



10. APRIL 2024

PRINT MANAGER VER. 4.0
INSTALLATION, KONFIGURATION UND BEDIENUNG

ANDREAS HARTMANN
HMEDIA
mail@hmedia.de

PRÄAMBEL

Die Informationen in diesem Dokument und in allen anderen Dokumenten, die zur PrintManager-Software gehören, können ohne vorherige Ankündigung geändert werden. Sie enthalten keine von Hmedia eingegangenen Verpflichtungen. Hmedia übernimmt keine Verantwortung für Fehler, die in diesen Dokumenten erscheinen könnten. Irrtümer und Auslassungen sind zu erwarten. Die zur PrintManager Software gehörende Dokumentation ist nur für interne, nicht-kommerzielle Referenzzwecke lizenziert.

Das in dieser Dokumentation beschriebene System unterliegt der technischen Weiterentwicklung. Es kann zu Abweichungen zwischen der Dokumentation und der tatsächlichen Implementierung des Systems kommen.

Kein Teil der mit der PrintManager-Software gelieferten oder auf der Hmedia-Website zugänglichen Dokumentation darf verändert werden.

Kein Teil der PrintManager-Software oder dieses Dokuments darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder in einem anderen Verfahren) ohne vorherige schriftliche Genehmigung von Hmedia an Dritte weitergegeben werden. Dem Anwender ist es gestattet, dieses Dokument für seine Zwecke zu vervielfältigen.

Hmedia behält sich alle nicht ausdrücklich in dieser Lizenz gewährten Rechte vor.

Die in diesem Dokument beschriebene Software unterliegt der folgenden Lizenz:

1. Die Software wird zur freien Verwendung auf den Systemen des Kunden überlassen.
2. Jegliche Veränderung oder Erweiterung der PrintManager-Software ist nicht gestattet.
3. Kein Teil der PrintManager-Software oder von PrintManager-Software-Artefakten darf in irgendeiner Form an Dritte weitergegeben werden.
4. Sie dürfen diese Software nicht zurückentwickeln, dekompileieren oder disassemblieren.

Diese Lizenz gilt für alle Updates und Ergänzungen, die Hmedia zur Verfügung stellt.

Hmedia kann Ihre Lizenz kündigen, wenn Sie die Bedingungen dieser Lizenz nicht einhalten.

In diesem Fall müssen Sie alle Instanzen der PrintManager-Software deinstallieren und alle Kopien der Software sowie alle ihre Artefakte vernichten.

DIESE SOFTWARE UND DIE BEGLEITENDE DOKUMENTATION WERDEN OHNE MÄNGELGEWÄHR UND OHNE GARANTIE JEDLICHER ART BEREITGESTELLT.

DARÜBER HINAUS ÜBERNIMMT HMEDIA KEINE GEWÄHRLEISTUNG, GARANTIE ODER HAFTUNG IN BEZUG AUF DIE NUTZUNG UND DIE ERGEBNISSE DER NUTZUNG DER SOFTWARE UND DER DOKUMENTATION IN BEZUG AUF RICHTIGKEIT, GENAUIGKEIT, ZUVERLÄSSIGKEIT, RECHTZEITIGKEIT ODER ANDERES. DAS GESAMTE RISIKO IN BEZUG AUF DIE ERGEBNISSE UND DIE LEISTUNG DER SOFTWARE LIEGT BEIM BENUTZER. SOLLTE DIE SOFTWARE ODER DIE BENUTZERDOKUMENTATION

FEHLERHAFT SEIN, SO TRÄGT ALLEIN DER BENUTZER UND NICHT HMEDIA DIE GESAMTEN KOSTEN FÜR ALLE NOTWENDIGEN REPARATUREN UND KORREKTUREN.

WEDER HMEDIA NOCH IRGEND EINE ANDERE PARTEI, DIE AN DER ENTWICKLUNG, PRODUKTION ODER LIEFERUNG DIESES PRODUKTS BETEILIGT IST, HAFTET FÜR SCHÄDEN JEDLICHER ART, SEIEN ES DIREKTE, INDIREKTE, FOLGE- ODER BEILÄUFIG ENTSTANDENE SCHÄDEN (EINSCHLIESSLICH SCHÄDEN FÜR GEWINNVERLUSTE, GESCHÄFTSUNTERBRECHUNGEN, DATENVERLUSTE, VERLUST VON GESCHÄFTSINFORMATIONEN UND ANDERE VERMÖGENSSCHÄDEN), DIE SICH AUS DER NUTZUNG ODER DER UNFÄHIGKEIT ZUR NUTZUNG DIESES PRODUKTS ERGEBEN, SELBST WENN HMEDIA AUF DIE MÖGLICHKEIT SOLCHER SCHÄDEN HINGEWIESEN WURDE.

Dieses Produkt enthält Software, die von der Apache Software Foundation (<http://www.apache.org/>) entwickelt wurde.

© Copyright 2024

Hmedia

Andreas Hartmann Medienlösungen

Vorwerkstraße 1

01936 Königsbrück - Deutschland

mail@hmedia.de

Alle Rechte vorbehalten. Gedruckt in Europa.

Die in diesem Dokument verwendeten Bezeichnungen und Markennamen der jeweiligen Unternehmen unterliegen in der Regel dem Warenzeichen-, Marken- oder Patentschutz.

INHALTSÜBERSICHT

Präambel	1
Vorwort	5
Allgemein.....	5
Über dieses Dokument.....	5
Symbole und Konventionen	5
System-Übersicht	7
Merkmale und Komponenten	7
Widerstandsfähigkeit	Fehler! Textmarke nicht definiert.
PrintServer - iNEWS.....	8
PrintClient - PrintServer.....	8
Netzwerk	8
Der PrintServer	9
Allgemein.....	9
Voraussetzungen	9
Das Betriebssystem	9
Java	9
Kater	10
Log4J Sicherheitslücke.....	10
Einrichtung	10
Vorbereitung des Tomcat.....	10
PrintServer-Bereitstellung.....	11
PrintServer-Konfiguration	11
PrintStyle Konfiguration	13
Status Seite.....	15
Lizenzvergabe	16
Operation	Fehler! Textmarke nicht definiert.
Log-Dateien	17
Der PrintClient	18
Allgemein.....	30
Voraussetzungen	30
Betriebssystem	30
.NET-Rahmenwerk.....	31

Java	31
Adobe Acrobat Reader	31
Einrichtung	32
Konfiguration.....	32
PrintServer.....	32
PrintClient-Eigenschaften.....	33
Operation	Fehler! Textmarke nicht definiert.
PrintClient im grafischen Modus	35
PrintClient Kommandozeilenparameter	35
Ein-Klick-Modus.....	37
Druckstile.....	39
PDF-Erstellungsprozess	39
Druckstile erstellen.....	39
Anhang	41
Quellcode der Standard-Druckstile	41
produktion_cues.xsl	41
rundown_body.xsl.....	43
rundown_cues.xsl.....	46

VORWORT

ALLGEMEIN

Der Hmedia PrintManager ergänzt das Avid Newsroom Management System um vielseitige PDF-Rendering- und Druckfunktionen. Er ermöglicht PDF-Renderings bestimmter iNEWS-Queues oder ausgewählter, einzelner Beiträge. Die erstellten PDF-Dokumente können gedruckt, mit Adobe Acrobat in der Vorschau angezeigt oder direkt aus der PrintClient-Software heraus per E-Mail verschickt werden.

ÜBER DIESES DOKUMENT

Die Kapitel dieses Dokuments behandeln die folgenden Themen:

- System-Übersicht: Gibt eine allgemeine Beschreibung der Komponenten des PrintManagers und wie diese Komponenten zusammenwirken.
- Der PrintServer - Native Bereitstellung: Erläutert die Installation und Konfiguration der PrintServer-Komponente direkt in einem Java Servlet Dienst.
- Der PrintServer - Containerisierte Anwendung: Erläutert die verschiedenen Möglichkeiten, den PrintServer als Containeranwendung einzusetzen.
- Der PrintClient: Installation, Konfiguration und Bedienung des PrintClients.
- Druckstile: Erklärt, wie Printstyles verwendet werden, um den Inhalt einer iNEWS-Warteschlange in ein druckfähiges Dokument umzuwandeln und wie solche PrintStyles strukturiert sind...

SYMBOLE UND KONVENTIONEN

Terminologie

Das Avid iNEWS System wurde in MediaCentral | Newsroom Management umbenannt. Der Begriff iNEWS ist nach wie vor gebräuchlich und wird in vielen Dokumenten verwendet. In dieser PrintManager-Dokumentation werden die beiden Begriffe **iNEWS** und **Newsroom Management** synonym verwendet.

Die Datenstruktur von iNEWS besteht aus drei grundlegenden Elementen: Verzeichnis - **Queue** - Story. Verzeichnisse spielen für den PrintManager keine Rolle. Eine Queue ist eine Sammlung von einzelnen Stories und das am häufigsten zu druckende Element. Der PrintManager kann jedoch auch einzelne oder mehrere ausgewählte Stories verarbeiten.

Ein Printstyle ist der Regelsatz zur Umwandlung der Quelldaten in ein PDF-Layout. Technisch gesehen sind die Printstyles XSLT-Stylesheet-Dokumente.

Fett

Fett gedruckter Text hebt wichtige Begriffe, Buchstaben, Zeichen oder Zahlen hervor.

Tipp

AUF DIESE WEISE FORMATIERTE INFORMATIONEN HELFEN DEM LESER, KEINE ZUSÄTZLICHEN FAKTEN ZU ÜBERSEHEN.

Warnung

EINE WARNUNG WIE DIESE HILFT IHNEN, TYPISCHE FEHLER UND BEKANNTE PROBLEME ZU VERMEIDEN.

Links

Hypertext-Links sehen so aus: <https://hmedia.de>

Konsolenbefehle und Konsolenausgabe

Eine andere Schriftart wird zur Kennzeichnung von Befehlszeilenausdrücken und zugehörigen Ausgaben verwendet, wie hier für den Befehl **java -version**:

```
c:\>java -version
```

```
openjdk Version "16" 2021-03-16
```

```
OpenJDK-Laufzeitumgebung (Build 16+36-2231)
```

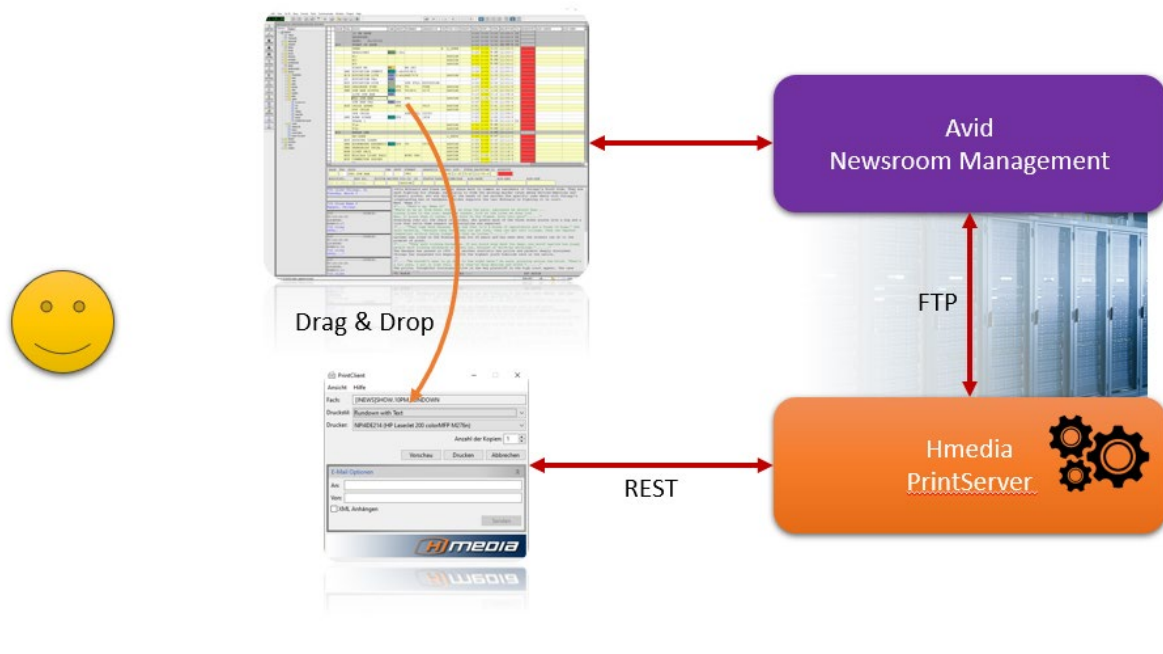
```
OpenJDK 64-Bit Server VM (Build 16+36-2231, gemischter Modus,  
gemeinsame Nutzung)
```

SYSTEM-ÜBERSICHT

MERKMALE UND KOMPONENTEN

Ursprünglich als iNEWS-Erweiterung entwickelt, um beliebige Daten in verschiedenen Formaten zu drucken, kann man nun die Ergebnisse in Adobe Acrobat ansehen und von dort aus weiterverarbeiten oder direkt aus dem PrintClient heraus als E-Mail versenden.

Der PrintManager besteht aus zwei Komponenten - dem PrintServer und dem PrintClient. Der PrintServer wird auf einem zentralen Host ausgeführt und verarbeitet alle Druckaufträge. Der PrintClient wird zusammen mit der Avid iNEWS Workstation Software installiert. Er sorgt für die Kommunikation mit iNEWS, dem Backend, Adobe Acrobat und der Druckinfrastruktur.



Ein typischer Druckvorgang besteht aus den folgenden Schritten:

1. Der Benutzer gibt an, welche Daten aus iNEWS gerendert werden sollen (eine iNEWS-Warteschlange oder eine beliebige Anzahl von Meldungen) und zieht sie per Drag&Drop in den PrintClient. Alternativ dazu bietet der PrintClient Befehlszeilenparameter zur Übergabe der zu druckenden Objekte.
2. In einem nächsten Schritt wählt der Benutzer den Druckstil aus.
3. Wenn alle Parameter definiert sind, sendet der PrintClient eine Anfrage an den PrintServer.
4. Der PrintServer liest die angeforderten Inhalte aus dem iNEWS-System und wandelt sie unter Verwendung des Printstyles in ein PDF-Dokument um. Dieses PDF-Dokument wird dann zurück an den PrintClient übertragen.
5. Der PrintClient zeigt die PDF-Datei in Adobe Acrobat an, druckt sie auf einem beliebigen angeschlossenen Drucker aus oder versendet sie per E-Mail.

AUSFALLSICHERHEIT

Der PrintServer verfolgt ein hochverfügbares und skalierbares Konzept.

PrintServer - iNEWS

Die Kommunikation mit dem Newsroom Management System ist fehlertolerant implementiert und unterstützt die typische iNEWS-Architektur mit zwei gespiegelten Servern. Wenn ein Server in einem Dual-iNEWS-System ausfällt oder aus irgendeinem Grund nicht reagiert, versucht der PrintServer, eine Verbindung zum anderen iNEWS-Server herzustellen.

PrintClient - PrintServer

Der PrintClient kann mit mehreren PrintServern kommunizieren. Wenn der PrintClient eine Anfrage für ein bestimmtes iNEWS-System erhält, versucht er, eine Verbindung zu allen für dieses System konfigurierten PrintServern herzustellen - einen nach dem anderen, bis er von einem der Server eine erfolgreiche Antwort erhält.

Zusätzlich kann ein einzelner PrintServer auch für mehrere iNEWS-Systeme arbeiten, und mehrere PrintServer können mit demselben iNEWS-System verbunden werden. Diese Architektur bietet Skalierbarkeit und Redundanz.

NETZWERK

Der PrintClient kommuniziert mit der REST-API des PrintServers über HTTP oder HTTPS. Der Standard-Port des PrintServers ist 8080, kann aber angepasst werden. Für eine verschlüsselte und gesicherte Verbindung kann der PrintServer mit einer Reverse-Proxy-Software wie Nginx gekoppelt werden. Dann wird die Kommunikation zwischen PrintClient und PrintServer über diesen Reverse-Proxy geleitet. Diese Komponente ist dann auch für die Verwaltung der TLS-Zertifikate zuständig. In einem solchen Szenario kann die REST-API auch mit Http Basic Authentication geschützt werden.

Der PrintServer ruft die Daten von den iNEWS-Servern über deren FTP(S)-Schnittstelle ab.

DER PRINTSERVER - NATIVE BEREITSTELLUNG

ALLGEMEINES

Dieser Abschnitt behandelt den PrintServer, falls er direkt auf einem Host-System innerhalb eines Java-Servlet-Dienstes installiert ist.

Hmedia entwickelt und qualifiziert die Software für Apache Tomcat. Obwohl auch andere Java-Applikationsserver den Hmedia PrintServer hosten können, behandelt diese Dokumentation nur die Installation und den Betrieb in Apache Tomcat.

Die Installation und die Funktionen von Apache Tomcat sind nicht Teil dieses Dokuments. Der Text kann Verweise auf öffentliche Internet-Seiten zu den entsprechenden Themen enthalten.

VORAUSSETZUNGEN

Betriebssystem

Apache Tomcat und Hmedia PrintServer sind Java-Programme. Daher sind Sie frei in der Wahl des zugrunde liegenden Betriebssystems.

Bei Hmedia testen wir den PrintServer 4.0 mit Apache Tomcat 9 auf **Windows Server 2019** und **Ubuntu 22.04 LTS**.

Java

Der PrintServer benötigt eine **Java Runtime Environment (JRE) 11** oder höher. Die aktuelle Version (März 2024) ist Java 21. Die Java-Installation muss vor der Installation des Tomcat abgeschlossen sein.

Der Eigentümer von Java, Oracle, hat sein Lizenzmodell für die Java Runtime im Jahr 2019 geändert. Seitdem ist die Oracle JRE nur noch mit einer kommerziellen Lizenz erhältlich.

Hmedia empfiehlt die Verwendung der Community-Version namens OpenJDK. Das OpenJDK kann unter <https://jdk.java.net> heruntergeladen werden. Das OpenJDK besitzt kein eigenes Installationsprogramm. Sie müssen die Installationsprozedur entsprechend Ihres Server-Betriebssystems befolgen, wie sie in verschiedenen Internet-Tutorials erklärt wird.

Wenn Sie Hilfe bei der Installation von Java auf Ihrem PrintServer-Computer benötigen, kontaktieren Sie uns bitte unter mail@hmedia.de.

STELLEN SIE SICHER, DASS DER BIN-ORDNER VON JAVA IN DER UMGEBUNGSVARIABLEN PATH EINGETRAGEN IST.

Der Befehl `java` muss auf der Betriebssystem-Shell ausführbar sein. Überprüfen Sie die aktive Version Ihrer Java-Installation mit dem Befehl `java -version`. Das Ergebnis sollte wie folgt aussehen:

```
openjdk Version "16" 2021-03-16
OpenJDK-Laufzeitumgebung (Build 16+36-2231)
```

OpenJDK 64-Bit Server VM (Build 16+36-2231, gemischter Modus, gemeinsame Nutzung)

Tomcat

Wie bereits erwähnt, wird der PrintServer in einem Java-Servlet-Container ausgeführt. Hmedia entwickelt und testet die Software mit Apache Tomcat. Wir setzen die Versionen Tomcat 7 und 9 ein.

TOMCAT 10 KANN NICHT VERWENDET WERDEN! IN DIESER VERSION HABEN VIELE KOMPONENTEN UNTERSCHIEDLICHE NAMENSÄUME (MIGRIERT VON JAVA ZU JAKARTA). DAHER IST ER NICHT ABWÄRTSKOMPATIBEL.

Unter Linux wird Tomcat normalerweise mit distributionsspezifischen Paketen geliefert. Das macht die Installation recht einfach und folgt den üblichen Verfahren Ihres Linux-Paketmanagers (wie yum oder apt). Wenn Sie eine neuere Version als die von Ihrer Distribution bereitgestellte installieren möchten, können Sie vorkompilierte Binärdateien installieren. Sie finden diese im Download-Bereich der Apache Tomcat-Website <https://tomcat.apache.org>. Wenn Sie Hilfe bei einer solchen Installation benötigen, wenden Sie sich bitte an mail@hmedia.de.

Für Microsoft Windows finden Sie die Binärpakete im Download-Bereich der Apache Tomcat-Website <https://tomcat.apache.org>. Wenn Sie Hilfe bei der Installation benötigen, wenden Sie sich bitte an mail@hmedia.de.

Log4J Sicherheitslücke

Die Version 4.0 des PrintManagers verwendet eine aktualisierte Version des Log4J-Frameworks.

PRÜFEN SIE AUßERDEM IHRE JAVA- UND TOMCAT-INSTALLATION AUF EVENTUELL ENTHALTENE LOG4J-BIBLIOTHEKEN UND ERSETZEN SIE DIESE DURCH EINE NEUERE VERSION >= 2.17!

EINRICHTUNG

Das folgende Kapitel beschreibt den Installationsvorgang der PrintServer-Software. Wir verwenden hier die Linux-Pfadsyntax mit dem Trennzeichen /. Wenn Sie die Anleitung auf ein Windows-System anwenden, passen Sie bitte die Pfadnamen entsprechend an.

Vorbereitung des Tomcat

1. Lokalisieren Sie den Ordner, in dem Tomcat installiert ist. Dieser Ordner wird als **<TOMCAT_HOME>** bezeichnet.
2. Erstellen Sie den Ordner **<TOMCAT_HOME>/shared/classes**.
3. Bearbeiten Sie die Datei **<TOMCAT_HOME>/conf/catalina.properties** und ersetzen Sie die Zeile **shared.loader=** durch **shared.loader=\${catalina.base}/shared/classes**.

WENN SIE SPÄTER DATEIEN INNERHALB DES GEMEINSAMEN ORDNERS ÄNDERN, MÜSSEN SIE TOMCAT NEU STARTEN, UM DIE ÄNDERUNGEN KORREKT ZU ÜBERNEHMEN!

PrintServer-Bereitstellung

Um die PrintServer-Software in Tomcat einzusetzen, führen Sie die folgenden Schritte aus:

1. Entpacken Sie das Installationsarchiv **printserver-xyz.zip**. xyz steht für die Versionsnummer wie 4.0-512. Das entpackte PrintServer-Archiv enthält die Datei **printserver.war** und einen Unterordner **shared/classes**.
2. Kopieren Sie die Datei **printserver.war** in den Ordner **<TOMCAT_HOME>/webapps**.
3. Kopieren Sie den gesamten Inhalt des Ordners **shared/classes** des Archivs in den Ordner **<TOMCAT_HOME>/shared/classes**.

Herzlichen Glückwunsch, Sie haben den Hmedia PrintServer installiert! Nun müssen Sie den PrintServer wie im nächsten Abschnitt beschrieben konfigurieren.

PrintServer-Konfiguration

Alle Konfigurationsdateien des PrintServers befinden sich innerhalb des Tomcat-Installationsordners. Wie oben beschrieben, wird dieser als **<TOMCAT_HOME>** bezeichnet. Sie finden die Standardversionen der Konfigurationsdateien in dem entpackten PrintServer-Archiv.

iNEWS Konfiguration

Richten Sie die Verbindung des PrintServers zu einem oder mehreren Newsroom Management Systemen mit der Datei **inewsconf.xml** ein. Diese muss sich im Ordner **<TOMCAT_HOME>/shared/classes/printserver** befinden. Der Inhalt der Datei sieht wie folgt aus:

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

<!--
The connectionTimeout variable is the maximum time in milliseconds this program waits for an
iNEWS Server response when connecting. If an iNEWS server does not respond in the given time,
this program will assume the server to be down.
-->

<iNEWSConfiguration>
  <system name="iNEWS" maxConnections="6" loadBalance="2">
    <server host="inews-a" maxConnections="6"
      defaultUser="avstar" defaultPassword="avstar" connectionTimeout="3000"/>
    <server host="inews-b" maxConnections="6"
      defaultUser="avstar" defaultPassword="avstar" connectionTimeout="3000"/>
  </system>
  <system name="NRCS" maxConnections="6" loadBalance="2">
    <server host="172.21.10.31" maxConnections="6"
      defaultUser="avid" defaultPassword="inews" connectionTimeout="3000"/>
  </system>
</iNEWSConfiguration>
```

In obigem Beispiel arbeitet der PrintServer für zwei iNEWS-Systeme INEWS und NRCS. Bei INEWS handelt es sich um eine Dual-Server-Konfiguration, bei NRCS um eine Single-Server-Konfiguration. Jeder Server muss mit einem Netzwerknamen und Anmeldedaten (Benutzername und Passwort) konfiguriert werden. Diese Zugangsdaten werden zur Authentifizierung für die FTP(S)-Verbindung (RXNET) verwendet.

Der PrintServer kann eine beliebige Anzahl von iNEWS-Systemen mit jeweils einem bis zu vier Servern verwalten.

Elemente der Datei **inewsconf.xml**:

Name des Elements: iNEWSConfiguration

Subelemente: system (beliebige Anzahl)

Attribute: keine

Bedeutung: Das XML Wurzelement.

Name des Elements: system

Subelemente: server (1 - 4)

Attribute:

- **name:** Der Name des iNEWS-Systems. Muss mit der Konfiguration im PrintClient übereinstimmen (siehe Kapitel Der PrintClient - Konfiguration). In der Regel ist es derselbe Name, der für die iNEWS-Datenbank konfiguriert ist.
- **maxConnections:** Die maximale Anzahl der FTP(S)-Verbindungen, die der PrintServer zu diesem iNEWS-System herstellen soll.
- **loadBalance:** Wenn das iNEWS-System mehrere Server online hat, verteilt der PrintServer die FTP(S)-Verbindungen so, dass die Anzahl der Verbindungen zu einem Server nicht größer ist als der loadBalance-Wert, verglichen mit jedem anderen Server in diesem bestimmten System.

Bedeutung: Repräsentiert ein iNEWS-System.

Name des Elements: server

Subelemente: keine

Attribute:

- **host:** Der Netzwerkname des Servers (IP-Adresse, Hostname oder FQDN)
- **maxConnections:** Die maximale Anzahl der FTP(S)-Verbindungen, die der PrintServer zu diesem iNEWS-Server herstellen soll. Dieser Wert kann nicht größer werden als die Gesamtzahl der maxConnections für das gesamte System.
- **defaultUser:** Der Benutzername für die Authentifizierung auf dem iNEWS-Server.
- **defaultPassword:** Das Passwort für den defaultUser.
- **connectionTimeout:** Timeout für Verbindungsversuche in Millisekunden. Wenn innerhalb dieser Zeit keine Verbindung hergestellt werden kann, geht der PrintServer davon aus, dass dieser Server offline ist.

Bedeutung: Repräsentiert einen Newsroom-Management-Server.

PrintStyle Konfiguration

Alle Printstyle-Dateien müssen sich im Ordner

<TOMCAT_HOME>/shared/classes/printserver/stylesheets befinden. Um die Druckstile zu aktivieren, müssen Sie sie in der Konfigurationsdatei

<TOMCAT_HOME>/shared/classes/printserver/iNEWSSystems.txt registrieren. Diese Konfiguration bietet folgende Vorteile:

1. Sie können jeder Printstyle-Datei einen benutzerfreundlichen Namen geben, der dann im PrintClient angezeigt wird. Dieser Name kann Leerzeichen und Sonderzeichen enthalten.
2. Sie können Druckstilen einzelnen iNEWS-Systemen zuweisen, so dass sie nur dort verwendet werden.
3. Sie können einzelnen iNEWS-Queues spezifische Druckstile zuweisen.

In der Konfigurationsdatei steht jeweils eine printstyle-Datei auf einer Zeile. Jeder Eintrag beginnt mit dem Dateinamen der printstyle-Datei. Dann folgt ein Leerzeichen (Space oder Tab) als Trenner, gefolgt vom Anzeigenamen.

Eine Zeile, die mit einer eckigen Klammer [beginnt, wird als iNEWS-System behandelt, wie [INEWS] oder [NRCS]. Der PrintClient zeigt alle Druckstile, die auf eine solche Zeile folgen (vor der nächsten iNEWS-Systemzeile) nur an, wenn er mit diesem iNEWS-System arbeitet.

Eine Zeile, die mit einem @ beginnt, gefolgt von einem regulären Ausdruck, zielt auf einen iNEWS-Queue-Namen. Der PrintClient zeigt alle folgenden Druckstile (vor der nächsten Zeile mit System- oder Queue-Namen) nur an, wenn die zu druckende Queue mit dem regulären Ausdruck übereinstimmt.

Auf eine Zeile, die mit **@ nomatch** beginnt, können Druckstile folgen, die angezeigt werden, wenn die zu druckende Queue mit keinem Eintrag in den darüberliegenden Queue-Namen Zeilen übereinstimmt.

Beispiel 1

Die folgende Beispiel iNEWSSystems.txt enthält drei Druckvorlagen. Die erste Datei ist **rundown_body.xml** und der Anzeigenname ist **Rundown mit Text**, usw. PrintClients, die sich mit diesem PrintServer verbinden, zeigen alle drei Stile unabhängig vom iNEWS-System und der Warteschlange an.

```
rundown_body.xml Rundown mit Text
rundown_cue.xml Rundown mit Anweisungen
production_cues.xml Stories mit Text und Anweisungen
```

Beispiel 2

Die nächste Abbildung zeigt eine iNEWSSystems.txt, die für zwei iNEWS-Systeme gilt. Die erste Druckvorlage (rundown_body.xml) ist für beide iNEWS-Systeme verfügbar. Der zweite Druckstil (rundown_cue.xml) wird nur angezeigt, wenn aus dem System NRCS gedruckt wird. Die dritte Druckvorlage (production_cues.xml) wird nur dann angezeigt, wenn das Quellsystem INEWS ist.

```
rundown_body.xml Rundown mit Text
```

[NRCS]

rundown_cue.xml Rundown mit Stichwörtern

[INEWS]

production_cues.xml Geschichten mit Text und Hinweisen

Beispiel 3

Die letzte Beispielkonfiguration weist eine komplexere Einrichtung auf:

Das erste iNEWS-System **NRCS** verfügt über drei Stile.

Für das zweite System **NRCS26** zeigt der PrintClient verschiedene Druckstile an, abhängig von der zu druckenden Queue.

- Für eine Queue, die den Begriff RUNDOWN irgendwo in ihrem Namen/Pfad enthält, zeigt der PrintClient
Presenter, Production Cues, RAI Cues, Rundown Cue und Rundown Cue Body.
- Für eine Queue, die den Begriff TRAINING enthält, zeigt der PrintClient
Presenter, Production Cues, RAI Cues und Training Stylesheet.
- Für alle anderen Warteschlangen, die weder RUNDOWN noch TRAINING enthalten, zeigt der PrintClient
Presenter, Production Cues, RAI Cues, RAI Title und RAI Text an

[NRCS]

```
production_cues.xml    Produktion_Cues
rundown_cue.xml       Rundown_Cue
rundown_cue_body.xml  Rundown_Cue_Body
```

[NRCS26]

```
presentator.xml      Presenter
production_cues.xml  Production Cues
rai_cues.xml         RAI Cues
```

@ .*RUNDOWN.*

```
rundown_cue.xml      Rundown Cue
rundown_cue_body.xml Rundown Cue Body
```

@ .*TRAINING.*

```
rai_training.xml    Training Stylesheet
```

```
@ nomatch
rai_title.xsl          RAI-Title
rai_text.xsl          RAI Text
```

STATUS SEITE

Nach der Installation und Konfiguration muss der Tomcat neu gestartet werden. Der entsprechende Befehl hängt von Ihrem Betriebssystem und der Tomcat-Installation ab.

Um den Status des PrintServers nach dem Start zu überprüfen, können Sie einen beliebigen Webbrowser auf einem beliebigen Computer verwenden, der über einen Netzwerkzugang zur folgenden URL verfügt:

http://<printserver_host>:8080/printserver.

Hier ist **<printserver_host>** der Netzwerkname des PrintServer-Computers.

Die Ausgabe im Browser sieht dann so aus:



NACH DEM START DES PRINTSERVERS IST DIE STATUSSEITE ERST VERFÜGBAR, NACHDEM DER STATUS ALLER KONFIGURIERTEN INEWS-SYSTEME ERMITTELT WURDE. WENN EIN ODER MEHRERE INEWS-SYSTEME OFFLINE SIND, VERSUCHT DER PRINTSERVER, EINE VERBINDUNG HERZUSTELLEN, BIS DER KONFIGURIERTE TIMEOUT ERREICHT IST.

Im obigen Beispiel ist das iNEWS-System NRCS nicht verfügbar, während das System mit dem Namen INEWS bereit ist.

Der PrintServer im obigen Beispiel kann nicht verwendet werden, da er nicht lizenziert ist.

LIZENZVERGABE

Für den vollständigen Betrieb des PrintServers ist eine Lizenzdatei namens **license.lic** erforderlich. Hmedia liefert diese Datei separat zum Softwarepaket aus. Sie müssen die Lizenzdatei in den Ordner **<TOMCAT_HOME>/shared/classes/printserver** kopieren.

Beim nächsten Tomcat-Start wird der PrintServer die Lizenz validieren und den Status aktualisieren:



Es ist keine Online-Aktivierung erforderlich.

Mit der Validierung der Lizenz auf dem PrintServer ist das gesamte PrintManager-System lizenziert und voll funktionsfähig. Es gibt keine separate Lizenz für den PrintClient. Die Anzahl der angeschlossenen PrintClients, iNEWS-Systeme oder Druckstile ist nicht begrenzt.

Die Lizenz wird nicht mit der Hard- oder Software des PrintServer-Rechners verschränkt und kann für neue oder weitere Installationen wiederverwendet werden.

Die Lizenz ist jedoch auf den Kunden limitiert und darf nicht an Dritte oder andere Kunden weitergegeben werden.

Die Lizenz bezieht sich auf eine bestimmte PrintServer-Version oder -Versionsreihe und kann nur mit der entsprechenden Software verwendet werden.

Es gibt zeitlich befristete und unbefristete Lizenzen.

BEDIENUNG

Für den PrintServer selbst gibt es keine spezifischen Kommandos. Start, Stopp und Neustart werden auf der Tomcat-Ebene durchgeführt.

Wenn Sie die iNEWS-Systeme neu konfigurieren (**inewsconf.xml**), starten Sie den Tomcat neu.

Wenn Sie bestehende Druckstile ändern/debuggen, brauchen Sie Tomcat nicht neu zu starten. Der PrintServer liest eine bestimmte Printstyle-Datei zur Laufzeit auf Anfrage und speichert sie nicht im Cache.

Wenn Sie jedoch die Datei **iNEWSsystems.txt** ändern, müssen Sie den Tomcat anschließend neu starten.

LOG-DATEIEN

Die Protokolldateien von Tomcat befinden sich im Ordner **<TOMCAT_HOME>/logs**.

Spezifische Meldungen des PrintServers finden Sie in der Datei **<TOMCAT_HOME>/Hmedia.log**.

Diese Protokolldatei wird wie üblich rotiert.

ES GIBT KEINE AUTOMATISCHE PROTOKOLLBEREINIGUNG FÜR TOMCAT/PRINTSERVER. FALLS ERFORDERLICH, MUSS DER ADMINISTRATOR ALTE PROTOKOLLDATEN MANUELL LÖSCHEN.

HTTPS UND AUTHENTIFIZIERUNG

Mit Version 4.0 kann der PrintClient über HTTPS auf die REST-API zugreifen. Er unterstützt auch Basic HTTP Authentication. Siehe Kapitel Der PrintClient - Konfiguration, auf Seite 32 für weitere Einzelheiten.

Auf der PrintServer-Seite werden diese Funktionen nicht vom PrintServer selbst bereitgestellt. Sie können entweder in der Servlet-Container-Software (Tomcat) konfiguriert werden oder, was noch einfacher ist, indem der PrintServer-REST-API ein Reverse-Proxy vorgeschaltet wird. Eine gängige Lösung für einen solchen Reverse Proxy ist der Apache Nginx.

Wenn Sie Hilfe bei der Installation und Konfiguration von Nginx für TLS und Authentifizierung benötigen, kontaktieren Sie uns bitte unter mail@hmedia.de.

DER PRINTSERVER - EINE CONTAINERISIERTE ANWENDUNG

ÜBERSICHT

Mit Version 4.0 ist PrintServer als containerisierte Anwendung verfügbar. Dies bietet eine wesentlich robustere und stabilere Bereitstellung, die leicht skalierbar und für Failover-Funktionen konfigurierbar ist.

Es gibt mehrere Möglichkeiten, den PrintServer als containerisierte Anwendung zu installieren, was Ihnen verschiedene Grade an Flexibilität bietet. Auf diese Weise können Sie den PrintServer optimal in Ihre Infrastruktur integrieren.

Die verfügbaren Einsatzoptionen sind:

- Als einfacher Docker-Container in einer Docker-kompatiblen Container-Laufzeitumgebung. In diesem Fall können Sie weitere Details auf <https://hub.docker.com/r/hmediade/printserver> erhalten und das Image einfach mit `docker pull hmediade/printserver` abrufen.

Zusätzlich zum PrintServer-Container muss ein Reverse-Proxy-Container (wie Nginx) konfiguriert und ausgeführt werden, falls ein verschlüsselter und authentifizierungsgeschützter Zugriff auf die PrintServer-REST-API gefordert ist.

Die Bereitstellung der Daten (Konfiguration und Druckstile) ist völlig frei wählbar.

- Die Bereitstellung in einem Kubernetes-Cluster lässt sich am besten über das Helm-Chart steuern, das wir von Hmedia für Sie erstellt haben. Es ist inklusive einiger Anleitungen auf <https://artifacthub.io/packages/helm/hmediade/printserver> verfügbar. Mit diesem Chart erhalten Sie ein vollständiges, skalierbares Deployment, optional inklusive Verschlüsselung und Authentifizierung für die PrintServer REST API. Die Konfiguration und Installation ist auf der genannten Seite detailliert beschrieben.
- Die bei weitem einfachste und unkomplizierteste Methode, den PrintServer zu installieren: Wenn Sie das Helm Chart nicht manuell vorbereiten möchten, bieten wir ein Deployment-Skript an, das Sie durch eine interaktive Setup-Routine führt, um alle erforderlichen Daten einzugeben und die PrintServer-Software nahtlos zu installieren, zu aktualisieren und zu entfernen. Sie können dieses Skript (`printserverctl`) von der [PrintServer-Produktseite](#) herunterladen.

Kopieren Sie es einfach auf den Kontroll-Client für Ihren Kubernetes-Cluster (wo Sie `kubectl` ausführen). Weitere Hilfe erhalten Sie mit dem Skript und diesem Dokument.

Dieses Dokument konzentriert sich nur auf die Installation mit unserem printserverctl-Skript. Wenn Sie Ihre Umgebung mit einem der anderen Einsatzszenarien anpassen möchten, wenden Sie sich bitte an mail@hmedia.de, um weitere Hilfe zu erhalten.

KONZEPTE UND ARCHITEKTUR

Dieser Abschnitt beschreibt die Architektur einer voll ausgestatteten PrintServer-Umgebung.

Die Kernkomponente einer jeden containerisierten Anwendung ist das Image. Hmedia verwendet Docker, um das Image für den PrintServer zu erstellen. Dieses Image enthält eine grundlegende Linux-Schicht, Java, Tomcat und die PrintServer-Software selbst.

Damit der PrintServer arbeitsfähig ist, müssen alle spezifischen Daten in den Container injiziert werden und der Container zusätzlich mit anderen Komponenten (wie einem Reverse-Proxy) gekoppelt werden.

Dateninjektion

Der PrintServer erfordert zwei Arten von Konfigurationen und Einstellungen, die standortspezifisch sind und nicht mit der Software selbst geliefert werden können:

- PrintServer-Konfiguration
- PrintStyles und zusätzliche Dateien (wie Grafiken)

Grundsätzlich können Daten für einen Container über Dateimounts oder über Umgebungsvariablenbereitgestellt werden. Der PrintServer-Container selbst arbeitet ausschließlich mit einer dateibasierten Bereitstellung (keine Umgebungsvariablen), da alle benötigten Daten als Dateien vorliegen.

Wenn der PrintServer-Container einzeln bereitgestellt wird (nicht über Kubernetes, Helm oder das printserverctl-Skript), kann man einfach zwei Verzeichnisse im Container einhängen, eines für die Konfigurationsdateien und eines für die Druckstile. Das Verfahren wird auf <https://hub.docker.com/r/hmediade/printserver> ausführlich erklärt.

Wird der PrintServer in einem Kubernetes-Cluster verteilt, besteht jeder PrintServer-Pod aus einem printserver-init-Container und dem Haupt-Printserver-Container. Abbildung 1 zeigt, wie der Pod die benötigten Daten erhält.

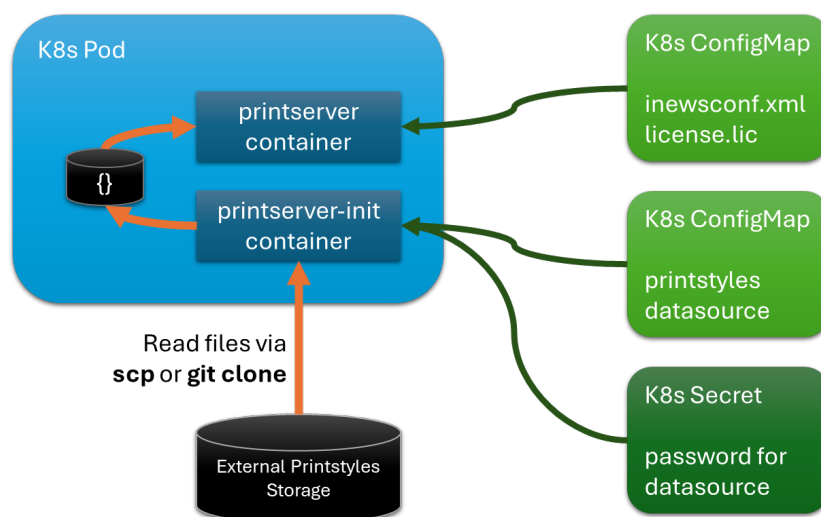


Abbildung 1: Dateninjektion in den PrintServer Container

Beim Erstellen des Pods führt Kubernetes zunächst den Init-Container aus. Dieser liest die Printstyle-Dateien aus einer externen Quelle. Die aktuelle Version des PrintServers kann die Dateien über scp oder git lesen. Alle Details sind auf <https://hub.docker.com/r/hmediade/printserver-init> beschrieben.

Der init-Container speichert printstyles und nachfolgende Dateien in einem ephemeren Speicher innerhalb des Kubernetes-Pods.

Sobald der Init-Container beendet ist, startet der Hauptcontainer mit der PrintServer REST API.

Bereitstellung der PrintServer REST API

Wenn der PrintServer als eigenständiger Container (nicht in Kubernetes) läuft, wird ein Host-Port auf den Port des PrintServer-Containers (8080) verbunden. Die Kommunikation ist rein http - keine Verschlüsselung, keine Authentifizierung.

Bei einer vollständigen Kubernetes-Bereitstellung spielen verschiedene Kubernetes-Artefakte eine Rolle. Die Reihenfolge ist von innen nach außen:

1. Pod - stellt die Kernfunktionalität bereit und bietet die REST-API in einem internen Pod-Netzwerk an. Pods sind mortal/volatil. Sie können jederzeit zerstört und neu erstellt werden. Ein neuer Pod ist ein anderes Kubernetes-Objekt und hat eine andere IP-Adresse im Pod-Netzwerk als der vorherige. Die Arbeitslast kann durch den Betrieb mehrerer Instanzen skaliert werden.
2. Service - Ein Service hat eine feste IP im internen Servicenetzwerk und bietet einen Zugangspunkt zur API über eine stabile Adresse/DNS-Name. Allerdings ist dieser Zugang immer noch rein Kubernetes-intern.
3. Ingress - Ein Reverse-Proxy (Nginx) muss im Kubernetes-Cluster als Ingress-Controller bereit stehen. Ein Ingress-Objekt verbindet dann das äußere Netzwerk mit dem Service (siehe Punkt 2). Zusätzlich unterstützt der Ingress-Controller Verschlüsselung (HTTPS/TLS) und Authentifizierung (Basic Http Authentication).

Abbildung 2 zeigt die vollständige Einrichtung in einem Kubernetes-Cluster.

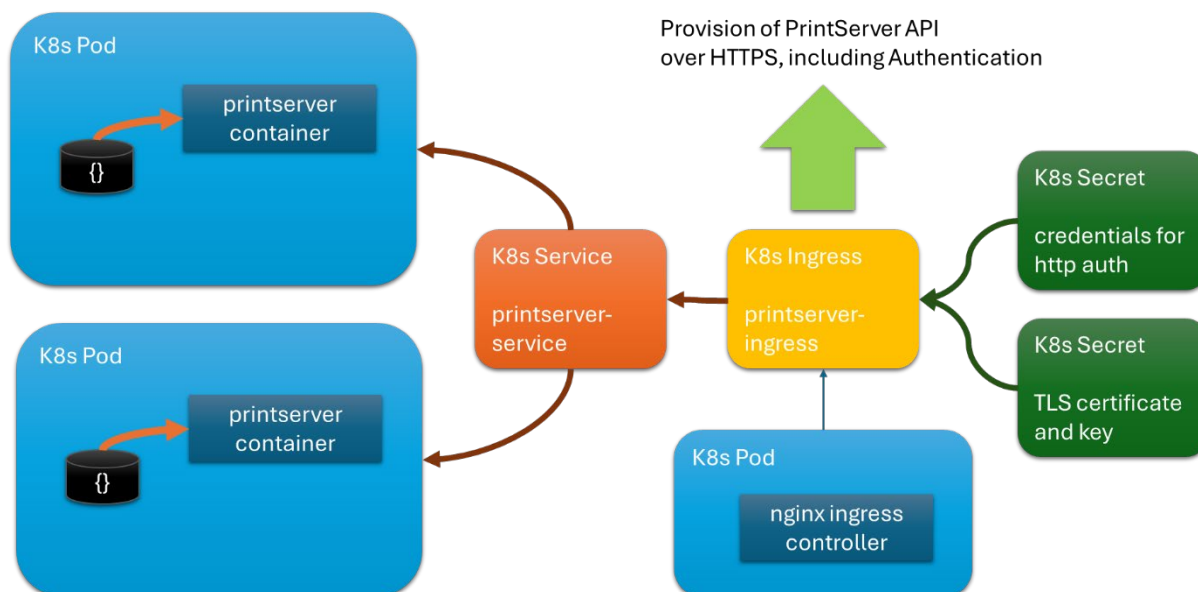


Abbildung 2: Kubernetes-Objekte für eine PrintServer-Bereitstellung

Im Anhang finden Sie eine Musterdeklaration aller erforderlichen Artefakte. Sie können den gesamten PrintServer mit einem einzigen Befehl `kubectl apply -f <declaration.yaml>` bereitstellen, wobei `declaration.yaml` die Beschreibung aller erforderlichen Objekte in der Kubernetes-spezifischen Deklarationsprache ist.

Hmedia bietet – wie oben bereits erwähnt – ein interaktives, dialoggesteuertes Deployment an. Siehe Kapitel Bereitstellung auf Seite 22 für weitere Informationen.

SOFTWARE-SPEZIFIKATIONEN

PrintServer Container

Das Image für den Printserver-Container wird auf dem Image `tomcat:9` aufgebaut.

Als Basisschicht verwendet dieses Image ein Ubuntu 22.04 LTS-System. Das Java ist in der neuesten Version - JDK 21, erstellt vom Eclipse Temurin Projekt. Der Tomcat selbst ist in der Version 9.0.87, zum Zeitpunkt der Erstellung des Images.

PrintServer-Init Container

Das Image für den `printserver-init`-Container wird auf `ubuntu:latest` erstellt. Dieses Image enthält eine Basisversion von Ubuntu 22.04 LTS zum Zeitpunkt der Erstellung des Images. Zusätzliche Software, die in dem Container verwendet wird, ist:

- `openssh-client`
- `sshpas`
- `git`

BEREITSTELLUNG MIT DEM HMEDIA PRINTSERVERCTL

Übersicht

Das Skript `printserverctl` vereinfacht die Installation des Hmedia PrintServers. Es bietet verschiedene Befehle für unterschiedliche Operationen:

- `prepare`
- `clean`
- `review`
- `dryrun`
- `deploy`
- `remove`
- `version`
- `help`

Das Skript `printserverctl` bereitet die Konfiguration vor und initiiert Aufträge mit dem Werkzeug `helm`. Es speichert alle Konfigurationsdaten in dem Ordner `/etc/hmedia/printserver`. Alle Dateien in diesem Ordner werden nur für die Bereitstellung benötigt und nicht in einer laufenden Instanz des PrintServers. Das Programm `printserverctl` erstellt diesen Ordner, falls er nicht vorhanden ist.

Wenn Sie möchten, können Sie die Daten im Einstellungsordner `/etc/hmedia/printserver` überprüfen und manuell anpassen. Dies hat keine Auswirkungen auf eine laufende Instanz des PrintServers, bis Sie `printserverctl deploy` ausführen.

Voraussetzungen

Erforderliche Werkzeuge

Die folgenden Linux-Tools und -Programme werden vom Deployment-Skript verwendet:

- `mkdir`
- `sed`
- `awk`
- `weniger`
- `base64`
- `Steuerrad`
- `kubectl`

Das Skript `printserverctl` prüft bei jeder Ausführung, ob alle diese Werkzeuge und Programme vorhanden sind.

Kubernetes muss in Betrieb sein und mit dem Befehl `kubectl` auf demselben Rechner erreicht werden können.

Kubernetes muss die Version 1.28 oder höher haben.

Helm muss installiert und verfügbar sein. Helm Version 3.0 und höher wird unterstützt.

Netzzugang

Es wird dringend empfohlen, während der Installation den Internetzugang zu aktivieren. Der Befehl `helm` ruft das PrintServer Helm-Chart aus dem Internet ab, und Kubernetes zieht, zumindest beim ersten Start, die Container-Images von der Docker Hub-Website.

Die Installation des PrintServers ist auch komplett offline möglich. Wenn Sie an einem solchen Einsatzszenario interessiert sind, kontaktieren Sie uns bitte unter mail@hmedia.de für weitere Informationen.

Konfigurationsdaten und Dateien

Um alle erforderlichen Informationen erfolgreich zusammen zu stellen, fragt das interaktive Setup nach den folgenden Parametern. Bitte halten Sie Ihre Werte bereit und kopieren Sie die erforderlichen Dateien auf den Installationsrechner, bevor Sie den Befehl `printserverctl prepare` ausführen:

Parameter	Beschreibung
deployment.name	Name des Helm-Objekts. Die Vorgabe ist printserver .
deployment.namespace	Ziel-Namensraum in Kubernetes. Standard ist default . <hr/> WENN SIE ANDERE SOFTWARE IM SELBEN KUBERNETES LAUFEN LASSEN UND DIE PRINTSERVER-OBJEKTE TRENNEN MÖCHTEN, VERWENDEN SIE EINEN SPEZIFISCHEN NAMESPACE WIE PRINTSERVER. <hr/>
ingressNginx.install	Möchten Sie, dass das NGINX Ingress-Chart automatisch installiert wird? Antwort: true oder false. Standard ist true . <hr/> DER PRINTSERVER HÄNGT VOM INGRESS-CONTROLLER AB. SETZEN SIE DEEN WERT NUR AUF FALSE, WENN DER INGRESS-CONTROLLER BEREITS BEREITGESTELLT WURDE. <hr/>
ingressNginx.className	Name der Nginx-Ingress-Klasse. Standard ist nginx .
ingressNginx.auth.enabled	Soll die Verbindung zum PrintServer mit HTTP Basic Authentication geschützt werden? Antworten Sie true oder false. Standard ist true .
ingressNginx.auth.benutzername	Benutzername für die HTTP-Basisauthentifizierung. Standard ist printclient_user . Wird nicht verwendet, wenn ingressNginx.auth.enabled auf false gesetzt ist.
ingressNginx.auth.passwort	Kennwort für die HTTP-Basisauthentifizierung. Standard ist printclient_passwd .

	Wird nicht verwendet, wenn ingressNginx.auth.enabled auf false gesetzt ist.
ingressNginx.tls.enabled	Soll die Verbindung zum PrintServer über TLS 1.3 verschlüsselt werden? Antwort: true oder false. Standard ist true .
ingressNginx.tls.secretName	Name des Secret-Objekts zum Speichern des TLS-Zertifikats. Standard ist printserver-tls .
Dateipfad zum TLS-Zertifikat	Absoluter Pfad und Dateiname des TLS-Zertifikats. printserverctl kopiert die Datei nach /etc/hmedia/printserver und verarbeitet sie weiter. Kein Standardwert. Wird nicht verwendet, wenn ingressNginx.tls.enabled auf false gesetzt ist.
Dateipfad zum TLS-Schlüssel	Absoluter Pfad und Dateiname des TLS-Schlüssels. Die Schlüsseldatei muss entschlüsselt und darf NICHT durch ein Passwort geschützt sein. printserverctl kopiert die Datei nach /etc/hmedia/printserver und verarbeitet sie weiter. Kein Standardwert. Wird nicht verwendet, wenn ingressNginx.tls.enabled auf false gesetzt ist.

Für das TLS-Zertifikat: Bei dieser Installation stellen wir die Zertifikatsdaten als Dateien zur Verfügung. Da sie ein Verfallsdatum haben, müssen sie regelmäßig erneuert werden (die maximale Lebensdauer sollte 397 Tage betragen). Die PrintServer-Installation kann auch mit CertManager ausgestattet werden, um die Zertifikate automatisch durch eine Root CA zu erneuern. Die vielen Varianten einer PKI-Infrastruktur sprengen den Rahmen dieses Dokuments. Wenn Sie Hilfe bei der Einrichtung einer automatischen Zertifikatsverwaltung in Ihrem Kubernetes benötigen, kontaktieren Sie uns bitte unter mail@hmedia.de.

Parameter	Beschreibung
printserver.pullPolicy	Wann sollte Docker das Printserver-Image aus der Docker Hub-Registry abrufen? Antworten: Always - Helm ruft das Abbild jedes Mal ab, wenn dieses Diagramm installiert wird. IfNotPresent - Helm ruft das Abbild aus der Registrierung ab, wenn es nicht lokal verfügbar ist. Never - Helm ruft das Abbild überhaupt nicht ab. Es muss bereits lokal vorhanden sein. Die Voreinstellung ist Always .
printserver.dockerTag	Welcher Docker-Tag des PrintServer-Containers soll für die Installation verwendet werden. Docker-Tags beziehen sich in

	<p>der Regel auf die Software-Version oder -Variante. Verfügbare Tags finden Sie unter https://hub.docker.com/r/hmediade/printserver/tags. Standard ist 4.0.0.</p>
printserver.numberOfInstances	<p>Wie viele Instanzen des PrintServers sollen im Kubernetes-Cluster bereitgestellt werden? Die Antwort ist eine ganze Zahl. Standard ist 1.</p>
Dateipfad zur iNEWS-Konfigurationsdatei	<p>Absoluter Pfad und Dateiname von PrintServer inewsconf.xml. printserverctl kopiert die Datei nach /etc/hmedia/printserver und verarbeitet sie weiter. Kein Standardwert.</p> <hr/> <p>OHNE DIESE KONFIGURATIONSDATEI WIRD PRINTSERVER NICHT FUNKTIONIEREN. UM DIE RICHTIGE INEWSCONF.XML ZU ERSTELLEN, LESEN SIE BITTE DAS KAPITEL PRINTSERVER-KONFIGURATION, AUF SEITE 11.</p> <hr/>
Dateipfad zu Ihrer PrintServer-Lizenzdatei	<p>Absoluter Pfad und Dateiname der PrintServer license.lic. printserverctl kopiert die Datei nach /etc/hmedia/printserver und verarbeitet sie weiter. Kein Standardwert.</p> <hr/> <p>OHNE DIESE LIZENZ IST DER PRINTSERVER NICHT VOLL FUNKTIONSFÄHIG. SIE KÖNNEN EINE SOLCHE LIZENZ VON HMEDIA ERHALTEN.</p> <hr/>

Parameter	Beschreibung
printserverInit.pullPolicy	<p>Wann sollte Docker das printserver-init-Image aus der Docker Hub-Registry abrufen? Antworten: Always - Helm ruft das Abbild jedes Mal ab, wenn dieses Diagramm installiert wird. IfNotPresent - Helm ruft das Abbild aus der Registrierung ab, wenn es nicht unmittelbar verfügbar ist. Never - Helm ruft das Abbild überhaupt nicht ab. Es muss bereits lokal vorhanden sein. Die Voreinstellung ist Always.</p>
printserverInit.dockerTag	<p>Welcher Docker-Tag des PrintServer Init-Containers soll für die Installation verwendet werden. Docker-Tags beziehen sich in der Regel auf die Software-Version oder -Variante. Verfügbare Tags finden Sie unter https://hub.docker.com/r/hmediade/printserver-init/tags. Standard ist 1.0.0.</p>

printstyleSource.connectionType	Wie soll der PrintServer seine Printstyle-Dateien beim Starten abrufen? Antworten: scp - PrintServer verbindet sich mit einem Linux-Rechner über SCP. git - PrintServer checkt die benötigten Dateien aus einem Git-Repository über HTTPS aus. Standard ist scp .
printstyleSource.username	Benutzername für den Abruf von printstyles. Standard ist printstyles_user .
printstyleSource.password	Passwort für den Abruf von Druckstilen. Standard ist printstyles_passwd .
printstyleSource.host	IP-Adresse, Hostname oder FQDN der Printstyle-Quelle. Standard ist printstyles_host .
printstyleSource.path	Der lokale Pfad, aus dem die Druckstile gelesen werden. Die Vorgabe ist /path/to/printstyles . Im Falle einer SCP-Verbindung sollte der Pfad ein absoluter Pfad im Dateisystem sein. Im Falle eines Git-Klons ist der Pfad der URL-Teil nach dem Servernamen. Insgesamt sieht die URL wie folgt aus: https://username:password@host/path .

Parameter	Beschreibung
cluster.fqdn	FQDN des Kubernetes-Clusters. Die PrintServer-Webseite und die REST-API werden unter der URL <a href="https://<cluster.fqdn>/printserver">https://<cluster.fqdn>/printserver verfügbar sein. Der Standard ist k8s.cluster.fqdn .
service.port	Der Netzwerk-Port für das Service-Objekt (Verbindung vom Ingress zum Service). Standard ist 8080 .

Erster Aufruf

1. Stellen Sie sicher, dass alle Voraussetzungen erfüllt sind, dass alle Daten verfügbar sind und dass alle Konfigurationsdateien auf dem Installationsrechner erstellt/kopiert wurden.
2. Rufen Sie das `printserverctl`-Skript von <https://hmedia.de/helm/printserverctl> ab. Speichern Sie das Skript unter dem Namen `printserverctl` in einem beliebigen Ordner Ihres Kubernetes-Kontrollrechners (von dem aus Sie `kubectl` und `helm` ausführen), und machen Sie

die Datei ausführbar mit dem Befehl
`chmod a+x <Pfad>/printserverctl.`

DAS PRINTSERVERCTL-SKRIPT MUSS ZUGRIFF AUF DIE RICHTIGE UMGEBUNG HABEN. NORMALERWEISE IST DAS DIEJENIGE DES AKTUELLEN BENUTZER. WENN DAS SKRIPT KUBERNETES NICHT ERREICHEN KANN, FÜHREN SIE ES MIT `sudo -E printserverctl` AUS.

3. Führen Sie das Skript mit
`sudo -E <Pfad>/printserverctl prepare,`
 und geben Sie alle angeforderten Daten ein. Sie können das Skript jederzeit sicher mit `Strg+C` unterbrechen.
4. (Optional) Überprüfen Sie die Konfiguration mit den folgenden Befehlen
`sudo -E <Pfad>/printserverctl review,` oder
`sudo -E <Pfad>/printserverctl dryrun.`
5. Stellen Sie schließlich den PrintServer mit der angegebenen Konfiguration bereit, indem Sie `sudo -E <Pfad>/printserverctl deploy` ausführen.

Änderung

Sie können das Bereitstellungsskript jederzeit erneut ausführen, um die Konfiguration anzupassen oder zu ändern, z. B. wenn Sie die TLS-Zertifikatsdateien aktualisieren oder die Datei `inewsconf.xml` ändern.

WENN SIE NUR DIE EINSTELLUNGSDATEIEN AUF DEM LINUX-HOST AKTUALISIEREN, BETRIFFT DAS NICHT DIE LAUFENDE PRINTSERVER-INSTANZ. SIE MÜSSEN DEN BEFEHL `printserverctl deploy` ERNEUT AUSFÜHREN.

1. Führen Sie das Skript mit
`sudo -E <Pfad>/printserverctl prepare,`
 und aktualisieren/ändern Sie alle angeforderten Daten. Das Skript zeigt alle aktuellen Einstellungen als Standardwerte an. Sie können diese Werte mit `Enter` überspringen und müssen nicht jede Einstellung erneut eingeben. Sie können das Skript jederzeit sicher mit `Strg+C` unterbrechen.
2. (Optional) Überprüfen Sie die Konfiguration mit den folgenden Befehlen
`sudo -E <Pfad>/printserverctl review,` oder
`sudo -E <Pfad>/printserverctl dryrun.`
3. Stellen Sie den PrintServer mit der angegebenen Konfiguration bereit, indem Sie `sudo -E <Pfad>/printserverctl deploy.`
 Das Skript teilt mit, dass es eine laufende Instanz gefunden hat und diese Bereitstellung aktualisieren wird.

Löschen

Wenn Sie eine laufende Instanz des PrintServers deinstallieren wollen, führen Sie den Befehl
`sudo -E <Pfad>/printserverctl remove,`

und beantworten Sie die entsprechenden Bestätigungsfragen für die Software selbst und den Namensraum.

Dadurch werden alle Artefakte der PrintServer-Bereitstellung aus Ihrem Kubernetes-Cluster entfernt.

Ändern von Druckstilen oder iNEWSSystems.txt

Der PrintServer als containerisierte Anwendung benötigt Zugriff auf einen Ordner, der alle Druckstile enthält. Dieser Ordner muss auch die Datei `iNEWSSystems.txt` enthalten. Außerdem kann er Dateien oder Unterordner/Dateien enthalten, auf die die Druckstile verweisen (wie Symbole oder andere Grafiken).

Das Format der Printstyles und der iNewsSystems.txt ist genau dasselbe wie bei der nativen Installation und dort ausführlich beschrieben (Kapitel PrintStyle-Konfiguration, auf Seite 13).

Wenn Sie `printstyle`-Dateien oder die Datei `iNEWSSystems.txt` ändern, müssen Sie dies auf der `printstyle`-Quelle tun - entweder auf einem externen Linux-Rechner oder in einem Git-Repository.

DIE VERWENDUNG EINES GIT-REPOSITORYS FÜR DIE PRINTSTYLE-DATEIEN HAT DEN ZUSÄTZLICHEN VORTEIL DER VERSIONSKONTROLLE. SO KÖNNEN SIE ÄLTERE VERSIONEN ÜBERPRÜFEN ODER UNERWÜNSCHTE ÄNDERUNGEN NAHTLOS RÜCKGÄNGIG MACHEN.

Um die Änderung zu aktivieren, löschen Sie einfach den `printserver`-Pod mit dem Befehl `kubectl delete pod printserver -n <namespace>`, wobei `namespace` der Wert der **`deployment.namespace`** Einstellung ist. Wenn der Namespace `default` ist, können Sie den Parameter `-n` weglassen.

Kubernetes erstellt sofort einen neuen Pod, der mit dem neuen Satz von `printstyle`-Dateien und/oder `iNewsSystems.txt` initialisiert wird.

BEDIENUNG

Für den PrintServer gibt es keine eigenen Kommandos. Sie können die Statusseite des PrintServers über <https://<cluster.fqdn>/printserver> oder <http://<cluster.fqdn>/printserver> einsehen, je nachdem, ob TLS aktiviert ist oder nicht.

Siehe Kapitel Status-Seite, auf Seite 15 für weitere Informationen.

Der PrintServer hat eine Liveness-Probe konfiguriert. Das heißt, Kubernetes prüft regelmäßig (alle 10 Sekunden), ob die PrintServer-REST-API noch erreichbar ist. Wenn diese Prüfung dreimal fehlschlägt, wird Kubernetes den PrintServer-Pod löschen und neu erstellen.

Ein PrintServer-Pod kann jederzeit gelöscht werden. Kubernetes wird sofort automatisch einen neuen Pod erstellen.

BEACHTEN SIE, DASS ALLE PROTOKOLLINFORMATIONEN DES AKTUELLEN PODS VERLOREN GEHEN, WENN DER POD BEENDET WIRD.

PROTOKOLLDATEN

Innerhalb eines PrintServer-Pods befinden sich die gleichen Log-Dateien wie bei der nativen Installation. Sie können die gesamten Pod-Protokolle mit dem folgenden Befehl anzeigen `kubectl logs printserver -n <Namespace>`, wobei namespace der Wert der **deployment.namespace**-Einstellung ist. Wenn der Namespace default ist, können Sie diesen Parameter weglassen.

Alternativ können Sie den Kubernetes-Cluster mit einem grafischen Admin-Tool wie dem Kubernetes Dashboard oder mit einer Lösung zur Aggregation von Metriken und einem grafischen Dashboard (Prometheus und Grafana) ausstatten. Derzeit exportiert der PrintServer selbst keine Metriken. Der Nginx-Ingress-Controller hingegen schon.

Wenn Sie an weiteren Erweiterungen für Ihren Kubernetes-Cluster interessiert sind, kontaktieren Sie uns bitte unter mail@hmedia.de.

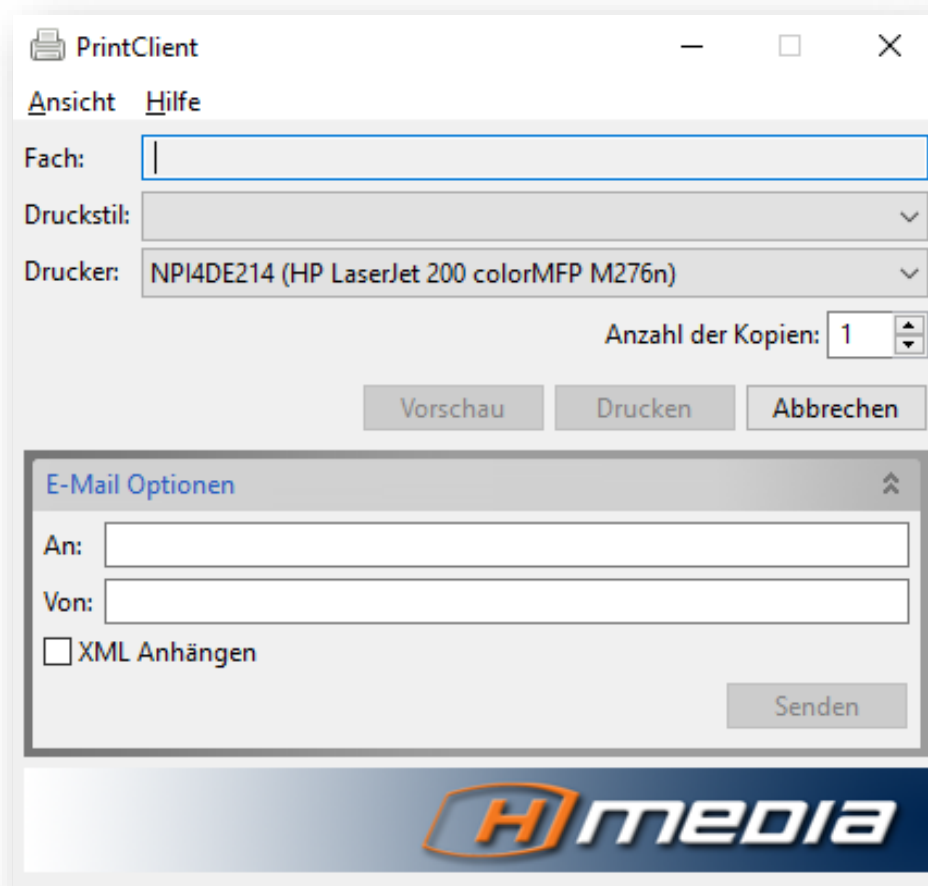
DER PRINTCLIENT

ALLGEMEINES

Der PrintClient ist ein Java-Programm. Für eine bessere Nutzerfreundlichkeit versteht ein .NET-Wrapper die Software mit einem gewöhnliches .EXE-Programm.

Der PrintClient benötigt daher beide Laufzeitumgebungen - .NET und JRE.

Der PrintClient hat keine spezifischen Hardware-Anforderungen. Er kann mit einer grafischen Benutzeroberfläche, aber auch rein über Kommandozeilenparameter bedient werden.



VORAUSSETZUNGEN

Betriebssystem

Der PrintClient läuft zusammen mit der iNEWS Workstation-Software auf einem Windows-Desktop-Betriebssystem.

Sowohl Windows 10 als auch Windows 11 wurden getestet und qualifiziert.

.NET-Framework

Die .NET CLR Version 4.0 oder höher ist erforderlich. Das .NET-Framework Version 4.5 oder höher ist erforderlich. Sie können das .NET-Framework von <https://www.microsoft.com/downloads> herunterladen. Auf dieser Webseite finden Sie auch weitere Details zur Installation der .NET-Umgebung.

DAS PRINTCLIENT-INSTALLATIONSPROGRAMM IST IN DER LAGE, DAS BENÖTIGTE .NET FRAMEWORK WÄHREND DER INSTALLATIONSPROZEDUR ALS VORAUSSETZUNG ABZURUFEN UND ZU INSTALLIEREN.

Java

Für den PrintClient gelten hinsichtlich der Java-Installation die gleichen Regeln wie für den PrintServer (siehe oben). Die minimal unterstützte Version ist Java 11. Hmedia empfiehlt die Verwendung des OpenJDK, das auf <https://jdk.java.net> zum Download zur Verfügung steht.

Eingebettete JRE

In dieser Version kommt der PrintClient mit einem eingebetteten OpenJDK. Damit ist der Hmedia PrintClient unabhängig von einer ansonsten installierten JRE im Betriebssystem.

DIE EINGEBETTETE JAVA RUNTIME RUFT KEINE SICHERHEITSUPDATES AB UND MUSS MANUELL GEPATCHT WERDEN.

PrintClient Version 4.0 verwendet das OpenJDK 21.0.2. Es befindet sich im Unterordner **Java** im PrintClient-Stammverzeichnis (normalerweise **C:\Programme\Hmedia\PrintClient**).

System Standard JRE

Durch das Löschen des Unterordners **Java** im PrintClient-Hauptverzeichnis wird der PrintClient gezwungen, mit der Standard-JRE des Systems zu arbeiten. Stellen Sie in diesem Fall sicher, dass der Java-Befehl von jedem Benutzer, der den PrintClient ausführen muss, ausgeführt werden kann.

STELLEN SIE SICHER, DASS DER ORDNER **BIN** IHRER JAVA-INSTALLATION IN DER UMGEBUNGSVARIABLEN PATH ENTHALTEN IST. DER PRINTCLIENT-BENUTZER MUSS IN DER LAGE SEIN, DEN JAVA-BEFEHL IN DER BEFEHLSZEILE AUSZUFÜHREN.

Mit **java -version** können Sie überprüfen, ob Java generell auf dem Computer verfügbar ist und welche Version der JRE installiert ist.

Adobe Acrobat Reader

Der PrintClient verwendet Adobe Acrobat für die Vorschau der gerenderten Ausgaben. Die Installation des Acrobat Readers ist nicht Teil des PrintClient-Installationsverfahrens und muss separat durchgeführt werden. Sie können den Acrobat Reader von der Webseite <https://www.adobe.com/acrobat/pdf-reader.html> herunterladen und installieren.

Der PrintClient bietet einen Konfigurationsparameter in der Datei **<PRINTCLIENT_HOME>/printclient.properties**. (Siehe Kapitel PrintClient-Eigenschaften, auf Seite 33)

```
viewer.acrobat.exe=C:\\Program Files (x86)\\Adobe\\Acrobat DC\\Acrobat\\Acrobat.exe
```


Dieser Wert enthält den Programmpfad des Acrobat Reader. Sie brauchen den Pfad nicht in Anführungszeichen zu setzen, wenn er Leerzeichen enthält. Allerdings muss jeder Backslash mit einem weiteren Backslash escaped werden.

Wenn dieser Konfigurationswert nicht verfügbar oder leer ist, wertet der PrintClient die Zuordnung der ausführbaren Datei des Adobe Acrobat Reader aus dem Windows-Registrierungsschlüssel

`HKLM\Software\Classes\acrobat\shell\open\command` aus.

EINRICHTUNG

Stellen Sie sicher, dass alle Voraussetzungen (siehe vorheriges Kapitel) erfüllt sind.

Das PrintManager-Installationsmedium enthält die Setup-Datei **PrintClient Java Setup.msi** im Unterordner **PrintClient**.

Führen Sie dieses Programm mit administrativen Rechten aus. Befolgen Sie die Anweisungen. Das Setup-Tool installiert den PrintClient in dem gewählten Ordner und erstellt Verknüpfungen im Startmenü und auf dem Desktop.



Der Ordner, in dem die PrintClient-Software installiert ist, wird als **<PRINTCLIENT_HOME>** bezeichnet.

KONFIGURATION

PrintServer

Sie richten alle PrintServer, die vom PrintClient verwendet werden, in der Datei **<PRINTCLIENT_HOME>\printserver.xml** ein. Diese XML-Datei ist eine Standard-Java-Bean-Konfiguration.

Server-Spezifikation

Jeder PrintServer ist als eine Bean der Klasse

de.hmedia.printmanager.printclient.inews.INewsPrintServer definiert. Sie gibt die RESTful URL des jeweiligen Servers als Untererelement **des Konstruktors-Args** an, wie in den folgenden Beispielen.

Der PrintServer wird über das XML-Attribut **name** identifiziert. Sein Inhalt ist frei wählbar, muss aber in der gesamten Konfiguration eindeutig sein - in den Beispielen unten: **server1** und **server2**.

Zusätzlich müssen die PrintServer mit ihren Namen in das serverList-Element aufgenommen werden.

Verschlüsselung

Der PrintManager 4.0 ermöglicht eine sichere und verschlüsselte Kommunikation zwischen PrintClient und PrintServer. Auf der Serverseite wird diese Funktion normalerweise durch einen zusätzlichen Reverse-Proxy (wie Nginx) bereitgestellt.

Der PrintClient verwendet denselben Schlüsselspeicher wie das Betriebssystem. Je nach vorhandener PKI-Infrastruktur muss die Schlüsselkette der Root CA, die das PrintServer-Zertifikat signiert hat, im Speicher "Vertrauenswürdige Stammzertifizierungsstellen" in den Internetoptionen des Windows-Computers gespeichert werden.

Authentifizierung

Wenn die PrintServer REST API mit Basic Http Authentication geschützt ist, muss die Bean XML zusätzlich zwei Eigenschaftswerte für user und password enthalten. Der Wert für den Benutzer ist Klartext. Der Passwort-Wert muss base64-kodiert sein.

Zusammengefasst sieht die PrintServer-Konfigurationsdatei wie folgt aus:

Beispiel 1

Eine einfache PrintServer-Konfiguration mit nur einem angegebenen PrintServer. Der Zugriff auf die PrintServer-API erfolgt über HTTPS. Die Authentifizierung ist aktiviert.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://www.springframework.org/schema/util
    http://www.springframework.org/schema/util/spring-util-2.0.xsd">

  <bean name="server1" class="de.hmedia.printmanager.printclient.inews.INewsPrintServer">
    <constructor-arg value="https://toolserver/printserver"/>
    <property name="user" value="printclient">
    <property name="password" value="cHJpbmRjbGllbnQ=">
  </bean>

  <util:list id="serverList" list-class="java.util.ArrayList">
    <ref bean="server1"/>
  </util:list>
</beans>
```

Beispiel 2

In dieser Konfiguration verbindet sich der PrintClient mit zwei PrintServern. Diese können entweder für verschiedene iNEWS-Systeme arbeiten, Redundanz für dasselbe System bieten oder beides. Die Verschlüsselung und Authentifizierung ist nur für Server2 aktiviert.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:util="http://www.springframework.org/schema/util"
```

```

xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
    http://www.springframework.org/schema/util
    http://www.springframework.org/schema/util/spring-util-2.0.xsd"

<bean name="server1" class="de.hmedia.printmanager.printclient.inews.INewsPrintServer">
    <constructor-arg value="http://toolserver:8080/printserver"/>
</bean>

<bean name="server2" class="de.hmedia.printmanager.printclient.inews.INewsPrintServer">
    <constructor-arg value="https://toolserver2.buero.hmedia.de/printserver"/>
    <property name="user" value="printclient">
    <property name="password" value="cHJpbmRjbGllbnQ=">
</bean>

<util:list id="serverList" list-class="java.util.ArrayList">
    <ref bean="server1"/>
    <ref bean="server2"/>
</util:list>
</beans>

```

PrintClient-Eigenschaften

Die Datei **<PRINTCLIENT_HOME>/printclient.properties** enthält zusätzliche Parameter zur Steuerung des PrintClient-Verhaltens:

- **Localdb.encoding=UTF-16LE** - Kodierung der lokalen iNEWS-Datenbank. Die Version 4.0 des PrintManagers unterstützt das Drucken der lokalen Datenbank nicht. Daher wird diese Einstellung ignoriert.
- **SMTP-Parameter** - Der PrintClient ist in der Lage, das gerenderte PDF direkt als E-Mail zu versenden. Für diese Funktion sind drei wesentliche Einstellungen erforderlich:
 - **mail.smtp.host** - Netzwerkname des SMTP-Servers.
 - **mail.smtp.user** - Benutzer zur Authentifizierung gegenüber dem SMTP-Server.
 - **mail.smtp.password** - Passwort für den SMTP-Benutzer.

DIE SMTP-KONFIGURATION KANN UM WEITERE PARAMETER ERWEITERT WERDEN. WEITERE DETAILS FINDEN SIE AUF DER WEBSEITE [HTTPS://JAVAEE.GITHUB.IO/JAVAMAIL/DOCS/API/COM/SUN/MAIL/SMTP/PACKAGE-SUMMARY.HTML](https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html).

Im PrintClient selbst kann der Benutzer eine Absenderadresse angeben. Der SMTP-Server muss so konfiguriert sein, dass er solche E-Mails weiterleitet, die eine andere Absenderadresse als die Authentifizierung haben.

Beispiel

```

localdb.encoding=UTF-16LE

# SMTP settings used by the PrintClient email function.
mail.smtp.host=smtplib.hmedia.de
mail.smtp.user=mail@hmedia.de

```

mail.smtp.password=secret

BEDIENUNG

PrintClient im grafischen Modus

Wird das PrintClient-Desktop-Symbol angeklickt oder der Befehl PrintClient.exe ohne Parameter aufgerufen, öffnet sich die grafische Benutzeroberfläche (GUI) des PrintClient.

Das PrintClient-Fenster bleibt so lange im Vordergrund, wie es auf Benutzereingaben wartet.

Ein typisches Verfahren umfasst die folgenden Schritte:

1. Der Benutzer zieht den Druckbereich (ausgewählte Stories oder eine bestimmte iNEWS-Queue) per Drag&Drop auf das PrintClient-Fenster. Das Drop-Ziel ist die gesamte PrintClient-App.
2. Als Nächstes ruft der PrintClient die Liste der verfügbaren Druckstile vom PrintServer ab und filtert diese Liste je nach dem ausgewählten Druckbereich.
3. Der Benutzer wählt den Druckstil aus und legt fest, was als nächstes geschieht:
 - a. Die Schaltfläche Vorschau öffnet die gerenderte PDF-Datei in Acrobat Reader.
 - b. Mit der Schaltfläche Drucken wird das Dokument auf dem ausgewählten Drucker ausgedruckt.
 - c. Die Schaltfläche Abbrechen ist immer verfügbar. Sie stoppt den aktuellen Vorgang und schließt den PrintClient.
 - d. Im E-Mail-Panel kann der Benutzer Absender und Empfänger festlegen und die E-Mail verschicken. Das E-Mail-Panel ist nur verfügbar, wenn E-Mail für den PrintClient konfiguriert ist.

Solange der PrintManager Daten verarbeitet und der PrintClient ein Dokument im Acrobat Reader anzeigt, ist das PrintClient-Fenster minimiert und nicht sichtbar.

Der PrintClient beendet sich nach dem erfolgreichen Senden einer E-Mail oder dem Drucken eines Dokuments auf einem Drucker.

PrintClient-Befehlszeilenparameter

Die **PrintClient Java.exe** kann mit Parametern aufgerufen werden, um bestimmte Daten an den PrintClient zu übergeben.

Wenn alle notwendigen Daten (iNEWS-System, iNEWS-Warteschlange, Druckstil) in den Parametern enthalten sind, wird der PrintClient den Prozess sofort starten, ohne die grafische Benutzeroberfläche anzuzeigen.

Der PrintClient startet die grafische Benutzeroberfläche mit vorgegebenen Parametern als Vorauswahl, wenn die Parameter nicht ausreichen, um den Druckvorgang abzuschließen.

Parameter	Beschreibung
-s [iNEWS-System]	Das iNEWS-System, wie im PrintServer konfiguriert (typischerweise der iNEWS-DB-Name). Beispiel: -s NRCS
-q [Pfad einer Queue]	Der Pfad für die zu druckende Warteschlange. Beispiel: -q NEWS.SHOW.1400.RUNDOWN
-t [Printstyle]	Der Druckstil, wie die Quelldaten dargestellt werden sollen. Hier ist es der Dateiname der Druckstil-Datei, nicht der Anzeigename im PrintClient! Beispiel: -t rundown_body.xsl
-c	Wenn dieser Parameter gesetzt ist, versucht der PrintClient, den iNEWS-System- und Warteschlangenpfad aus der Windows-Zwischenablage zu lesen. In diesem Fall werden die Parameter -s und -q ignoriert. Die Daten in der Zwischenablage müssen folgendem Muster entsprechen (einschließlich der eckigen Klammern): -c [iNEWS-Systemname]Queue Name
-p [Drucker]	Gibt das Druckergerät an. Er muss in doppelte Anführungszeichen gesetzt werden, wenn der Druckername Leerzeichen enthält. Ohne diesen Parameter nimmt der PrintClient automatisch den Windows-Standarddrucker. Beispiel: -p "Microsoft XPS Document Writer"
-show	Zeigen Sie die grafische Benutzeroberfläche an, auch wenn die Parameter ausreichen, um einen Druckvorgang durchzuführen.
-o	Wenn dieser Parameter gesetzt ist, öffnet der PrintClient das gerenderte Dokument in Adobe Acrobat und beendet sich.
-from [Email-Adresse]	Gibt die Absender-E-Mail-Adresse an. Beispiel: -from andreas@hmedia.de
-to [Email-Adresse]	Gibt die Empfänger-E-Mail-Adresse an. Beispiel: -to andreas@hmedia.de
-type [Datentyp]	Dieser Parameter steuert, welche Daten per E-Mail versendet werden: <ul style="list-style-type: none"> • PDF_REQUEST - die PDF-Datei wird gesendet. Dies ist die Standardeinstellung. • PDF_AND_XML_REQUEST - das PDF und zusätzlich die Rohdaten XML werden gesendet. Dies ist nützlich für technische Implementierungen oder zur Fehlerbehebung.

Beispiel 1

Sie wollen die Queue NEWS.1200.RUNDOWN des iNEWS-Systems NRCS mit dem Druckstil rundown.xsl drucken. Dann muss Ihre Befehlszeile wie folgt aussehen:

```
PrintClient Java.exe -s NRCS - q NEWS.1200.RUDOWN -t rundown.xml
```

Beispiel 2

Sie möchten mit denselben Quelldaten wie in Beispiel 1 drucken, aber über den Netzwerkdrucker [\\buero.hmedia.de\hp_laser](http://buero.hmedia.de/hp_laser):

```
PrintClient Java.exe -s NRCS - q NEWS.1200.RUDOWN -t rundown.xml -p \\buero.hmedia.de\hp_laser
```

Beispiel 3

Sie wollen wieder mit den gleichen Daten wie in Beispiel 1 drucken, aber diesmal soll die Quelle aus der Windows-Zwischenablage stammen. Sie erstellen die Daten der Zwischenablage als normalen Text mit dem Muster:

```
[iNEWS-Systemname]Name der Queue
```

Unter Verwendung der Quelldaten aus Beispiel 1 sehen die Daten in der Zwischenablage dann wie folgt aus:

```
[NRCS]NEWS.1200.RUDOWN
```

Schließlich sieht die Befehlszeile dann so aus:

```
PrintClient Java.exe -c -t rundown.xml
```

UM DEN INEWS-SYSTEMNAMEN UND DIE QUEUE IN DIE ZWISCHENABLAGE ZU KOPIEREN, KÖNNEN SIE AUCH EIN INEWS-MAKRO VERWENDEN. IN INEWS KANN DIESES MAKRO EINER SCHALTFLÄCHE IN DER SYMBOLLEISTE ODER EINEM TASTATURKÜRZEL ZUGEWIESEN WERDEN.

Ein-Klick-Modus

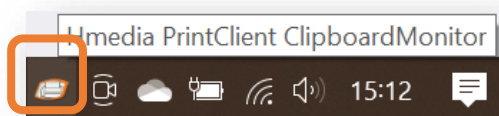
Für einen typischen Druckvorgang muss der Benutzer in der Regel zwei Schritte ausführen, auch wenn er die Befehlszeilenparameter verwendet. Diese sind:

1. Kopieren Sie die Queue in die Zwischenablage. Dieser Vorgang könnte zum Beispiel mit einem iNEWS-Tastenkürzel durchgeführt werden.
2. Starten Sie das Rendern/Drucken mit einem bestimmten Druckstil. Dieser Vorgang wird in der Regel über eine Schaltfläche in der iNEWS-Symbolleiste ausgeführt.

Mit dem PrintClientClipboardMonitor - einer Windows System Tray Anwendung - können diese Schritte in einem einzigen Vorgang zusammengefasst werden.

Der PrintClientClipboardMonitor läuft zu diesem Zweck permanent im System Tray. Wenn Daten in einem bestimmten Format in der Windows-Zwischenablage ankommen, startet der PrintClientClipboardMonitor den PrintClient mit allen benötigten Parametern.

Der PrintClientClipboardMonitor wird automatisch in den Autostart-Ordner des Benutzers installiert.



Die Daten in der Zwischenablage müssen dem Muster folgen:

```
HMEDIA_PRINT[iNEWS-Systemname]Queue Name <Optionen>
```

Um einen Druckvorgang für das obige Beispiel zu initiieren, muss der folgende Text in die Zwischenablage kopiert werden:

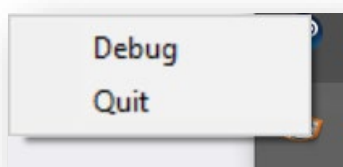
```
HMEDIA_PRINT[NRCS]NEWS.1200.RUNDOWN -t rundown.xml
```

Sie können diesen Text zum Beispiel mit dem folgenden iNEWS-Makro in die Zwischenablage bringen (englischer iNEWS-Client). Der Benutzer muss sich in der Queue **NEWS.1200.RUNDOWN** befinden (die aktuelle Queue wird mit diesem Makro gedruckt).

```
{alt {right}{right}{left}}{alt wo}{home}HMEDIA_PRINT[NRCS]{end}{space}-t rundown.xml {shift {home}} {ctrl c} {esc}
```

Optionen

Das Programm in der Taskleiste bietet zwei Optionen über das Kontextmenü (rechter Mausklick).



Mit **Debug** aktiviert der ClipboardMonitor den Debug-Modus und zeigt die resultierende ausführbare Kommandozeile für einen identifizierten Druckauftrag an. Der Druckbefehl wird dann als Windows-Benachrichtigung angezeigt. Sie können den Debug-Modus mit einem weiteren Klick auf diesen Menüeintrag deaktivieren.

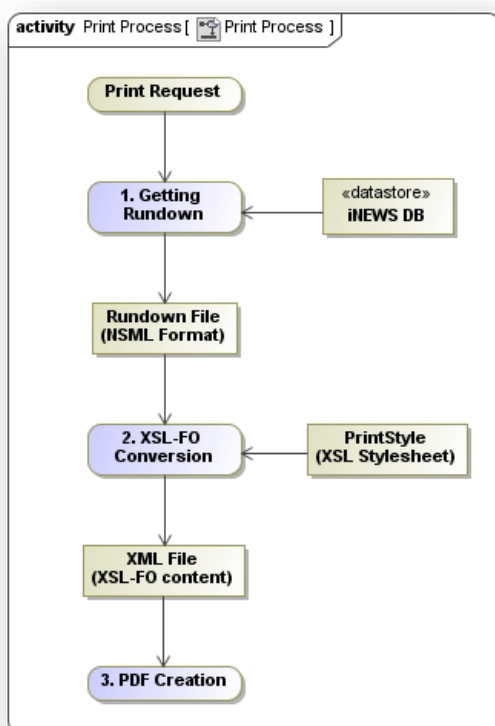


Mit **Quit** kann der Benutzer den PrintClientClipboardMonitor für diese Windows-Sitzung deaktivieren.

DRUCKSTILE

PDF-ERSTELLUNGSPROZESS

Um eigene PrintStyles zu erstellen, ist es notwendig zu verstehen, wie Daten aus einer iNEWS-Queue oder von individuellen Stories in ein druckfähiges Dokument umgewandelt werden. Abbildung 1 zeigt diesen Prozess, der aus drei Schritten besteht:



1. Nach Erhalt eines Druckauftrags holt der PrintServer die kompletten Daten aus der iNEWS-Datenbank und schreibt sie in eine NSML-Datei. Während frühere PrintManager-Versionen die NSML-Version 2 verwendeten, arbeitet der PrintManager seit Version 3.2 mit dem XML-konformen NSML 3.
2. Ein XSLT-Prozessor kann diese XML-Datei und ein XSL-Stylesheet (den PrintStyle) verwenden, um eine XSL-FO-Datei zu erstellen. Die XSL-FO-Datei enthält die Rundown-Daten zusammen mit Formatierungsinformationen, den sogenannten Formatierungsobjekten (FO).
3. Der Apache Formatting Objects Processor (FOP) wird dann verwendet, um die XSL-FO-Datei in eine PDF-Datei umzuwandeln, die schließlich heruntergeladen und mit dem PrintClient/Adobe Acrobat Reader gedruckt oder angezeigt wird.

ERSTELLEN VON DRUCKSTILEN

Eigene Druckvorlagen zu erstellen bedeutet also, eine XSL-Stylesheet-Datei zu schreiben, die vom XSLT-Prozessor in Schritt 2 (siehe oben) verwendet werden kann.

Daher sind fundierte Kenntnisse über XML, XSL und XSL-FO erforderlich, um eigene Druckstile zu entwickeln. Diese Themen werden in dieser Dokumentation nicht behandelt, da sie nicht PrintManager-spezifisch sind und es viele Websites und Bücher gibt, die sich damit beschäftigen. Weitere Informationen zu diesen drei Themen finden Sie auf einer der Websites mit Tutorials des W3C.

- XML - <https://www.w3schools.com/xml/>
- XSL - https://www.w3schools.com/xml/xsl_intro.asp
- XPath - https://www.w3schools.com/xml/xpath_intro.asp
- XSL-FO - <https://w3schools.sinsixx.com/xslfo/default.asp.htm>

Der PrintManager wird mit einigen Standard-Druckstilen ausgeliefert. Diese können Basis für eigene Erweiterungen und Implementierungen verwendet werden. Der Standard-Druckstile-Ordner besteht aus:



- hmedia_scripts.xml - Keine Druckvorlage, sondern verschiedene Funktionen für Zeitberechnungen usw.
- production_cues.xml - Druckt Stories mit zwei Spalten für Produktionsanweisungen und Text der Story.
- rundown_body.xml - Druckt eine Queue als Rundown-Tabelle, einschließlich des Textkörpers.
- rundown_cue.xml - Druckt eine Queue als Rundown-Tabelle mit den Produktionsanweisungen der Stories.

Den Inhalt dieser drei Druckstile finden Sie im nachstehenden Anhang.

ANHANG

QUELLCODE VON DEFAULT PRINTSTYLES

production_cues.xsl

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  xmlns:hmedia="xalan://hmedia.extensions"
  xmlns:java="http://xml.apache.org/xalan/java"
  exclude-result-prefixes="java hmedia" version="1.0">
  <xsl:import href="hmedia_scripts.xsl" />

  <xsl:param name="SYSTEM"/>
  <xsl:param name="QUEUE"/>

  <xsl:template match="/NSMLRoot">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master master-name="alleSeiten"
          page-height="29.7cm" page-width="21cm" margin-top="1cm"
          margin-bottom="1cm" margin-left="1cm" margin-right="1cm">
          <fo:region-body margin-top="0.8cm"
            margin-bottom="1.5cm" />
          <fo:region-before extent="0.8cm" />
          <fo:region-after extent="0.8cm" />
        </fo:simple-page-master>
      </fo:layout-master-set>
      <fo:page-sequence master-reference="alleSeiten">
        <fo:static-content flow-name="xsl-region-after">
          <fo:block font-size="8pt">
            <fo:table table-layout="fixed">
              <fo:table-column column-width="5cm"
                column-number="1" />
              <fo:table-column column-width="9cm"
                column-number="2" />
              <fo:table-column column-width="5cm"
                column-number="3" />
              <fo:table-body>
                <fo:table-row>
                  <fo:table-cell text-align="left" padding-top="5pt">
                    <fo:block>
                      Printed
                      <xsl:value-of
                        select="java:format(java:java.text.SimpleDateFormat.new('d-MM-yyyy, HH:mm:ss'),
                          java:java.util.Date.new())" />
                    </fo:block>
                  </fo:table-cell>
                  <fo:table-cell text-align="center">
                    <fo:block>
                      <fo:external-graphic
                        src="stylesheets/icons/hmedia.jpg" width="3cm" />
                    </fo:block>
                  </fo:table-cell>
                  <fo:table-cell text-align="right" padding-top="5pt">
                    <fo:block>
                      Page
                      <fo:page-number />
                      of
                      <fo:page-number-citation ref-id="last-page"/>
                    </fo:block>
                  </fo:table-cell>
                </fo:table-row>
              </fo:table-body>
            </fo:table>
          </fo:block>
        </fo:static-content>
        <fo:flow flow-name="xsl-region-body" font-size="10pt">

```

```

<xsl:variable name="t2" select="java:put($props, 'end', -1)"/>
<xsl:apply-templates select="nsml" mode="endtime" />
<xsl:call-template name="banner"/>
<fo:table table-layout="fixed" start-indent="2pt"
end-indent="2pt" space-before="0.5cm" hyphenate="false">
  <fo:table-column column-number="1"
column-width="9cm" />
  <fo:table-column column-number="2"
column-width="9cm" />
  <fo:table-header>
    <fo:table-row text-align="center"
font-weight="bold" background-color="gray">
      <fo:table-cell
border="solid thin black">
        <fo:block space-after="2pt"
space-before="2pt">
          Text
        </fo:block>
      </fo:table-cell>
      <fo:table-cell
border="solid thin black">
        <fo:block space-after="2pt"
space-before="2pt">
          Cues
        </fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-header>
  <fo:table-body>
    <xsl:apply-templates select="nsml" />
  </fo:table-body>
</fo:table>
<fo:block id="last-page"></fo:block>
</fo:flow>

</fo:page-sequence>
</fo:root>
</xsl:template>

<xsl:template match="nsml">
  <xsl:variable name="video-id" select="fields/string[@id='video-id']" />
  <xsl:choose>
    <xsl:when test="head/meta/@break = 1">
      <xsl:call-template name="inhalt">
        <xsl:with-param name="rowbackground" select="'lightgray'" />
        <xsl:with-param name="formatbackground" select="'lightgray'" />
      </xsl:call-template>
    </xsl:when>
    <xsl:when
test="string-length(normalize-space($video-id) ) > 0">
      <xsl:call-template name="inhalt">
        <xsl:with-param name="rowbackground" select="'white'" />
        <xsl:with-param name="formatbackground" select="'#FFFF99'" />
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="inhalt">
        <xsl:with-param name="rowbackground" select="'white'" />
        <xsl:with-param name="formatbackground" select="'white'" />
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

<xsl:template name="inhalt">
  <xsl:param name="rowbackground" />
  <xsl:param name="formatbackground" />
  <xsl:variable name="total-time" select="fields/duration[@id='total-time']" />
  <fo:table-row background-color="{ $rowbackground }">
    <fo:table-cell border="solid thin black">
      <fo:block space-after="15pt" space-before="2pt">
        <xsl:apply-templates select="body">
          </xsl:apply-templates>
        </fo:block>
      </fo:table-cell>
    <fo:table-cell border="solid thin black">
      <fo:block space-after="15pt" space-before="2pt" font-weight="bold">
        <xsl:call-template name="inserts">

```

```

        <!-- uncomment the next line if you don't want to print the name of
the insert -->
        <!-- <xsl:with-param name="printInsertName" select="false()"/> -->
        <!-- uncomment the next line if you don't want to print production
cues between inserts -->
        <!-- <xsl:with-param name="printProductionCues" select="false()"/> -->
        </xsl:call-template>
    </fo:block>
</fo:table-cell>
</fo:table-row>
</fo:table>
</xsl:template>

<xsl:template name="banner">
    <fo:table table-layout="fixed" start-indent="4pt" end-indent="4pt">
        <fo:table-column column-number="1" column-width="19cm" />
        <fo:table-body font-size="14pt">
            <fo:table-row>
                <fo:table-cell text-align="center">
                    <fo:block space-before="10pt">
                        <xsl:text>Text and Production Cues</xsl:text>
                    </fo:block>
                    <fo:block font-weight="bold" space-before="5pt" space-after="10pt">
                        <xsl:value-of select="$SYSTEM"/> - <xsl:value-of select="$QUEUE"/>
                    </fo:block>
                </fo:table-cell>
            </fo:table-row>
        </fo:table-body>
    </fo:table>
</xsl:template>
</xsl:stylesheet>

```

rundown_body.xsl

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format"
    xmlns:hmedia="xalan://hmedia.extensions"
    xmlns:java="http://xml.apache.org/xalan/java"
    exclude-result-prefixes="java hmedia" version="1.0">
    <xsl:import href="hmedia_scripts.xsl" />

    <xsl:param name="SYSTEM"/>
    <xsl:param name="QUEUE"/>

    <xsl:template match="/NSMLRoot">
        <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
            <fo:layout-master-set>
                <fo:simple-page-master master-name="alleSeiten"
                    page-height="29.7cm" page-width="21cm" margin-top="1cm"
                    margin-bottom="1cm" margin-left="1cm" margin-right="1cm">
                    <fo:region-body margin-top="0.8cm"
                        margin-bottom="1.5cm" />
                    <fo:region-before extent="0.8cm" />
                    <fo:region-after extent="0.8cm" />
                </fo:simple-page-master>
            </fo:layout-master-set>
            <fo:page-sequence master-reference="alleSeiten">
                <fo:static-content flow-name="xsl-region-after">
                    <fo:block font-size="8pt">
                        <fo:table table-layout="fixed">
                            <fo:table-column column-width="5cm"
                                column-number="1" />
                            <fo:table-column column-width="9cm"
                                column-number="2" />
                            <fo:table-column column-width="5cm"
                                column-number="3" />
                            <fo:table-body>
                                <fo:table-row>
                                    <fo:table-cell text-align="left" padding-top="5pt">
                                        <fo:block>
                                            Printed
                                            <xsl:value-of
select="java:format(java:java.text.SimpleDateFormat.new('d-MM-yyyy, HH:mm:ss'),
java:java.util.Date.new())" />

```

```

        </fo:block>
      </fo:table-cell>
      <fo:table-cell text-align="center">
        <fo:block>
          <fo:external-graphic
            src="stylesheets/icons/hmedia.jpg" width="3cm"
          />
        </fo:block>
      </fo:table-cell>
      <fo:table-cell text-align="right" padding-top="5pt">
        <fo:block>
          Page
          <fo:page-number />
          of
          <fo:page-number-citation ref-id="last-page"/>
        </fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-body>
</fo:table>
</fo:block>
</fo:static-content>
<fo:flow flow-name="xsl-region-body" font-size="10pt">
  <xsl:call-template name="banner"/>
  <fo:table table-layout="fixed" start-indent="2pt"
    end-indent="2pt" space-before="0.5cm" hyphenate="false">
    <fo:table-column column-number="1"
      column-width="0.8cm" />
    <fo:table-column column-number="2"
      column-width="2cm" />
    <fo:table-column column-number="3"
      column-width="7cm" />
    <fo:table-column column-number="4"
      column-width="7cm" />
    <fo:table-column column-number="5"
      column-width="2cm" />
    <fo:table-header>
      <fo:table-row text-align="center"
        font-weight="bold" background-color="gray">
        <fo:table-cell
          border="solid thin black">
          <fo:block space-after="2pt"
            space-before="2pt">
            Pos.
          </fo:block>
        </fo:table-cell>
        <fo:table-cell
          border="solid thin black">
          <fo:block space-after="2pt"
            space-before="2pt">
            Backtime
          </fo:block>
        </fo:table-cell>
        <fo:table-cell
          border="solid thin black">
          <fo:block space-after="2pt"
            space-before="2pt">
            Title
          </fo:block>
        </fo:table-cell>
        <fo:table-cell
          border="solid thin black">
          <fo:block space-after="2pt"
            space-before="2pt">
            Text
          </fo:block>
        </fo:table-cell>
        <fo:table-cell
          border="solid thin black">
          <fo:block space-after="2pt"
            space-before="2pt">
            Total
          </fo:block>
        </fo:table-cell>
      </fo:table-row>
    </fo:table-header>
  </fo:table-body>
  <xsl:apply-templates select="nsm1" />

```

```

        </fo:table-body>
    </fo:table>
    <fo:block id="last-page"></fo:block>
</fo:flow>

</fo:page-sequence>
</fo:root>
</xsl:template>

<xsl:template match="nsm1">
    <xsl:variable name="video-id" select="fields/string[@id='video-id']" />
    <xsl:choose>
        <xsl:when test="head/meta/@break = 1">
            <xsl:call-template name="inhalt">
                <xsl:with-param name="rowbackground" select="'lightgray'" />
                <xsl:with-param name="formatbackground" select="'lightgray'" />
            </xsl:call-template>
        </xsl:when>
        <xsl:when
            test="string-length(normalize-space($video-id) ) > 0">
            <xsl:call-template name="inhalt">
                <xsl:with-param name="rowbackground" select="'white'" />
                <xsl:with-param name="formatbackground" select="'#FFFF99'" />
            </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name="inhalt">
                <xsl:with-param name="rowbackground" select="'white'" />
                <xsl:with-param name="formatbackground" select="'white'" />
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template name="inhalt">
    <xsl:param name="rowbackground" />
    <xsl:param name="formatbackground" />
    <xsl:variable name="total-time" select="fields/duration[@id='total-time']" />
    <fo:table-row background-color="{ $rowbackground}" keep-together.within-page="auto"
keep-with-next="always">
        <fo:table-cell border="solid thin black">
            <fo:block space-after="15pt" space-before="2pt"
                text-align="center" font-weight="bold">
                <xsl:value-of select="fields/string[@id='page-number']" />
            </fo:block>
        </fo:table-cell>
        <fo:table-cell border="solid thin black">
            <fo:block space-after="15pt" space-before="2pt"
                text-align="right" font-weight="bold">
                <!-- calculating backtime -->
                <xsl:variable name="sum">
                    <xsl:variable name="back-time" select="substring-
after(fields/duration[@id='back-time'], '@')" />
                    <xsl:choose>
                        <xsl:when
                            test="string-length($back-time) > 0">
                            <xsl:value-of select="$back-time" />
                        </xsl:when>
                        <xsl:when
                            test="string-length($back-time) = 0">
                            <xsl:variable name="sum">
                                <xsl:call-template name="calculateBacktime">
                                    <xsl:with-param name="nsm1Nodes"
                                        select="following-sibling::*" />
                                </xsl:call-template>
                            </xsl:variable>
                            <xsl:value-of
                                select="$sum - fields/duration[@id='total-time']" />
                            </xsl:when>
                    </xsl:choose>
                </xsl:variable>
                <xsl:if test="not(head/meta/@float = '1') and not($sum <= 0)">
                    <xsl:value-of select="hmedia:DateTime.getTime('H:mm:ss',
number($sum))" />
                </xsl:if>
            </fo:block>
        </fo:table-cell>
    </fo:table-row>
</xsl:template>

```

```

        <fo:block space-after="15pt" space-before="2pt"
            font-weight="bold">
            <xsl:value-of select="fields/string[@id='title']" />
        </fo:block>
    </fo:table-cell>
    <fo:table-cell border="solid thin black">
        <fo:block space-after="15pt" space-before="2pt">
            <!-- <xsl:call-template name="inserts" /> -->
        </fo:block>
    </fo:table-cell>
    <fo:table-cell border="solid thin black">
        <fo:block space-after="15pt" space-before="2pt"
            text-align="right" font-weight="bold">
            <xsl:value-of
                select="hmedia:DateTime.getTime('mm:ss', number($total-time))" />
        </fo:block>
    </fo:table-cell>
</fo:table-row>
<fo:table-row keep-together="auto">
    <fo:table-cell border="solid thin black">
        <fo:block/>
    </fo:table-cell>
    <fo:table-cell border="solid thin black" number-columns-spanned="4">
        <fo:block space-after="15pt" space-before="2pt">
            <xsl:apply-templates select="body">
            </xsl:apply-templates>
        </fo:block>
    </fo:table-cell>
</fo:table-row>
</xsl:template>

<xsl:template name="banner">
    <fo:table table-layout="fixed" start-indent="4pt" end-indent="4pt">
        <fo:table-column column-number="1" column-width="19cm" />
        <fo:table-body font-size="14pt">
            <fo:table-row>
                <fo:table-cell text-align="center">
                    <fo:block space-before="10pt">
                        <xsl:text>Run-down with text</xsl:text>
                    </fo:block>
                    <fo:block font-weight="bold" space-before="5pt" space-after="10pt">
                        <xsl:value-of select="$SYSTEM"/> - <xsl:value-of select="$QUEUE"/>
                    </fo:block>
                </fo:table-cell>
            </fo:table-row>
        </fo:table-body>
    </fo:table>
</xsl:template>
</xsl:stylesheet>

```

rundown_cues.xml

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:fo="http://www.w3.org/1999/XSL/Format"
    xmlns:hmedia="xalan://hmedia.extensions"
    xmlns:java="http://xml.apache.org/xalan/java"
    exclude-result-prefixes="java hmedia" version="1.0">

    <xsl:import href="hmedia_scripts.xsl" />

    <xsl:param name="SYSTEM"/>
    <xsl:param name="QUEUE"/>

    <xsl:template match="/NSMLRoot">
        <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
            <fo:layout-master-set>
                <fo:simple-page-master master-name="alleSeiten"
                    page-height="29.7cm" page-width="21cm" margin-top="1cm"
                    margin-bottom="1cm" margin-left="1cm" margin-right="1cm">
                    <fo:region-body margin-top="0.8cm"
                        margin-bottom="1.5cm" />
                    <fo:region-before extent="0.8cm" />
                    <fo:region-after extent="0.8cm" />
                </fo:simple-page-master>
            </fo:layout-master-set>
        </fo:root>
    </xsl:template>

```

```

</fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence master-reference="alleSeiten">
  <fo:static-content flow-name="xsl-region-after">
    <fo:block font-size="8pt">
      <fo:table table-layout="fixed">
        <fo:table-column column-width="9.5cm"
          column-number="1" />
        <fo:table-column column-width="9.5cm"
          column-number="2" />
        <fo:table-body>
          <fo:table-row>
            <fo:table-cell text-align="left">
              <fo:block>
                Printed
                <xsl:value-of
select="java:format(java:java.text.SimpleDateFormat.new('d-MM-yyyy, HH:mm:ss'),
java:java.util.Date.new())" />
              </fo:block>
            </fo:table-cell>
            <fo:table-cell text-align="right">
              <fo:block>
                Page
                <fo:page-number />
                of
                <fo:page-number-citation ref-id="last-page"/>
              </fo:block>
            </fo:table-cell>
          </fo:table-row>
        </fo:table-body>
      </fo:table>
    </fo:block>
  </fo:static-content>
  <fo:flow flow-name="xsl-region-body" font-size="10pt">
    <xsl:call-template name="banner"/>
    <fo:table table-layout="fixed" start-indent="2pt"
      end-indent="2pt" space-before="0.5cm" hyphenate="false">
      <fo:table-column column-number="1"
        column-width="1.5cm" />
      <fo:table-column column-number="2"
        column-width="7cm" />
      <fo:table-column column-number="3"
        column-width="6.6cm" />
      <fo:table-column column-number="4"
        column-width="2.0cm" />
      <fo:table-column column-number="5"
        column-width="1.7cm" />
      <fo:table-header>
        <fo:table-row text-align="center"
          font-weight="bold" background-color="gray">
          <fo:table-cell
            border="solid thin black">
            <fo:block space-after="2pt"
              space-before="2pt">
              Pos.
            </fo:block>
          </fo:table-cell>
          <fo:table-cell
            border="solid thin black">
            <fo:block space-after="2pt"
              space-before="2pt">
              Title
            </fo:block>
          </fo:table-cell>
          <fo:table-cell
            border="solid thin black">
            <fo:block space-after="2pt"
              space-before="2pt">
              Cue
            </fo:block>
          </fo:table-cell>
          <fo:table-cell
            border="solid thin black">
            <fo:block space-after="2pt"
              space-before="2pt">
              Total
            </fo:block>
          </fo:table-cell>
        </fo:table-row>
      </fo:table-header>
    </fo:table>
  </fo:flow>

```



```

        </fo:table-cell>
        <fo:table-cell
            border="solid thin black">
            <fo:block space-after="2pt"
                space-before="2pt">
                Backtime
            </fo:block>
        </fo:table-cell>
    </fo:table-row>
</fo:table-header>
<fo:table-body>
    <xsl:apply-templates select="nsm1" />
</fo:table-body>
</fo:table>
<fo:block id="last-page"></fo:block>
</fo:flow>

</fo:page-sequence>
</fo:root>
</xsl:template>

<xsl:template match="nsm1">
    <xsl:variable name="video-id" select="fields/string[@id='video-id']" />
    <xsl:variable name="break" select="head/meta/@break" />
    <xsl:choose>
        <xsl:when test="head/meta/@break = 1 or head/meta/@break = 'true'">
            <xsl:call-template name="inhalt">
                <xsl:with-param name="rowbackground" select="'lightgray'" />
                <xsl:with-param name="formatbackground" select="'lightgray'" />
            </xsl:call-template>
        </xsl:when>
        <xsl:when
            test="string-length(normalize-space($video-id) ) > 0">
            <xsl:call-template name="inhalt">
                <xsl:with-param name="rowbackground" select="'white'" />
                <xsl:with-param name="formatbackground" select="'#FFFF99'" />
            </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name="inhalt">
                <xsl:with-param name="rowbackground" select="'white'" />
                <xsl:with-param name="formatbackground" select="'white'" />
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>

<xsl:template name="inhalt">
    <xsl:param name="rowbackground" />
    <xsl:param name="formatbackground" />
    <xsl:variable name="float" select="head/meta/@float" />
    <xsl:variable name="total-time" select="fields/duration[@id='total-time']" />
    <xsl:variable name="back-time" select="substring-after(fields/duration[@id='back-
time'], '@')" />

    <xsl:variable name="foreground">
        <xsl:choose>
            <xsl:when test="$float = 1 or $float = 'true'">
                <xsl:value-of select="'blue'"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="'black'"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>

    <xsl:variable name="backtime-foreground">
        <xsl:choose>
            <xsl:when test="string-length($back-time) > 0">
                <xsl:value-of select="'green'"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="'black'"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>

    <fo:table-row keep-together="always" background-color="{ $rowbackground }">

```

```

<fo:table-cell border="solid thin black">
  <fo:block space-after="15pt" space-before="2pt"
    text-align="center" font-weight="bold" color="{ $foreground}">
    <xsl:value-of select="fields/string[@id='page-number']" />
  </fo:block>
</fo:table-cell>

<fo:table-cell border="solid thin black">
  <fo:block space-after="15pt" space-before="2pt"
    font-weight="bold" color="{ $foreground}">
    <xsl:value-of select="fields/string[@id='title']" />
  </fo:block>
</fo:table-cell>
<fo:table-cell border="solid thin black">
  <fo:block space-after="15pt" space-before="2pt">
    <xsl:call-template name="inserts" />
  </fo:block>
</fo:table-cell>
<fo:table-cell border="solid thin black">
  <fo:block space-after="15pt" space-before="2pt"
    text-align="right" font-weight="bold" color="{ $foreground}">
    <xsl:value-of
      select="hmedia:DateTime.getTime('mm:ss', number($total-time))" />
  </fo:block>
</fo:table-cell>
<fo:table-cell border="solid thin black">
  <fo:block space-after="15pt" space-before="2pt"
    text-align="right" font-weight="bold" color="{ $backtime-foreground}">
    <xsl:if test="not($float = 1 or $float = 'true')">
      <!-- calculating backtime -->
      <xsl:variable name="sum">
        <xsl:choose>
          <xsl:when test="string-length($back-time) > 0">
            <xsl:value-of select="$back-time" />
          </xsl:when>
          <xsl:when
            test="string-length($back-time) = 0">
            <xsl:variable name="sum">
              <xsl:call-template name="calculateBacktime">
                <xsl:with-param name="nsm1Nodes"
                  select="following-sibling::*" />
              </xsl:call-template>
            </xsl:variable>
            <xsl:value-of
              select="$sum - fields/duration[@id='total-time']" />
          </xsl:when>
        </xsl:choose>
      </xsl:variable>
      <xsl:if test="not(head/meta/@float = '1') and not($sum < 0)">
        <xsl:value-of select="hmedia:DateTime.getTime('H:mm:ss',
          number($sum))" />
      </xsl:if>
    </xsl:if>
  </fo:block>
</fo:table-cell>
</fo:table-row>
</xsl:template>

<xsl:template name="banner">
  <fo:table table-layout="fixed" start-indent="4pt" end-indent="4pt">
    <fo:table-column column-number="1" column-width="13cm" />
    <fo:table-column column-number="2" column-width="6cm" />
    <fo:table-body font-size="14pt">
      <fo:table-row>
        <fo:table-cell text-align="center">
          <fo:block space-before="10pt">
            <xsl:text>Rundown</xsl:text>
          </fo:block>
          <fo:block font-weight="bold" space-before="5pt" space-after="10pt">
            <xsl:value-of select="$SYSTEM"/> - <xsl:value-of select="$QUEUE"/>
          </fo:block>
        </fo:table-cell>
        <fo:table-cell text-align="right">
          <fo:block>
            <fo:external-graphic
              src="stylesheets/icons/hmedia.jpg" width="6cm" />
          </fo:block>
        </fo:table-cell>
      </fo:table-row>
    </fo:table-body>
  </fo:table>

```

```
        </fo:table-row>
      </fo:table-body>
    </fo:table>
  </xsl:template>
</xsl:stylesheet>
```