>
    </title>
  </head>

  <body>
    <npsobj name="body" insertvalue="var"/>
  </body>
</html>

```

Alle weiteren Dateien werden ebenfalls mit Hilfe dieser Layoutdatei exportiert, sodass ihr jeweiliger Haupttext im gleichen Inhaltsframe angezeigt wird, wenn im entsprechenden Link kein anderes Target definiert wurde.

Bitte beachten Sie bei der Konzeption eines solchen Framelayouts mit dem Content Management Server, dass die Framesets nicht innerhalb eines HTML-Bodys definiert werden d rfen, also keinesfalls in einem Dokument oder einem Ordner und auch nicht in einer Layoutdatei, die in einem Dokument oder einem Ordner referenziert wird. Dies w rde dazu f hren, dass die Frameset-Definition auch auf die Seiten des Content Navigators angewendet wird.

Wenn Sie mehrere Framesets verwenden, k nnen Sie sie einzeln in den Basislayouts der jeweiligen Unterordner ablegen. Alternativ k nnen mehrere Framesets in einem globalen Basislayout definiert sein, wobei jedes dieser Framesets von einer vergleichenden Abfrage auf einen Feldwert umschlossen wird. Dies erm glicht den Redakteuren, ein Frameset zu ihrem Dokument festzulegen, indem sie das Feld mit dem entsprechenden Wert belegen.

Wenn Sie das Aussehen der Website mit der Vorschau des Content Management Servers  berpr fen, muss wenigstens einmal im Vorschaufenster das komplette Frameset geladen worden sein, weil die Vorschau bei einer untergeordneten Datei nur den Haupttext der Datei ohne die umgebenden Frames anzeigt.

13

13 Die NPSOBJ-Syntax

NPSOBJ-Anweisungen sind ein mächtiges Instrument zur Erzeugung von HTML-Code. Die Anweisungen können im Hauptinhalt von Layoutdateien, Dokumenten und Ordnern (nicht jedoch in anderen Feldern) verwendet werden, um beim Export auf Dateien und ihre Felder zuzugreifen und die ausgelesenen Daten (oder daraus erzeugten HTML-Text) in die Ausgabedatei einzufügen.

NPSOBJ-Anweisungen werden ausgewertet, sobald Sie sich über die Benutzeroberfläche des Content Management Servers eine Vorschau anzeigen lassen oder den Export von Dateien auslösen. Auch die Template Engine verarbeitet NPSOBJ-Anweisungen. Wenn in diesem Abschnitt NPSOBJ-Anweisungen und die Export-Funktionalität des Content Management Servers behandelt werden, so ist immer auch die Template Engine gemeint.

NPSOBJ-Anweisungen werden bei der Auswertung durch das Ergebnis dieser Auswertung ersetzt. Bei der Auswertung dient zunächst die exportierte Datei als Datenquelle. Man kann mit NPSOBJ-Anweisungen jedoch auch beliebige andere Dateien als Datenquelle verwenden.

Der Content Management Server verwendet zum Export jeder Datei – ausgenommen Bilder und Ressourcen – das Basislayout. Wie andere Layouts auch, wird das Basislayout genau einmal von oben nach unten abgearbeitet. Jede auftretende NPSOBJ-Anweisung wird dabei entsprechend den hier aufgeführten Regeln durch HTML-Code oder weitere Anweisungen ersetzt.

In den folgenden Abschnitten erfahren Sie, welche syntaktischen und semantischen Regeln der Content Management Server beim Export anwendet. Die Ausführungen beschreiben den Ablauf des Exportvorgangs und spezifizieren zugleich die Regeln, nach denen das NPSOBJ-Sprachelement verwendet werden muss, um einen erfolgreichen Export zu bewirken.

13.1 Konventionen

Für die formale Darstellung von NPSOBJ-Elementen wird eine leicht verständliche Notation verwendet. Sämtliche Syntaxbeschreibungen sind in der Schriftart *Courier* gedruckt. Symbole werden ohne Auszeichnung wiedergegeben, literale Zeichen dagegen sind **fett gedruckt**. Für variable Zeichenketten, die durch einen sinnvollen Wert ersetzt werden müssen, wird *Kursivschrift* verwendet. Im folgenden Beispiel wird die NPSOBJ-`includetext`-Anweisung definiert:

```
npsobj_includetext ::=  
  <npsobj includetext = "internal_url" />
```

Die verwendeten Symbole werden aufgelöst, es sei denn, es handelt sich dabei um terminale Symbole wie `string`. `string` bezeichnet eine Zeichenkette, die nicht weiter aufgelöst werden kann.

Symbole können mit dem Zeichen | (vertikaler Strich) disjunkt verknüpft sein. Dies bedeutet, dass exakt eines der verknüpften Symbole verwendet werden muss. So verknüpfte Elemente können mit runden Klammern zu einer Gruppe zusammengefasst sein.

Optionale Bestandteile werden in eckige Klammern gefasst. Bestandteile, die mehr als einmal vorkommen dürfen, sind von geschweiften Klammern umgeben.

13.2 Wertbestimmung im Kontext

NPSOBJ-Elemente werden verwendet, um beispielsweise Feldwerte einer Datei auszulesen und in den exportierten HTML-Code einzufügen. In diesem Abschnitt wird der Mechanismus zur Wertbestimmung für alle NPSOBJ-Anweisungen beschrieben.

13.2.1 Was ist ein Kontext?

Wenn eine CMS-Datei exportiert wird, dann dient sie dem Layout, das für den Export verwendet wird (beispielsweise dem `mastertemplate`), als Datenquelle. Diese Datenquelle enthält benannte Werte wie den mittels `title` abrufbaren Titel der Datei. Die Datenquelle bildet den Kontext, in dem sich NPSOBJ-Anweisungen befinden, wenn sie sich auf einen Namen beziehen, um dessen Wert zu ermitteln:

```
<npsobj insertvalue="var" name="title" />
```

Hier wird im aktuellen Kontext nach dem Wert für `title` gefragt. Fragt man in einem Kontext nach einem Wert für einen in diesem Kontext unbekannten Namen, so erhält man den leeren Wert.

Der Content Management Server kennt zwei Typen von Kontexten, Datei- und Linkkontexte.

Während des Exports einer Datei gibt es immer mindestens zwei Kontexte, den initialen und den aktuellen Kontext. Zu Beginn des Exports eines Dokuments oder eines Ordners liefert die betreffende CMS-Datei den initialen und den aktuellen Kontext. Während der initiale Kontext im Verlauf des Exports der CMS-Datei immer gleich bleibt, kann sich der aktuelle Kontext ändern. Der aktuelle Kontext wird durch NPSOBJ-Anweisungen geändert, die keinen einfachen Wert wie `title` liefern, sondern eine Liste von Kontexten. Die folgenden Anweisungen liefern eine solche Kontextliste: `list`, `newslist`, `table` und `context`.

Mit Hilfe dieser Anweisungen lässt sich für jeden Kontext in der Kontextliste NPSOBJ-Code ausführen. Beispiel:

```
<npsobj list="toclist">
  <npsobj insertvalue="anchor" name="self">
    <npsobj insertvalue="var" name="title" /><br />
  </npsobj>
</npsobj>
```

Die mittlere NPSOBJ-Anweisung, `insertvalue="anchor"`, wird für jeden in `toclist` enthaltenen Kontext ausgeführt. Bei jedem Durchlauf der `list`-Anweisung wird der betreffende Kontext aus `toclist` zum aktuellen Kontext. Nachdem alle Kontexte aus `toclist` durchlaufen wurden - also nach Abschluss der `list`-Anweisung - wird der ursprüngliche Kontext, in dem die `list`-Anweisung ausgeführt wurde, wieder zum aktuellen Kontext.

Jede Frage nach einem Wert wird im aktuellen Kontext ausgewertet. Die Liste der darin verfügbaren Namen hängt vom Typ des Kontexts ab. Bei Dateikontexten ergibt sich ein Teil der verfügbaren Namen aus dem Typ der Datei und aus ihrer Dateivorlage.

13.2.2 Der Aufbau von Namen

Um mit einer NPSOBJ-Anweisung in einem Kontext nach einem Wert zu fragen, gibt man den Namen des Wertes an. Im einfachsten Fall ist dies ein einfacher Name (*simple_name*) wie beispielsweise `title` oder `lastChanged`:

```
<npsobj insertvalue="var" name="title" />
```

Der gelieferte Wert ist in diesen Fällen eine Zeichenkette. Bei anderen Namen (wie `parent`) ist der gelieferte Wert jedoch eine Datei. Fügt man einen solchen Wert beim Export direkt in die Ausgabe ein, erhält man eine Zeichenkettenrepräsentation der Datei, was in der Regel nicht das gewünschte Ergebnis ist. Es ist daher möglich, auch diese Datei wiederum nach einem Wert zu fragen. Hierfür hängt man den Namen des zu ermittelnden Wertes als Komponente an den ersten Namen an, getrennt durch einen Punkt. Dies kann auch mehrfach geschehen, um nach dem Wert eines aus mehreren Komponenten bestehenden Namens zu fragen:

```
<npsobj insertvalue="var" name="parent.title" /><br />
<npsobj insertvalue="var" name="parent.next.title" />
```

Formell ist ein Name folgendermaßen aufgebaut:

```
composite_name ::= [ exporter_prefix_sequence ] [ simple_name_sequence ]
exporter_prefix_sequence ::=
    initial. | self. | export. | ( context. { context. } )

simple_name_sequence ::= [ simple_name . ] simple_name
```

Die Zahl der Komponenten in einem zusammengesetzten Namen ist prinzipiell nicht begrenzt. Durch die Angabe einer Komponente wird während der Wertbestimmung der aktuelle Kontext für die sich anschließende Komponente temporär umgesetzt. Bei einem zusammengesetzten Namen wie `parent.title` wird demnach der `parent`-Ordner des aktuellen Dateikontexts zum aktuellen Kontext gemacht, in dem dann der Wert von `title` ermittelt wird. Abschließend wird der ursprünglich aktuelle Kontext wieder zum aktuellen Kontext gemacht.

Präfix-Sequenzen

Mit einer Präfix-Sequenz (*exporter_prefix_sequence*) kann der Kontext während des Exportvorgangs oder zwischen Geltungsbereichen von Namen gewechselt werden:

Präfix	Wirkung
<code>context</code>	Der Kontext des umgebenden <code>list</code> -, <code>newslst</code> - oder <code>table</code> -Elements verwendet wird. Er kann mehrfach eingesetzt werden, um mehrere Ebenen aufzusteigen.
<code>initial</code>	Der initiale Kontext wird verwendet.
<code>export</code>	Der Namensraum wird temporär gewechselt: der auf <code>export</code> folgende Name wird als Name einer <u>Export-Variablen</u> interpretiert.

self	Der aktuelle Kontext wird referenziert.
------	---

Ein Präfix ist nur wirksam, während der Wert der auf den Präfix folgenden Namenskomponente ermittelt wird. Im folgenden Beispiel bezieht sich daher nur die erste `insertvalue="var"`-Anweisung auf den Kontext, in dem die `list`-Anweisung ausgeführt wird. Die zweite `insertvalue="var"`-Anweisung bezieht sich dagegen auf den Kontext des aktuellen `toclist`-Elements:

```
<npsobj list="toclist">
  <b><npsobj insertvalue="var" name="context.title" /></b> -
  <npsobj insertvalue="var" name="title" /><br />
</npsobj>
```

13.2.3 Namen in Datei-Kontexten

Ist der aktuelle Kontext eine Datei (im Folgenden kurz die aktuelle Datei genannt), sind die im Folgenden aufgeführten Namen verfügbar. Sie liefern den Wert des hier angegebenen Typs (*Datei*, *Zeichenkette* usw.), sofern es keine Export-Variable mit demselben Namen gibt, deren Wert einen anderen Typ hat (siehe auch [npsobj_modifyvar_set](#) sowie [Export-Variablen](#)).

Namen, deren Wert eine Datei ist

- `parent`: liefert den über der aktuellen Datei liegenden Ordner.
- `next`: liefert entsprechend der anzuwendenden Sortierung die nächste Datei in dem Ordner, in dem sich die aktuelle Datei befindet.
- `previous`: liefert entsprechend der anzuwendenden Sortierung die vorherige Datei in dem Ordner, in dem sich die aktuelle Datei befindet.
- `self`: liefert die aktuelle Datei.
- `original`: liefert die Originaldatei, wenn die aktuelle Datei eine Spiegeldatei ist, ansonsten die aktuelle Datei.
- `up`: liefert die erste Datei (auf dem Weg von der aktuellen Datei zum Basisordner), in der der Wert der auf `up` folgenden Komponenten nicht der leere Wert ist. `up` darf nicht die letzte Komponente eines Namens sein.

Namen, deren Wert ein Link ist

- `link.link_title`: liefert einen freien Link der aktuellen Datei mit dem Titel `link_title`.

Namen, deren Wert eine Zeichenkette ist

- `body`: liefert den Hauptinhalt der aktuellen Datei.
- `contentType`: liefert die Dateierdung der aktuellen Datei.
- `hasSuperLinks`: liefert 1, wenn Links auf die aktuelle Datei verweisen, andernfalls 0.
- `id`: liefert die ID der aktuellen Datei.
- `isRoot`: liefert 1, wenn die aktuelle Datei der Basisordner ist, andernfalls 0.
- `name`: liefert den Namen der aktuellen Datei.
- `objClass`: liefert die Vorlage der aktuellen Datei.
- `objType`: liefert den Dateityp der aktuellen Datei.

- `path`: liefert den Pfad der aktuellen Datei.
- `permissions.permissionName`: liefert die Namen der Benutzergruppen, die das dateispezifische Recht `permissionName` haben.
- `prefixPath`: Wenn die aktuelle Datei ein Ordner ist, wird der Pfad, gefolgt von einem Schrägstrich geliefert, andernfalls wird der Pfad geliefert.
- `title`: liefert den Titel der aktuellen Datei.
- `version`: liefert die Revisionsnummer der aktuellen Datei.
- `visibleName`: Wenn die aktuelle Datei ein Dokument ist, wird der Dateiname, gefolgt von der Dateiendung geliefert, ist es ein Ordner, wird nur der Dateiname geliefert.
- `visiblePath`: Bei Dokumenten der komplette Pfad einschließlich Dateiname, mit dem die Datei exportiert wird; bei Ordnern der komplette Pfad einschließlich Dateiname, gefolgt von der Zeichenkette `"/index"`, gefolgt von der Dateiendung des Ordners.
- alle in der Dateivorlage der aktuellen Datei definierten Felder der Typen `string`, `text`, `html` und `enum`. Bei Versionsfeldern kann auf den Namen der String `.displayValue` folgen, um den Anzeigewert des Feldes zu ermitteln.
- alle Export-Variablen, denen eine Zeichenkette zugewiesen wurde.

Namen, deren Wert eine Datumszeichenkette ist

- `lastChanged`: liefert das Datum der letzten Änderung der aktuellen Datei. Änderungen in der Version der Datei werden hierbei nicht berücksichtigt. Beim Export einer freigegebenen Datei ist die letzte Änderung die Freigabe.
- `validFrom`: liefert das Datum, ab dem der Inhalt der Datei gültig ist. Dieser Wert ist immer gesetzt. Bei Layoutdateien wird er jedoch ignoriert.
- `validUntil`: liefert das Datum, bis zu dem der Inhalt der Datei gültig ist. Wenn dieser Wert der leere Wert ist, ist das Dokument unbegrenzt gültig. Der Wert wird bei Layoutdateien ignoriert.
- alle in der Vorlage der aktuellen Datei definierten Felder des Typs `date`.

Namen, deren Wert eine Liste von Zeichenketten ist

- alle in der Vorlage der aktuellen Datei definierten Felder des Typs `multienum`.

Namen, deren Wert eine Liste von Links ist

- `freeLinks`: liefert die Liste aller freien Links in der Datei, d. h. alle Links, die in `linklist`-Feldern enthalten sind.
- `subLinks`: liefert die Liste aller Links in der Datei.
- `superLinks`: Liefert die Liste von Links, die auf die aktuelle Datei verweisen.
- `textLinks`: Liefert die Liste von Links, die im Hauptinhalt und in HTML-Feldern der Datei enthalten sind (außer Einfügelinks und Kontextlinks).
- alle in der Vorlage der aktuellen Datei definierten Felder des Typs `linklist`.
- alle Export-Variablen, die eine Liste von Links enthalten.

Namen, deren Wert eine Liste von Dateien ist

- `children`: Liefert die Liste aller Dateien in einem Ordner (unabhängig von ihren Versionen). Wenn die aktuelle Datei kein Ordner ist, wird die leere Liste geliefert.
- `objectsToRoot`: Liefert eine Liste aller Ordner auf dem Weg vom Basisordner zur aktuellen Datei. Der Basisordner ist das erste, die aktuelle Datei das letzte Element der Liste.

- `superObjects`: Liefert die Liste von Dateien, in denen ein Link auf die aktuelle Datei gesetzt ist.
- `toclist`: Liefert eine Liste von sortierten Dateien in einem Ordner, die keine Bilder und keine Layoutdateien sind.
- alle Export-Variablen, die eine Liste von Dateien enthalten.

Namen, deren Wert eine Kontextliste ist

- alle Export-Variablen, die eine Kontextliste enthalten.

Die Namen `toclist`, `previous` und `next` liefern ihr Ergebnis entsprechend der vorgegebenen Reihenfolge der Dateien in einem Ordner. Dies trifft auch dann zu, wenn der aktuelle Kontext von einem `list` – oder `table`-Element erzeugt wurde, in dem die Reihenfolge der Dateien von der Standardreihenfolge abweicht, weil sie sortiert wurden.

13.2.4 Namen in Link-Kontexten

Ist der aktuelle Kontext ein Link, so sind die folgenden Namen verfügbar:

Namen, deren Wert eine Datei ist

- `destination`: die Datei, die das Ziel des Links ist. Dies ist bei externen Links der leere Wert.
- `source`: die Datei, die der Ausgangspunkt des Links ist.

Namen, deren Wert eine Zeichenkette ist

- `destinationUrl`: Eine URL vom initialen Kontext auf das Linkziel. Zeigt der Link nicht auf ein externes Ziel, sondern auf eine Datei im CMS, ist der Pfad relativ.
- `displayTitle`: Wenn der aktuelle Link einen nichtleeren `title` hat, wird dieser `title` geliefert. Andernfalls wird der Titel der freigegebenen Version der Zieldatei zurückgegeben. Bei externen Links wird immer `title` zurückgegeben.
- `target`: Das Fenster (*frame*), in dem der Browser den Link öffnen soll.
- `title`: Wenn der aktuelle Link einen `title` hat, wird der Titel geliefert.
- alle Export-Variablen, denen eine Zeichenkette zugewiesen wurde.

Namen, deren Wert eine Kontextliste ist

- alle Export-Variablen, die eine Kontextliste enthalten.

13.2.5 Namen in allen Kontexten

Die folgenden Namen existieren in allen Kontexten:

Namen, deren Wert eine Zeichenkette ist

- `exportCharset`: liefert den in der Systemkonfiguration definierten und in der Liste `exportCharsetMap` nachgeschlagenen Namen des Export-Zeichensatzes.
- `exportMode`: liefert den Wert `preview`, wenn der Name während der Vorschau evaluiert wird. Andernfalls (während des Exports) wird der Wert `normal` geliefert. Sie können diesen Wert verwenden, um den Export bestimmter Daten vom Exportmodus abhängig zu machen.

- `fsPathPrefix`: liefert den Wert des Systemkonfigurationseintrags `export.absoluteFsPathPrefix`, d. h. die Zeichenkette, die Dateipfade beim Export mit absoluten Pfaden vorangestellt wird. Siehe auch [Absolute Pfade in Skripten ausgeben](#).
- `urlPrefix`: liefert den Wert des Systemkonfigurationseintrags `export.absoluteUrlPrefix`, d. h. die Zeichenkette, die beim Export mit absoluten Pfaden den Pfaden in URLs vorangestellt wird. Siehe auch [Absolute Pfade in Skripten ausgeben](#).

13.2.6 @-Referenzen

@-Referenzen ermöglichen es, die in Kontexten verfügbaren Werte auch als Werte von HTML-Attributen zu verwenden. Diese Funktion kann nicht durch den Einsatz von NPSOBJ-Anweisungen wie der folgenden erreicht werden:

```
<!-- NICHT ZULÄSSIG! -->
<a href="/public/file" title="<npsobj insertvalue="var" name="myField" />">...</a>
```

Hierfür steht eine @-Referenz zur Verfügung:

```
<!-- Bis Version 6.7.0 -->
<a href="/public/file" title="@myField">...</a>

<!-- Ab Version 6.7.1 -->
<a href="/public/file" title="@{myField}">...</a>
```

In @-Referenzen können alle Namen verwendet werden, die auch in NPSOBJ-insertvalue-var-Anweisungen eingesetzt werden können, beispielsweise die Namen von Feldern und Exportvariablen.

Für @-Referenzen gelten die folgenden Regeln und Beschränkungen:

- @-Referenzen sind nur in den Attributwerten von HTML- und NPSOBJ-Tags verfügbar;
- Bis Version 6.7.0: Nur Tag-Attributwerte, deren erstes Zeichen das @-Zeichen ist, und deren zweites Zeichen ein alphanumerisches Zeichen (0 - 9, a - z, A - Z) oder der Unterstrich ist, werden als @-Referenz betrachtet.

Ab Version 6.7.1: @-Referenzen beginnen mit dem @-Zeichen, auf das die in geschweifte Klammern eingeschlossene referenzierte Variable folgt. (siehe das Beispiel oben). @-Referenzen können in Attributwerten an beliebiger Stelle stehen, wodurch letztere aus statischem und dynamischem Text zusammengesetzt werden können.

- Nur bis Version 6.7.0: Sind die beiden ersten Zeichen eines Tag-Attributwertes @-Zeichen, so werden diese in ein einzelnes @-Zeichen umgewandelt, das nicht als der Beginn einer @-Referenz betrachtet wird.
- @-Referenzen funktionieren weder in Processing-Instructions (PHP, JSP, usw.) noch in HTML-Kommentaren.
- Das Ergebnis einer @-Referenz-Ersetzung ist immer eine Zeichenkette oder ein Kontext.
- Ist der mit einer @-Referenz bezeichnete Wert eine Liste (Felder vom Typ *Mehrfachauswahl*, Kontextlisten), so werden die einzelnen Listenwerte mit dem Zeichen | verbunden.
- @-Referenzen in URI-Attributen wie `href` werden beim Import in das CMS nicht in die Linkverwaltung aufgenommen.
- Kann ein Wert nicht ermittelt werden, so wird die @-Referenz durch die leere Zeichenkette ersetzt.
- Auch zusammengesetzte Namen wie `parent.visiblePath` werden unterstützt.

- @-Referenzen sind nicht dafür geeignet, die Werte von Mehrfachauswahlfeldern zu behandeln. Hierfür sollten Formatter verwendet werden.
- Wenn mit einer @-Referenz eine Exportvariable referenziert wird, so wird der Wert der Exportvariablen nicht erst ermittelt, wenn die @-Referenz ausgewertet wird, sondern bereits, wenn die Exportvariable gesetzt wird.
- @-Referenzen, die sich in `title`- oder `target`-Attributen von Links im Text (also in Feldern wie dem Hauptinhalt) befinden, haben Vorrang vor den Titeln oder Targets, die über die Linkverwaltung (also beispielsweise mittels Content Navigator) gesetzt werden.
- Eine über die Linkverwaltung eingetragene @-Referenz wird nicht ausgewertet, sondern als konstante Zeichenkette exportiert. Beim Import wird diese Zeichenkette jedoch zu einer @-Referenz, weil sie nun direkt im Hauptinhalt steht.

Beispiele

Die Zeilenumbrüche und Einrückungen zwischen den NPSOBJ-Anweisungen in den folgenden Beispielen dienen nur der Übersichtlichkeit und dürfen nicht eingegeben werden, weil die Ergebnisse (URLs) sonst defekt wären.

Der folgende Code erzeugt aus einem Feldwert den letzten Teil einer URL und hängt an ihn einen Parameter an:

```
<!-- Bis Version 6.7.0 -->
<npsobj modifyvar="set" varname="urlpart">
  <npsobj insertvalue="var" name="varname" />?print=YES
</npsobj>
<a href="@urlpart">...</a>

<!-- Ab Version 6.7.1 -->
<a href="@{varname}?print=YES">...</a>
```

Der folgende Code erzeugt eine URL in Abhängigkeit vom Ergebnis einer konditionalen Auswertung:

```
<npsobj modifyvar="set" varname="jaOderNein">
  <npsobj condition="isEqual" name1="z" value2="1">Ja-Url</npsobj>
  <npsobj condition="isEqual" name1="z" value2="0">Nein-URL</npsobj>
</npsobj>

<!-- Bis Version 6.7.0 -->
<a href="@jaOderNein">...</a>

<!-- Ab Version 6.7.1 -->
<a href="@{jaOderNein}">...</a>
```

13.2.7 Tcl-Prozeduren zur Formatierung von Werten

Bei den NPSOBJ-Anweisungen `insertvalue` `meta` und `var` können Sie das CMS während des Exports eine Tcl-Prozedur ausführen lassen, um das Ergebnis der NPSOBJ-Anweisung zu modifizieren, bevor es in das Ausgabedokument eingefügt wird. Eine solche Tcl-Prozedur wird *Formatter* genannt.

Der anzuwendende Formatter kann mit einem `formatter`-Tag in der NPSOBJ-Anweisung spezifiziert werden. Beispiel:

```
<npsobj insertvalue="var" name="title" formatter="formatter_procedure_alias" />
```

Auch bei insertvalue `dynamiclink` kann ein Formatter angegeben werden. Link-Formatter sind jedoch ein spezieller Formatter-Typ, weil ihnen andere Argumente übergeben werden als den Standard-Formattern (letztere werden normalerweise auf Feldwerte angewendet).

Die Standard-Formatter befinden sich im Verzeichnis `share/script/cm/serverCmds`. Sie müssen im Systemkonfigurationseintrag `export.tclFormatterCommands` registriert worden sein, bevor sie verwendet werden können.

Die Prozedur, deren Alias `formatter_procedure_alias` ist, wird aufgerufen, wenn die NPSOBJ-Anweisung ausgewertet wird. Die Anweisung wird durch den Rückgabewert der Prozedur ersetzt.

Definition der Formatter-Prozedur

Eine Standard-Formatter-Prozedur wird folgendermaßen definiert (die Standard-Formatter befinden sich erst ab Version 6.7.1 im Namensraum `formatter`):

```
namespace eval ::formatter {
  proc meinFormatter {value type params} {
    return "meinFormatter wurde mit dem Wert $value, dem Typ $type\
          und den Parametern $params aufgerufen ."
  }
}
```

Das CMS übergibt der Prozedur die folgenden Argumente in der angegebenen Reihenfolge:

- Das erste Argument, `value`, ist der Wert des durch `name` referenzierten Feldes; enthält die NPSOBJ-Anweisung auch das Tag-Attribut `format` oder `separator`, so wird der formatierte Wert verwendet.
- Als zweites Argument, `type`, wird der Typ des Wertes übergeben. Hier gilt:
 - Ergibt der Wert eine Kontextliste (bei Datei- und Linklisten wie `children` oder `superlinks`, aber auch bei `parent` usw.), ist der Typ `list`.
 - Ist der Wert mit `npsobj modifyVar` zugewiesen worden, ist der Typ `html`.
 - Ergibt der Wert eine Link-Instanz, ist der Typ `link`.
 - Ergibt der Wert einen gültigen 14-stelligen Datumstring, so ist der Typ `date`.
 - In allen anderen Fällen ist der Typ `string`.
- Das dritte Argument, `params`, ist eine Liste mit allen anderen Tag-Attributen und ihren Werten (einschließlich `name`, jedoch ohne `insertvalue`, `formatter`, `format` und `separator`); die Liste enthält paarweise die Tag-Attribut-Namen mit ihren jeweiligen Werten.

Infopark CMS Fiona wird mit den im Folgenden beschriebenen Formatter-Prozeduren ausgeliefert. Es sind hier die Alias-Namen aufgeführt, die in der NPSOBJ-Anweisung als Wert des `formatter`-Attributs angegeben werden können.

formatBlobLength

Aufgabe

Formatiert Dateigrößen menschenlesbar. Das Ergebnis ist eine möglichst kleine Zahl (<1000), die mit der entsprechenden Maßeinheit (KByte, MByte etc.) versehen ist.

Parameter (ab Version 6.7.1)

- `useBinaryPrefix`: Ein Wahrheitswert (`true` oder `false`), mit dem die Art des Präfixes für die Maßeinheit bestimmt wird. Ist dieser Wert wahr (`true`), wird die Dateigröße (Bytes) durch 1024

geteilt und es werden binäre Präfixe verwendet (Ki, Mi, Gi, Ti). Ist der Wert `false` oder nicht angegeben, ist der Divisor 1000 und es werden SI-Präfixe verwendet (K, M, G, T).

- `maxSteps`: Gibt die Anzahl der maximal durchgeführten Divisionen an. Wird hier beispielsweise 2 angegeben, werden auch Werte von mehr als 1000 (bzw. 1024) M(i)Bytes weiterhin als M(i)Byte angezeigt. Der Standardwert ist 2.
- `precision`: Gibt die Anzahl der Nachkommastellen an. Der Standardwert ist 0.
- `separators`: Gibt die Zeichen an, die als 1000er-Trennzeichen und Dezimalpunkt verwendet werden sollen (in dieser Reihenfolge). Wird nur ein Trennzeichen angegeben, wird es als Zeichen für den Dezimalpunkt verwendet, und es werden keine 1000er-Trennzeichen eingefügt. Der Standardwert ist `'.'`.

Beispiele

```
<npsobj ... formatter="formatBlobLength"/>
```

Diese NPSOBJ-Anweisung erzeugt aus dem ermittelten Wert 12345678 die Ausgabe 12 MByte.

Das folgende Beispiel erzeugt aus dem ermittelten Wert 12345678 die Ausgabe 12.056,3 KiByte.

```
<npsobj ... formatter="formatBlobLength" useBinaryPrefix="true" maxSteps="1"
precision="1" separators="."/ >
```

phpVarDef

Aufgabe

Erzeugt eine PHP-Variablendefinition, mit der PHP-Variablen auf Werte gesetzt werden können, die während des Exports berechnet wurden.

Parameter

- `name`: Der Name des Feldes, dessen Wert ermittelt werden soll. Dieser Name wird zugleich als Variablenname in der erzeugten PHP-Zuweisung verwendet.

Beispiel

```
<npsobj insertvalue="var" name="title" formatter="phpVarDef" />
```

Mit der obigen Anweisung wird die Variablendefinition `<?php $title = "Dokumenttitel"; ?>` erzeugt.

formatToRFC822

Aufgabe

Formatiert ein Datum nach RFC-822. Dieses Standardformat wird beispielsweise in RSS-Feeds verwendet.

Parameter (ab Version 6.7.1)

- **withSeconds:** Ein Wahrheitswert (*yes* oder *no*), der bestimmt, ob die Sekunden mit ausgegeben werden. Die Standardeinstellung ist *no*. Da der Standard ein sekundengenaues Format fordert, werden die Sekunden mit *00* angegeben, wenn dieser Wert *no* ist.
- **withTwoDigitYear:** Ein Wahrheitswert (*yes* oder *no*), der bestimmt, ob die Jahreszahl zwei- statt vierstellig ausgegeben werden soll. Die Standardeinstellung ist *no*.

Beispiele

```
<npsobj ... formatter="formatToRFC822"/>
```

Diese Anweisung erzeugt ein Datum der Form 15 Apr 2011 14:36:00 GMT.

Die folgende Anweisung erzeugt ein Datum der Form 15 Apr 11 14:36:23 GMT.

```
<npsobj ... formatter="formatToRFC822" withSeconds="yes" withTwoDigitYear="yes"/>
```

dehtmlize

Aufgabe

Wandelt die HTML-Entitäten `<`, `>`, `"` und `&` in ihre Textform `<`, `>`, `"` und `&` um.

Parameter

Dieser Formatter hat keine Parameter.

Beispiel

```
<npsobj ... formatter="dehtmlize"/>
```

Die obige Anweisung konvertiert den Eingabewert `<div>Groß & "klein"</div>` zu `<div>Groß & "klein"</div>`.

13.3 npsobj break

13.3.1 Syntax

```
npsobj_break ::=
  <npsobj break />
```

13.3.2 Aufgabe

Innerhalb einer `NPSOBJ-list`-Anweisung bewirkt dieses Element, dass der Rest des `NPSOBJ-list`-Elements ignoriert wird und keine weiteren Iterationen für die restlichen Elemente der Liste durchgeführt werden.

Außerhalb eines `NPSOBJ-list`-Elements führt die Anweisung dazu, dass der Export der Datei beendet wird.

13.3.3 Beispiel

```
<npsobj list="toclist" sortKey1="attribute-name">
  <npsobj condition="isEqual" name1="name" value2="news">
    <npsobj break />
  </npsobj>
</npsobj>
```

13.4 npsobj condition

13.4.1 Syntax

```
npsobj_condition ::=
  ( <npsobj condition = "unary_condition" data [ negate = "" ]
    [ separator = "separator" ] [ format = "format_name" ] >
    content </npsobj> ) |
  ( <npsobj condition = "binary_condition" data1 data2
    [ negate = "" ] [ separator = "separator" ]
    [ format = "format_name" ] >
    content </npsobj> )

unary_condition ::= isEmpty | isEmpty | isNil | isNotNil

binary_condition ::= isEqual | isNotEqual | isGreaterThan |
  isLessThan | hasPrefix | hasSuffix | isCaseEqual | matches

data ::= ( value = "value" ) | ( name = "name" )

data1 ::= ( value1 = "value" ) | ( name1 = "name" )

data2 ::= ( value2 = "value" ) | ( name2 = "name" )
```

13.4.2 Aufgabe

Diese Anweisung erlaubt es, den Export eines Textes davon abhängig zu machen, dass eine Bedingung erfüllt ist. Die Bedingung wird mit Hilfe eines Namens wie beispielsweise `isEqual` als Wert des Tag-Attributs `condition` angegeben (siehe die folgende Liste der Namen für Bedingungen); die Operanden der Bedingung werden über weitere Attribute festgelegt.

Die Bedingung wird zunächst ausgewertet. Wenn sie erfüllt ist, wird das gesamte NPSOBJ-Element durch `content` ersetzt. Wenn sie nicht erfüllt ist, wird das NPSOBJ-Element durch die leere Zeichenkette ersetzt.

Ist der Parameter `negate` angegeben, wird der Wert der Bedingung, der sich entsprechend der folgenden Spezifikation der einzelnen Werte für `condition` ergibt, negiert. Dann erst wird entschieden, ob das Element durch `content` oder durch die leere Zeichenkette ersetzt wird.

Die Auswertung von `data`, `data1` und `data2` folgt demselben Schema. Es wird hier für `data` definiert:

Ist `value="value"` angegeben, so ist der Wert von `data` die Konstante `value`. Ist `value` nicht angegeben und ist `name="name"` angegeben, so ist der Wert von `data` das Ergebnis der Wertbestimmung von `name` als Zeichenkette mit dem Separator `separator` und dem Datumsformat `format_name`. Ist (fehlerhafterweise) weder das `value`- noch das `name`-Attribut angegeben, ist der

Wert von `data` die leere Zeichenkette. Sind (fehlerhafterweise) `value="value"` und `name="name"` angegeben, wird `name="name"` ignoriert.

Der Separator muss angegeben werden, wenn `name` angegeben wurde und `name` eine Liste referenziert. Dies ist bei den Werten eines Mehrfachauswahlfeldes der Fall.

Bitte beachten Sie, dass bei binären Bedingungen `name` und `value` mit den Indizes 1 und 2 verwendet werden müssen. Die Indizes legen bei den Bedingungstypen `hasPrefix`, `hasSuffix`, `isGreaterThan`, `isLessThan` und `matches` fest auf welcher Seite des Operators sich der jeweilige Operand befindet.

isEmpty, isNil

Die Bedingungen sind wahr, wenn der Wert von `data` eine leere Zeichenkette oder eine leere Liste ergibt oder nicht existiert. Die beiden Bedingungen sind gleichbedeutend, `isNil` ist aus Kompatibilitätsgründen noch verfügbar.

isNotEmpty, isNotNil

Die Bedingungen sind wahr, wenn der Wert von `data` existiert und eine nicht leere Zeichenkette oder eine nicht leere Liste ergibt. Die beiden Bedingungen sind gleichbedeutend, `isNotNil` ist aus Kompatibilitätsgründen noch verfügbar.

isEqual

Die Bedingung ist genau dann wahr, wenn der Wert von `data1` zeichenweise mit dem Wert von `data2` übereinstimmt.

isNotEqual

Die Bedingung ist genau dann falsch, wenn der Wert von `data1` zeichenweise mit dem Wert von `data2` übereinstimmt.

isGreaterThan

Die Bedingung ist genau dann wahr, wenn der Wert von `data1` größer als der Wert von `data2` ist. Die Werte werden zeichenweise von links nach rechts miteinander verglichen. Datumswerte werden vor dem Vergleich in ihre 14-stellige ISO-Darstellung und Listen in Zeichenketten umgewandelt. Durch die Negation der Bedingung kann ein Kleiner-gleich-Vergleich durchgeführt werden.

isLessThan

Analog zu `isGreaterThan` ist die Bedingung genau dann wahr, wenn der Wert von `data1` kleiner als der Wert von `data2` ist. Durch die Negation der Bedingung kann ein Größer-gleich-Vergleich durchgeführt werden.

hasPrefix

Die Bedingung ist genau dann wahr, wenn der Anfang von `data1` zeichenweise mit `data2` übereinstimmt.

hasSuffix

Die Bedingung ist genau dann wahr, wenn die Länge `n` von `data2` kleiner ist als die von `data1` und die letzten `n` Zeichen von `data1` zeichenweise mit `data2` übereinstimmen.

isCaseEqual

Die Bedingung ist genau dann wahr, wenn der Wert von `data1` zeichenweise mit dem Wert von `data2` übereinstimmt, wobei Groß- und Kleinschreibung nicht unterschieden wird.

matches

Die Bedingung ist genau dann wahr, wenn der Wert von `data1` auf den regulären Ausdruck `data2` passt. Ist `data2` kein gültiger regulärer Ausdruck, so ist die Bedingung falsch. Ist `data2` die leere Zeichenkette, so ist die Bedingung wahr.

Reguläre Ausdrücke müssen der erweiterten POSIX-Syntax gehorchen. Beachten Sie bitte, dass der tatsächlich verfügbare Funktionsumfang von der POSIX-ERE-Implementation in dem verwendeten Betriebssystem abhängt. In der folgenden Aufstellung werden die Spezialsymbole und ihre Bedeutung erläutert:

- [] In eckigen Klammern wird ein Bereich von Symbolen angegeben, in dem sich das untersuchte Zeichen befinden muss, damit der reguläre Ausdruck passt. Wenn das untersuchte Zeichen sich nicht in dem Bereich befinden darf, so muss dem Bereich das Symbol `^` vorangestellt werden. Das Minuszeichen kann nur als Literal verwendet werden, wenn es als erstes oder letztes Symbol auftritt. `[=-]` trifft auf die Zeichen `=` und `-` zu. Beispiele: `[abc]`, `[a-z]`, `[^0-9]`
- * Der `*` nach einer Komponente eines regulären Ausdrucks bedeutet: kann beliebig oft vorkommen (auch gar nicht). Beispiel: `1[a-z]*[0-9]`
- .
- ^ Das Symbol `^` am Anfang eines Ausdrucks bewirkt, dass die zu vergleichende Zeichenkette vom Anfang her mit dem regulären Ausdruck übereinstimmen muss. Innerhalb eckiger Klammern bedeutet das Symbol, dass das verglichene Zeichen nicht in dem angegebenen Bereich enthalten sein darf.
- \$ Das Symbol `$` am Ende eines Ausdrucks bewirkt, dass die zu vergleichende Zeichenkette vom Ende her mit dem regulären Ausdruck übereinstimmen muss. `\n` entspricht einer neuen Zeile.
- + Das Pluszeichen nach einem regulären Ausdruck bedeutet *einmal oder öfter*. Beispielsweise ist `[0-9]+` gleichbedeutend mit `[0-9][0-9]*`.
- () Runde Klammern werden zur Gruppierung verwendet. Ein Operator wie `*` oder `+` kann auf ein einzelnes Zeichen angewendet werden oder auf einen von Klammern umschlossenen regulären Ausdruck. Beispiel: `(a*(cb+)*)`.
- { } Geschweifte Klammern werden zur Quantifizierung verwendet. Nach einem Element kann mit `{n}`, `{n,m}` oder `{n,}` verlangt werden, dass das Element *n*-mal, *n*-mal bis *m*-mal bzw. wenigstens *n*-mal vorkommen muss. Beispiel: `[0-9]{14}`
- | Der vertikale Strich bedeutet *oder*. Beispiel: `(abc|def)`

Sämtlichen oben aufgeführten Symbole muss ein umgekehrter Schrägstrich (Backslash) vorausgehen, damit sie als Literal verwendet werden können.

13.4.3 Beispiel

Im folgenden Beispiel wird in einem Link-Kontext nur dann ein Link erzeugt, wenn die URL die Zeichenkette `://` enthält, d. h. wenn es sich um einen externen Link handelt.

```
<npsobj name1="destinationUrl" value2="://" condition="matches">
  <npsobj insertvalue="anchor" name="destinationUrl">
```

```

    <npsobj insertvalue="var" name="displayTitle"/>
  </npsobj>
</npsobj>

```

13.5 npsobj context

13.5.1 Syntax

```

npsobj_context ::=

  <npsobj context = "path">
    content
  </npsobj>

```

13.5.2 Aufgabe

Diese Anweisung bewirkt, dass *content* im Kontext von *path* ausgewertet wird. Als *path* wird der Pfad zu einer Datei angegeben. Dieser Pfad wird in die Linkverwaltung aufgenommen.

13.5.3 Beispiel

Der folgende Code wechselt zur Datei mit dem Pfad `/newspub/news/news1`, wodurch diese zur Datenquelle für die darauf folgenden Anweisungen wird. Anschließend wird ein Link auf diese Datei erzeugt, bei dem der `displayTitle` als verlinkter Text verwendet wird.

```

<npsobj context="/newspub/news/news1">
  <npsobj insertvalue="anchor" name="destinationUrl">
    <npsobj insertvalue="var" name="displayTitle"/>
  </npsobj>
</npsobj>

```

13.6 npsobj frame

13.6.1 Syntax

```

npsobj_frame ::=
  <npsobj frame = "name"
    [ defaulttarget = "target" ]
    { other_attribute } >
    content
  </npsobj>

other_attribute ::= string = "string"

```

13.6.2 Aufgabe

Diese Anweisung kann nur in einer Layoutdatei verwendet werden (siehe auch [Framesets mit dem NPSOBJ-Frame-Kommando automatisch erzeugen](#)). Neben der Ausgabedatei, die das CMS beim Export einer Datei erzeugt, wird bei der Auswertung dieser Anweisung eine weitere Ausgabedatei generiert. Diese Datei erhält den Namen `objName.name.contentType`, wobei `objName` der Name der exportierten Datei, `name` der im `frame`-Attribut des Tags angegebene Name und `contentType` die Dateiendung der exportierten Datei ist. Die Ausgabedatei enthält den ausgewerteten `content` der Anweisung.

Die Anweisung wird durch ein `frame`-Tag ersetzt, das wenigstens die Tag-Attribute `name` und `src` enthält. Das `name`-Attribut wird auf `name` und das `src`-Attribut auf den Namen der durch die NPSOBJ-frame-Anweisung erzeugten Ausgabedatei gesetzt. Sämtliche anderen in der NPSOBJ-frame-Anweisung angegebenen Attribute (`other_attribute`) werden unverändert in das `frame`-Tag übernommen.

Enthält die Anweisung das Tag-Attribut `defaulttarget="target"`, so werden im ausgewerteten `content` alle Link-Tags, die das Attribut `target` unterstützen, mit diesem Attribut versehen, wobei sein Wert auf `"target"` gesetzt wird. Ein im Link-Tag bereits vorhandenes `target`-Attribut wird jedoch nicht verändert. Ist in der NPSOBJ-Anweisung kein `defaulttarget` angegeben, so wird `_top` als Wert für `defaulttarget` verwendet.

NPSOBJ-frame-Anweisungen dürfen nicht verschachtelt werden.

13.6.3 Beispiel

Beispiele zum NPSOBJ-Frame-Befehl finden Sie in Abschnitt [Framesets mit dem NPSOBJ-Frame-Kommando automatisch erzeugen](#).

13.6.4 Verwendung von `target="_top"` in der Vorschau

In einem Link haben die Frame-Werte `_top` und `_parent` die besondere Eigenschaft, das Linkziel in einem übergeordneten Frame innerhalb des Browserfensters darzustellen. Da der Content-Navigator ab Version 6.5 für die Vorschau Frames verwendet, werden beim Klicken auf einen Link in der Vorschau unter Umständen Bedienelemente wie das Menü und die Werkzeugleiste unzugänglich.

Dieses Verhalten lässt sich umgehen, indem der Content angepasst wird. Erweitern Sie hierfür das Layout, mit dem HTML-Head und -Body erzeugt werden, um folgende Anweisungen:

```
<head>
...
<npsobj condition="isEqual" name1="exportMode" value2="preview">
  <npsobj modifyvar="set" varname="adjustTargetsInPreview">adjustTargetsInPreview() </npsobj>
  <script type="text/javascript">
    function <npsobj insertvalue="var" name="adjustTargetsInPreview" />
    {
      var previewFrameName = "nps_browser_frame";
      var elements = document.getElementsByTagName('a');
      for (var i = 0; i < elements.length; i++) {
        var target = elements[i].target;
        if ("_top"==target || ("_parent"==target && previewFrameName == self.name)) {
          target = previewFrameName;
        }
        elements[i].target = target;
      }
    }
  </script>
</npsobj>
```

```

    </script>
  </npsobj>
  ...
</head>

<body onload="@{adjustTargetsInPreview}">
  ...
</body>

```

Dieses Skript sorgt innerhalb der Vorschau dafür, dass der Frame-Wert eines Links nie den eigentlichen Vorschaubereich verlässt.

13.7 npsobj includetext

13.7.1 Syntax

```

npsobj_includetext ::=
  <npsobj includetext = "internal_url" />

```

13.7.2 Aufgabe

Bei der Auswertung wird diese Anweisung durch den ausgewerteten Hauptinhalt der Datei ersetzt, die über *internal_url* referenziert wird. Existiert diese Datei nicht, wird das Element durch die leere Zeichenkette ersetzt. Die Datei muss ein Dokument oder ein Ordner sein.

Intern wird diese Anweisung als Link des Typs *Einfügelink* verwaltet. Dadurch unterliegt *internal_url* der [Linkverwaltung](#) und wird automatisch angepasst, wenn das Linkziel oder die Datei, die diese Anweisung enthält, verschoben wird.

13.7.3 Beispiel

```

<npsobj includetext="/docs/technical/index.html" />

```

13.8 npsobj insertvalue anchor

13.8.1 Syntax

```

npsobj_insertvalue_anchor ::=
  <npsobj insertvalue = "anchor"
    name = "name"
    { other_attribute } >
    content
  </npsobj>

other_attribute ::= string = "string"

```


13.8.2 Aufgabe

Diese Anweisung erzeugt ein a-Element. Der ausgewertete `content` der Anweisung wird zum Inhalt des erzeugten a-Elements. Das a-Tag enthält die folgenden Attribute:

- Es wird ein `href`-Attribut erzeugt. Zur Bestimmung des Wertes von `href` wird nach einer Url für `name` gefragt. `name` muss einen Dateikontext ergeben.
- Alle Attribute des NPSOBJ-Tags außer `insertvalue` und `name` werden in das resultierende a-Tag übernommen, gegebenenfalls nachdem die in ihren Werten enthaltenen @-Referenzen aufgelöst wurden.

Um einen Link auf die aktuelle Datei zu erzeugen, können Sie eine `insertvalue=anchor`-Anweisung mit `name="self"` verwenden.

13.8.3 Beispiel

Der folgende Code verlinkt den Ordner, in dem die aktuelle Datei enthalten ist (also den `parent`-Ordner). Als der verlinkte Text wird der Titel dieses Ordners verwendet.

```
<npsobj insertvalue="anchor" name="parent">
  <npsobj insertvalue="var" name="parent.title"/>
</npsobj>
```

13.9 npsobj insertvalue dynamiclink

13.9.1 Syntax

```
npsobj_insertvalue_dynamiclink ::=

  <npsobj insertvalue = "dynamiclink"
    destination = "destination"
    [ formatter = "formatter_procedure_alias" ]
    {other_attribute} />

  other_attribute ::= string = "string"
```

13.9.2 Aufgabe

Das Element dient dazu, `include`-Anweisungen in der Syntax beliebiger Skriptsprachen (PHP, JSP, SSI) zu erzeugen und die Referenz in die automatische [Linkverwaltung](#) aufzunehmen. Das Element sollte aus Performance-Gründen nur zu seinem eigentlichen Zweck – die serverseitige Inklusion von Dateien – und nicht zur Berechnung relativer Pfade verwendet werden. Für letzteres ist ein Formatter in einer `insertvalue-var`-Anweisung oder eine `systemexecute`-Anweisung ausreichend. Damit dynamische Links in der Vorschau funktionieren, muss die [Vorschau dynamisch](#) sein.

Der erzeugte Link hat den Typ `dynamic`.

Die Attribute haben die folgende Bedeutung:

- Wenn `formatter` angegeben ist, wird die Tcl-Prozedur ausgeführt, die im Systemkonfigurationseintrag `export.tclDynamicLinkFormatterCommands` den Alias `formatter_procedure_alias` hat. Dieser Prozedur werden die unten aufgeführten Argumente übergeben. Der Rückgabewert dieser Prozedur wird anstelle der gesamten NPSOBJ-Anweisung in die exportierte Seite eingefügt. Wenn `formatter` nicht angegeben ist, so wird anstelle der Anweisung `destination` eingefügt.
- Mit `destination` wird der Pfad oder die URL angegeben, die die Formatierungsprozedur verarbeiten soll. `destination` kann ein relativer oder absoluter Pfad oder eine URL sein. Pfade werden als interne, URLs als externe Links in die Linkverwaltung übernommen.
- Als `other_attribute` können beliebige Attribute angegeben werden.

Da die referenzierte Datei von beliebigen Dateien in der Live-Server-Verzeichnishierarchie inkludiert werden und auch selbst wieder Links enthalten kann, lassen sich der Content Manager und die Template Engine mit dem Systemkonfigurationseintrag `export.exportAbsolutePaths` so einstellen, dass interne Pfade und URLs in der referenzierten Datei absolut exportiert werden. Ist dieser Wert auf `YES` gesetzt, werden die im Eintrag `export` konfigurierbaren Präfixe für absolute Pfade und URLs verwendet.

Der Formatierungsprozedur werden die folgenden Argumente in der angegebenen Reihenfolge übergeben:

- Die URL des Ziels, d. h. der CMS-Dateipfad, gegebenenfalls mit URL-Präfix;
- der Dateisystem-Pfad des Ziels, gegebenenfalls mit Dateisystem-Präfix;
- die ID der exportierten Datei;
- eine Liste mit allen `other_attribute`-Attributen und ihren Werten als Name-Wert-Paare.

13.9.3 Beispiel

Die folgende Anweisung erzeugt in der exportierten Datei eine PHP-include-Anweisung, die die Datei mit dem Pfad `../some/path` inkludiert. Es wird ein Link auf die referenzierte Datei erzeugt, der in die Linkverwaltung aufgenommen wird. Wird – bei Referenzen auf CMS-Dateien – die referenzierte Datei verschoben, so werden der Link und die NPSOBJ-Anweisung automatisch angepasst.

```
<npsobj insertvalue="dynamicLink" destination="../some/path" formatter="phpInclude" />
```

Die Anweisung gibt die folgende PHP-Anweisung zurück:

```
<? include("/from/root/to/some/path"); ?>
```

13.10 npsobj insertvalue image

13.10.1 Syntax

```
npsobj_insertvalue_image ::=
  <npsobj insertvalue = "image" name = "name"
  { other_attribute } >
  </npsobj>

other_attribute ::= string = "string"
```

13.10.2 Aufgabe

Diese Anweisung erzeugt ein `img`-Tag mit den unten angegebenen Attributen. Mit dieser Anweisung können keine Bilder-Links mit externen URLs generiert werden. Verwenden Sie hierfür in Ihrem Layout ein manuell eingefügtes `img`-Tag.

- Wenn `name` angegeben ist, wird nach einer URL für `name` gefragt. Wenn diese nicht leer ist, wird das Attribut `src` mit der URL als Wert eingefügt. Der Wert von `name` muss einen Dateikontext ergeben.
- Alle Attribute des NPSOBJ-Tags außer `insertvalue` und `name` werden in das resultierende `img`-Tag übernommen, gegebenenfalls nachdem die in ihren Werten enthaltenen `@`-Referenzen aufgelöst wurden.

13.10.3 Beispiel

Im folgenden Beispiel wird zunächst in den Kontext einer Bild-Datei gewechselt. Dieser Kontext wird dann in der `insertvalue=image`-Anweisung mit `self` referenziert.

```
<npsobj context="/resources/banners/large">
  <npsobj insertvalue="image" name="self" height="50%" width="50%" />
</npsobj>
```

13.11 npsobj insertvalue link

13.11.1 Syntax

```
npsobj_insertvalue_link ::=
  <npsobj insertvalue="link" name = "name" />
```

13.11.2 Aufgabe

`link`-Elemente können sich im `head`-Element einer Webseite befinden. Sie ermöglichen es dem Browser (sofern dieser das Feature unterstützt), zusätzliche Navigationselemente anzuzeigen, mit denen man zu anderen Seiten gelangt, die in der Website-Struktur relevant sind (etwa das übergeordnete Dokument oder die Suchseite).

Diese NPSOBJ-Anweisung fügt drei `link`-Elemente in die Ausgabedatei ein. Das erste `link`-Tag enthält folgende Attribute:

- `rel` mit dem Wert `up`
- `href` mit dem Wert, der geliefert wird, wenn man im Kontext `name` nach der URL von `parent` fragt.

Das zweite `link`-Tag enthält folgende Attribute:

- `rel` mit dem Wert `next`
- `href` mit dem Wert, der geliefert wird, wenn man im Kontext `name` nach der Url von `next` fragt.

Das dritte `link`-Tag enthält folgende Attribute:

- `rel` mit dem Wert `previous`
- `href` mit dem Wert, der geliefert wird, wenn man im Kontext `name` nach der Url von `previous` fragt.

13.11.3 Beispiel

```
<npsobj insertvalue="link" name="self"/>
```

13.12 npsobj insertvalue meta

13.12.1 Syntax

```
npsobj_insertvalue_meta ::=
<npsobj insertvalue = "meta"
  [ content = "name" ]
  ( name = "varname" ) | ( http-equiv = "varname" )
  [ format = "format_name" ]
  [ formatter = "formatter_name" ]
  [ separator = "separator" ] >
</npsobj>
```

13.12.2 Aufgabe

Das Element wird durch ein `meta`-Tag mit zwei Attributen ersetzt:

- Das Attribut `name` oder `http-equiv` mit dem Wert `varname`.
- Das Attribut `content` mit einem im Folgenden beschriebenen Wert.

Des Wert des `content`-Attributs wird zunächst genauso ermittelt, wie bei `npsobj_insertvalue_var` beschrieben; allerdings wird der Wert `name` des Attributs `content` für die Wertermittlung verwendet, nicht etwa der Wert `varname` des Attributs `name` oder `http-equiv`. Anschließend werden reservierte HTML-Zeichen (wie die spitzen Klammern) im so ermittelten Wert in ihr HTML-Äquivalent umgewandelt.

Bezieht sich `name` auf eine Liste, so muss ein Separator angegeben werden. `separator` kann jedoch die leere Zeichenkette sein.

Ein `formatter` kann erst ab Version 6.7.0 angegeben werden. Er kann dazu verwendet werden, den Rückgabewert der NPSOBJ-Anweisung zu formatieren.

13.12.3 Beispiel

```
<npsobj insertvalue="meta" name="keywords" content="keywordList" separator="" />
<npsobj modifyvar="set" varname="metaValue">text/html; charset="@exportCharset" </npsobj>
<npsobj insertvalue="meta" http-equiv="Content-Type" content="metaValue" />
```

13.13 npsobj insertvalue systemexecute

13.13.1 Syntax

```
npsobj_insertvalue_systemexecute ::=

<npsobj insertvalue = "systemexecute" name = "name" >
  content
</npsobj>
```

13.13.2 Aufgabe

Mit dieser Anweisung lässt sich während des Exports einer Datei ein berechneter Wert in die erzeugte Ausgabedatei einfügen. Die Anweisung wird häufig dazu verwendet, Feldwerte der exportierten Datei (beispielsweise ein Datum) umzuformatieren oder mehrere Feldwerte miteinander zu verrechnen.

Hierzu wird die über *name* referenzierte Tcl-Prozedur mit zwei Argumenten aufgerufen, der betreffenden Datei- oder Link-ID sowie dem ausgewerteten *content*. Der Rückgabewert der Prozedur wird anstelle der NPSOBJ-Anweisung in die Ausgabedatei eingefügt.

name wird als der Alias eines Prozedurnamens aufgefasst, der im Systemkonfigurationseintrag [export.tclSystemExecuteCommands](#) hinterlegt ist. Kann der Alias in `tclSystemExecuteCommands` nicht gefunden werden, so wird kein Tcl-Befehl ausgeführt, sondern das NPSOBJ-Element wird durch die Zeichenkette `Error: Execution of 'name' not allowed` ersetzt.

Wenn Sie die Template Engine einsetzen, so beachten Sie bitte, dass die beim Export auszuführenden Prozeduren auch auf dem Live-Server verfügbar sein müssen.

13.13.3 Beispiel

Der folgende Code ruft die Funktion mit dem Alias `command_alias` auf. Ihr werden als Argumente die ID der zu exportierenden Datei sowie der Wert des kundenspezifischen Attributs `attr_name` übergeben. Anschließend wird der Code durch den Rückgabewert der Funktion ersetzt.

```
<npsobj insertvalue="systemexecute" name="command_alias">
  <npsobj insertvalue="var" name="attr_name" />
</npsobj>
```

13.14 npsobj insertvalue template

13.14.1 Syntax

```
npsobj_insertvalue_template ::=
```

```

<npsobj insertvalue = "template"
  ( name = "name" ) | ( key = "key" )
  { other_attribute } >
  content
</npsobj>

other_attribute ::= string = "string"

```

13.14.2 Aufgabe

Diese Anweisung ersetzt das Element durch eine ausgewertete Layoutdatei. Ist das Attribut `name` angegeben, wird dessen Wert als Name der Layoutdatei verwendet, andernfalls wird der Name im Feld `key` des aktuellen Kontexts nachgeschlagen.

Die Suche nach einer Layoutdatei mit dem ermittelten Namen beginnt im aktuellen Kontext. (Beim Export einer Datei entspricht der aktuelle Kontext anfangs dieser Datei. Der Kontext kann jedoch im Verlauf des Exports durch weitere Exportanweisungen umgesetzt werden.) Ist der aktuelle Kontext ein Ordner, beginnt die Suche in diesem Ordner, andernfalls in dem Ordner, der die CMS-Datei enthält, die den aktuellen Kontext bereitstellt. Ist der aktuelle Kontext ein Link, beginnt die Suche bei der CMS-Datei, die den Link enthält.

Existiert die Layout-Datei nicht an der Stelle, die wie beschrieben ermittelt wurde, wird immer weiter in den darüber liegenden Ordnern gesucht, bis sie gefunden wird oder der Basisordner erreicht ist. Gibt es auch im Basisordner keine Layoutdatei mit dem gesuchten Namen, wird eine entsprechende Fehlermeldung als Ergebnis verwendet.

Bis Version 6.7.0 werden andere Attribute als `name` und `key` sowie der `content` des Elements ignoriert.

Erweiterungen ab Version 6.7.1

Im NPSOBJ-Tag können neben `name` und `key` weitere Attribute angegeben werden, um der inkludierten Layout-Datei Argumente zu übergeben. Der Wert eines solchen Arguments kann nur Text sein, keine Kontextliste. Bei Bedarf kann der Wert eines Arguments mit einer @-Referenz berechnet werden.

Die Argumente werden wie Exportvariablen behandelt, sind jedoch nur innerhalb des aufgerufenen Layouts sichtbar, d.h. sie sind lokale Variablen. Um Namenskonflikte mit globalen, mittels `modifyvar-set` oder `modifyvar-append` erzeugten Exportvariablen zu vermeiden, sollten die Namen der lokalen Variablen eindeutig sein. Eine lokale Variable kann mit `modifyvar-set` oder `modifyvar-append` modifiziert und mit `modifyvar-clear` gelöscht werden. Existiert die in diesen Anweisungen angegebene lokale Variable nicht, werden die Anweisungen auf die gleichnamige globale Exportvariable angewendet, falls diese existiert.

Zusätzlich zu den Argumenten wird der nicht ausgewertete `content` des Elements der inkludierten Layoutdatei als Wert der ebenfalls lokalen Variablen `npsobjContent` übergeben. Im `content` enthaltene NPSOBJ-Anweisungen werden erst ausgewertet, wenn `npsobjContent` ausgelesen wird, d.h. im jeweiligen lokalen Kontext.

Auf die Argumente und `npsobjContent` können Sie innerhalb des inkludierten Layouts so zugreifen wie auf andere Felder und Exportvariablen auch, also beispielsweise mit der Anweisung [`insertvalue-var`](#).

13.14.3 Beispiele

Der folgende Code prüft, ob die exportierte Datei oder einer ihrer darüber liegenden Ordner ein Feld mit dem Namen `listTemplateName` enthält. Ist der Wert dieses Feldes nicht der leere Wert, so wird er als Name einer Layoutdatei aufgefasst, die dann verwendet wird.

```
<npsobj condition="isNotEmpty" name="up.listTemplateName">
  <npsobj insertvalue="template" key="up.listTemplateName" />
</npsobj>
```

Das folgende Beispiel zeigt, wie der inkludierten Layoutdatei ein Argument als Attribut (`showbody`) sowie ein Text im Elementinhalt übergeben werden. Beide Werte werden von der Layoutdatei ausgegeben. Diese beiden Techniken der Übergabe von Werten an Layouts sind ab Version 6.7.1 verfügbar.

```
<!-- Anweisungen im inkludierenden Layout -->
<npsobj insertvalue="template" name="TemplateName" showbody="1">
  <h1>Bereichsüberschrift</h1>
  <p><strong><npsobj insertvalue="var" name="title"/></strong></p>
</npsobj>

<!-- Inhalt des Layouts TemplateName -->
<npsobj insertvalue="var" name="npsobjContent"/>
<npsobj condition="isEqual" name1="showbody" value2="1">
  <p><npsobj insertvalue="var" name="body"/></p>
</npsobj>
```

Beim Aufruf des inkludierenden Layouts erzeugt das obige Fragment die folgende Ausgabe (die Farben dienen nur zur Veranschaulichung):

```
<h1>Bereichsüberschrift</h1>
<p><strong>Titel des exportierten Dokuments</strong></p>
<p>Dies ist der Body des exportierten Dokuments.</p>
```

13.15 npsobj insertvalue var

13.15.1 Syntax

```
npsobj_insertvalue_var ::=

<npsobj insertvalue = "var" name = "name"
  [ format = "format_name" ]
  [ formatter = "formatter_procedure_alias" ]
  [ separator = "separator" ] />
```

13.15.2 Aufgabe

Die NPSOBJ-Anweisung wird durch den Wert eines Feldes oder einer Exportvariablen ersetzt. Dazu wird im aktuellen Kontext nach dem Zeichenkettenwert für *name* mit dem Format *format_name* und dem Separator *separator* gefragt. Falls angegeben, wird der Separator verwendet, wenn nach dem Wert eines Feldes vom Typ *Mehrfachauswahl* gefragt wird und dieser Wert aus mindestens zwei

Elementen besteht. In diesem Fall wird der Separator als Trennzeichenkette zwischen den Elementen verwendet.

Das Format *format_name* wird nur angewendet, wenn der Wert von *name* vom Typ *Datum* ist. Es kann einer der Formatnamen angegeben werden, die im Systemkonfigurationseintrag [validDateTimeOutputFormats](#) definiert sind.

Mit dem Tag-Attribut *formatter* kann der Wert von *name* mit einer Tcl-Prozedur formatiert werden.

13.15.3 Beispiel

Den Titel der exportierten Version ausgeben:

```
<npsobj insertvalue="var" name="title" />
```

Die folgende Anweisung erzeugt eine PHP-Variablendefinition, wenn *phpVardef* der Alias einer Prozedur ist, die den erforderlichen PHP-Code erzeugt.

```
<npsobj insertvalue="var" name="title" formatter="phpVardef" />
```

Die Prozedur *phpVardef* ist als Serverkommando bereits in der Datei *serverCmds/formatter.tcl* enthalten. Wird diese Prozedur durch die obige NPSOBJ-Anweisung aufgerufen, so wird folgende Ausgabe erzeugt:

```
<?php $title = "Titel der aktuellen Datei"; ?>
```

13.16 npsobj list

13.16.1 Syntax

```
npsobj_list ::=

<npsobj list = "name"
  [ sortkey1 = "sortKey1"
    [ sortmodifier1 = sortModifier ]
    [ sortkey2 = "sortKey2"
      [ sortmodifier2 = sortModifier ]
      [ sortkey3 = "sortKey3"
        [ sortmodifier3 = sortModifier ]]]]
  [ start = "startIndex" ]
  [ ( length = "length" ) | ( end = "end" ) ]
  [ reverse = "" ] >
content
</npsobj>

sortModifier ::=
  "alpha" |
  "numeric" |
  "ascending" |
  "descending" |
  "alpha ascending" |
  "alpha descending" |
  "numeric ascending" |
  "numeric descending"
```


13.16.2 Aufgabe

Die Anweisung wird häufig verwendet, um Inhaltsübersichten automatisch zu generieren. Sie erzeugt eine Liste von Datei- oder Linkkontexten, indem *name* ausgewertet wird. *name* steht hierbei für einen Bezeichner, der eine Kontextliste ergibt, beispielsweise `toclist` oder `children`. Als *name* kann auch der Name eines Linklistenfeldes oder einer Exportvariablen angegeben werden, der vorher eine Kontextliste zugewiesen wurde.

Für jedes Element in der Liste wird *content* genau einmal ausgewertet (siehe jedoch auch [npsobj_break](#)). Für die Auswertung von *content* wird der aktuelle Kontext temporär auf das jeweilige Listenelement gesetzt. Dadurch beziehen sich sämtliche Namen (beispielsweise Feldnamen) auf das aktuelle Listenelement, solange der Kontext innerhalb von *content* nicht gewechselt wird.

Sortierung der Liste

Die Kontextliste kann sortiert werden, indem die Werte `sortKey1` der Kontexte verglichen werden. Sind zwei Werte gleich, so werden die Werte von `sortKey2` herangezogen. Sind auch diese Werte gleich, wird die Reihenfolge der Kontexte anhand von `sortKey3` bestimmt. Die Reihenfolge von Kontexten, bei denen alle drei Sortierschlüssel jeweils denselben Wert haben, ist nicht definiert. Ist `sortkey1` nicht angegeben, so wird die Kontextliste nach den Sortierschlüsseln sortiert, die für den darüber liegenden Ordner angegeben wurden.

Mit dem Modifikator `sortModifier` kann man angeben, wie die Werte zweier Sortierschlüssel miteinander verglichen werden sollen. `alpha` sortiert nach Zeichenwert, `numeric` numerisch, d. h. die Zeichenfolge wird als Zahl interpretiert. `ascending` sortiert aufsteigend, `descending` absteigend. Jeweils ein Schlüsselwort der beiden Modifikatorpaare (`alpha/numeric` und `ascending/descending`) kann miteinander kombiniert werden (Beispiel: `numeric descending`). Der voreingestellte Sortiermodus ist `alpha`, die voreingestellte Sortierrichtung `ascending`. Die voreingestellte Modifikatorkombination ist also `alpha ascending`.

Auswahl von Listenelementen

Ist das Attribut `start` angegeben und hat `startIndex` einen Wert größer 0, so ist `startIndex` der Index des ersten zu berücksichtigenden Elements der Kontextliste. Das erste Element hat den Index 1. Hat `startIndex` einen Wert kleiner 0, so wird das erste zu berücksichtigende Element vom Ende der Kontextliste her ermittelt. Ist `start` gleich 0, so wird das Attribut ignoriert.

Mit dem Wert des Attributs `length` kann die Anzahl der Elemente in der Kontextliste festgelegt werden. Ist `length` gleich 0 (Voreinstellung), so werden die Elemente ab `start` bis zum Ende der Liste zurückgegeben. Dagegen bestimmt ein Wert größer 0 die maximale Anzahl der ab `start` zu liefernden Elemente. Ist `length` kleiner als 0, so bezeichnet `start` das letzte Element der Liste, die höchstens so viele Elemente hat, wie es der Absolutwert von `length` festlegt. Mit `start="-1"` und `length="-2"` erhält man also das vorletzte und das letzte Element der Liste.

Alternativ zu `length` kann das Attribut `end` angegeben werden. Ist `end` eine positive Zahl, so bezeichnet diese Zahl den Index des letzten zu berücksichtigenden Elements der ursprünglichen Liste. Das erste Element hat den Index 1. Ist `end` negativ, so werden vom Ende der Liste her so viele Elemente ausgespart wie der Absolutwert von `end` angibt.

Die Wirkung von `start` und `length` bzw. `end` zeigt die folgende Übersicht. Das durch `start` gewählte Element ist grün gefärbt.

Ausgewählte Elemente	start	length	end	
1■□□□□6				
1□□■□□6	3	2		
1□□□□■6	5	-2		
1□□□■□6	-3	2		
1□□□□■6	-1	-3		
1□□■□□6	3		5	
1□■□□□6	2		-2	
1□□□■□6	-3		5	
1□□■□□6	-4		-1	

Legende
□ Nicht ausgewähltes Element
■ Erstes ausgewähltes Element
■ Weiteres ausgewähltes Element

Ist das Attribut `reverse` angegeben, so wird die Reihenfolge der Elemente umgekehrt, nachdem sie gegebenenfalls sortiert wurden und bevor diese mit `start` und `length` bzw. `end` gekürzt werden.

13.16.3 Beispiel

```
<npsobj list="toclist" start="2" length="3" sortkey1="title" sortmodifier1="ascending">
  <!-- Dies wird für jedes Listenelement abgearbeitet -->
  <npsobj insertvalue="anchor" name="self">
    <npsobj insertvalue="var" name="title" /><br />
  </npsobj>
</npsobj>
```

13.17 npsobj micronavigation

13.17.1 Syntax

```
npsobj_micronavigation ::=
  <npsobj micronavigation = "fieldname" [ levels = "number" ] >separator</npsobj>
```

13.17.2 Aufgabe

Dieses Element wird durch eine Folge von Links ersetzt. Besteht die Folge aus mehr als einem Link, wird der *separator* ausgewertet und zwischen den Links eingefügt. Ist *separator* leer, wird der Schrägstrich eingefügt.

Um die Folge von Links zu erzeugen, wird zunächst eine Folge von Dateien berechnet. Diese Folge besteht aus dem über der aktuellen Datei liegenden Ordner als letztem Element, dem Ordner, der über diesem Ordner liegt, als vorletztem Element usw., bis der Basisordner erreicht ist. Ist

`levels="number"` angegeben, werden nur *number* Elemente dieser Folge verwendet. Ist *number* größer null, werden die Elemente von vorne abgezählt, andernfalls von hinten.

Für jede dieser Dateien wird ein `a`-Element erzeugt. Das `href`-Attribut des `a`-Tags wird mit einem Link auf die jeweilige Datei belegt. Der Inhalt des `a`-Elements ist das Ergebnis einer Wertbestimmung für *fieldname* im Kontext der jeweiligen Datei.

13.17.3 Beispiel

```
<npsobj micronavigation="title" levels="-3"> / </npsobj>
```

13.18 npsobj modifyvar append

13.18.1 Syntax

```
npsobj_modifyvar_append ::=
<npsobj modifyvar = "append"
  varname = "name"
  name = "varValueKey"
  [ local = "( true | false )" ] />
```

13.18.2 Aufgabe

Dieses Element prüft zunächst, ob die [Export-Variable](#) *name* existiert und eine Kontextliste ist. Existiert die Variable und ist sie keine Kontextliste, wird ein Fehler erzeugt. Existiert sie nicht, so wird ihr eine leere Kontextliste zugewiesen. Anschließend wird an die Export-Variable *name* die Kontextliste angehängt, die sich aus der Evaluierung von *varValueKey* ergibt.

Ist das ab Version 6.7.1 verfügbare Attribut `local` mit dem Wert `true` angegeben und existiert die Exportvariable noch nicht, so wird sie angelegt und ihr Geltungsbereich auf das aktuelle Layout beschränkt, d.h. sie wird zu einer lokalen Variablen. Dadurch ist sie weder im aufrufenden Layout noch in Sublayouts sichtbar. Verwenden Sie eindeutige Namen für Variablen oder eine [modifyvar-set](#)-Anweisung mit lokalem Geltungsbereich vor `modifyvar append`, um sicherzustellen, dass die verwendete Variable lokal ist. Bitte beachten Sie, dass ab Version 6.7.1 Variablen als Argumente an Layouts übergeben werden können, die mit [insertvalue-template](#)-Anweisungen inkludiert werden.

An die Export-Variable kann eine Linkkontext- oder Dateikontext-Liste angehängt werden, unabhängig von den Kontexten, die bereits in der Export-Variablen gespeichert sind. Wenn Ihre Liste potenziell sowohl Link- als auch Dateikontexte enthält, sollte Ihr NPSOBJ-Code eine entsprechende Fallunterscheidung machen, indem beispielsweise der Wert eines nur bei Dateien vorhandenen und nie leeren Feldes (wie `objClass`) geprüft wird. Ist der Wert von `objClass` leer, handelt es sich um einen Linkkontext.

13.18.3 Beispiel

```
<npsobj modifyvar="append" varname="mixedList" name="freelinks2" />
<npsobj list="mixedList">
```

```

<npsobj condition="isEmpty" name="objClass">
  <!-- Muss Linkkontext sein -->
</npsobj>
<npsobj condition="isNotEmpty" name="objClass">
  <!-- Muss Dateikontext sein -->
</npsobj>
</npsobj>

```

13.19 npsobj modifyvar clear

13.19.1 Syntax

```

npsobj_modifyvar_clear ::=
  <npsobj modifyvar = "clear" varname = "name" />

```

13.19.2 Aufgabe

Dieses Element löscht die [Export-Variable](#) mit dem Namen *name*. Es wird nicht nur der Wert der Variablen gelöscht, sondern die Variable als solche wird entfernt und ist daher anschließend nicht mehr definiert.

13.19.3 Beispiel

```

<npsobj modifyvar="clear" varname="string25" />

```

13.20 npsobj modifyvar range

13.20.1 Syntax

```

npsobj_modifyvar_range ::=
  <npsobj modifyvar = "range" varname = "name"
    [ start = "startIndex" ]
    [ ( length = "length" ) | ( end = "end" ) ] />

```

13.20.2 Aufgabe

Dieses Element kürzt die Kontextliste, die in der Export-Variablen mit dem Namen *name* gespeichert ist. Die Attribute *start*, *length* und *end* haben die gleiche Aufgabe wie bei der [NPSOBJ-list-Anweisung](#).

13.20.3 Beispiel

```

<npsobj modifyvar="range" varname="mylist" start="2" end="4" />

```

13.21 npsobj modifyvar set

13.21.1 Syntax

```
npsobj_modifyvar_set ::=
( <npsobj modifyvar = "set"
  varname = "name"
  name = "varValueKey"
  [ local = "( true | false )" ] /> ) |
( <npsobj modifyvar = "set"
  varname = "name"
  [ local = "( true | false )" ]>varValue</npsobj> )
```

13.21.2 Aufgabe

Export-Variablen können dazu verwendet werden, während des Exports einer Datei Zeichenketten oder Kontextlisten unter einem Namen zu speichern. Auf die Werte kann man zugreifen, indem man die Namen so referenziert wie andere während des Exports gültige Namen auch.

Wird das ab Version 6.7.1 verfügbare Attribut `local` mit dem Wert `true` angegeben, so sind die mit dieser Anweisung erzeugten Exportvariablen nur in genau der Layoutdatei sichtbar, in der sie erzeugt wurden. Ihr Geltungsbereich ist in diesem Fall also auf das aktuelle Layout beschränkt, d.h. die Variablen sind weder im aufrufenden Layout noch in Sublayouts verfügbar. Bitte beachten Sie, dass ab Version 6.7.1 Variablen auch als Argumente an Layouts übergeben werden können, die mit insertvalue-template-Anweisungen inkludiert werden.

Bis Version 6.7.0, oder wenn `local` mit dem Wert `true` nicht angegeben wurde, sind die Exportvariablen von dem Moment ihrer Erzeugung bis zum Abschluss des Exports der jeweiligen Datei global verfügbar (sofern die Variable nicht zwischenzeitlich mit einer modifyvar-clear-Anweisung gelöscht wird). Der Wert einer gesetzten Variablen kann also auch bei einem Kontextwechsel, beispielsweise innerhalb einer NPSOBJ-list- oder NPSOBJ-context-Anweisung, abgerufen werden.

Die Namen von Export-Variablen haben Vorrang vor anderen Namen (beispielsweise den Namen von Feldern). Legt man also beispielsweise eine Export-Variable mit dem Namen `author` an und weist ihr mit

```
<npsobj modifyvar="set" varname="author">Max Mustermann</npsobj>
```

die Zeichenkette `Max Mustermann` zu, so liefert `<npsobj insertvalue="var" name="author" />` stets `Max Mustermann`, und zwar auch dann, wenn die aktuell exportierte Datei ein Feld `author` mit einem anderen Wert hat. Es ist jedoch möglich, mit dem Präfix `self` explizit den Wert eines Feldes zu referenzieren. Analog dazu kann man mit dem Präfix `export` ausdrücklich den Wert einer Export-Variablen inkludieren.

Die `modifyvar-set`-Anweisung gibt es in den beiden oben aufgeführten Varianten.

Variante 1: Wertzuweisung mittels Tag-Attribut

Bei der ersten Variante wird der Variablen der Wert des Feldes oder der Exportvariablen namens `varValueKey` zugewiesen. `varValueKey` kann (wie beispielsweise `parent.title`) eine Zeichenkette oder eine Liste von Link- oder Dateikontexten ergeben. Die Namen, die Datei- oder Linkkontexte ergeben, sind in den Abschnitten [Verfügbare Namen in einem Kontext vom Typ Datei](#) und [Verfügbare Namen in einem Kontext vom Typ Link](#) aufgeführt. Die folgende Anweisung setzt die Variable `siblings` auf die Liste der Geschwisterdateien der aktuellen Datei (und schließt sich selbst dabei ein):

```
<npsobj modifyvar="set" varname="siblings" name="parent.children" />
```

An Export-Variablen, die Link- oder Dateikontexte enthalten, können mit der `modifyvar-append`-Anweisung weitere Kontextlisten angehängt werden (siehe [npsobj_modifyvar_append](#)). Ferner können die Kontexte in Export-Variablen sortiert werden (siehe [npsobj_modifyvar_sort](#)).

Variante 2: Wertzuweisung über den Inhalt des Elements

Mit der zweiten Variante der `modifyvar-set`-Anweisung kann einer Export-Variablen eine Zeichenkette zugewiesen werden. NPSOBJ-Elemente in dieser Zeichenkette werden ausgewertet. Dadurch lassen sich Layouts, in denen dasselbe Konstrukt mehrfach benötigt wird, effizienter schreiben.

Beachten Sie bitte, dass die `modifyvar-set`-Anweisung bei dieser Verwendungsweise einen neuen Kontext erzeugt. Dieser Kontext entspricht dem darüberliegenden Kontext, so dass normalerweise keine zusätzlichen Maßnahmen erforderlich sind. Möchten Sie jedoch einen Kontext oberhalb des aktuellen Kontexts adressieren, so benötigen Sie einen zusätzlichen `context`-Präfix wie im folgenden Beispiel:

```
<npsobj list="relatedLinks">
  <npsobj modifyvar="set" varname="theLinkListCode"><p>
    <npsobj list="context.context.linkImage">
      <npsobj insertvalue="image" name="destination" />
    </npsobj>
    <npsobj insertvalue="anchor" name="self">
      <npsobj insertvalue="var" name="destination.title"/>
    </npsobj></p>
  </npsobj>
</npsobj>
```

Um die CMS-Datei zu ermitteln, die einen Link enthält, kann auch die Linkeigenschaft `source` abgefragt werden.

13.21.3 Beispiel

Der folgende Code speichert einen Link zum Ordner, der über der aktuellen Datei liegt, in einer Variablen und gibt ihn anschließend aus.

```
<npsobj modifyvar="set" varname="uplink">
  <npsobj insertvalue="anchor" name="parent">
    <npsobj insertvalue="var" name="parent.title" />
  </npsobj>
</npsobj>
<npsobj insertvalue="var" name="uplink" />
```

13.22 npsobj modifyvar sort

13.22.1 Syntax

```
npsobj_modifyvar_sort ::=
<npsobj list = "name"
  sortkey1 = "sortKey1"
  [ sortmodifier1 = sortModifier ]
  [ sortkey2 = "sortKey2"
    [ sortmodifier2 = sortModifier ]
    [ sortkey3 = "sortKey3"
      [ sortmodifier3 = sortModifier ]]] />

sortModifier ::=
"alpha" |
"numeric" |
"ascending" |
"descending" |
"alpha ascending" |
"alpha descending" |
"numeric ascending" |
"numeric descending"
```

13.22.2 Aufgabe

Dieses Element sortiert den Wert der Export-Variablen `name`, die eine Kontextliste enthält. Die Bedeutung der Felder `sortkey1`, `sortkey2`, `sortkey3`, `sortmodifier1`, `sortmodifier2`, `sortmodifier3` sowie des Wertes von `sortModifier` ist bei der [NPSOBJ-list-Anweisung](#) beschrieben.

13.22.3 Beispiel

```
<npsobj modifyvar="set" varname="siblings" name="parent.children" />
<npsobj modifyvar="sort" varname="siblings" sortkey1="name" sortkey2="id"
sortmodifier2="numeric" />
```

13.23 npsobj newslist all

13.23.1 Syntax

```
npsobj_newslist_all ::=
<npsobj newslist = "all" [ ( length = "length" ) ] >
  content
</npsobj>
```

13.23.2 Aufgabe

Die Anweisung wird verwendet, um Newslisten zu erzeugen. `all` zeigt an, dass alle News, unabhängig von den Channels, denen sie zugeordnet sind, berücksichtigt werden. Die Anweisung erzeugt eine

Liste von maximal *length* Dateikontexten und ersetzt das Element durch eine Folge von Texten. Die Texte ergeben sich aus der Auswertung von *content* für jedes Element der Kontextliste.

13.23.3 Beispiel

```
<npsobj newslst="all" length="20">
  Wird für jede News evaluiert
</npsobj>
```

13.24 npsobj newslst selected

13.24.1 Syntax

```
npsobj_newslst_selected ::=
  <npsobj newslst = "selected"
    ( name = "channel_attribute" | value = string )
    [ ( length = "length" ) ] >
    content
  </npsobj>

string = "string"
```

13.24.2 Aufgabe

Die Anweisung wird verwendet, um Newslisten zu erzeugen. *selected* zeigt an, dass nur News der angegebenen Channels berücksichtigt werden. Die Channels werden entweder indirekt über das Feld *channel_attribute* oder direkt als String angegeben. *channel_attribute* kann vom Typ *Zeichenkette*, *Text*, *Auswahl* oder *Mehrfachauswahl* sein. Direkt oder indirekt als Zeichenkette angegebene Channels müssen mit einem Komma voneinander getrennt sein. Die Anweisung erzeugt eine Liste von maximal *length* Dateikontexten und ersetzt das Element durch eine Folge von Texten. Die Texte ergeben sich aus der Auswertung von *content* für jedes Element der Kontextliste.

13.24.3 Beispiel

```
<npsobj newslst="selected" name="channels" length="20">
  Wird für jede News evaluiert
</npsobj>
```

13.25 npsobj switch

13.25.1 Syntax

```
npsobj_switch ::=
  <npsobj switch = "switchName">
    { <npsobj casecond = "caseCondition"
```



```
[ negate = "" ]
[ proceed = ( "yes" | "no" ) ]
( name = "caseName" | value = "caseValue" ) >
  content
</npsobj> }
[ <npsobj casecond = "default" [ proceed = ( "yes" | "no" ) ] >
  content
</npsobj> ]
</npsobj>
```

13.25.2 Aufgabe

Mit diesem Element lässt sich ein Wert (*switchName*) des aktuellen Kontexts mit mehreren anderen Werten (*caseName*) oder Konstanten (*caseValue*) vergleichen. Für jeden Vergleich kann einer der unter [npsobj_condition](#) aufgeführten Vergleichsoperatoren (*caseCondition*) verwendet werden. Wird ein unärer Vergleichsoperator (wie *isEmpty*) auf *switchName* angewendet, so werden *caseName* und *caseValue* ignoriert, andernfalls wird *switchName* mit *caseName* oder *caseValue* verglichen, je nachdem, ob *name* oder *value* angegeben wurde. Trifft ein Vergleich zu, wird der *content* des betreffenden *caseCond*-Elements ausgewertet.

switchName und *caseName* können alle im aktuellen Kontext verfügbaren Namen sein, während mit *caseValue* eine konstante Zeichenkette angegeben werden kann. Mit dem optionalen Attribut *negate* kann das Ergebnis der Vergleichsoperation umgekehrt werden. Wird als *caseCondition* kein Vergleichsoperator, sondern *default* angegeben, so trifft der betreffende Vergleich immer zu. Typischerweise ist *default* in einer *switch*-Anweisung der letzte Vergleich, um den Fall zu behandeln, dass keiner der vorausgegangenen Vergleiche zutrifft.

Mit dem optionalen Attribut *proceed* kann man steuern, ob die nächste *casecond*-Anweisung im *switch*-Block auch dann abgearbeitet werden soll, wenn der Vergleich zutrifft (Voreinstellung: *no*).

Innerhalb eines *switch*-Elements werden nur *casecond*-Elemente ausgewertet. Text sowie weitere Elemente auf der gleichen Ebene werden ignoriert.

13.25.3 Beispiel

```
<npsobj switch="name">
  <npsobj casecond="isEqual" value="news1">
    Newsordner 1
  </npsobj>
  <npsobj casecond="isEqual" value="news2">
    Newsordner 2
  </npsobj>
  <npsobj casecond="default">
    Keine Newsordner
  </npsobj>
</npsobj>
```

13.26 npsobj table

13.26.1 Syntax

```
npsobj_table ::=
```

```

<npsobj table = "name"
[ sortedby = "sortkey" ]
[ sortkey1 = "sortKey1"
  [ sortmodifier1 = sortModifier ]
  [ sortkey2 = "sortKey2"
    [ sortmodifier2 = sortModifier ]
    [ sortkey3 = "sortKey3"
      [ sortmodifier3 = sortModifier ]]]]
[ start = "startIndex" ]
[ ( length = "length" ) | ( end = "end" ) ]
[ reverse = "" ]
[ columns = "columns" ]
[ direction = ( "horizontal" | "vertical" ) ]
{ other_attribute } >
content
</npsobj>

sortModifier ::=
  "alpha" |
  "numeric" |
  "ascending" |
  "descending" |
  "alpha ascending" |
  "alpha descending" |
  "numeric ascending" |
  "numeric descending"

other_attribute ::= string = "string"

```

13.26.2 Aufgabe

Die Anweisung erzeugt eine Tabelle und füllt die einzelnen Zellen mit Texten. Die einzelnen Texte ergeben sich aus der Auswertung der Werte *name*, *sortkey*, *startIndex*, *length* und *end* sowie *sortKey1*, *sortKey2*, *sortKey3*, *sortModifier1*, *sortModifier2*, *sortModifier3* und *reverse* wie im Abschnitt [npsobj_list](#) beschrieben. Enthält ein Text nur Leerzeichen (also Leerzeichen, Tabulator oder Zeilenumbrüche), so wird die Zelle mit dem Text ` ` gefüllt.

Die Tabelle enthält *columns* Spalten. Ist *columns* nicht angegeben, wird eine einzige Spalte erzeugt. Das erzeugte *table*-Tag erhält als Attribute alle als *other_attribute* angegebenen Attribute des NPSOBJ-Tags. Für alle Zeilen wird ein *tr*-Element mit einem öffnenden und einem schließenden Tag erzeugt. Für jede Zelle wird ein *td*-Element mit einem öffnenden und einem schließenden Tag erzeugt. Die erzeugten öffnenden *tr*- und *td*-Tags haben keine Attribute.

Die Zellen der Tabelle werden reihenweise von links nach rechts, beginnend in der obersten Zeile, mit den berechneten Texten gefüllt, wenn *direction*="horizontal" angegeben ist. Andernfalls werden die Zellen spaltenweise von oben nach unten, beginnend mit der ersten Spalte, gefüllt. Es wird eine Zahl von Zeilen bzw. Spalten erzeugt, die gerade ausreicht, um für alle Texte eine eigene Zelle erzeugen zu können. Zellen am Ende der letzten Zeile bzw. Spalte, für die kein Text mehr vorhanden ist, werden mit dem Text ` ` gefüllt.

13.26.3 Beispiel

```

<npsobj table="children" columns="2" direction="vertical" border="1">
  <npsobj insertvalue="var" name="title"/>
</npsobj>

```

14

14 Die Bearbeitungsmöglichkeiten in der Vorschau erweitern

Die [Vorschau](#) gibt Redakteuren die Möglichkeit, Feldinhalte und Links direkt zu bearbeiten, d. h. ohne die Vorschau zu verlassen. Hierfür werden in der Vorschau neben Links und Feldern Bearbeitungselemente (so genannte Marker) angezeigt. Diese Marker bindet das Redaktionssystem automatisch in die Vorschau ein, d. h. ohne dass ein Redakteur hierfür eine Option zu aktivieren braucht. Ab Version 6.5.0 lassen sich die Marker jedoch mit Hilfe eines `npsgui`-Elements, das Sie in Vorlagen einbetten können, explizit ein- und ausschalten.

Zusätzlich zu den automatischen Markern können Sie Redakteuren – ebenfalls ab Version 6.5.0 mit Hilfe von `npsgui`-Elementen – weitere Arbeitshilfen zur Verfügung stellen.

Das `npsgui`-Element hat den folgenden Funktionsumfang:

- Sie können [Teile des Layouts in der Vorschau anzeigen lassen](#), um Redakteuren Bearbeitungshinweise oder ähnliche Informationen zu geben. Diese Abschnitte im Layout können beliebigen HTML-Text enthalten. Sie werden nicht exportiert.
- Sie können Redakteure per Klick auf einen Marker oder verlinkten Text [Aktionen ausführen](#) lassen. Als Aktionen können Menübefehle, Assistenten oder die Bearbeitung eines Feldes definiert werden. Beispielsweise können Sie Redakteuren in der Vorschau eines Newsartikels anbieten, den Artikel mit einem Assistenten zu bearbeiten oder einen neuen Artikel anzulegen.
- Ferner können Sie, wie oben bereits angedeutet, die [automatische Erzeugung von Markern ein- oder ausschalten](#), so dass die Vorschau das Aussehen der entsprechenden Live-Seite hat (sofern Sie keine eigenen Marker oder Bearbeitungshinweise anzeigen lassen).

`npsgui`-Elemente werden analog zu `NPSOBJ`-Elementen verwendet.

14.1 Bearbeitungshinweise in der Vorschau anzeigen

In der Vorschau können Bearbeitungshinweise angezeigt werden, indem Sie mit einem `npsgui`-Element einen Bereich definieren und in diesen Bereich HTML-Text einfügen. Der Bereich wird beim Export ignoriert. Beispiel:

```
<npsgui>
  Dieser Bereich ist nur in der redaktionellen Vorschau sichtbar.
</npsgui>
```

Kommentare verwenden

Das `npsgui`-Element kann auch dazu verwendet werden, Kommentare in Layouts aufzunehmen. Die Kommentare werden im Content Navigator nicht angezeigt und sind in den veröffentlichten Seiten nicht vorhanden. Beispiel:

```
<npsgui><!-- Dieses Layout berechnet die sekundäre (linke) Navigation --></npsgui>
```

14.2 Aktionen anbieten

Sie können Redakteuren in der integrierten Vorschau [Menübefehle oder registrierte Assistenten](#) zur Ausführung oder die Bearbeitung einzelner Felder anbieten.

Eine Aktion erscheint in der Vorschau als Link, dessen Aussehen Sie beeinflussen können, indem Sie ein Icon oder HTML-Text angeben. Haben Sie weder ein Icon noch HTML-Code angegeben, wird das konfigurierte Icon bzw. der lokalisierte Titel der Aktion verwendet. Der folgende Screenshot zeigt, wie mit dieser Funktionalität Menüeinträge von einem Redakteur umsortiert werden können:



Die Aktionen, die ausgelöst werden, wirken auf die Datei, in deren [Kontext](#) sie definiert sind. Dadurch ist es zum Beispiel möglich, in einer generierten Inhaltsübersicht die aufgelisteten Dateien zu bearbeiten.

Bearbeitungselemente (so genannte Marker) werden auch für die ab Version 6.5.0 verfügbaren [Spiegeldateien](#) angeboten. Diese wirken sich auf die jeweiligen Originaldateien aus.

14.2.1 Syntax

```
<npsgui insertMarker="Typ" name="Name" icon="Icon-Datei">HTML-Code</npsgui>
```

Parameter

- *Typ*: Gibt den Typ des einzufügenden Links an:
 - *item*: Mit dem eingefügten Link kann ein Menübefehl oder ein Assistent ausgeführt werden.
 - *attribute*: Mit dem eingefügten Link kann ein Feld bearbeitet werden.
- *Name*: Ist *Typ* gleich *item*, so ist dies die [ID einer registrierten Aktion](#). Wenn *Typ* gleich *attribute* ist, gibt *Name* den Namen des Feldes an, das zur Bearbeitung angeboten werden soll.
- *Icon-Datei*: Ist der Dateiname eines Icons im Schemaverzeichnis der GUI-Webapplikation. Die Größe des Icons sollte 31×31 Pixel betragen. Dieser Parameter ist optional.

- *HTML-Code*: Der HTML-Code, der verlinkt werden soll, damit die Aktion ausgelöst werden kann. Der Text hat gegenüber einem eventuell ebenfalls angegebenem Icon Vorrang. Auch *HTML-Code* ist optional.

Beispiel

Das im obigen Screenshot gezeigte Bearbeitungselement, mit dem ein Menüeintrag nach oben verschoben werden kann, wird durch den folgenden Code erzeugt (beachten Sie bitte die ab Version 6.7.1 geänderte Syntax der @-Referenzen):

```
<npsgui insertMarker="item" name="changeSortOrder" objId1="@{_previous}"
objId2="@{self}">
  
</npsgui>
```

14.2.2 Assistenten parametrisieren

Wenn Sie einen Link, mit dem ein Assistent ausgeführt werden kann, in die integrierte Vorschau einbinden, können Sie dem [Assistenten](#) Parameter übergeben. Um beispielsweise den Assistenten zur [Bearbeitung eines bestimmten Feldes](#) einzubinden, geben sie das zusätzliche Tag-Attribut `attributeName` an:

```
<npsgui insertMarker="item" name="attributeEditWizard" attributeName="Name">Name
bearbeiten</npsgui>
```

Alle Tag-Attribute bis auf `insertMarker`, `name`, `icon` und `context` werden zu den Parametern des Assistenten hinzugefügt. Den Parameternamen wird `wizard.` vorangestellt, so dass der Assistent beispielsweise auf `attributeName` unter `wizard.attributeName` zugreifen kann.

14.2.3 Bearbeitungselemente dynamisch positionieren

Um störende Auswirkungen der angezeigten Marker auf das Layout zu minimieren, gibt es ab Version 6.7.1 die Möglichkeit, Marker vom Typ `attribute` dynamisch zu positionieren. Das GUI stellt die Marker dann nicht mehr an Ort und Stelle dar, sondern platziert an den betreffenden Stellen nur noch `span`-Elemente, die als Anker fungieren. Die Bearbeitungselemente selbst werden in unsichtbaren Containern platziert und bei der Anzeige mit Hilfe von JavaScript positioniert und sichtbar gemacht. Dabei wird das jeweilige Bearbeitungselement über den zu bearbeitenden Inhalt gelegt:

Diese Form der Positionierung funktioniert nicht mit Inhalten, die mit Hilfe von JavaScript ein- oder ausgeblendet werden (etwa JavaScript-Menüs).

Um Marker dynamisch zu positionieren, geben Sie das Attribut `placeWithJS` mit dem Wert `true` an:

```
<npsgui insertMarker="attribute" placeWithJS="true" ... />
```

Bitte beachten Sie, dass dynamisch positionierte Marker nicht aus individuellem HTML-Code bestehen können. Auch kann das dargestellte Icon nur über das [Darstellungsschema](#) und nicht über das Attribut `icon` geändert werden.

Um die dynamische Positionierung zu verbessern oder Effekte (wie Hervorhebung) auf den zu bearbeitenden Inhalt anwenden zu können, kann das `npsgui`-Element den zu bearbeitenden Inhalt umschließen. Das GUI wird das Anker-`span`-Element dann ebenfalls um den Inhalt positionieren:

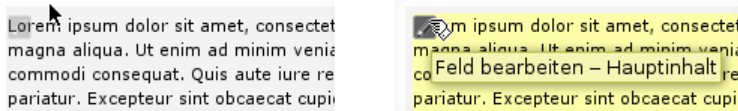
```
<npsgui ... placeWithJS="true"><npsobj insertvalue="var" name="title" /></npsgui>
```

Mit dem Attribut `containerTag` können Sie ein alternatives Ankerelement festlegen, wenn `span` nicht zulässig sein sollte (beispielsweise weil der umschlossene HTML-Code Blockelemente enthält):

```
<npsgui ... placeWithJS="true" containerTag="div"><p>...</p></npsgui>
```

Marker-Darstellung anpassen

Die Darstellung dynamisch positionierter Marker lässt sich mit Hilfe von JavaScript beeinflussen. Das folgende Beispiel zeigt, wie Marker durchscheinend dargestellt werden und der zu bearbeitende Inhalt hervorgehoben wird, wenn die Maus über dem Marker schwebt.



Für diese Effekte können entweder die JavaScript-Bibliotheken [Prototype](#) und [Scriptaculous](#) oder [jQuery](#) verwendet werden. Diese müssen jeweils als Inhalte im CMS vorliegen und im `head`-Bereich der HTML-Seiten eingebunden werden. Der folgende Auszug aus einer Layout-Datei stellt den erforderlichen Code in diesem Bereich zur Verfügung.

```
<!-- Variante mit Prototype und Scriptaculous -->

<npsgui>
  <script type="text/javascript" language="JavaScript"
    src="/javascripts/prototype.js"></script>
  <script type="text/javascript" language="JavaScript"
    src="/javascripts/scriptaculous.js?load=effects"></script>

  <script type="text/javascript">

    /*****
    /* Hilfsfunktionen für die Hervorhebung */

    function rememberHighlightingStyle(myElement)
    {
      myElement.rememberBackgroundColor = myElement.getStyle('backgroundColor');
      myElement.rememberZoom = myElement.getStyle('zoom');
      myElement.rememberColor = myElement.getStyle('color');
    }

    function highlight(myElement)
    {
      // setting the zoom works around incomplete redraws on ie6
      myElement.setStyle({
        backgroundColor: '#ffffaa',
        zoom: 1,
        color: '#000000'
      });
    }

    function startHighlighting(myElement)
    {
      rememberHighlightingStyle(myElement);
      highlight(myElement);
    }
  </script>
</npsgui>
```

```

function stopHighlighting(myElement)
{
    myElement.setStyle({
        backgroundColor: myElement.rememberBackgroundColor,
        zoom: myElement.rememberZoom,
        color: myElement.rememberColor
    });
}

/*****
/* Funktion zum Positionieren und Anpassen der Marker */

function placeEditMarker(marker, anchor, offsetLeft, offsetTop)
{
    marker.onmouseover = function() {
        new Effect.Opacity(this, { from: 0.3, to: 1.0, duration: 0.3 });
        anchor.descendants().each(function(e) { rememberHighlightingStyle(e) });
        anchor.descendants().each(function(e) { highlight(e) });
        return startHighlighting(anchor);
    };
    marker.onmouseout = function() {
        new Effect.Opacity(this, { from: 1.0, to: 0.3, duration: 0.3 });
        anchor.descendants().each(function(e) { stopHighlighting(e) });
        return stopHighlighting(anchor);
    };
    new Effect.Opacity(marker, { from: 1.0, to: 0.3, duration: 0 });
    nps.editMarker.placeEditMarker(marker, anchor, offsetLeft, offsetTop);
}
</script>
</npsgui>

<!-- Variante mit jQuery -->

<npsgui>
<script type="text/javascript" language="JavaScript"
    src="/javascripts/jquery.js"></script>

<script type="text/javascript">

/*****
/* Hilfsfunktionen für die Hervorhebung */

function rememberHighlightingStyle(myElement)
{
    myElement.rememberBackgroundColor = $(myElement).css('backgroundColor');
    myElement.rememberZoom = $(myElement).css('zoom');
    myElement.rememberColor = $(myElement).css('color');
}

function highlight(myElement)
{
    // setting the zoom works around incomplete redraws on ie6
    $(myElement).css({backgroundColor: '#ffffaa', zoom: 1, color: '#000000'});
}

function startHighlighting(myElement)
{
    rememberHighlightingStyle(myElement);
    highlight(myElement);
}

function stopHighlighting(myElement){
    $(myElement).css({
        backgroundColor: myElement.rememberBackgroundColor,
        zoom: myElement.rememberZoom,
        color: myElement.rememberColor
    });
}

/*****

```

```

/* Funktion zum Positionieren und Anpassen der Marker */

function placeEditMarker(marker, anchor, offsetLeft, offsetTop)
{
    marker.onmouseover = function() {
        $(this).fadeTo("slow", 1);
        $(anchor).find(":visible").each(function(i) {
            rememberHighlightingStyle(this)
        });
        $(anchor).find(":visible").each(function(i) {
            highlight(this)
        });
        return startHighlighting(anchor);
    };
    marker.onmouseout = function() {
        $(this).fadeTo("slow", 0.3);
        $(anchor).find(":visible").each(function(i) {
            stopHighlighting(this);
        });
        return stopHighlighting(anchor);
    };
    $(marker).fadeTo(1, 0.3);
    nps.editMarker.placeEditMarker(marker, anchor, offsetLeft, offsetTop);
}
</script>
</npsgui>

```

Falls die JavaScript-Funktion `placeEditMarker` definiert ist, ruft das GUI sie auf, um die Marker zu positionieren. Andernfalls ruft das GUI zu diesem Zweck die interne JavaScript-Funktion `nps.editMarker.placeEditMarker` auf.

Die Parameter `marker` und `anchor` sind die HTML-Elemente des Markers bzw. dessen Anker. Die Parameter `offsetLeft` und `offsetTop` enthalten Werte, mit denen die errechnete Position korrigiert werden kann. Sie können als Attribute im `npsgui`-Element angegeben werden:

```

<!-- Marker 10px nach unten und 20px nach links verschieben -->
<npsgui ... placeWithJS=" true" offsetTop="10" offsetLeft="-20">...</npsgui>

```

Positive Werte für `offsetTop` verschieben die Position nach unten und negative nach oben. Analog dazu verschieben positive Werte für `offsetLeft` die Position nach rechts und negative nach links.

14.3 Anzeige automatisch erzeugter Marker steuern

Sie können die automatische Anzeige von Markern in der Vorschau gezielt aus- oder einschalten, um beispielsweise das Aussehen der Seiten zu erhalten oder nur Ihre eigenen Aktionen anzubieten.

Die automatische Anzeige von Markern kann in Bereichen Ihrer Website individuell gesteuert werden. Auch können Sie innerhalb eines Website-Bereichs mit ausgeschalteter Anzeige diese wieder aktivieren oder umgekehrt. Solche Bereiche lassen sich folgendermaßen definieren:

```

<npsgui generateAutomaticEditMarkers="Modus">...</npsgui>

```

Modus kann einen der folgenden Werte annehmen:

- **all**: Alle automatisch generierten Marker werden angezeigt.
- **brokenLinks**: Nur Marker, mit denen nicht aufgelöste Links bearbeitet werden können, werden angezeigt.
- **none**: Es werden gar keine automatisch generierten Marker angezeigt.

Die Bearbeitungsmöglichkeiten in der Vorschau erweitern