

---

## Blob-Service 3.2



## Inhaltsverzeichnis

<b>1 Überblick</b>	<b>3</b>
<b>2 Installation</b>	<b>3</b>
2.1 Startparameter . . . . .	3
2.2 Konfigurationsdatei "blobservice.ini" . . . . .	3
2.3 Installation als Windows-Dienst . . . . .	8
<b>3 Von interner Speicherung auf Blobservice-Speicherung umstellen</b>	<b>8</b>
<b>4 Blobservices umziehen</b>	<b>9</b>



# 1 Überblick

Der Blob-Service dient der Datenhaltung von grossen Dateien ausserhalb des Wissensnetzes, aber verknüpft mit den Datei-/URL-/Dokument-Attributen, in denen diese Dateiinhalte abgelegt werden sollen. Dies hat mehrere Vorteile:

- Das Wissensnetz enthält dadurch nur noch die semantischen Informationen, die auf den Dateien aufsetzen und bleibt gut sicher- und übertragbar.
- Speicherorte von Wissensnetz und Dateiinhalten können unterschiedlich konfiguriert werden.
- Es lassen sich mehrere Blob-Services an ein Wissensnetz anschliessen, so dass theoretisch je Attributdefinition ein Speicherort vorgehalten werden kann.

Im folgenden Kapitel wird das Einrichten und der Betrieb von Blob-Services erläutert.

## 2 Installation

### 2.1 Startparameter

Unter Windows kann der BlobService manuell gestartet werden (einfacher Aufruf von "blob-service.exe" oder als Dienst eingerichtet werden (s.u.).

Unter Unix wird der Blob-Service mit „visual blob-service.im &“ gestartet.

Gestoppt werden kann der Blob-Service auf zwei Arten:

- Der Windows-Dienst kann mit den üblichen Windows-Mechanismen gestoppt werden
- Ansonsten kann mit dem zusätzlichen Kommandozeilenparameter "-stop hostname[:port]" der Dienst beendet werden. "Hostname" ist dabei der Name des Rechners, auf dem der Dienst läuft, Port der bei der Konfiguration des Blob-Services angegebene Port. Wenn der Port nicht angegeben wird, wird der Standardport 30005 verwendet.

### 2.2 Konfigurationsdatei "blob-service.ini"

Der BlobService wird über die Datei "blob-service.ini" konfiguriert. Diese kann folgende Einträge enthalten:

```
[Default]
host=mediator-hostname[:port]
volume=volume-name
stores=store-names
garbageCollectTime=gcTime
garbageCollectInterval=gcInterval
detectHostname=true
standby=standbyInterval
```

```
[store-name]
```



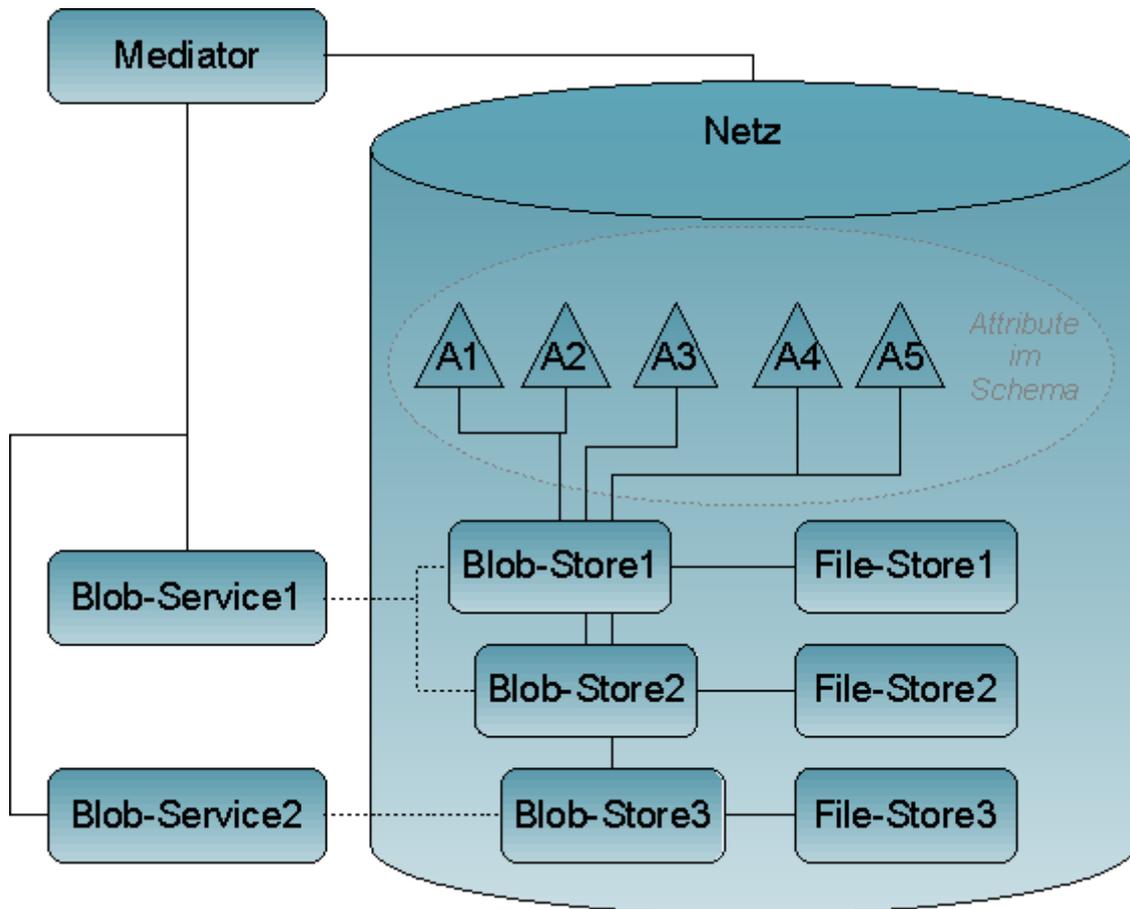
directory=C:\\daten\\blobs

- **mediator-hostname** ist der Name des K-Infinity-Servers (mit optionaler Portangabe)
- **volume-name** ist der Name des Wissensnetzes
- **store-names** ist eine durch Komma getrennte Liste mit den internen Namen der Blob-Stores, die von diesem Dienst bedient werden. Wenn kein Eintrag „stores“ existiert, werden alle Blob-Stores bedient.
- **garbageCollectTime** ist die Uhrzeit (Format "hh:mm:ss"), zu der eine automatische Blob-Garbagecollection gestartet wird. Wenn dieser Parameter fehlt, wird keine automatische GC ausgeführt.
- **garbageCollectInterval** gibt den Intervall in Tagen an, in dem eine GC ausgeführt wird. Defaultwert ist 1 (täglich)
- **detectHostname** ist per default true und weist den startenden Blob-Service an, den Namen des Rechners auf dem er läuft selbstständig zu ermitteln und im Netz zu hinterlegen. In komplizierten Netzwerkkonfigurationen kann es notwendig sein, detectHostname auf false zu setzen und im Netz den Namen des Rechners auf dem die jeweiligen Blob-Stores laufen werden manuell zu konfigurieren.
- **standbyInterval** gibt die Zeit in Sekunden an, die bei Verbindungsabbruch zum Mediator zwischen Neustarts gewartet werden soll.

Für jeden externen Speicher kann man in der Kategorie [**store-name**] noch weitere Einstellungen vornehmen:

- **directory** überschreibt das Verzeichnis, das bei der Konfiguration des File-Stores angegeben wurde. Wichtig: Backslashes ("\") müssen wie im Beispiel oben doppelt angegeben werden ("\\").

Bei der Konfiguration muss beachtet werden, dass immer nur genau ein Blob-Service ein Blob-Store und dessen Datenverzeichnis bedienen darf. Auch wenn eine Dateiresource über das Netzwerk zugreifbar wäre, würden parallel betriebene Blob-Services auf dem selben Blob-Store zu Schreibkonflikten führen.



### Logging-Einstellungen:

`loglevel = <Integer>`

Konfiguriert, welche Meldungen im Log erscheinen sollen:

- 0: Alle Meldungen inclusive Debug-Ausgaben
- 10 (Standardwert): Alle Meldungen ausser Debug-Ausgaben
- 20: Nur Warnungen und Fehlermeldungen
- 30: Nur Fehlermeldungen

`debug = true/false`

Setzt den Log-Level bei true auf 0, bei false auf 10

`channels = <Channel1> [,<Channel2>,...]`

Namen von Channelfiltern. Mit Hilfe der Channelfilter werden nur Log-Meldungen ausgegeben, die zu den angegebenen Channelfiltern gehören. Der Name eines Channelfilters deutet darauf hin, zu welchem Themengebiet die Log-Ausgaben gehören. Welche Channelfilter möglich sind, erfährt man in der Kommandozeile mit Hilfe des Parameters `-availableChannels`.



```
logfile = <Dateiname>  
log = <Dateiname>
```

Name der Log-Datei, die statt der Standard-Log-Datei verwendet wird.

```
maxLogSize = <size>
```

die maximale Grösse des Logfiles, ab der die alte Logdatei archiviert wird und eine neue angebrochen wird. Bei Werten kleiner als 1024 wird die Angabe als in MB verstanden

```
notifiers = <Name1> [, <Name2>, ...]
```

Namen von Notifiern. Ein Notifier wird bei Log-Ausgaben ab einem bestimmten Level aktiviert (typischerweise bei Fehlermeldungen). Siehe Abschnitt „Notifier“.

```
logprefix = <Prefix1> [, <Prefix2>, ... ]
```

Zusätzliche Daten, die bei jeder Log-Ausgabe hinzugefügt werden

- \$timestamp\$ : Zeitstempel der Log-Ausgabe
- \$pid\$ : Prozess-ID der Anwendung
- \$proc\$ : ID des aktuellen Smalltalk-Threads
- \$alloc\$ : belegter Speicher der VM (in Megabyte)
- \$free\$ : Freier Speicher der VM (in Megabyte)
- \$incGC\$ : Status inkrementelle GCs
- \$os\$ : Information über OS
- \$cmd\$ : Kommandozeile
- \$build\$ : Build-Version
- \$coast\$ : COAST-Version

Bei einem Prefix, der nicht in dieser Liste enthalten ist, wird der Prefix unverändert ausgegeben

### **Speicher-Einstellungen:**

Die folgenden drei Parameter dienen zur Konfiguration der Speicherzuteilung und -nutzung. Erlaubt ist die Angabe von Werten entweder in Megabyte oder in tatsächlichen Byte, wobei die Annahme gilt, dass sich Werte kleiner als 1048576 auf Megabyte-Angaben beziehen.

```
maxMemory=<Integer, in MB>
```

Maximal erlaubte Hauptspeicherbelegung. Minimal 50 MB, standardmäßig gesamter physikalisch vorhandener Hauptspeicher (unter Windows) bzw. 512 MB.

```
growthRegimeUpperBound=<Integer, in MB>
```

Hauptspeicherbelegung ab der verstärkt versucht wird, Speicher freizugeben. Standardmäßig  $0.6 * \text{maxMemory}$ .

```
freeMemoryBound=<Integer, in MB> [10]
```

Falls belegter, aber nicht mehr benötigter Speicher diese Grenze überschreitet, wird er wieder freigegeben.



`minAge=<Integer> [30]`

Minstdauer (in Sekunden), die ein Cluster im Speicher bleibt. Ein Cluster ist eine Menge von Objekten, die immer zusammen am Stück geladen werden (z.B. ein Individuum mit all seinen (Meta)eigenschaften. Cluster, die längere Zeit nicht mehr verwendet werden, werden bei Bedarf ausgelagert.

`unloadInterval=<Integer> [10]`

Minstdauer (in Sekunden) zwischen zwei Cluster-Auslagerungen

`unloadSize=<Integer> [4000]`

Mindestanzahl an geladenen Cluster, ab der ausgelagert wird

`keepSize=<Integer> [3500]`

Zahl der Cluster, die beim Auslagern behalten werden

### **Mail-Notifier Einstellungen:**

Log-Ausgaben können über Mail-Notifier zusätzlich zum Eintrag in die Log-Datei per Mail versendet werden. Sinnvollerweise wählt man einen nicht zu geringen Schwellwert (Log-Level = 30), ab dem eine Mail versendet wird.

Für jeden angegebenen Namen aus der Notifier-Liste muss eine Konfiguration angegeben werden, die im Abschnitt [Name] angegeben wird.

Folgende Angaben sind für alle Notifier gleich:

- `class = <Klassenname>` : Art des Notifiers (z. Zt. nur `COAST.CoastMailNotifier`)
- `level = <Integer>` : Log-Level, ab dem der Notifier aktiv wird

Spezifische Angaben für `COAST.CoastMailNotifier`;

- `attachLog = <Boolean>` : Wenn `true` wird ein Error-Log als Attachment an die Mail angehängt
- `sender = <Mailadresse>` : Absender der Mail
- `recipient = <Mailadresse>` : Empfänger der Mail
- `smtpHost = <Hostname>` : Mailserver
- `smtpPort = <Integer>` : Port des Mailservers. Standardwert = 25 / 465 (TLS/SSL)
- `retries = <Integer>` : Anzahl der Versuche, die Mail bei einem Fehlschlag nochmals zu versenden
- `retryDelay = <Integer>` : Wartezeit zwischen zwei fehlgeschlagenen Versuchen, eine Mail zu versenden
- `username = <Nutzername>` : optionaler Anmeldename, benötigt für Authentifizierung
- `password = <Password>` : optionales Passwort, benötigt für Authentifizierung
- `tls = true/false` : gesicherte Verbindung (TLS/SSL) verwenden. Standardwert = `false`. Falls `true` müssen `username` und `password` gesetzt sein.

TLS/SSL ist für Authentifizierung nicht zwingend erforderlich, aber dringend zu empfehlen. Die Kompatibilitätserweiterung `STARTTLS` wird nicht unterstützt.

Beispiel:

```
notifiers = errorMailNotifier
```



```
[errorMailNotifier]
class = COAST.CoastMailNotifier
level = 30
;Spezifische Angaben für Mail-Versand
attachLog = true
sender = sender@hostname.de
recipient = receiver@hostname.de
smtpHost = mailserver.hostname.de
retries = 3
retryDelay = 30
```

Die angegebene Konfiguration definiert einen Mail-Notifier, die alle Fehlermeldungen per Mail verschickt.

## 2.3 Installation als Windows-Dienst

Der Blob-Service kann als Windows-Dienst installiert werden:

```
blobservice -installAsService <Name> [<Dienst_1> ... <Dienst_n>]
```

„Name“ gibt den Namen des Windows-Dienstes an. Das aktuelle Verzeichnis, aus dem die Installation als Dienst gestartet wird, wird als Arbeitsverzeichnis verwendet

Die (optionale) angegebenen Dienste werden als benötigte Dienste eingetragen. Insbesondere der K-Infinity Server-Dienst bietet sich hier an.

Der Starttyp des Dienstes kann danach noch bei Bedarf in der Windows-Dienstverwaltung auf „Automatisch“ geändert werden.

Der Dienst kann mit

```
blobservice -deinstallService <Name>
```

wieder entfernt werden. Vorsicht: Es wird nicht überprüft ob der angegebene Dienst tatsächlich ein Blob-Service oder ein anderer Dienst ist. Alternativ kann der Dienst auch mit dem Service Control-Tool von Microsoft deinstalliert werden:

```
sc delete <Name>
```

## 3 Von interner Speicherung auf Blobservice-Speicherung umstellen

Eine kleine Anleitung, um von interner Speicherung auf die Speicherung mittels Blobservice umzustellen:

- Im Admin-Tool einen File-Store einrichten, diesem ein schreibbares Verzeichnis einrichten.
- Auf diesen File-Store einen Blob-Store einrichten, indem ein Blob-Store hinzugefügt wird und im unteren Listenbereich der zuvor eingerichtete File-Store selektiert wird. Ebenfalls eingestellt werden können der Port, unter dem der Blob-Store kontaktiert wird, und der interne Name.
- Den Blobservice starten (ohne stores-Zeile in der .ini-Datei), Blobs werden via Blobservice im angegebenen Verzeichnis verwaltet.



- Im KB auf dem jeweiligen Blob-Attribut auf die externe Speicherung (Blob-Store) umschalten.

Variante:

- In der .ini-Datei des Blobservices eine "stores"-Zeile einfügen, in der der interne Name des Blob-Stores aufgeführt ist

- In den Einstellungen für den Blob-Store ein anderes Verzeichnis spezifizieren, auf das der Blobservice zugreifen kann.

- Die im bisherigen Verzeichnis gespeicherten Dateien in das neue kopieren.

- Blobservice starten, Dateien werden nun auf dem neuen Verzeichnis verwaltet.

## 4 Blobservices umziehen

Um die von einem Blobservice verwalteten Blobs an eine andere Stelle zu transferieren, müssen folgende Schritte ausgeführt werden:

- Blobservice beenden
- Blobservice an neuer Installationstelle einrichten
- Blobs an die neue Installationsstelle verschieben/kopieren, dabei ist darauf zu achten, dass der Blobservice dieses neue Verzeichnis sehen kann und entsprechende Rechte besitzt
- ini-Datei des Blobservice an neuer Stelle muss Ort des Blobs-Verzeichnisses enthalten, wie folgt:  
[storename]  
directory=verzeichnispfad  
Hierbei ist zu beachten, dass backslashes in der Pfadangabe verdoppelt werden müssen
- Falls der hostname des Servers, auf welchem der Blobservice eingerichtet wird, für den Mediator nicht sichtbar ist, muss in der [Default]-Sektion der ini-Datei noch der Parameter "detectHostname" auf false gesetzt werden und im Wissensnetz per Admin-Tool am Blobstore der korrekte hostname eingetragen werden
- In der Datei "KFileSystemStore-[0-9]\*.ini" im Blobs-Verzeichnis muss die Zeile mit dem Eintrag "volumeID=..." gelöscht werden
- Beim Starten des Blobservices stehen dem Wissensnetz dann wie zuvor die Daten zur Verfügung

Um das Wissensnetz an eine andere Stelle zu transferieren und anschliessend weiterhin Zugriff zum weiterbestehenden Blobservice zu haben, sind folgendes zu tun:

- Wissensnetz an neuen Ort transferieren
- Blobservice herunterfahren
- im Blobspeicherverzeichnis die Datei "KFileSystemStore-[0-9]\*.ini" in einem Editor öffnen und die Zeile mit folgendem Eintrag löschen: "volumeID=...", speichern
- In der ini-Datei des Blobservice in der [Default]-Sektion folgenden Eintrag einfügen: "detectHostname=false"
- Den Blobservice wieder starten
- Mit dem Admin-Tool auf das Wissensnetz zugreifen und unter "Systemkonfiguration-



>Blobspeicherung" den bisherigen Eintrag für den gewünschten Blobservice selektieren und "Einstellungen" klicken

- Im Editor für die Blobservice-Einstellungen unter Hostname den per DNS auflösbaren Hostnamen des Blobservice-Hosts eintragen oder dessen IP-Adresse
- Mit OK quittieren