



Infopark CMS Fiona



Fiona 6.7.1 Release Notes

Infopark CMS Fiona

Fiona 6.7.1 Release Notes

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 Neue und geänderte Funktionen	6
1.1 Verbesserte Benutzerfreundlichkeit	6
1.1.1 Überarbeitetes Design der Assistenten-Auswahlseite	6
1.1.2 TinyMCE HTML-Editor leichter in Assistenten einzubinden	6
1.1.3 Weitere vorbelegbare Felder	6
1.1.4 Bilder als Ressourcen ablegen	7
1.1.5 Markdown-Editor zeigt den Feld-Titel an	7
1.1.6 Detailübersicht mit Registerkarten	7
1.2 Erweiterte Template-Sprache	7
1.2.1 Verbesserte Bearbeitungselemente in der Vorschau	7
1.2.2 Bessere @-Referenzen	7
1.2.3 Lokale Exportvariablen	8
1.2.4 Parametrierbare Layouts	8
1.3 Handhabung der Assistenten	9
1.3.1 Beschreibung und Sortiermöglichkeit für Assistenten	9
1.3.2 Ein einzelnes Auswahlkästchen erzeugen	9
1.3.3 Stile für Texteingabefelder	9
1.3.4 Assistenten werden zentral eingelesen	9
1.3.5 Formularvalidierung bei Assistenten-Buttons abschaltbar	10
1.4 Systemfunktionalität	10
1.4.1 Erhöhte Stabilität	10
1.4.2 Datenbank-Unterstützung	10
1.4.3 Neues Zertifikat für externen Editor	10
1.4.4 Vorschau berücksichtigt optional Leserecht der CMS Dateien	10
1.4.5 Strukturierte Link-Funktionen (Callbacks)	10
1.4.6 Einfache Integration von Benachrichtigungsfunktionen (Post-Action-Callbacks)	11
1.4.7 Erweiterte Tcl-Formatter	11
1.4.8 Interface von externen Präprozessoren erweitert	11
1.4.9 Erweiterte LDAP-Anbindung des Content Management Servers	11
1.4.10 Konfigurationsdateien der Benutzeroberfläche	11
1.5 Infopark Rails Connector	12
1.6 Software von Drittanbietern	13
1.7 Weitere Änderungen	13

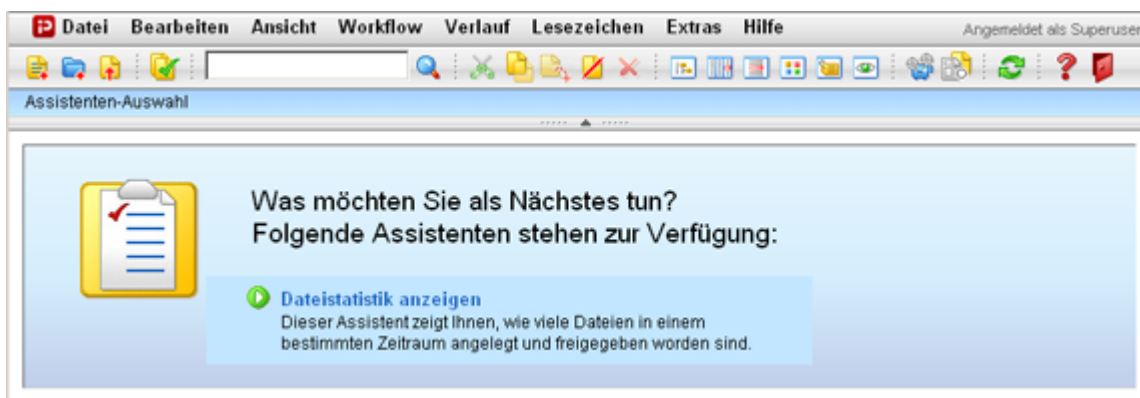
2 Behobene Fehler	14
2.1 CMS-Funktionalität	14
2.2 Systemfunktionalität	15
2.3 Rails Connector	16
2.4 Portal Manager	16
3 Wichtige Hinweise zum Upgrade	17
3.1 Java-API wurde auf Java Generics umgestellt	17
3.2 Speicherung der Portlet-Einstellungen von Benutzern	17
3.3 Rails-Tabellen des CMS aktualisieren	17
3.4 Abwicklung von Datenbank-Transaktionen im Portal Manager	17
3.5 Leserechte in der Vorschau	18
3.6 Geänderter Namensraum der Standard-Formatter	18
3.7 Notwendige Datenbankoption bei der Verwendung von Sybase	18
3.8 Offizielle Unterstützung für Microsoft SQL Server 2008	18
4 Bekannte Fehler	19

1 Neue und geänderte Funktionen

1.1 Verbesserte Benutzerfreundlichkeit

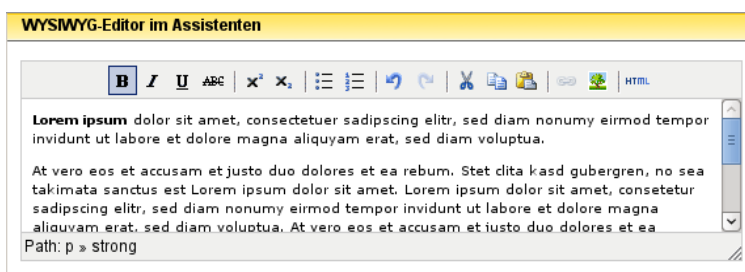
1.1.1 Überarbeitetes Design der Assistenten-Auswahlseite

Das Design des Startbereichs *Assistenten-Auswahl* in der Redaktionsoberfläche wurde überarbeitet:



1.1.2 TinyMCE HTML-Editor leichter in Assistenten einzubinden

Der in Assistenten nutzbare JavaScript-WYSIWYG-Editor TinyMCE kann jetzt einfacher eingebunden werden.



1.1.3 Weitere vorbelegbare Felder

Das Dateifeld *Bei Export ignorieren* (`suppressExport`) kann in Dateivorlagen jetzt vorbelegt werden.

Die Sortierkriterien (`sortKey1`, `sortKey2`, `sortkey3`) sowie der jeweilige Sortiermodus (`numeric`, `alphanumeric`) können nun in Dateivorlagen ebenfalls vorbelegt werden.

1.1.4 Bilder als Ressourcen ablegen

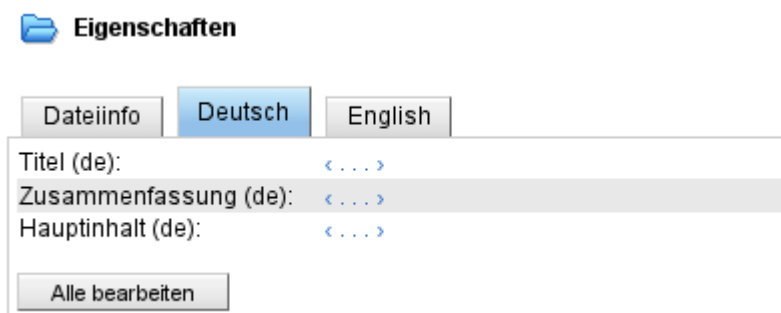
Bilder können wieder in Dateien vom Typ *Ressource* (*generic*) gespeichert werden, beispielsweise, um sie zum Download anzubieten. Um dies zuzulassen, setzen Sie bitte den Konfigurationseintrag `content.genericDocumentsCanContainImages` auf `true`.

1.1.5 Markdown-Editor zeigt den Feld-Titel an

Im Markdown-Editor wird nun der Titel des Feldes angezeigt, das gerade bearbeitet wird.

1.1.6 Detailübersicht mit Registerkarten

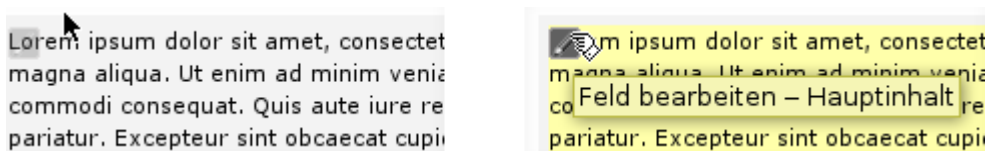
Die Detailübersicht kann nun auch mit Reitern versehen werden, um Feldgruppen platzsparender anzuordnen.



1.2 Erweiterte Template-Sprache

1.2.1 Verbesserte Bearbeitungselemente in der Vorschau

Bearbeitungselemente können in der Vorschau jetzt dynamisch positioniert und mit Effekten versehen werden. Solche Marker beeinflussen das angezeigte Layout nicht mehr, da Sie über den zu bearbeitenden Inhalten schweben.



1.2.2 Bessere @-Referenzen

@-Referenzen können nicht mehr nur am Anfang eines Attributwerts verwendet werden, sondern an beliebiger Stelle im Attributwert, auch mehrfach. Dadurch können Attributwerte auf einfache Art und Weise aus statischem und dynamisch berechnetem Text zusammengesetzt werden. Dies ist durch eine andere Syntax möglich. @-Referenzen werden nun nicht mehr nach dem Muster `@variable` gebildet, sondern haben das Format `@{variable}`.

Bisher gingen Layout-Entwickler folgendermaßen vor, um Tag-Attributen eine Kombination aus statischem und dynamischem Text zuzuweisen:

```
<!-- Bisher übliche Vorgehensweise -->
<npsobj modifyvar="set"
  varname="cssClass">box obj_<npsobj insertvalue="var" name="id"/></npsobj>
<div class="@cssClass">
```

Mit den neuen @-Referenzen kann derselbe Effekt viel einfacher erzielt werden:

```
<!-- Neue Vorgehensweise -->
<div class="box obj_{id}">
```

Bei den neuen @-Referenzen wird der Feld- oder Variablenname immer in geschweifte Klammern eingeschlossen. Die bisherige Schreibweise wird nicht mehr unterstützt.

Bei der Migration werden @-Referenzen, die sich in Layouts und HTML-Dokumenten befinden und das bisherige Format haben, automatisch in das neue Format umgewandelt.

1.2.3 Lokale Exportvariablen

In der NPSOBJ-Anweisung `modifyvar set` können Exportvariablen als lokal markiert werden. Beispiel:

```
<npsobj modifyvar="set" varname="foo" local="true">bar</npsobj>
```

Das Attribut `local` ist auch bei `modifyvar append` erlaubt. Bei dieser Anweisung wird eine Variable jedoch nur dann zu einer lokalen, wenn sie noch nicht existiert. Bestehende globale Variablen bleiben also global.

In Verbindung mit der unten beschriebenen Parametrierbarkeit von Layout-Aufrufen lassen sich Sublayouts durch lokale Variablen besser von den übergeordneten Layouts abgrenzen, wodurch eine klarere und leichter wartbare Layout-Struktur entsteht.

1.2.4 Parametrierbare Layouts

Es ist nun möglich, einem Layout, das über eine `insertvalue-template`-Anweisung aufgerufen (inkludiert) wird, einen statischen oder (ganz oder teilweise) berechneten Text zu übergeben. Hierzu wird der Text im Inhalt der `insertvalue-template`-Anweisung platziert. Im aufgerufenen Layout kann der Text über die Variable `npsobjContent` abgerufen werden. Ein Beispiel finden Sie in der Dokumentation zu der genannten Anweisung `insertvalue template`.

Diese neue Funktion ermöglicht es, Layouts in unterschiedlichen Kontexten zu inkludieren, ohne globale Variablen zur Übergabe von Argumenten verwenden zu müssen. Dadurch lässt sich die Komplexität von Layoutdateien reduzieren.

Ferner können reine Textdaten einem einzufügenden Layout jetzt auch als Argumente in Form von Tag-Attributen übergeben werden. Beispiel:

```
<npsobj insertvalue="template" name="TemplateName" showbody="1" person="@{author}" />
```

Die Argumente und auch die Variable `npsobjContent` werden wie lokale Exportvariablen behandelt; sie sind nur während der Auswertung des inkludierten Layouts gesetzt und nur in diesem sichtbar. Ein Beispiel und weitere Erläuterungen finden Sie in der Dokumentation zur `insertvalue-template`-Anweisung.

1.3 Handhabung der Assistenten

1.3.1 Beschreibung und Sortiermöglichkeit für Assistenten

Bei der Konfiguration von Assistenten kann jetzt das Property `descriptions` angegeben werden, um zusätzlich zum Namen des Assistenten auch eine Kurzbeschreibung verfügbar zu machen und anzeigen zu können. Beispiel:

```
<bean id="createStatistics"
  class="com.infopark.cm.htmlgui.browse.menuaction.CustomCommand">
  <property name="titles"><map>
    <entry key="de" value="Dateistatistik anzeigen"/>
    <entry key="en" value="Show File Statistics"/>
    <entry key="fr" value="Afficher les statistiques du fichier"/>
    <entry key="it" value="Mostrare le statistiche del file"/>
    <entry key="es" value="Mostrar estadísticos del fichero"/>
  </map></property>
  <property name="descriptions"><map>
    <entry key="de"><value>
      Wie viele Dateien wurden in einem bestimmten Zeitraum angelegt und freigegeben?
    </value></entry>
    <entry key="en"><value>
      How many files have been created and released in a given period of time?
    </value></entry>
  </map></property>
  <property name="selectionType" value="none"/>
  <property name="command" value="createStatistics::render"/>
  <property name="dialog" value="true"/>
</bean>
```

Ferner kann nun mit dem Property `sortKey` die Reihenfolge der Assistenten im Startbereich *Assistenten-Auswahl* definiert werden. Die Assistenten werden alphabetisch aufsteigend nach dem Wert von `sortKey` sortiert.

1.3.2 Ein einzelnes Auswahlkästchen erzeugen

In Assistenten kann die neue Assistenten-Funktion `checkBox` verwendet werden, um ein einzelnes Auswahlkästchen zu erzeugen.

1.3.3 Stile für Texteingabefelder

In Assistenten können sowohl die CSS-Klasse als auch das `style`-Attribut für Texteingabefelder (Datum, File, Text, Textarea) angegeben werden. Beispiel:

```
::layout::fileField myUpload "Dateifeld" class "myCSS" \
style "background-color: red"
```

1.3.4 Assistenten werden zentral eingelesen

Assistenten werden jetzt von einem zentralen Skript eingelesen: `share/script/cm/serverCmds/wizards.tcl`. Diese Tcl-Datei sorgt dafür, dass alle Assistenten in den Verzeichnissen `share/script/cm/serverCmds/wizards` und `instance/script/cm/serverCmds/wizards` geladen werden.

Die instanzenspezifischen Dateien `instance/script/cm/serverCmds/wizards.tcl` werden nicht mehr benötigt und sollten gelöscht werden, nachdem die darin enthaltenen kundenspezifischen Erweiterungen (die nicht dazu dienen, Assistenten zu laden) gegebenenfalls in eine andere Skriptdatei übernommen wurden. Existieren instanzenspezifische Versionen dieser Datei, so wird ein entsprechender Eintrag im CM-Serverprotokoll erzeugt.

1.3.5 Formularvalidierung bei Assistenten-Buttons abschaltbar

Voreingestellt validiert das GUI die Formularinhalte von Assistenten-Dialogen, wenn der Anwender auf einen Assistenten-Button klickt. Diese Validierung ist nun abschaltbar, damit beispielsweise Unterdialoge trotz fehlerhafter Eingaben sofort geöffnet werden.

1.4 Systemfunktionalität

1.4.1 Erhöhte Stabilität

Die Stabilität von Massenoperationen wurde erhöht. Dies betrifft Operationen wie die Datensicherung und -wiederherstellung (Dump und Restore), die vollständige Indizierung, große Migrationsschritte und die Erstellung der Datenbanktabellen für den Rails Connector. Solche Operationen verbrauchen nun weniger Speicher und nutzen Subprozesse.

1.4.2 Datenbank-Unterstützung

Der mitgelieferte Oracle-Adapter für Infopark CMS Fiona unterstützt jetzt Oracle 11.

Der mitgelieferte Sybase-Adapter linkt jetzt gegen die Open Client Library von Sybase 15. Ältere Sybase-Versionen werden nicht mehr unterstützt.

1.4.3 Neues Zertifikat für externen Editor

Das Java-Applet, mit dem eine clientseitige Anwendung als Editor gestartet werden kann (externer Editor) hat ein neues Zertifikat erhalten.

1.4.4 Vorschau berücksichtigt optional Leserecht der CMS Dateien

In der Vorschau des Redaktionssystems können die Leserechte der CMS-Benutzer oder der Live-Benutzer oder beide berücksichtigt werden. Für beides ist nun der `authorizationManager` zuständig. Siehe auch die Hinweise zum Upgrade.

1.4.5 Strukturierte Link-Funktionen (Callbacks)

Die `linkCallback`-Prozedur wurde erweitert und kann ihrerseits je zu verarbeitendem Link-Tag eine kundenspezifische Funktion aufrufen. Dies ist geeignet, den verarbeitenden Code transparenter und leichter wartbar zu gestalten.

1.4.6 Einfache Integration von Benachrichtigungsfunktionen (Post-Action-Callbacks)

Sogenannte Notification-Commands können registriert werden, indem sie entsprechend der Beschreibung in der Datei `share/script/cm/serverCmds/notificationCmd.tcl` im Instanzenverzeichnis `instanceName/script/cm/serverCmds/notifications/` abgelegt werden.

Das für die E-Mail-Benachrichtigung mitgelieferte Skript `share/script/cm/serverCmds/notifications/emailNotification.tcl` funktioniert bereits nach diesem Prinzip und kann als Beispiel für eigene Notification-Commands dienen.

1.4.7 Erweiterte Tcl-Formatter

Mit einem so genannten Formatter können in Layouts, d.h. in NPSOBJ-Anweisungen, Feldwerte formatiert werden, bevor sie in die Ausgabedatei eingefügt werden.

Der mitgelieferte Formatter `blobLength`, mit dem man Datei-Größenangaben formatieren kann, wurde um Optionen erweitert, mit denen unter anderem die verwendete Maßeinheit beeinflusst werden kann.

Der RFC-822-Formatter (zur Formatierung von Datumsangaben) wurde um zwei Optionen erweitert, die die Ausgabe der Sekunden und der Jahreszahl betreffen.

1.4.8 Interface von externen Präprozessoren erweitert

Externe Präprozessoren können einen Feldwert nun auch mit `encoding="stream"` als Datei oder mit `encoding="base64"` zurückgeben. Der MIME-Typ der konvertierten oder erzeugten Daten muss gesetzt werden.

In der Konfiguration eines externen Präprozessors können die Argumente, die dem Präprozessor-Programm übergeben werden sollen, jetzt auf eine eindeutige Art einzeln angegeben werden. Sofern dem Programm bisher mehr als ein Argument übergeben wurde, ist diese Konfigurationsänderung zwingend erforderlich.

1.4.9 Erweiterte LDAP-Anbindung des Content Management Servers

Eigenschaften von Benutzern oder Gruppen im LDAP, die über die LDAP-Konfigurationsoptionen `userAttributeMapping` bzw. `groupAttributeMapping` Feldern im CM zugeordnet wurden, können jetzt mit den Benutzermanager-API-Funktionen `userWithLoginGet` bzw. `groupWithNameGet` abgefragt werden.

1.4.10 Konfigurationsdateien der Benutzeroberfläche

Die GUI-Konfigurationsdateien `itemRegistry.xml`, `inspectorRegistry.xml` und `searchRegistry.xml` sind jetzt UTF-8-kodiert. Wenn Sie individuelle (geänderte) Versionen dieser Dateien haben, so konvertieren Sie sie bitte nach UTF-8 und passen Sie die XML-Deklaration an:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Zur Konvertierung können Sie unter Unix den Befehl `recode` und unter Windows einen beliebigen Character Encoding Converter verwenden, der Text vom bestehenden Encoding nach UTF-8 konvertieren kann.

Ferner wurde in der Datei `searchRegistry.xml` ein Bezeichner geändert. Anstelle von

```
<property name="labels">
```

wird nun einheitlich

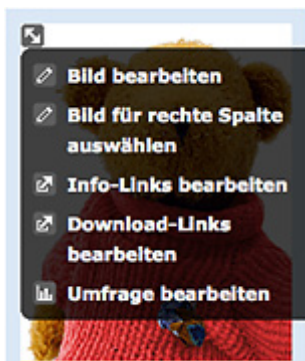
```
<property name="titles">
```

für die lokalisierten Titel von Suchparametern verwendet. Wenn Sie diese Datei geändert haben, so nehmen Sie diese Anpassung bitte manuell vor.

1.5 Infopark Rails Connector

Der Infopark Rails Connector liegt in einer neuen Version vor, mit der schneller als bisher das Grundgerüst einer modernen Web-Anwendung erstellt und durch individuelle Funktionalität erweitert werden kann. Die neue Version besteht aus zwei leicht zu installierenden und zu konfigurierenden Komponenten, je einer Komponente für die Grundfunktionalität und für die zahlreichen mitgelieferten Website-Features.

Bei der neuen Version des Rails Connectors wurde großen Wert auf sofortige Nutzbarkeit sämtlicher Funktionen gelegt. Mit den als "Add-ons" im Connector enthaltenen Website-Features können auf unkomplizierte Art und Weise beispielsweise Webseiten kommentierbar und bewertbar gemacht sowie RSS-Feeds eingebunden werden. Mit dem PDF-Generator lassen sich aus CMS-Inhalten unter Verwendung individueller Vorlagen PDF-Dateien erstellen. Ferner stehen Besucher-Registrierung, Anmeldung und Abmeldung sowie Passwort-Anforderung zur Verfügung. Für diese, den individuellen Besucher betreffenden Funktionen kann die neue integrierte Schnittstelle zum Infopark Online Marketing Cockpit genutzt werden. Lebendige Portale, bei denen die Kommunikation mit und zwischen den Besuchern ein wesentlicher Aspekt ist, lassen sich mit dem Rails Connector zügig entwickeln.



Auch das Zusammenspiel zwischen dem Rails Connector und Infopark CMS Fiona wurde verbessert. In der Vorschau können transparente Bearbeitungselemente und individuell gestaltbare Marker-Menüs eingesetzt werden, um die Arbeit im Redaktionssystem angenehmer und effizienter zu gestalten, auch unter Rückgriff auf vorhandene Ressourcen wie beispielsweise Assistenten zur Erfassung von Inhalten.

Ganz im Sinne erhöhter Produktivität wurde der Rails Connector um eine Template-Sprache, Liquid, erweitert. Liquid bietet dem Web-Entwickler einen ähnlichen Komfort wie die Layout-Sprache von

CMS Fiona. Hervorzuheben sind hier vor allem der dynamische Zugriff auf Inhalte und deren flexible Integration in Webseiten.

Insbesondere in Verbindung mit dem Online Marketing Cockpit bietet der Rails Connector interessante Möglichkeiten der Personalisierung und der gezielten Kommunikation mit den Nutzern von Diensten im Web. Die neuen Funktionen zur Suchmaschinenoptimierung runden das Bild ab. Sie können dazu beitragen, die Sichtbarkeit von Webangeboten zu verbessern.

1.6 Software von Drittanbietern

In Infopark CMS Fiona 6.7.1 wurde die folgende Software von Drittanbietern auf die angegebene Version aktualisiert.

- Autonomy (ehemals Verity) KeyView Export SDK: 10.8.0
- ImageMagick: 6.4.4
- Edit-on pro (EOP, integrierter HTML-Editor): 5.3.245
- Trifork Application Server: 4.1.32

- Apache Velocity: 1.6.2

Die Template-Sprache Velocity wird für die Anzeige der Detailansichten im GUI und als Standard-Template-Sprache für den Portal Manager verwendet. Zu Änderungen an der Syntax seit der Version 1.5 lesen Sie bitte das Protokoll der Änderungen in Velocity.

Die Software von Drittanbietern, die für den Betrieb von Infopark CMS Fiona 6.7.1 zwingend erforderlich ist (Systemvoraussetzungen), wird bei der Installation des CMS nun immer mitinstalliert. Es ist nicht mehr möglich, diese Software abzuwählen. Andere Versionen der mitgelieferten Software können bei Bedarf mit symbolischen Links eingebunden werden.

1.7 Weitere Änderungen

- Um die Größe der Software-Pakete zu verringern, wird die PDF-Version der Dokumentation jetzt als separates Package (das für alle unterstützten Plattformen gleich ist) bereitgestellt.
- Wenn man sich mit dem Tcl-Client am Search Server (SES) anmeldet, wird im Prompt als Benutzername `anonymous` angezeigt. Beispiel:

```
anonymous@localhost:3012 default (SES) >
```

- Die Konfigurationsdatei `config/mimeTypes.xml` enthält jetzt auch Einträge für die Dateiendungen `tif` und `tiff`.
- Das Skript `preview_mongrel` wurde bereits in Version 6.7.0 durch `mongrel_rails start` ersetzt.

2

2 Behobene Fehler

2.1 CMS-Funktionalität

- Durch ein falsches Konfigurationsbeispiel konnte es zu Problemen mit Basic Authentication oder NTLM-Authentifizierung kommen. Bei fehlgeschlagener Authentifizierung wird nun keine leere Seite anstelle einer Fehlerseite mehr angezeigt. (ID 11167)
- In der Suche des CMS funktioniert der Kalender für die Eingabe von Datumswerten wieder. (ID 11451)
- Feldwerte können jetzt auch aus einer Detailansicht heraus mit einem Assistenten bearbeitet werden. (ID 11536)
- Der Titel der Feldgruppe `baseGroup` kann wieder bearbeitet werden. (ID 11550)
- Im GUI wird jetzt korrekt angezeigt, ob die Felder `permaLink`, `title`, `channels` und `mimeType` Pflichtfelder sind. (ID 11551)
- Bei der Kombination aus Internet Explorer 6 (clientseitig) und Tomcat (serverseitig) ist es wieder möglich, einen Benutzer auszuwählen, um ihn beispielsweise zu einer Benutzergruppe hinzuzufügen. (ID 11596)
- Das GUI liefert nun wieder gültige `Expires`-Header aus. Die durch den ungültigen Header verursachten Einschränkungen bei der Bearbeitung von Inhalten oder der Konfiguration mit dem Internet Explorer bestehen nicht mehr. Ferner werden die GUI-Seiten nach Änderungen wieder aktualisiert. (ID 11610)
- Eingereichte Versionen zählen in der Statistik jetzt als Arbeitsversionen und nicht mehr als archivierte Versionen. (ID 11616)
- Die "Fehler in der Version" werden auch für leere Linklisten-Felder ausgegeben, wenn die Datei deshalb unvollständig ist. (ID 11673)
- Beim Anlegen von neuen Dateien werden nur noch Zugriffsrechte vererbt, die für den Typ der neuen Datei relevant sind. (ID 11732)
- Mehrfachauswahl-Felder lassen sich nun korrekt mit Auswahlkästchen bearbeiten. (ID 11807)
- Im GUI wurde ein Speicherleck behoben. Bei häufigem Anmelden und Abmelden konnte dieses Problem dazu führen, dass das GUI nicht mehr schnell genug reagierte oder stehen blieb. (ID 11817)
- Fehler während des Anmeldevorgangs im GUI werden nun wieder korrekt angezeigt und führen nicht mehr zu leeren Seiten oder Umleitungsschleifen. (ID 11857)
- Die Einstellungen zur Sortierung können jetzt auch im GUI auf die Standardwerte zurückgesetzt werden. (ID 11904)
- Durch die Aktualisierung der Velocity-Engine beantwortet das GUI parallele Anfragen zur Vorschau deutlich schneller. (ID 11922)

- Die Layoutprüfung funktioniert nun auch, wenn ein Layout eine `insertvalue-template-`Anweisung mit dem Parameter `key` enthält, dessen Wert einen ungültigen Namen (beispielsweise den Namen eines nicht existenten Feldes) enthält. Die Layoutprüfung weist mit einer entsprechenden Meldung auf diesen Fehler hin. (ID 12223)

2.2 Systemfunktionalität

- Die Master-Slave-Verwaltung von CM, TE und SES wurde verbessert und erzeugt nun keine unbegründeten Protokoll-Einträge über "Broken Pipes" mehr. (ID 6636)
- Nach Fehlern während der Indizierung liegen die zu indizierenden Daten weiterhin unverändert vor. (ID 7379)
- Die CM-, TE- oder SES-Master-Prozesse entfernen Ihre Slave-Prozesse nun immer korrekt. (ID 8345)
- Spiegeldateien werden beim inkrementellen Export nun korrekt aus dem Suchindex entfernt. (ID 10761)
- Bei der Migration werden nicht mehr gültige Rollen aus den Benutzereinstellungen entfernt. (ID 11402)
- Die Aktualisierung der Datenbanktabellen für den Rails Connector (mittels `CM -railsify`) kann jetzt abgebrochen werden und wird nach der Wiederaufnahme fortgesetzt. (ID 11527)
- Die LDAP-Anbindung im Tcl-Benutzermanager akzeptiert jetzt Bindestriche in Filtern (und damit auch in LDAP-Attributnamen). (ID 11554)
- Floating Licenses funktionieren wieder. (ID 11578)
- Nachdem Links gelöscht wurden, schreibt der Content Manager (CM) jetzt immer die aktualisierten Content-Daten in die Datenbank-Tabellen für den Rails Connector. Daher kann der Rails Connector keine bereits gelöschten Links von der Datenbank mehr erhalten. (ID 11583)
- Lizenzen werden vom GUI nicht mehr als ungültig betrachtet, wenn sie Umlaute enthalten, etwa im Element `owner`. (ID 11587)
- Die Anzahl der Protokollausgaben der HTML-Benutzerschnittstelle (GUI) im Log-Level `info` wurde verringert, um die Systemlast zu reduzieren. (ID 11650)
- Das Präprozessor-API des Search Servers (SES) kann wieder zur Aufbereitung der Indizierungsrequests verwendet werden. (ID 11669)
- Der Tcl-Befehl `indexAllObjects` verbraucht nun nicht mehr so viel Speicher. Dadurch läuft er auch mit sehr großen Datenmengen problemlos. Der Befehl überprüft nicht mehr, ob der ausführende Benutzer das Administrationsrecht für die Dateien hat. Daher kann `indexAllObjects` nur noch von einem Superuser ausgeführt werden. (ID 11700)
- Die CMS-Prozesse erkennen eine unterbrochene Datenbankverbindung zum MySQL-Server nun besser und versuchen nicht, weitere Datenbankoperationen über die unterbrochene Verbindung durchzuführen. (ID 11725)
- Das XML-Interface wandelt die HTML-Entitäten `>`, `<`, `&` und `"` wieder korrekt um. (ID 11745)
- Bei der Indizierung leerer Dateien vom Typ `ressource` mit einem Text-MIME-Typ (z.B. `text/html`) stürzt der Search Server nicht mehr ab. (ID 11779)
- Bei Verwendung des Applikationsservers Tomcat werden gültige Lizenzen nicht mehr zurückgewiesen. (ID 11802)
- Bei der Anbindung des Tcl-Benutzermanagers über LDAP/ADS werden Verbindungen wiederverwertet und serverseitig gestörte Verbindungen wiederhergestellt. (ID 11947)
- Änderungen an Spiegeldateien wirken sich beim inkrementellen Export nun korrekt auf den darüber liegenden Ordner aus. (ID 12205)

- Nachdem die Live-Leserechte einer Datei geändert wurden, werden die dazu gehörenden Einträge in den Rails-Tabellen jetzt aktualisiert. (ID 12208)
- Der inkrementelle Export der Template Engine wird durch Export-Fehler nicht mehr verlangsamt. (ID 12221)
- Das Startskript `rc.npsd` wartet jetzt fünf Sekunden lang, bevor es den Trifork-Applikationsserver startet. Dies erhöht die Wahrscheinlichkeit, dass der Port 8070, der von einer vorher laufenden Trifork-Instanz belegt wurde, wieder frei ist. (Keine ID)
- Dateien, die mit Assistenten angelegt wurden, können wieder mittels `npsGuiControl/npsSelect` ausgewählt werden. (Keine ID)

2.3 Rails Connector

- Die Rails-Connector-Tabellen `objs` und `preview_objs` haben jeweils die neuen Spalten `valid_from` und `valid_until` einschließlich passender Indizes erhalten. Dadurch kann die zeitliche Gültigkeit von Dokumenten schneller ermittelt werden. (ID 11539)
- Beim Anlegen eines Ordners werden in die Rails-Connector-Tabellen nun die richtigen Dateipfad-Informationen eingetragen. (ID 11569)
- Bei Einsatz des Rails Connectors und dessen Bearbeitungselementen ("action markers") kommt es in der Vorschau nicht mehr zu der folgenden Fehlermeldung: "java.lang.IllegalArgumentException: Class not supported: org.json.JSONObject\$Null". (ID 11589)
- Die Werte der Felder `sortKey1`, `sortKey2` und `sortKey3` werden nun richtig ermittelt, so dass nach Datum sortierte Listen wie gewünscht angezeigt werden. (ID 11907)
- Die Rails-Tabellen `objs` und `preview_objs` haben jeweils eine neue Spalte `is_edited` erhalten. (Keine ID)

2.4 Portal Manager

- Die Anzahl der im Dateisystem speicherbaren benutzerspezifischen Portlet-Einstellungen ist nicht mehr auf 31999 beschränkt. (ID 11559)
- Im `FilesystemPreferencesStorage`-Bean werden Threads nicht mehr so häufig synchronisiert, um Deadlocks zu verhindern. (Keine ID)

3

3 Wichtige Hinweise zum Upgrade

3.1 Java-API wurde auf Java Generics umgestellt

Wenn Sie eigene Java-Projekte haben, so beachten Sie bitte folgende Hinweise beim Übersetzen von Java-Projekten, die das bisherige API nutzen:

- Bei Fehlermeldungen der Art "Generics are not supported in 1.4" ändern sie bitte die Target-Version ihres Projekts auf Java 6.
- Bei Warnungen der Form "References to generic type should be parameterized" oder "Some input files use unchecked or unsafe operations" empfehlen wir Ihnen, Ihren Java-Code ebenfalls auf Generics umzustellen.

Alternativ können sie die Warnungen entweder durch die Annotation `@SuppressWarnings("unchecked")` im Quellcode oder mit Hilfe des Compiler-Parameters `-nowarn` unterdrücken.

3.2 Speicherung der Portlet-Einstellungen von Benutzern

Aufgrund von Einschränkungen durch Dateisysteme wird die Speicherung von Benutzereinstellungen im Dateisystem mittels `FileSystemPreferencesStorage`-Bean im Produktiveinsatz nicht mehr unterstützt. Als Ersatz wird empfohlen, die Benutzereinstellungen in einer Datenbank zu speichern.

3.3 Rails-Tabellen des CMS aktualisieren

Falls bei Ihrem CMS die Rails-Unterstützung aktiviert ist, so reinitialisieren Sie bitte mit dem Kommando `CM -railsify` die Rails-Tabellen in der CMS-Datenbank.

3.4 Abwicklung von Datenbank-Transaktionen im Portal Manager

Im Portal Manager werden Datenbank-Transaktionen jetzt über das Java Transaction Handling des Spring Frameworks durchgeführt. Aktivieren Sie bitte je nach verwendeten Datenbankfunktionen die benötigten Beans in der Datei `instanceName/webapps/*/WEB-INF/pm.xml` und konfigurieren Sie sie:

- `com.infopark.pm.user.DatabaseUserDirectory`
- `com.infopark.pm.DatabasePreferencesStorage`
- `dataSource`

- `transactionManager`

Wenn Sie eigenen Programmcode haben, der auf eine Datenbank zugreift, so ist dieser von der obigen Änderung nicht betroffen und braucht nicht angepasst zu werden.

3.5 Leserechte in der Vorschau

Aus der Datei `instanceName/webapps/GUI/WEB-INF/pm-filter.xml` wurde der Permission-Filter `com.infopark.cm.htmlgui.browse.preview.PermissionFilter` entfernt. Verwenden Sie stattdessen den Authorizer `com.infopark.cm.htmlgui.browse.preview.Authorizer` im `authorizationManager`, damit in der Vorschau die Datei-Leserechte der CMS-Benutzer (gegebenenfalls zusätzlich zu den Leserechten der Live-Benutzer) berücksichtigt werden.

3.6 Geänderter Namensraum der Standard-Formatter

Alle Standard-Formatter (jedoch nicht diejenigen zur Formatierung dynamischer Links) befinden sich jetzt im Namensraum `::formatter`. So heißt beispielsweise die Tcl-Prozedur des Formatters `phpVardef` jetzt `::formatter::phpVardef`.

Bei der Migration wird die Registrierung der mitgelieferten Formatter in der Systemkonfiguration angepasst. Kundenspezifische Formatter (im `script`-Unterverzeichnis der CMS-Instanzen) und deren Registrierung in der Systemkonfiguration müssen gegebenenfalls manuell angepasst werden.

3.7 Notwendige Datenbankoption bei der Verwendung von Sybase

Bei Verwendung des Sybase-Datenbankservers ist es erforderlich, für die CMS-Datenbanken die Option `select into` zu aktivieren.

3.8 Offizielle Unterstützung für Microsoft SQL Server 2008

Infopark unterstützt die Version 2008 des Microsoft SQL Servers jetzt offiziell als Datenbank für Infopark CMS Fiona. Ab diesem Release entfällt der Support für ältere Versionen von SQL Server. Es sprechen jedoch keine technischen Gründe gegen den Einsatz von SQL Server 2000 oder 2005.

4

4 Bekannte Fehler

- Wenn der Pfad einer Portlet-Web-Applikation im Pfad einer zweiten enthalten ist (wie beispielsweise bei `/default/PM-PL` und `/default/PM-PL-Copy`), so werden die Portlets beider Web-Applikationen nicht mehr angezeigt. (Id: 10531)
- Sucht man mit `obj where condition {objClasses is objType}` nach Dateien, so enthält das Suchergebnis keine Spiegeldateien, deren Originaldatei die angegebene Vorlage zugewiesen ist. (Id: 10553)
- Wenn eine CMS-Datei, die eine weitere Datei in einem `iframe` öffnet, in der Vorschau angezeigt wird, erscheint auch die im `iframe` angezeigte Datei in der Verlaufsliste. (Id: 10758)
- Bei Assistenten, die `npsField type="file"` verwenden, um eine Datei hochzuladen, führt der zweite und jeder weitere Versuch, die entsprechende Formularseite zu verwenden dazu, dass eine leere Ticket-ID erzeugt wird. Dadurch kann die Datei in diesem Fall nicht hochgeladen werden. (Id: 11404)
- Bei dem Versuch, einen Spiegelordner zu löschen, der selbst nicht verlinkt ist, jedoch mindestens eine verlinkte implizite Spiegeldatei enthält, wird eine falsche Fehlermeldung ausgegeben. Diese Meldung besagt, dass der Spiegelordner nicht gelöscht werden könne, weil er ein Linkziel sei. (Id: 11396)
- Bearbeitet man Links über den HTML-Editor (EOP) mit dem Link-Bearbeiten-Dialog des GUIs, so sind Umlaute oder Sonderzeichen im Titel anschließend falsch kodiert. (Id: 11315)
- Die Auswahl der CMS-Dateien, auf die ein Assistent anwendbar sein soll, kann mit der Assistenten-Eigenschaft `selectionType` definiert werden (`none`, `single`, `list`). Wird ein Assistent, dessen `selectionType` gleich `none` ist, über ein mittels `npsgui` erzeugtes Bearbeitungselement aufgerufen, so wird der Assistentenprozedur dennoch die ID der CMS-Datei übergeben, d.h. der Assistent wird so aufgerufen als wäre sein `selectionType` gleich `single`. Wenn der Assistent auch ins Menü oder in die Werkzeugleiste eingebunden ist, wird ihm je nach Art des Aufrufs also eine unterschiedliche Anzahl von Argumenten übergeben. Definieren Sie als Workaround den Assistenten mit `single` als `selectionType`. Alternativ kann die Assistentenprozedur so ergänzt werden, dass sie ihr zweites Argument, dessen Inhalt vom `selectionType` abhängt, richtig interpretiert. (Id: 11842)
- Mit dem Microsoft HTML Editor können keine Links oder Bilder in den HTML-Text eingefügt werden. Auch können bestehende Links und Bilder nicht bearbeitet werden. (Id: 11935)