



infoAsset Broker

Administrationshandbuch für Windows und Unix-Plattformen

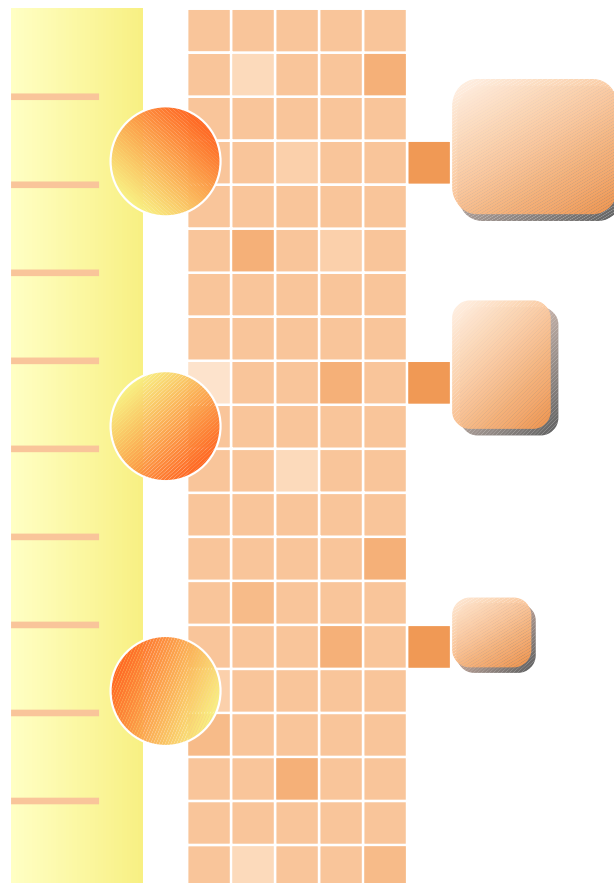
Stand: Mai 2002

a
d
d
i
n
g

v
a
l
u
e

t
o

i
n
f
o
r
m
a
t
i
o
n





Inhaltsverzeichnis

1	Aufbau des <i>infoAsset Brokers</i>.....	5
1.1	Client-Server Struktur.....	5
1.2	Verzeichnisse und Konfigurationsdateien des infoAsset Brokers.....	6
2	Standard-Installation, Konfiguration und Start des <i>infoAsset Brokers</i>.....	9
2.1	Standard-Installation des infoAsset Brokers	9
2.1.1	Die allgemeinen Installationsvoraussetzungen prüfen	9
2.1.2	Die Software installieren (Windows)	10
2.1.3	Die Software installieren (Unix).....	10
2.2	Konfiguration des infoAsset Brokers	11
2.2.1	Konfigurationen in der Konfigurationsdatei <code>properties.txt</code>	12
2.2.1.1	Allgemeines zum Änderung in der Konfigurationsdatei	12
2.2.1.2	Initialisieren des Stores.....	13
2.2.1.3	Projekthierarchie eines Projektes festlegen	14
2.2.1.4	Die Fehlerprotokollierungsoptionen des infoAsset Broker konfigurieren	14
2.2.1.5	Die Spracheinstellung des infoAsset Brokers vornehmen	17
2.2.1.6	ContentManager Configuration	17
2.2.1.7	Java-Datenbanktreiber von Drittherstellern	18
2.2.1.8	Volltextindizes.....	19
2.2.1.9	Filter	19
2.2.1.10	Port	20
2.2.1.11	Automatisches Abmelden	21
2.2.1.12	Mehrere Serverinstanzen des infoAsset Broker konfigurieren	21
2.2.1.13	Einen anderen Browser beim Start des infoAsset Brokers starten.....	23



2.3	Den infoAsset Broker starten.....	23
3	Administration	25
3.1	Funktionen der Web-Oberfläche.....	25
3.1.1	Als Administrator anmelden.....	25
3.1.2	Manuelles Abmelden	25
3.1.3	Server herunterfahren	25
3.1.4	Aktive Sitzungen anzeigen.....	25
3.1.5	Domänen bearbeiten	26
3.1.6	Bildclips bearbeiten	26
3.1.7	Alle Bewertungen ansehen	26
3.1.8	Bewertungsregeln bearbeiten	26
3.1.9	Ähnlichkeitssuchmaschine.....	26
3.1.10	Klassifikatoren	27
3.1.11	Verzeichnisstatistik.....	27
3.1.12	Volltextindizes	27
3.1.13	Indizierungsfehler	27
3.2	Freischalten einer Person	28
3.3	Von einem anderen Computer als Benutzer auf den infoAsset Broker zugreifen.	28
3.3.1	Datenbanken älterer <i>infoAsset Broker</i> -Versionen nach einem Software-Upgrade nutzen	29



Zusammenfassung

Der *infoAsset Broker* ist eine skalierbare und flexible Serversoftware, die als Broker Inhalte zwischen Clients und Content Stores vermittelt. Die Bedienung unterschiedlicher Clients und die Anbindung unterschiedlicher Content Stores erfordert unternehmensspezifische Konfiguration der Schnittstellen des *infoAsset Brokers* zu diesen Komponenten. Die möglichen Konfigurationen der Schnittstellen bilden den Hauptteil dieses Handbuchs. Als einleitendes Kapitel wird das Zusammenspiel der drei Komponenten Client, Broker und Content Store beschrieben. Für die Installation und Konfiguration der Clients bzw. Content Stores wenden Sie sich bitte an den Herstellern der Komponenten.

Zielpublikum

Dieses Administrationshandbuch richtet sich an diejenigen Personen im Unternehmen, die für die Installation und den Serverbetrieb des *infoAsset Brokers* (oder von Systemlösungen basierend auf dem *infoAsset Broker*) verantwortlich sind.

Es werden Grundkenntnisse auf den Gebieten Betriebssysteme, Datenbanken und Internet-Protokolle vorausgesetzt sowie Erfahrung beim Umgang und der Administration der jeweiligen Ziel-Betriebssysteme und ggf. Datenbanken.

Zum Verständnis dieses Handbuchs

Der *infoAsset Broker* ist ein Kernsystem auf dem firmenspezifische Projekte (Systemlösungen) aufbauen. Im Kernsystem sind firmenübergreifende Dienste realisiert, die in den Projekten mit firmenspezifischen Diensten erweitert werden. Die firmenspezifischen Projekte erhalten eindeutige Namen, z.B. *webac*, *finb*, *smapco*, *contentmap*. Da dieses Handbuch für alle firmenspezifische Projekt gilt, werden die firmenspezifischen Einstellungen auf das fiktive Projekt *abc* dargestellt.

Wichtige Begriffe werden fett-formatiert dargestellt. Ein Begriffsverzeichnis am Ende dieses Handbuchs bietet die Möglichkeit wichtige Begriffe nachzuschlagen.

1 Aufbau des *infoAsset Brokers*

In diesem Kapitel wird der grundlegende Aufbau des *infoAsset Brokers* beschrieben. Es dient dazu, die wesentlichen Komponenten, die mit dem *infoAsset Broker* eingesetzt werden können und die dazu benötigten Konfigurationsdateien darzustellen.

1.1 Client-Server Struktur

Der *infoAsset Broker* ist eine zu 100% in Java realisierte plattformunabhängige Server-Software. Der *infoAsset Broker* kann seine Daten in verschiedenen Formaten in Datenbanken, im Dateisystem oder in Content-Management-Systemen von Drittherstellern (z.B. CoreMedia Content Application Platform) ablegen und verschiedenen Endgeräten / Clients (HTML, WML, ...) zur Verfügung stellen.

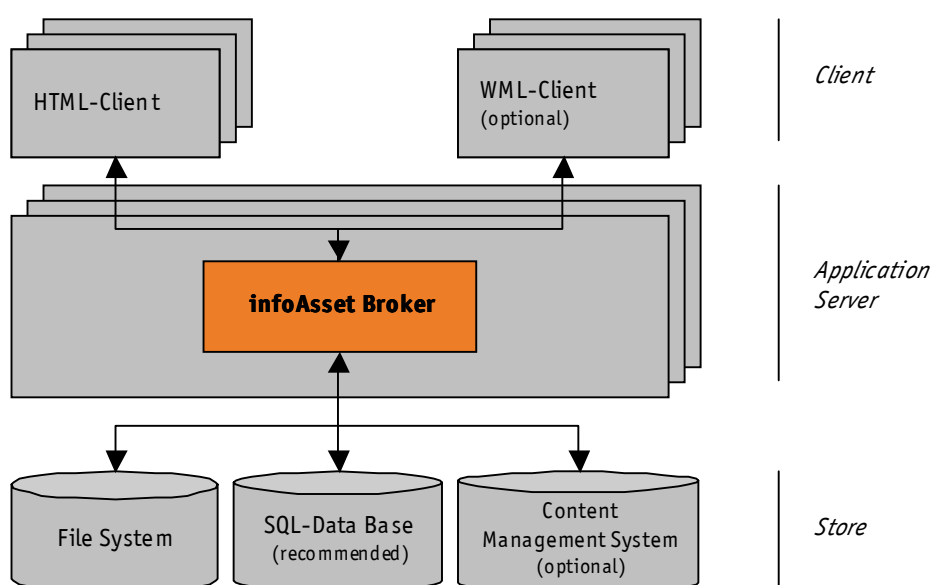


Abbildung 1: Client-Server Struktur

Der **Application Server** in Abbildung 1 stellt einen Computer dar, auf dem der *infoAsset Broker* installiert ist. Der Server ist über das HTTP-Internet-Protokoll für **Clients** erreichbar. Im Falle des Brokers können PCs und mobile Endgeräte (Handies, PDAs ...) als Clients fungieren. Die Clients können über ein Intranet, Extranet sowie über das Internet auf den Server zugreifen. Dazu muss jeder Benutzer einen **Web-Browser** (Internet Explorer oder Netscape Navigator) auf seinem Computer bzw. entsprechende Soft-

wäre auf anderen Endgeräte (z.B. Mobiltelefone mit WAP-Zugang) starten und einen Zugang über das Internet zum *infoAsset Broker* Server haben.

Durch die Vielzahl der bedienten Endgeräte und unterstützten Speichersysteme (**Stores**) entsteht die Flexibilität und Skalierbarkeit der Brokers. Diese Eigenschaften wiederum bedingen eine Anpassung des *infoAsset Brokers* an die gegebene IT-Infrastruktur. Die Konfiguration an die unternehmensspezifische IT-Infrastruktur wird in den folgenden Abschnitten beschrieben.

1.2 Verzeichnisse und Konfigurationsdateien des *infoAsset Brokers*

Die wichtigsten Systemkomponenten für die Konfiguration des *infoAsset Brokers* sind in Abbildung 2 dargestellt.

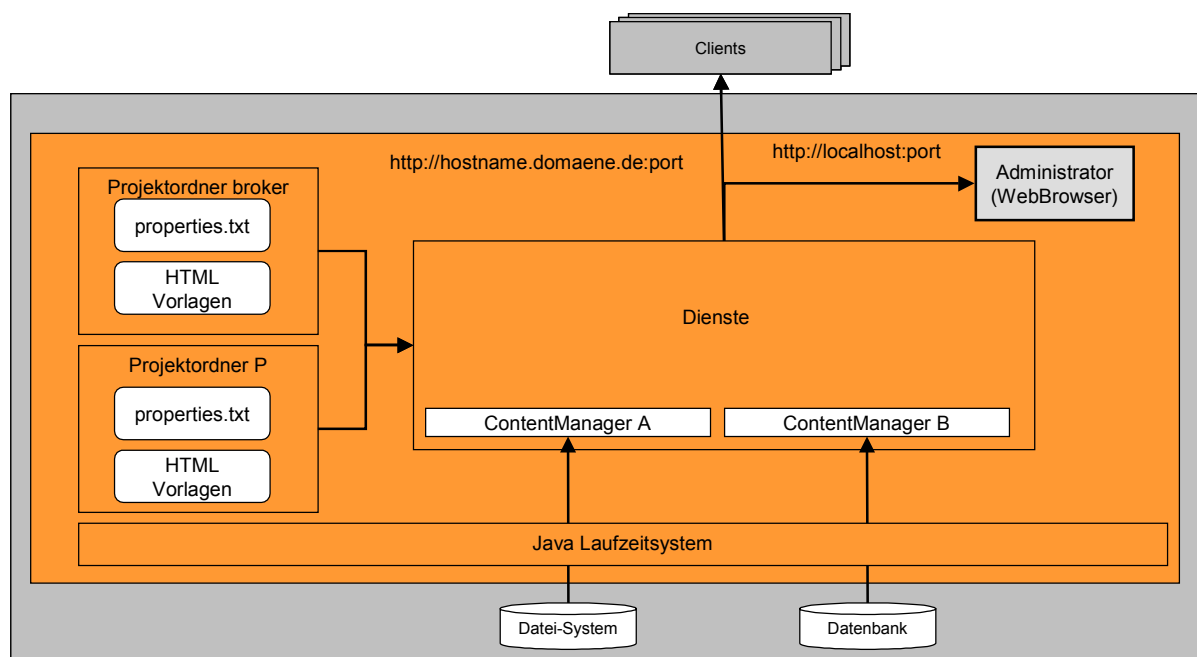


Abbildung 2: Systemkomponenten einer *infoAsset Broker* Installation im Überblick

Nachdem Sie die Software im **Installationsverzeichnis** entpackt haben (siehe Kapitel 2.1.2), enthält dieses die folgende Verzeichnisstruktur, in Abbildung 2 links angedeutet.

Verzeichnis	Inhalt
\$home\$	Installationsverzeichnis (z.B. c:\Programme\infoAssetBroker)
\$home\$/classes	Angepasste, binäre Programmcodes des <i>infoAsset Brokers</i> (Java-



	Klassen)
<code>\$home\$/doc</code>	Dokumentation zum infoAsset Broker
<code>\$home\$/lib</code>	Zum Betrieb des <i>infoAsset Brokers</i> benötigte Java-Bibliotheken
<code>\$home\$/projects</code>	Projektverzeichnisse werden innerhalb dieses Ordners angelegt
<code>\$home\$/projects/broker</code>	Projektverzeichnis des <i>infoAsset Brokers</i> Basissystems
<code>\$home\$/projects/abc</code>	Projektverzeichnis des Projekts "abc"
<code>\$home\$/projects/...</code>	Projektverzeichnisse für weitere Projekte

Tabelle 1: Verzeichnisstruktur

Bei der Installation des *infoAsset Broker* für das Projekt `abc` werden im Verzeichnis `$home$/projects/abc` folgende zwei Dateien angelegt:

- die Konfigurationsdatei `properties.txt`
- die Startdatei `start.bat` bzw. für UNIX-Betriebssystem `start.txt`

Die separat erhaltene Lizenzdatei muss ebenfalls in diesem Verzeichnis abgelegt werden.

Der folgende Text befasst sich nur mit der Datei `start.bat`. Es gelten aber sinngemäß die gleichen Aussagen für die Datei `start.sh`. Die von Ihnen gewählte Konfiguration wird in diesen beiden Dateien dem *infoAsset Broker* beim Start mitgeteilt. Um die Dateien zu anzupassen, rufen Sie die Dateien mit einem Texteditor auf.

Das Installationsverzeichnis wird innerhalb der Konfigurationsdatei `properties.txt` und Startdatei `start.bat` (Windows) `start.sh` (Unix) bzw. mit dem symbolischen Namen `$home$` (siehe auch 2.2) bezeichnet, daher wird der Name hier auch als Verzeichnisname verwendet.

Auf dem Server muss ein **Java Laufzeitsystem** (Java Runtime Environment, JRE, Version 1.2.2 oder besser) zur Ausführung des in Java realisierten *infoAsset Broker* installiert sein. Die Windows-Distribution des *infoAsset Broker* enthält bereits ein gebündeltes Java Laufzeitsystem.

Die Softwareschnittstelle des infoAsset Broker zu jedem Speichersystem (ORACLE-Datenbank, DB2-Datenbank, MySQL-Datenbank, CoreMedia CAP etc.) wird als **ContentManager** bezeichnet. Der *infoAsset Broker* kann einen oder mehrere ContentManager simultan benutzen.

Hinweise:

- Der *infoAsset Broker* ist ein Serverprogramm, das einmal auf dem Server gestartet wird, und das eine (prinzipiell) beliebig große Menge von Benutzern gleichzeitig bedient.



- Beim Start des *infoAsset Broker* geben Sie ein Startprojekt, z.B. Projekt `abc` und eine Liste von weiteren Projekten an (z.B. `abc, broker`). Jedem Projekt ist ein eigenes Projektverzeichnis in der Verzeichnisstruktur des *infoAsset Brokers* zugeordnet. Der *infoAsset Broker* liest seine Konfigurationsdatei aus dem Startprojektverzeichnis. Dabei "erbt" jedes Projekt von seinem zugrundeliegenden Projekt (seinem Vorgänger in der Projektliste). Als Basisprojekt muss in jedem Fall das Projekt `broker` verwendet werden. Durch diesen Mechanismus können Sie leicht (z.B. kundenspezifische) Varianten eines Produktes oder einer Installation erstellen.



2 Standard-Installation, Konfiguration und Start des *infoAsset Brokers*

In diesem Kapitel werden die Administratortaufgaben behandelt. Im Abschnitt "Installation" wird die Standard-Installation des *infoAsset Brokers* beschrieben. Diese Installation wird mit der Standard-Distribution des *infoAsset Broker* ausgeliefert. Diese Installation eignet sich für Demonstrationszwecke und bildet die Ausgangsinstallation für eine unternehmensspezifische Konfiguration des *infoAsset Brokers*. Die Konfigurationsdateien und ihre einzelnen Eigenschaften werden im Abschnitt "Konfiguration" behandelt. Der Start des *infoAsset Brokers* wird im letzten Abschnitt dieses Kapitels beschrieben.

Es wird empfohlen, dass Sie sich schrittweise mit dem infoAsset Broker vertraut machen und mit der Standardinstallation beginnend die von Ihnen gewünschten Installationsparameter ändern. Bei diesem Vorgehen fällt es Ihnen leichter, Konfigurationsprobleme im Zusammenspiel mit Drittsoftware (z.B. Datenbanken oder WAP-Gateways) einzugrenzen.

2.1 Standard-Installation des infoAsset Brokers

Bitte beachten Sie, dass diese Installationsform nicht für den Produktivbetrieb geeignet ist.

Das Installationsverzeichnis des infoAsset Brokers wird in diesem Handbuch auch als „workspace“ bezeichnet.

2.1.1 Die allgemeinen Installationsvoraussetzungen prüfen

1. Eine CPU mit mind. 233 MHz (500 MHz empfohlen) ist vorhanden.
2. Es sind mindestens 32 MB Hauptspeicher (64 MB empfohlen) zum simultanen Betrieb von Server und Client vorhanden.
3. Mindestens 55 MB freier Speicher auf der Festplatte verfügbar.
4. Betriebssystem Windows-95, Windows-98, Windows-NT 4.0, Windows-2000 Professional, Windows XP, Linux oder Sun Solaris 4.2/HP UX11 installiert.
5. Internet Explorer (Version 5 oder besser) oder Netscape Navigator (Version 4.7 oder besser) installiert.
6. Nur Unix: Java 1.2.2 Runtime Environment installiert und von demjenigen Benutzer ausführbar, unter dessen Kennung später der infoAsset Broker gestartet werden soll.
7. Nur Unix: Sprache des Benutzers, unter dessen Kennung später der infoAsset Broker gestartet werden soll, ist deutsch („setenv LANG DE“).



8. Für den Produktivbetrieb mit mehreren Anwendern wird der Einsatz einer Datenbank empfohlen. Es werden folgende Datenbanken unterstützt:

1. Oracle 8.1.7 Standard Edition
2. IBM DB2 Universal Database 7.2 mit Text Information Extender
3. MS-SQL 2000
4. MySQL

Für die entsprechenden Datenbanken wird weiterer Festplattenplatz und eine bessere CPU empfohlen. Die Datenbank kann alternativ auch auf einem entfernten Rechner installiert sein, soweit ein netzwerktechnischer Zugriff über JDBC (MS-SQL: ODBC) möglich ist. Ein evtl. vom Datenbankhersteller mitgelieferter Filter für Dateiformate muss auf Betriebssystemebene aufrufbar sein.

Für den notwendigen Datenbankplatz auf der Festplatte gilt: Notwendiger Platz \approx erwartetes Dokumentvolumen * 1,6.

2.1.2 Die Software installieren (Windows)

1. Laden Sie die ZIP-Datei `iABroker.zip` aus dem Internet (www.infoasset.de) oder von CD.
2. Öffnen Sie die ZIP-Datei durch Doppelklicken.
3. Entpacken Sie den gesamten Inhalt der ZIP-Datei in ein Verzeichnis, in das der infoAssetBroker als Unterverzeichnis installiert werden soll (Beispiel: „C:\Programme“).
4. Es wird ein Unterverzeichnis angelegt, das die benötigte Software enthält. Die Verzeichnisstruktur ist in Abschnitt 2.2 näher erläutert.

2.1.3 Die Software installieren (Unix)

5. Kopieren Sie die Datei „`iABroker.tgz`“ als Datei „`infoAssetBroker.tar.gz`“ in ein Verzeichnis, in dem der infoAsset Broker in einem Unterverzeichnis installiert werden soll.
6. Melden Sie sich als derjenige Benutzer an, unter dessen Kennung später der infoAsset Broker gestartet werden soll.
7. Entpacken Sie die Datei mit dem Programm `gzip` („`gzip -d infoAssetBroker.tar.gz`“).
8. Entpacken Sie das Archiv mit dem Programm `tar` („`tar xvf infoAssetBroker.tar`“). Als Ergebnis sollte jetzt ein neues Unterverzeichnis entstanden sein, das selbst wiederum diverse Unterverzeichnisse enthält.



2.2 Konfiguration des infoAsset Brokers

Im den folgenden Abschnitten werden die Konfigurationsdatei `properties.txt` und die Startdatei `start.bat` (Windows) bzw. `start.sh` (Unix) behandelt, die für Sie als Administrator relevant sind. In der Konfigurationsdatei `properties.txt` können u.a. folgende Einstellungen vorgenommen werden:

- definiert die Liste der Projekt, von denen das Startprojekt erbt
- konfiguriert die ContentManager, die der infoAsset Broker benutzt (z.B. Datenbankname, Datenbankbenutzer, Datenbankpasswort)
- legt die Ports fest, unter dem der infoAsset Broker auf Verbindungen von Web-Browsern wartet
- legt fest, wie lange Benutzer untätig sein dürfen, bevor sie automatisch vom Server abgemeldet werden
- legt den Namen und das Passwort des Systemadministrators fest
- definiert zahlreiche weitere Konfigurationsparameter des Broker, die in Abschnitt 2.2.1 im Detail beschrieben sind

In der Startdatei `start.bat` können u.a. folgende Einstellungen vorgenommen werden:

- legt das Installationsverzeichnis der infoAsset Broker Software im Dateisystem fest
- legt den Suchpfad für Java-Klassenbibliotheken von Drittherstellern fest (z.B. Datenbanktreiber)
- legt das Startprojekt fest
- ermöglicht eine Initialisierung der Datenbank beim erstmaligen Start des Servers

Zu den Einstellungen der Datei `start.bat` gelten analog für die Datei `start.sh`, mit deren Hilfe der Broker unter Unix gestartet wird.



Achtung: Da die Pfadangaben durch Java interpretiert werden, sollten Sie das Zeichen "/" statt des Zeichens "\" in den Pfadangaben verwenden (also `D:/myinstall` statt `D:\myinstall` wie bei Windows üblich).

2.2.1 Konfigurationen in der Konfigurationsdatei `properties.txt`

2.2.1.1 Allgemeines zum Änderung in der Konfigurationsdatei

Gehen Sie wie folgt vor, um den Wert einer Variablen zu verändern:


1. Zeigen Sie den Inhalt des Verzeichnisses des **Startprojekts** an. Das ist das Verzeichnis, in dem Sie den infoAsset Broker starten.
2. Öffnen Sie die Datei `properties.txt` durch Doppelklicken.
3. Es wird der Inhalt der Konfigurationsdatei angezeigt.
4. Suchen Sie die Zeile, die mit dem Namen der Variablen beginnt, deren Wert sie ändern wollen (z.B. `"browser ="`).
5. Löschen Sie den bisherigen Wert der Variablen, der hinter dem Gleichheitszeichen eingetragen ist und tragen Sie dort den neuen Wert für die Variable ein.
6. Speichern Sie die Änderungen in der Datei `properties.txt`.
7. Starten Sie den *infoAsset Broker* neu, damit Ihre Änderungen beim Start übernommen werden.



Hinweis Beachten Sie bei Änderungen in der Datei `properties.txt` die folgenden Regeln:

- Leerzeichen am Zeilenanfang, am Zeilenende sowie vor und nach dem Gleichheitszeichen werden ignoriert. Damit sind die folgenden Angaben äquivalent:

```
project = broker
project=broker
project  =   broker
```

- Als Zeichensatz wird ISO 8859_1 angenommen (Windows-Zeichensatz), dies ist besonders bei Umlauten relevant.
- Sie können die Notation `\uxxxx` verwenden, um ein Zeichen mit dem Zeichencode `xxxx` (z.B. eines Sonderzeichens) zu verwenden. Die Codierung erfolgt dabei in Unicode. Beispiel: `\u0020` bezeichnet das leere Zeichen „ “.
- Mit der Notation `"\t"`, `"\n"`, `"\r"`, `"\ "` können Sie die Zeichen Tab, CR, LF and Leerzeichen bezeichnen.
-  **Achtung**: Das Zeichen `"\"` müssen Sie durch `"\"` bezeichnen.



- Falls Sie den Wert einer Variablen (z.B. eine sehr lange Systemmeldung) in mehrere Zeilen aufbrechen wollen, beenden Sie jede Zeile bis auf die letzte Zeile mit einem `\` Zeichen, um anzuzeigen, dass der Wert in der folgenden Zeile fortgesetzt wird.
- Sie können an beliebigen Stellen in der Datei Kommentarzeilen einfügen. Jede Kommentarzeile beginnt mit dem Zeichen "#".
- Dateinamen sollten Sie am besten relativ zu dem Installationsverzeichnis des Broker angeben. Dazu können Sie innerhalb von `properties.txt` die Variablen `$home$`, `$broker$` und `$project$` verwenden.
- Die Zeichenkette `$project$` wird durch den Namen des Startprojektverzeichnisses `$home$/projects/abc` ersetzt, wobei `abc` durch den Wert des Parameters `project` in der Datei `start.bat` definiert ist.

2.2.1.2 Initialisieren des Stores

Wenn Sie sicherstellen wollen, dass sämtliche Datenbankinhalte und Datenbankstrukturen (Store) des *infoAsset Broker* gelöscht und neu initialisiert werden, gehen Sie wie folgt vor:

1. Löschen Sie das Wort `REM` in der Zeile

```
REM set OPTIONS=%OPTIONS% "deleteAllContainer=true"
```

so dass die Zeile folgenden Inhalt besitzt:

```
set OPTIONS=%OPTIONS% "deleteAllContainer=true"
```

2. Starten Sie den *infoAsset Broker* wie in Abschnitt 0 beschrieben.

Der *infoAsset Broker* löscht beim Start zunächst alle Datenbankstrukturen (z.B. mit dem Befehl „`DROP TABLE`“ in SQL) und legt sie direkt danach wieder an. Diese Operationen werden im Kommandozeilenfenster protokolliert. Falls die Datenbankstrukturen noch nicht existieren, werden Warnungsmeldungen angezeigt, die aber ignoriert werden können.



Achtung: Machen Sie nun die Änderung in der Datei `start.bat` wieder rückgängig, so dass die entsprechende Zeile wieder folgenden Inhalt besitzt:

```
REM set OPTIONS=%OPTIONS% "deleteAllContainer=true"
```

Wenn Sie eine SQL-Datenbank nutzen, werden alle vom *infoAsset Broker* benötigten Tabellen per SQL automatisch gelöscht und angelegt. Die Datenbank selbst muss aber mit den Administrationswerkzeugen des Datenbanksystems angelegt bzw. gelöscht werden.



Mit der Option `deleteAllContainer` werden auch vom *infoAsset Broker* benötigte Verzeichnisse im Dateisystem angelegt (`documentStore`, `tmp`, ...), soweit diese nicht schon vorhanden sind. Die Verzeichnisse werden *nicht* gelöscht.

Bei der Initialisierung der Datenbankinhalte wird insbesondere ein Benutzerprofil im *infoAsset Broker* für den Systemadministrator mit dem Namen „Max Mustermann“ und dem Passwort „otto“ angelegt. Der Login-Name des Systemadministrators wird durch den Wert der Variablen `adminMail` in der Datei `properties.txt` (mit dem Standardwert `admin@infoasset.de`) festgelegt.

Melden Sie sich daher unmittelbar nach der Initialisierung der Datenbank beim *infoAsset Broker* an und ändern Sie die Benutzerdaten des Administrators.

2.2.1.3 Projekthierarchie eines Projektes festlegen

Beim Start des *infoAsset Broker* geben Sie ein Startprojekt, z.B. Projekt `abc` und eine Liste von weiteren Projekten an (z.B. `abc,broker`). Jedem Projekt ist ein eigenes Projektverzeichnis in der Verzeichnisstruktur des *infoAsset Brokers* zugeordnet. Der *infoAsset Broker* liest seine Konfigurationsdatei aus dem Startprojektverzeichnis. Dabei „erbt“ jedes Projekt von seinem zugrundeliegenden Projekt (seinem Vorgänger in der Projektliste). Als Basisprojekt muss in jedem Fall das Projekt `broker` verwendet werden. Durch diesen Mechanismus können Sie leicht (z.B. kundenspezifische) Varianten eines Produktes oder einer Installation erstellen. Die Projekthierarchie wird am folgenden Beispiel veranschaulicht:

```
project = abc,xyz,broker
```

Die Reihenfolge der Vererbung würde wie folgt ablaufen: Das Startprojekt `abc` würde von Projekt `xyz` abstammen, dieses wiederum vom Projekt des Kernsystems `broker`. Damit würde zunächst im Projektverzeichnis `abc` nach einer Datei (z.B. einem HTML-Template) gesucht werden, danach im Verzeichnis `xyz` und erst am Ende im Verzeichnis `broker`.

2.2.1.4 Die Fehlerprotokollierungsoptionen des *infoAsset Broker* konfigurieren

Sollte bei der Generierung einer Seite ein Problem aufgetreten sein, kann man in 99% der Fälle auf die "Zurück"-Schaltfläche des Web-Browser klicken, um normal weiterzuarbeiten.

Eventuell auftretende Fehler sollten Sie möglichst genau dokumentieren und entsprechend Ihrer Lizenzbedingungen an *infoAsset* melden!

- Was war der Titel der zuletzt angezeigten Seite?
- Was war der Hyperlink, der ausgewählt wurde?
- Was steht am Ende des Server-Logs (als Auszug an die Fehlermeldung anhängen)



Standardmäßig werden alle Systemmeldungen des *infoAsset Brokers* über die Standardausgabe im Kommandozeilenfenster ausgegeben. Folgende Variablen in der Konfigurationsdatei `properties.txt` erlauben es Ihnen, die Protokollierung des *infoAsset Broker* zu konfigurieren:

- `log.*.file`: Falls ein Wert für diese Variable angegeben ist, werden Systemausgaben in die Datei im Startverzeichnis umgelenkt. Beispiel:

```
log.*.file = runtime.log
```

- `log.*.Tee`: Durch setzen dieses Wertes auf „true“ werden die Meldungen des infoAsset brokers nicht nur in die durch `log.*.file` gesetzte Datei geschrieben, sondern auch noch zusätzlich auf der Konsole ausgegeben. Die Voreinstellung ist, dies nicht der Fall ist. Beispiel:

```
log.*.Tee = true
```

- `genericExceptionMessage`: Falls diese Variable einen Wert `x` besitzt, so wird beim Auftreten eines Java-Laufzeitfehlers keine Java-Fehlermeldung (Stack-Trace), sondern nur der Meldungstext `x` ausgegeben. Dies ist für den Produktivbetrieb des Systems zu empfehlen.

Zusätzlich zu den eben beschriebenen Möglichkeiten besitzen die verschiedenen Komponenten des infoAsset Broker noch feiner einstellbare Fehler- und Meldungsprotokolle, die sogenannten „logs“. Jede Komponente benutzt eine eigenes Log, das unabhängig von den anderen konfiguriert werden kann. Die Komponenten können in diese Logs Fehlermeldungen, Informationen und Debug-Ausgaben schreiben. Alle drei Arten der Ausgabe können unabhängig voneinander ein- oder ausgeschaltet werden. So ist es im Produktivbetrieb etwa nicht nötig, sich Debug-Meldungen, die nur zur Fehlersuche bei der Installation hilfreich sind, anzuzeigen. Fehlermeldungen hingegen sollten in jedem Fall protokolliert werden.

Normalerweise werden die Meldungen aller Logs in die durch `log.*.file` bestimmte Datei geschrieben, dies kann aber für jedes Log umgestellt werden. Die Einstellungen für Logs heißen „`log.<name>.<eigenschaft> = <wert>`“, wobei `<name>` der Name des Komponente ist, die dieses Log verwendet. Jedes Log einer Komponente hat verschiedene Eigenschaften, die im folgenden beispielhaft beschrieben werden:

- `log.<name>.file = filename`: Hiermit wird die Datei angegeben, in die das Log geschrieben wird. Wird keine angegeben, so werden die Ausgaben in die per `log.*.file` gesetzte Datei bzw. auf die Konsole geschrieben.
- `log.<name>.append = true/false`: Stellt ein, ob die Ausgaben an die Datei angehängt werden sollen, falls sie schon existiert, oder ob sie beim Starten des infoAsset broker neu und leer begonnen werden soll. Diese Einstellung hat nur dann Wirkung, wenn `log.<name>.file` angegeben ist.
- `log.<name>.tee = true/false`: Stellt ein, ob die Ausgaben zusätzlich zum Schreiben in die Datei in die per `logFile` spezifizierte Datei geschrieben werden sollen (`true`) oder nicht (`false`). Der Standardwert ist `false`. Diese Einstellung hat nur dann Wirkung, wenn `log.<name>.file` angegeben ist.
- `Log.<name>.debug = on/off`: Stellt ein, ob Debug-Informationen geschrieben werden sollen (`on`) oder nicht (`off`). Der Standardwert ist `off`.



- `Log.<name>.info = on/off`: Stellt ein, ob normale Informationen (Statusmeldungen etc.) geschrieben werden sollen (`on`) oder nicht (`off`). Der Standardwert ist `on`.
- `Log.<name>.error = on/off`: Stellt ein, ob Fehlermeldungen geschrieben werden sollen (`on`) oder nicht (`off`). Der Standardwert ist `on`.

Die Standardwerte für `debug`, `info` und `error` können durch die Einstellungen

- `log.*.debug = on/off`
- `log.*.info = on/off`
- `log.*.error = on/off`

vorgegeben werden.

Jede Ausgabe in die Logs wird durch die Datum und Uhrzeit gekennzeichnet. Das verwendete Format dafür kann durch die Einstellung

- `log.timestampformat = format`

global für alle Logs geändert werden. Das Standardformat ist "[yyyy-MM-dd HH:mm:ss,SSSS]", wobei durch Umstellung der Platzhalter das Format geändert werden kann.

Die Einstellung

- `log.async = on/off`

stellt ein, ob die Logs asynchron (`on`) oder synchron (`off`) geschrieben werden. Asynchron bedeutet, dass die Ausgabe zeitverzögert von einem Hintergrundprozess geschrieben wird, sobald die Serverauslastung gering ist. Dies ist die voreingestellte und empfohlene Betriebsart, da so die Leistung des infoAsset Broker optimiert wird. Synchrones Schreiben kann zur Fehlersuche nützlich sein.

Folgende Komponenten des *infoAsset Broker* verwenden zur Zeit getrennte Logs:

<name>	Erklärung
ContentManager	Die vom broker benutzten Datenbanken oder Stores. Debug-Meldungen sind z.B. alle SQL-Kommandos an die Datenbank.
Dispatcher	Gibt im Debug-Modus aus, welche URL auf welche Handler abgebildet werden.
UserAgent	Gibt im Debug-Modus die erkannten User-Agents aus.
Template	Gibt im Debug-Modus analysierte HTML-Templates aus.
Session	Gibt aus, wann welche Session angelegt und beendet wurde.



WebServer	Gibt angefragte URLs und im Debug-Modus Parameter aus.
<Konfiguration>	Dito für einzelne Konfigurationen des Web-Servers.

2.2.1.5 Die Spracheinstellung des infoAsset Brokers vornehmen

Der Wert der Variablen `defaultLanguage` in der Datei `properties.txt` legt fest, welche Sprache zur Kommunikation mit einem nicht-angemeldeten anonymen Benutzer verwendet werden soll.

Beispiel:

```
defaultLanguage = de_DE
```

Der gewählte Wert muss einem verfügbaren Wert in der Domäne `Language` entsprechen. Die verfügbaren Werte werden durch Einträge der folgenden Form in der Datei `domainValues.txt` definiert.

Beispiele:

```
Language.de_DE = Deutsch  
Language.en_UK = English  
#da_DK Danish Denmark
```

Die möglichen Werte sind durch Kommentare in der Datei `domainValues.txt` beschrieben.

Jedes Benutzerprofil enthält ein Attribut, in dem die von dem Benutzer bevorzugte Sprache gespeichert ist. Die Attribute müssen einem aktiven Wert der Domäne `Language` entsprechen. Aktive Werte sind alle Nicht-auskommentierte Werte.

2.2.1.6 ContentManager Configuration

Die ContentManager Configuration stellt die Verbindung zum Content Store ein. Für folgende Content Stores sind bereits Voreinstellungen vorhanden:

- Coremedia
- ORACLE
- DB2
- MS-SQL
- MySQL



- FileSystem

2.2.1.7 Java-Datenbanktreiber von Drittherstellern

Wenn Sie einen Java-Datenbanktreiber von einem Drittherstellern mit dem infoAsset Broker nutzen möchten (z.B. für Oracle oder DB2), überprüfen Sie zunächst, ob nicht bereits ein solcher Treiber im Lieferumfang des *infoAsset Broker* enthalten ist. Alle Treiber und externen Java-Klassenbibliotheken sind im Verzeichnis `$home$/lib` gespeichert und müssen im Java-Suchpfad in der Datei `start.bat` aufgeführt werden.

Wenn Sie einen Java-Datenbanktreiber (JDBC-Treiber Version 1.2 oder besser) mit dem *infoAsset Broker* benutzen möchten (Beispiel: `xyz.jar`), der nicht mit der *infoAsset Broker* Distribution ausgeliefert wird, oder nicht im Java-Suchpfad aufgeführt wird, gehen Sie wie folgt vor.

1. Kopieren Sie das Java Archiv `xyz.jar` in den Ordner `$home$/lib`.
2. Fügen Sie am **Ende** des Suchpfads für Java-Klassenbibliotheken in der Datei `start.bat` Ihren Treiber ein, indem Sie mit einem Texteditor **nach** der letzten Zeile, die mit dem Befehl `set CLASSES` beginnt, folgende neue Zeile einfügen:

```
set CLASSES=%CLASSES%;%BROKER_HOME%\lib\xyz.jar
```

3. Ändern Sie die Einträge für den ContentManager in der Datei `properties.txt` mit einem Texteditor nach dem folgenden Schema:

```
contentManager.DefaultStore.class = OracleContentManager
contentManager.DefaultStore.driver = oracle.jdbc.driver.OracleDriver
contentManager.DefaultStore.user = admin
contentManager.DefaultStore.password = otto
contentManager.DefaultStore.url = jdbc:oracle:thin:@192.168.1.2:1521:db
```

Installationsvoraussetzungen für externe Datenquellen:

- Das Datenbankmanagementsystem ist installiert.
- Der Datenbankservers läuft
- Es existiert eine leere Datenbank
- Das Passwort und der Benutzername ist bekannt.
- Ein Java-Datenbanktreiber für die Datenbank ist vorhanden.


2.2.1.8 Volltextindizes

Zur effizienten (schnellen) Volltextsuche in großen Datenbeständen können verschiedene Einstellungen in der Datei `properties.txt` vorgenommen werden.

Die Eintragung

```
contentManager.DefaultStore.fulltext = true
```

schaltet die effiziente Volltextsuche ein.

 **Achtung:** Der unterliegende Content Manager (siehe Kapitel 2.2.1.6) muss entsprechende Optionen bieten. Dies ist derzeit nur bei dem Oracle Content Manager und dem DB2 Content Manager der Fall. Wenn ein anderer Content Manager verwendet wird, z.B. MySQL Content Manager oder der Properties Content Manager, so muss „`contentManager.DefaultStore.fulltext = false`“ gesetzt werden. Es ist dann nur eine weniger effiziente (schnelle) Volltextsuche möglich.

Die angelegten Volltextindizes (siehe Kapitel 3.1.12) müssen regelmäßig aktualisiert werden, um auch neu hinzu gekommene Dokumente zu durchsuchen. Der Eintrag

```
contentManager.DefaultStore.updateEvery
```


gibt die Häufigkeit der Aktualisierung in Minuten an. Eine häufige Aktualisierung drückt die Systemleistung, eine seltene Aktualisierung führt dazu, dass neu eingestellte Dokumente für längere Zeit nicht mit durchsucht werden. Im Allgemeinen sollte einmal pro Tag (alle 1440 Minuten) aktualisiert werden.

Die Eintragung

```
contentManager.DefaultStore.startAt
```

gibt die Uhrzeit an, zu der die erste Aktualisierung geschehen soll. Die Uhrzeit muss im Format `[hh] : [mm]` (00:00 bis 23:59) angegeben werden. Mit dieser Einstellung lässt sich die Aktualisierung der Indizes zu einem Zeitpunkt geringer Benutzeraktivität legen.

Die übrigen Einstellungen der Volltextindizes sollten nur in Absprache mit Ihrem Berater geändert werden.

 **Hinweis:** Wenn die DB2 als Content Manager verwendet wird, so muss noch die Eintrag „`contentManager.DefaultStore.onUnix`“ entsprechend der Systemumgebung gesetzt werden.

2.2.1.9 Filter

Für die Filterung von verschiedenen Dateiformaten kann ein externer Filter verwendet werden. Der (Kommandozeilen-)Befehl für die Filterung wird mit dem Eintrag

```
textFilter.command
```

festgelegt. Hier muss die Befehlsfolge eingetragen werden, mit der die Filterung einer Datei auf Kommandozeilenebene durchgeführt werden kann. Die Platzhalter „\$“ und „%“ stehen für die Quell- bzw. Zieldatei.



Für die Filter existieren weitere Einstellungen:

Einstellung	Standardwert	Bedeutung
<code>textFilter.class</code>	InsoFilter	Interne Klasse, die den Filter aufruft.
<code>textFilter.timeOutMilliseconds</code>	20000	Zeit in Millisekunden, ab der nicht mehr mit einem Ergebnis der Filterung gerechnet wird.
<code>textFilter.runsOnUnix</code>	true	gibt an, ob der Textfilter auf einem Unix- (<code>true</code>) oder Windows-System (<code>false</code>) aufgerufen wird.
<code>textFilter.plainTextFileExtensions</code>	<code>txt, java, c, sh, bat, bib, tex</code>	bei Dateien mit dieser Endung wird keine Filterung vorgenommen, da sie bereits als „plain-Ascii“ vorliegen.
<code>textFilter.filteredFileExtensions</code>	<code>pdf, doc, dot, ppt, pot, xls, html, htm, xml</code>	Dateien mit dieser Endung werden gefiltert, da sie nicht als „plain-Ascii“ vorliegen.

2.2.1.10 Port

Bei der Konfiguration des infoAsset Broker geben Sie einen Port (`http://hostname.domaene.de:port`) an, auf dem der infoAsset Broker auf einkommende Verbindungswünsche von Web-Browsern (oder anderen Endgeräten) wartet.

Sie können den infoAsset Broker so konfigurieren, dass er direkt nach dem Start auf dem Server einen lokalen Web-Browser (`http://localhost:port/`) startet, der den konfigurierten Port auf dem Server benutzt. Das ist nützlich für Demonstrationen auf einem Laptop und zur lokalen Administration des Brokers.

Einstellung in `properties.txt`:

```
port = 80
```



2.2.1.11 Automatisches Abmelden

Nach einer konfigurierbaren Zeit der Inaktivität werden Benutzer aus Sicherheitsgründen automatisch abgemeldet. Hierbei gehen nur solche Daten verloren, die noch nicht zum Server geschickt wurden.

Einstellung in `properties.txt`:

```
maxIdleMinutes = 60
```

2.2.1.12 Mehrere Serverinstanzen des infoAsset Broker konfigurieren

Der *infoAsset Broker* lässt sich so konfigurieren, dass er mehrere Web-Server-Instanzen, die auf verschiedenen Ports auf Anfragen von Web-Browsern warten, gleichzeitig betreibt. Diese Instanzen können dabei so konfiguriert werden, dass nur auf bestimmte Bereiche des Systems zugegriffen werden kann und dass nur bestimmte Domains (d.h. Bereiche im Inter- oder Intranet) darauf zugreifen können. Dies kann z.B. ausgenutzt werden, um gleichzeitig dem Internet und dem Intranet Zugriff auf verschiedene Informationen zu geben, wobei aber sichergestellt wird, dass die Internet-Nutzer nur eingeschränkte Funktionen haben.

Die verschiedenen Instanzen werden durch Eigenschaften der Form `server.<name>.<eigenschaft> = <wert>` konfiguriert. Mindestens eine solche Instanz muss definiert werden, damit auf den *infoAsset Broker* zugegriffen werden kann. Jede Web-Server-Instanz muss einen Namen haben, der statt `<name>` eingesetzt wird, z.B. „intranet“, „internet“ etc. Der Name bezeichnet zugleich den Namen eines Logs (vgl. Abschnitt 2.2.1.4)

Folgende Eigenschaften einer Instanz können konfiguriert werden:

- **server.instance.host:** Definiert den Namen des Rechners, auf dem der infoAsset Broker läuft. Dieser Name wird zusammen mit der Port-Nummer verwendet, um bestimmte Anfragen von Web-Browsern umzulenken (redirect). Wenn der Namen nicht oder falsch gesetzt ist, können bestimmte Seiten nicht erreicht werden! Wichtig für den Internet-Zugang ist daher, dass der Name so lautet, wie die jeweiligen Benutzer ihn schreiben müssten, also inklusive der vollständigen Domäne. Zum lokalen Testen reicht beispielweise „localhost“, aber dann kann natürlich kein Rechner von außerhalb diese Instanz verwenden.

Beispiel: `server.intranet.host = localhost`

- **server.instance.port:** Legt die Port-Nummer dieser Instanz fest, z.B. 80 oder 8080. Zu beachten ist, dass es keine zwei Instanzen mit derselben Portnummer geben kann.

Beispiel: `server.intranet.port = 80`

- **server.instance.domains.allowed** und **server.instance.domains.denied:** Hier können kommaseparierte Listen von IP-Adressen angegeben werden, die auf diese Instanz zugreifen dürfen bzw. es nicht dürfen. Jede Nummer kann an jeder Stelle den Joker `*` verwenden, um einen ganzen Bereich zu nennen. Ein Client darf nur auf eine Instanz zugreifen, wenn seine IP-



Nummer in der Liste der erlaubten Nummern vorkommt UND sie in der Liste der nichterlaubten nicht vorkommt. Zu beachten ist, dass auch der „localhost“ als 127.0.0.1 eingetragen werden muss, um die Instanz lokal benutzen zu können! Wenn ein Client auf eine Instanz zugreift, der Zugriff aber verwehrt wird, so erhält er die HTTP-Antwort 403 (Zugriff verweigert).

Beispiel:

```
server.intranet.domains.allowed = 134.28.70.*,127.0.0.1
server.intranet.domains.denied   = 192.168.*.*
```

- **server.instance.documents.root**: Definiert die Seite, die vom Server verwendet werden soll, wenn auf die Wurzel zugegriffen wird, also auf „/“. Hier darf kein Sprachpräfix angegeben werden, es wird die voreingestellte Sprache verwendet. Man kann hier auch die Seite einer nicht erlaubten Seite angeben (s.u.), und so den Zugriff auf die Wurzel sperren.
- **server.instance.documents.expired**: Gibt eine Seite an, die gesendet werden soll, wenn die Sitzung eines Benutzers ausgelaufen ist (expired). Zu beachten ist, dass die Seite zum Zugriff freigegeben (s.u.) und eine „Einstiegsseite“ (s.u.) sein muss, da ja gerade keine gültige Session mehr existiert.
- **server.instance.documents.allowed** und **server.instance.documents.denied**: Gibt durch Komma separierte Listen von Seiten an, die für den Zugriff freigegeben bzw. gesperrt sind. Die Angabe der Seite muss aus Bereich (optional) und Seite, z.B. „persons/edit“, bestehen, wobei für beides der Joker * verwendet werden kann. Die Liste „*/*,*“ gestattet beispielsweise Zugriff auf alle Seiten. Dabei dürfen keine Sprachpräfixe und Extensionen (z.B. .htm) benutzt werden. Ein Client kann nur dann auf eine Seite zugreifen, wenn sie in der Liste der erlaubten Seiten vorkommt UND sie in der Liste der nicht erlaubten nicht vorkommt.
- **server.instance.documents.entry**: Definiert eine Liste von Seiten, auf die ein Client zugreifen kann, ohne zuvor eine gültige Sitzung (session) zu haben. Dies sind gewissermaßen die erlaubten Einstiegsseiten. Alle Anderen Seiten können ohne gültige Sitzung nicht benutzt werden; der Versuch führt zu einer Umlenkung (redirect) auf die Wurzel der Instanz. Dies geschieht durch einen HTTP 302-Antwortcode; deswegen ist es wichtig, den Namen des Rechner per `server.instance.host` richtig zu konfigurieren. (s.o.) Zu beachten ist, dass das Sperren der Wurzel den Zugriff effektiv auf wohlbekannte Einstiegsseiten beschränkt.
- **server.instance.documents.free**: Definiert eine Liste von Seiten bzw. Dokumenten, die immer ohne eine gültige Sitzung zugreifbar sind. Diese müssen trotzdem per `server.instance.documents.allowed` zugreifbar gemacht worden sein. Wichtig: Diese Seiten (die auch Bilder etc. sein können) werden nicht über das übliche Verfahren der Handler-Tabellen verwaltet, sondern direkt ausgeliefert. Ferner unterstützt der Server hier *HTTP If-Modified-Since-Header*. Das dient dazu, die Leistung des Servers zu erhöhen. Gedacht sind diese Eigenschaften insbesondere für Bilder, Stylesheets und andere Gestaltungselemente, die keine Funktion im Sinne des *infoAsset Broker* haben.
- **server.instance.loggedPattern** und **server.instance.loggedFileName**: Legt ein Format und eine Datei fest, in die dem Common Log Format (CLF) bzw. dem Combined Logfile Format (DLF) folgend Web-Server-Zugriffslogs geschrieben werden. Das Format unterstützt folgende Parameter:



%h = host; %d = date; %r = request; %s = status; %u = user agent

Die Ausgabe wird stets an die ggf. schon bestehende Datei angehängt.

- **server.instance.forceLogin:** Legt fest, ob der Zugriff auf den infoAsset Broker grundsätzlich nur für bekannte Nutzer zugelassen ist. Greift z.B. ein Benutzer „von außen“ direkt auf eine Dokument zu, so muss er sich zunächst Authentifizieren (durch Login oder Cookie) bevor er ggf. das Dokument sehen darf. Anonyme Benutzer sind nicht gestattet. Die Rechteverwaltung *innerhalb* des infoAsset Brokers bleibt hiervon unberührt.

Ein Beispiel für eine Standardkonfiguration ist im folgenden wiedergegeben:

```
server.intranet.host = localhost
server.intranet.port = 8083
server.intranet.domains.allowed = 134.28.70.*, 127.0.0.1
server.intranet.domains.denied = 192.168.*.*
server.intranet.documents.root = redirect
server.intranet.documents.expired = home/illegalSession
server.intranet.documents.allowed = */*,*
server.intranet.documents.denied =
server.intranet.documents.entry =
redirect,main,home/default,home/illegalSession
server.intranet.documents.free = skin/*,skin/images/*,navigator/*,file/*,*
server.intranet.loggedPattern = %h - - [%d] "%r" %s - "-" "%u"
server.intranet.loggedFileName = $project$/httpLog.txt
server.intranet.forceLogin = false
```

Weitere Anmerkungen finden sich in der mitgelieferten Datei `properties.txt`.

2.2.1.13 Einen anderen Browser beim Start des infoAsset Brokers starten

Der Browser, der beim Start des *infoAsset Broker* gestartet wird, wird mit der Variablen `browser` in der Datei `properties.txt` festgelegt.

Beispiele für Werte dieser Variablen sind:

```
browser = C:/Programme/Internet Explorer/IEXPLORE.EXE
browser = C:/Programme/Plus!/Microsoft Internet/IEXPLORE.EXE
browser = C:/Programme/Netscape/Communicator/Program/navigator.exe
```

Gehen Sie wie in Abschnitt 2.2 beschrieben vor, um den Wert der Variablen `browser` zu ändern.

2.3 Den infoAsset Broker starten

Wie in der Einleitung erklärt, existiert für jedes Projekt ein eigenes Projektverzeichnis. Bitte starten Sie den *infoAsset Broker* in dem Projektverzeichnis des **Startprojekts**:



Beispiele:

- Starten Sie das Programm `C:\Programme\infoAssetBroker\projects\broker\start.bat` durch Doppelklicken, um den infoAsset Broker ohne Erweiterungen zu starten.
- Starten Sie das Programm `C:\Programme\infoAssetBroker\projects\abc\start.bat` durch Doppelklicken, um den infoAsset Broker mit den Erweiterungen des Projekts `abc` zu starten.

Es öffnet sich ein Kommandozeilen-Fenster mit schwarzem Hintergrund, in dem verschiedene Systemmeldungen des infoAsset Broker ausgegeben werden.

- Sie können das Kommandozeilen-Fenster minimieren, sie dürfen es aber nicht schließen, da sonst der infoAsset Broker beendet wird und dabei Daten verloren gehen können.
- Der *infoAsset Broker* startet automatisch den festgelegten Web-Browser und dieser öffnet ein neues Browser-Fenster mit einer URL der folgenden Form: `http://localhost:port/`.
- In diesem Browserfenster wird der *infoAsset Broker* angezeigt.

In Abschnitt 2.2.1.13 wird erklärt, wie Sie den *infoAsset Broker* so konfigurieren, dass ein anderer Web-Browser zur Anzeige benutzt wird.



3 Administration

3.1 Funktionen der Web-Oberfläche

Bitte beachten Sie, dass die Oberfläche des infoAsset Brokers typischerweise auf die individuellen Bedürfnisse des Kunden abgestimmt wird. Daher sieht „Ihr“ infoAsset Broker vermutlich ganz anders aus, als der hier gezeigte. Dies ändert jedoch nichts an der Funktionsweise.

Der folgende Abschnitt orientiert sich an der Seite „Server“ des Standard infoAsset Brokers.

3.1.1 Als Administrator anmelden

- Rufen Sie den Anmeldebildschirm mit Ihrem Web-Browser auf.
- Geben Sie den Anmeldenamen eines Administrators ein (Standardwert: `admin@infoasset.de`)
- Geben Sie das Passwort des Administrators ein (Standardwert: `otto`)
- Klicken Sie auf die Schaltfläche `Anmelden`.
- Es wird der Begrüßungsbildschirm angezeigt, auf dem auch zahlreiche Administrationsbefehle zur Verfügung stehen.

3.1.2 Manuelles Abmelden

- Die Schaltfläche "Ihr Profil" klicken.
- Die Funktion "[Abmelden]" wählen.

Weitere Hinweise zum Umgang mit dem infoAsset Broker enthält das Benutzungshandbuch.

3.1.3 Server herunterfahren

Über diese Funktion kann der infoAsset Broker über das HTML-Interface heruntergefahren werden. Nicht gesicherte Daten der Benutzer gehen dabei verloren, alle Benutzer werden automatisch abgemeldet.

Der Server kann so heruntergefahren werden, dass er sofort im Anschluss wieder neu startet.

3.1.4 Aktive Sitzungen anzeigen

Hier erscheint eine Seite mit allen aktiven Sitzungen.

Hinweis: Benutzer, die sich auf dem Anmeldebildschirm befinden, erscheinen als „anonymoe“ Benutzer mit kryptischem Namen. Dies ändert sich, sobald sich der Benutzer authentifiziert hat.

3.1.5 Domänen bearbeiten

In einer Domäne sind die möglichen Einträge für Auswahlfelder hinterlegt. Zum Beispiel werden in der Domäne "Sprache", die möglichen Sprachen festgelegt in den sich ein Benutzer anmelden kann. Dieses Einträge kann man, wie in Abbildung 4 dargestellt, in der Spalte Sprache im Optionsfeld auswählen.



The screenshot shows a web interface for editing user information. At the top, there is a search bar and a navigation menu with items like 'Home', '?', 'Dokumente', 'Verzeichnisse', 'Vorlagen', 'Begriffe', and 'Pe'. Below the navigation, the title 'Anmeldeinformation für Max Müller bearbeiten' is displayed. The form contains the following fields:

Anmeldename:	<input type="text" value="mueller@home.de"/>
Passwort:	<input type="password"/> (nur falls Änderung gewünscht)
Passwortwiederholung:	<input type="password"/> (zur Sicherheit)
Sprache:	<input type="text" value="Deutsch"/>
Registrierung:	<input type="text" value="Deutsch"/> <input type="text" value="English"/> in Benutzer erfolgt und noch nicht durch den Administrator freigegeben
Anmeldung:	<input checked="" type="checkbox"/> freigeschaltet

Abbildung 3: Anmeldeinformationen für Max Müller

Die Domänen die zur Bearbeitung vorhanden sind, sind projektabhängig. Die Bearbeitung der Domänenwerte verhält sich für alle Domänen und Projekte gleich.

3.1.6 Bildclips bearbeiten

Hier können die im System vorhandenen Bild-Clips zentral verwaltet werden.

3.1.7 Alle Bewertungen ansehen

Hier werden alle im System vorhandenen Bewertungen zentral angezeigt.

3.1.8 Bewertungsregeln bearbeiten

Hier wird die für alle Benutzer gültige Bewertungsregel (siehe *Recommendation Extension*) festgelegt. Jeder Benutzer hat die Möglichkeit, diese Vorgabe für sich selbst abzuändern.

3.1.9 Ähnlichkeitssuchmaschine

Hier kann die Ähnlichkeitssuchmaschine trainiert, ihr Status abgefragt und ihr Zustand abgespeichert werden.



Bitte beachten Sie folgendes:

- Das Training der Ähnlichkeitssuchmaschine kann bei großen Datenmengen eine längere Zeit dauern. Entsprechend sollte ein Training in Phase geringer Benutzeraktivität vorgenommen werden.
- Wenn Sie mit dem Ergebnis eines Training einverstanden sind (die Ähnlichkeitssuche liefert gute Treffer, die automatische Klassifikation macht vernünftige Vorschläge), müssen sie das Trainingsergebnis explizit abspeichern, sonst ist es beim nächsten Systemstart verloren. Das explizite Abspeichern verhindert ein versehentliches Löschen von brauchbaren Trainingsergebnissen, wenn mit neuen Einstellungen (siehe Handbuch zur *Asset Discovery Extension*) experimentiert wird.

3.1.10 Klassifikatoren

Die verschiedenen Klassifikatoren (siehe Handbuch zur *Asset Discovery Extension*) können hier trainiert und das Trainingsergebnis gespeichert werden.

3.1.11 Verzeichnisstatistik

Hier kann explizit eine Aktualisierung der Verzeichnisstatistik vorgenommen werden. Bei großen Datenmengen kann dieser Vorgang länger dauern, so dass er nur in Zeiten geringer Benutzung angestoßen werden sollte.

3.1.12 Volltextindizes

Hier können die Volltextindizes aktualisiert, optimiert, neu erzeugt oder gelöscht werden. Alle Befehle können auch über die Tools der entsprechenden Datenbankhersteller angestoßen werden. Die Befehle dienen hier lediglich der Bequemlichkeit.

Folgendes ist zu beachten:

- Nach dem ersten Systemstart müssen die Volltextindizes neu erzeugt werden. Wird eine Volltextsuche angestoßen, ohne dass die Indizes vorher angelegt wurden, so führt dies zu einer Fehlermeldung.
- Bei großen Datenmengen kann jede einzelne dieser Operationen recht lange dauern. Stoßen Sie die Operationen nur dann im laufenden Betrieb an, wenn sich die Leistung der Volltextsuche dramatisch verschlechtert.
- Eine manuelle Aktualisierung der Indizes ist im Allgemeinen nicht notwendig, da über eine entsprechende Eintragung in der Datei `properties.txt` die Aktualisierung regelmäßig im Hintergrund angestoßen werden kann (siehe Kapitel 2.2.1.8).

3.1.13 Indizierungsfehler

Sollten bei der Erstellung oder Optimierung der Volltext-Indizes Fehler aufgetreten sein, so werden diese hier ausführlich beschrieben.



3.2 Freischalten einer Person

Suchen Sie die Person im Bereich Personen, die Sie für die Nutzung des *infoAsset Brokers* "freischalten" wollen. Lassen Sie sich die Informationen zu der Person anzeigen (vergleiche Abbildung 4).

Herr Max Müller

- Dokumente
- Beiträge zur Pinnwand
- Bewertungen

Person [bearbeiten] [löschen]		Mitgliedschaften		
Name:	Müller, Max	Gruppe	Rolle	Bemerkung
Geburtstag:	1956-05-12	Diese Person ist Mitglied keiner Gruppe.		
Position:	Gebietsleiter			
Abteilung:	Töpfe			
Organisation:				
Kategorie:	Kunde			
Adresse:	Schwalbenstrasse 99 22567 Hamburg			
Telefon Büro:				
Telefon Mobil:				
Telefax Büro:				
E-Mail:	Mueller@home.de			
Homepage:				
Kürzel:				
Eigene Bewertung:	<input type="text"/> [alle Bewertungen]			

Anmeldeinformation [bearbeiten]	
Anmelden:	gesperrt [ändern] [freischalten und Passwort senden]

© Copyright 2001. Alle Rechte vorbehalten. Datenschutzinformation.

Abbildung 4: Herr Max Müller - Beispiel

- In der Abbildung 4 sind die Informationen zur einer fiktiven Person namens Max Müller dargestellt. Die *infoAsset Broker* Dialog sind Links und die Informationen bezüglich der Person dargestellt. Die Anmeldeinformationen sind im unteren Teil des *infoAsset Broker* Dialoges dargestellt. Über den Link [\[freischalten und Passwort senden\]](#) kann die Person freigeschaltet werden, d.h. die Person bekommt zu ihrer angegebenen E-Mail Adresse eine Benachrichtigung geschickt.

3.3 Von einem anderen Computer als Benutzer auf den infoAsset Broker zugreifen.

Wie in Abbildung 1 dargestellt, können simultan beliebig viele Benutzer mit ihrem Web-Browser über das Internet auf den *infoAsset Broker* zu greifen.

Um von Ihrem eigenen Computer als Benutzer auf den *infoAsset Broker* zuzugreifen, der auf einem anderen Computer läuft, gehen Sie wie folgt vor:



1. Starten Sie auf Ihrem Computer Ihren bevorzugten Web-Browser (z.B. Internet Explorer oder Netscape Navigator).
2. Geben Sie in der Adressleiste des Browsers die URL des *infoAsset Brokers* ein, z.B.

`http://test.infoasset.de:8080`

- `test` ist dabei der Name des Computers, auf dem der *infoAsset Broker* gestartet wurde (und aktuell aktiv läuft).
- `Infoasset.de` ist dabei die Internet- → Domäne, in der der Computer `test` im Internet registriert ist
- `8080` ist der Port, unter dem der *infoAsset Broker* gestartet wurde.

Falls der Computer, auf dem der *infoAsset Broker* gestartet wurde, in keiner Internet-Domäne registriert ist, müssen Sie in der Adressleiste des Browsers eine URL mit einer IP-Adresse eingeben, z.B.

`http://134.28.70.194:8080`

- `134.28.70.194` ist dabei die IP-Adresse des Computers, auf dem der *infoAsset Broker* gestartet wurde (und aktuell aktiv läuft).
- `8080` ist der Port, unter dem der *infoAsset Broker* gestartet wurde.

3.3.1 Datenbanken älterer *infoAsset Broker*-Versionen nach einem Software-Upgrade nutzen

Der *infoAsset Broker* überprüft bei jedem Start, ob die Datenbankstrukturen (z.B. Tabellen in einer SQL-Datenbank) mit den von dem aktuellen Software-Releasestand erwarteten Strukturen übereinstimmen. Ist dies nicht der Fall, so führt der *infoAsset Broker* beim Start automatisch die notwendigen Datenbank-schemamodifikationen durch (z.B. `ALTER TABLE` in SQL). Sollte es hierbei zu Problemen kommen, werden entsprechende Hinweise in das Log geschrieben.