

icenter web.media

Betriebshandbuch

Änderungshistorie

Version	Datum	Autor	Bemerkung
1.0	23.02.2010	Lars Kaufmann	erstellt
1.1	10.03.2010	Lars Kaufmann	Kapitel Website vervollständigt
1.2	19.03.2010	Lars Kaufmann	Kapitel Auslieferungszustand, Mehrsprachigkeit
1.3	09.04.2010	Lars Kaufmann	Masterpasswort
1.4.1	27.04.2010	Lars Kaufmann	URL-Snapshot Dienst, Applikationskonfiguration ergänzt
1.5	17.05.2010	Ali Moinzadeh	URL-Snapshot Webservice
1.5.1	08.07.2010	Lars Kaufmann	Applikationskonfiguration über Web.User.config
1.6	24.09.2012	Lars Kaufmann	Anpassung für Branch „classic“, Update auf Version 1.1

Autor: Lars Kaufmann
Stand: 24.09.2012
Version: 1.6

Geschäftsführer

Holger Bönisch
Aydin Mir Mohammadi
Lars Kaufmann

Sparkasse Karlsruhe
Kontonr. 22399828
BLZ 66050101

Inhalt

1	Produktbeschreibung.....	3
2	Komponenten	3
2.1	Datenbank.....	3
2.2	Website.....	3
2.3	URL-Snapshot Dienst	3
3	Installation.....	4
3.1	Datenbank.....	4
3.1.1	Datenbank Serverinstanz.....	4
3.1.2	Schema und initiale Betriebsdaten	4
3.2	Website.....	5
3.2.1	Datenbankverbindung	5
3.2.2	IIS-Website und Applikation	6
3.2.3	Verzeichnisstruktur	6
3.2.4	Konfiguration	7
3.2.5	Mehrsprachigkeit	11
3.3	URL-Snapshot Dienst	11
3.3.1	Installation	11
3.3.2	Konfiguration	12
3.3.3	Delegierte URL-Snapshot Generierung über URL-Snapshot Webservice	12
3.4	Übersicht.....	14
4	Einschränkungen	14
5	Ergänzungen.....	15
5.1	Gesicherte Übertragung.....	15
5.2	Update von Version 1.0 auf Version 1.1.12381.....	15

1 Produktbeschreibung

Für die Verwaltung von multimedialen Inhalten für Fahrgastinformationssysteme soll eine webbasierte Verwaltungssaplikation *icenter web.media* erstellt werden. Mit dieser wird gesteuert, welche (Werbe-) Inhalte auf den Informationsbildschirmen in den Fahrzeugen zu welchem Zeitpunkt angezeigt werden sollen. Es handelt sich somit um ein spezialisiertes Content Management System für Bild-, Video- und Nachrichteninhalte.

icenter web.media besteht aus mehreren zentralen Komponenten. Deren grundlegende Aufgaben, die Installation und die Betriebskonfiguration werden im Folgenden beschrieben.

2 Komponenten

2.1 Datenbank

Die organisatorischen Daten wie Inhalte, Playlisten, Abläufe, Fahrzeuge, Benutzer und Rechte, etc. sowie Konfigurationseinstellungen der Anwendung werden in einer SQL-Datenbank gespeichert. Zum Einsatz kommt das Datenbanksystem MySQL 5.1.

2.2 Website

Die webbasierte Verwaltungssaplikation ist als Microsoft ASP.NET Applikation realisiert. Das heißt, die Applikation benötigt beim Benutzer keine Installation, da sie zur Darstellung nur einen Webbrowser benötigt. Serverseitig werden die Microsoft Internet Information Services (IIS) ab Version 6 benötigt.

2.3 URL-Snapshot Dienst

Für die Webseitendarstellung bei Autocontent werden von den Webseiten in regelmäßigen Abständen Bilder generiert und in den Content-Verzeichnissen der Fahrzeuge abgelegt. Diese Aufgabe übernimmt ein Windows-Service, der im Hintergrund läuft und permanent die statischen Bilder erzeugt.

Dem URL-Snapshot Dienst steht optional ein URL-Snapshot Webservice zur Seite, der die Bildgenerierung kapselt.

3 Installation

3.1 Datenbank

Die Installation der „Datenbank“ gliedert sich in zwei Teile, die Datenbank Serverinstanz und die applikationsseitige Datenbank, bestehend aus ihrem Schema und den initialen Betriebsdaten in Katalogtabellen.

3.1.1 Datenbank Serverinstanz

Die Installation der Datenbank Serverinstanz erfolgt über das Setup von MySQL 5.1. Über den Installationsassistenten kann man die Installationsmethode **Typical** treffen. Der Servertyp darf **Server Machine** sein.

Die Datenbank Serverinstanz kann mit dem Webserver gemeinsam auf einen Rechner installiert werden oder auch eigenständig auf einem dedizierten Rechner betrieben werden.

Bei getrenntem Betrieb von Datenbank und Website muss die Firewall des Rechners, auf dem die Datenbank Serverinstanz läuft, den Zugriff auf TCP Port 3306 vom Website Rechner zulassen.

Aktuelle Installationsquellen (MSI Installer) sind direkt im Internet verfügbar:

MySQL 5.1 - <http://dev.mysql.com/downloads/mysql/>

3.1.2 Schema und initiale Betriebsdaten

Das Einspielen des Datenbankschemas für **icenter web.media** erfolgt über die mitgelieferte Schemadefinitionsdatei „mysql DB Schema.mwb“. Diese kann mit der Software MySQL Workbench 5.2 über die Funktion „Database -> Forward Engineeer...“ ausgeführt werden¹.

Alternativ kann die Datenbank auch über das mitgelieferte SQL-Skript mittels MySQL Konsole erzeugt werden.

Für den Zugriff der Website auf den Datenbestand wird über das Datenbankschema der Benutzer **gorba_webassess** mit der Rolle **gorba_web_media_webaccess** angelegt. Die Privilegien der Rolle sind so eingeschränkt, dass Benutzer in dieser Rolle keine Schema- und Katalogdatenänderungen durchführen dürfen.

Der Benutzer **gorba_webaccess** wird in der Konfiguration der Website referenziert. Werden auf der Datenbank die vorgegebenen Zugangsdaten verändert, so muss dass auch entsprechend in der Konfiguration der Website angepasst werden.

¹ Es ist darauf zu achten, dass im ersten Durchlauf die Datenbankbenutzer mit angelegt werden müssen. Bei einem Update dürfen die Benutzer nicht mehr angelegt werden. Dies wird über die Option „Do Not Create Users. Only Create Privileges“ geregelt.

Für den Zugriff des URL-Snapshot Dienstes auf den Datenbestand wird über das Datenbankschema der Benutzer **gorba_urlservice** ebenfalls mit der Rolle **gorba_web_media_webaccess** angelegt.

Der Benutzer **gorbar_urlservice** wird in der Konfiguration des URL-Snapshot Dienstes referenziert. Werden auf der Datenbank die vorgegebenen Zugangsdaten verändert, so muss dass auch entsprechend in der Konfiguration des Dienstes angepasst werden.

Aktuelle Installationsquellen (MSI Installer) sind direkt im Internet verfügbar:

MySQL Workbench - <http://dev.mysql.com/downloads/workbench/5.2.html>

3.1.2.1 Auslieferungszustand

Im Auslieferungszustand beinhalten die initialen Betriebsdaten einen Standardmandanten **Gorba AG** und den Mandanten übergreifenden Root-Benutzer. Das initiale Passwort für den Benutzer **root** lautet **1234**. Das Masterpasswort² für root lautet **-5@2p%FB%N_x**.

3.2 Website

Die Website muss auf einem Windows Server 2003 oder neuer betrieben werden. Es werden sowohl 32 als auch 64 Bit Betriebssysteme unterstützt. Das .NET Framework 2.0 muss installiert sein.

Für Windows Server 2008 (R2) müssen die Rollen Webserver (IIS) und Applikationsserver aktiviert sein.

Um die Vorschaufunktionalität für Videos nutzen zu können, muss zusätzlich das Desktopdarstellungsfeature installiert werden. Darüber hinaus empfiehlt sich die Installation von zusätzlichen Video-Codecs, am einfachsten z. B. über das K-Lite Codec Pack (Full)

Aktuelle Installationsquellen mit Updates sind direkt im Internet verfügbar:

K-Lite Codec Pack - <http://www.codecguide.com/>

3.2.1 Datenbankverbindung

Damit sich die Applikation mit der Datenbank verbinden kann, ist ein .NET Connector für MySQL notwendig. Die Installation erfolgt über das Setup.

Aktuelle Installationsquellen (ZIP Archiv mit Setup) sind direkt im Internet verfügbar:

MySQL Connector - <http://dev.mysql.com/downloads/connector/net/>

² Das Masterpasswort ist nicht veränderbar.

3.2.2 IIS-Website und Applikation

Die Installationsanleitung beschreibt das Szenario für eine Installation von icenter web.media, welche unter der Adresse <http://icenter.gorba.com/web.media> erreichbar sein soll.

Die Applikation wird als Zip-Archiv geliefert, das alle Bestandteile der Website enthält. Das Archiv muss an geeigneter Stelle entpackt werden. Diese Verzeichnis ist das [Applikationsverzeichnis](#).

Beispiel Pfad

```
C:\inetpub\wwwroot\gorba_icenter_web_media
```

Jetzt sollte mit dem Internetinformationsdienste-Manager eine eigene neue Website³ erstellt werden. Dafür wird ein voll qualifizierter Domainname (FQDN) benötigt, unter dem die Webapplikation später erreicht werden soll.

Beispiel FQDN

```
icenter.gorba.com
```

Anschließend wird in der Website eine neue Anwendung erstellt. Es muss ein Alias vergeben werden. Aus FQDN und Alias ergibt sich der vollständige URL, unter dem die Webapplikation erreichbar sein wird.

Beispiel Alias

```
web.media
```

Es empfiehlt sich, für die Applikation einen eigenen Anwendungspool zu erstellen und zu verwenden. Bei 64-Bit Betriebssystemen muss dann für diesen Anwendungspool in dessen erweiterten Einstellungen die Einstellung **32-Bit-Anwendung aktivieren** auf *true* gesetzt sein.

Der physikalische Pfad zeigt auf das Applikationsverzeichnis, sprich auf das oben beschriebene Verzeichnis, in das das Zip-Archiv entpackte wurde.

Als .NET Framework-Version muss 2.0 eingestellt werden.

3.2.3 Verzeichnisstruktur

Die Webapplikation beinhaltet und benötigt folgende Verzeichnisse⁴:

- **Applikationsverzeichnis**
Hier liegen die Programm-, HTML- Bild und CSS-Dateien, sowie auf oberster Ebene die Konfigurationsdateien.

³ Steht der Server ausschließlich für diese Webapplikation zur Verfügung, kann auch die Default Web Site verwendet werden.

⁴ Diese Verzeichnisse werden in der Konfiguration referenziert. Die zugehörigen Namen in der Konfiguration sind in Klammern mit angegeben.

Beispiel Applikationsverzeichnis

C:\inetpub\wwwroot\gorba_icenter_web_media

- Fahrzeuginhalte Archivverzeichnis (UpdateStoreArchive)
Im Applikationsverzeichnis ist auch das Fahrzeuginhalte Archivverzeichnis enthalten. Dort hin wird das Zip-Archiv der Fahrzeuginhalte geschrieben, das im Updatemanager heruntergeladen werden kann.
In diesem Verzeichnis benötigt die Webapplikation Schreibrechte.

Beispiel Fahrzeuginhalte Archivverzeichnis

C:\inetpub\wwwroot\gorba_icenter_web_media\VehicleData

- Arbeitsverzeichnis (BaseWorkingDirectory)
Dieses Verzeichnis beinhaltet alle dynamisch erzeugten Inhalte, wie etwa die Bilder des Static-Contents und die zugehörigen Vorschaubilder.
Das Verzeichnis muss manuell angelegt werden. In diesem Verzeichnis benötigt die Webapplikation Schreibrechte.
Die Applikation erstellt selbstständig die benötigten Unterordner:
 - Content
Hier werden die Inhalte mit Vorschau abgelegt.
 - TempContent
Diese Verzeichnis enthält lediglich temporäre Dateien, die für die Vorschau generierung benötigt werden. Es wird in regelmäßigen Abständen von der Applikation wieder geleert.
 - UpdateStore
Das ist das Synchronisationsverzeichnis für die Fahrzeuge. Hierhin werden die Fahrzeuginhalte beim Update geschrieben. Für jedes Fahrzeug wird ein eigenes Unterverzeichnis angelegt, in das die Inhalte, der statische Inhalt, die Message-Templates und eine wagen.xml geschrieben werden.
 - UpdateStore.old
Diese Verzeichnis enthält lediglich temporäre Dateien, die während des Updates benutzt werden.
 - Monitoring Datenverzeichnis (TFTDataDirectory)
Hier liegen die Protokolldateien des Synchronisationsprozesses mit den Fahrzeugen.
In diesem Verzeichnis benötigt die Webapplikation Leserechte.

3.2.4 Konfiguration

Für den Betrieb muss die Webapplikation konfiguriert werden.

3.2.4.1 Protokollierung

Die Applikation kann Ihre Aktivitäten in Protokolldateien schreiben. Die Protokollierung wird über die XML-Datei ***Bluehands.Repository.Diagnostics.Log.Config*** konfiguriert. Mit dem Attribut ***directory*** in der Sektion ***<file>*** wird das Verzeichnis festgelegt, in das das Logfile

geschrieben werden soll. Dieses Verzeichnis muss aus diesem Grund von der Webapplikation beschreibbar sein. Das Attribut *title* gibt den Dateinamen der Logdatei an.

Beispiel Log Konfiguration

```
<configuration>
  [...]
  <fileLogger>
    <file
      directory="C:\temp\Log\Gorba_Web.Media"
      title="GWM"
      [...]
    </file>
  </fileLogger>
  [...]
</logSettings>
</configuration>
```

3.2.4.2 Applikationskonfiguration

In der Datei *web.config* sind die Konfigurationseinstellungen für die Applikation in einer XML-Struktur hinterlegt. Die relevanten Parameter⁵ für die Anpassungen der Applikation finden sich in der Sektion *<appSettings>* und werden im Folgenden erläutert.

- *BaseUrl: Die Web-Adresse, unter der die Applikation erreichbar ist, muss hier eingetragen werden. Das ist genau die Adresse, unter der die Webapplikation im IIS angelegt wurde.
Standardwert/Beispiel: *http://icenter.gorba.com/web.media*
- Application.DefaultLanguage: Die Standardsprache der Applikation muss hier festgelegt werden. Es gelten die allgemeinen Kultur- und Länderkürzel-Kombinationen.
Standardwert: *de-CH*
- GoogleAnalyticsWebPropertyID: Soll der Zugriff auf die Webapplikation mittels Google Analytics auswertbar sein, kann hier die Web Property ID eingetragen werden.
Standardwert: keiner
- Content.HasCriteria: Legt fest, ob Content mit Kriterien gepflegt werden kann.
Standardwert: *true*
- Content.IsGlobal: Legt fest, ob bei neu erstelltem Content standardmäßig die Einstellung Global gesetzt ist.
Standardwert: *false*
- *BaseWorkingDirectory: Hier wird das Arbeitsverzeichnis angegeben, unter dem die Applikation ihre Inhalte und Updateverzeichnisse ablegt. Dieses Verzeichnis muss

⁵ Mit * gekennzeichnete Parameter sind essenziell und müssen auf alle Fälle überprüft und angepasst werden.

Schreibrechte für den Netzwerkdienst⁶ haben.

Standardwert/Beispiel: *C:\Temp\Gorba*

- *TFTDataDirectory: Das Verzeichnis, in dem die Protokolldateien des Synchronisationsprozesses mit den Fahrzeugen liegen, wird hier angegeben. Daraus werden die Daten für Monitoring gelesen.
Standardwert/Beispiel: *C:\Temp\Gorba\Monitoring Daten\Transfer*
- *TFTDataServerTagName: Legt den Namen für die DataServer Sektion fest, die für die Auswertung relevant ist.
Standardwert: *dserver11*
- TFTTimeShiftToleranceHours: Gibt die maximale Zeitdifferenz an, die noch als gültig betrachtet wird, falls eine Fahrzeuguhr vor geht und Zeitpunkte in der Zukunft produziert.
Standardwert: *10*
- Monitor_RefreshIntervalSeconds: Gibt das Intervall in Sekunden an, in dem sich die Fahrzeugliste in Monitor Manager aktualisiert.
Standardwert: *300*
- FileHelper_MaxRetries: Legt fest, wie oft versucht werden soll, eine gesperrte Datei zuzugreifen.
Standardwert: *10*
- FileHelper_RetryDelayMilliseconds: Gibt an, wie lange in Millisekunden zwischen wiederholten Zugriffsversuchen auf eine gesperrte Datei gewartet werden soll.
Standardwert: *250*
- Application.SnapshotSavingQuality: Gibt an, mit welcher Qualität, von 0 (Niedrig) bis 100 (Hoch), das Snapshot einer Webseite abgespeichert wird.
Standardwert: *95*
- WebService.EndPoint.Address: Die Adresse des WebServices, welcher die Snapshots einer Webseite generiert.
Beispiel: *<http://webmedia.gorba.com/urlsnapshot.webservice/Service1.asmx>*
- Application.UseUrlSnapshotWebService
Legt fest, ob die SnapShot Generierung von der Applikation ausgelagert über den WebService stattfinden soll. Alternativ wird in der Applikation generiert.
Standardwert: *false*

Es bietet sich an, die Anpassungen nicht direkt in der Datei **web.config** vorzunehmen, sondern in der Datei **web.user.config**⁷ zu überschreiben. Damit bleibt gewährleistet, dass die Datei **web.config** bei Updates und Erweiterungen der Applikation gefahrlos ersetzt

⁶ Es muss das Benutzerkonto Schreibrechte haben, unter dem die Webapplikation ausgeführt wird. Das Konto wird beim zugehörigen Anwendungspool konfiguriert. Im Allgemeinen ist das das lokale Konto NETZWERKDIENTST oder IIS_WPG.

⁷ Die Datei web.user.config wird parallel zur web.config gelegt.

werden kann, da alle Anpassungen an der Konfiguration in einer anderen Datei ausgelagert sind.

Die essenziell für den Betrieb anzupassenden Parameter sind in einer exemplarische web.user.config dargestellt:

```
<?xml version="1.0" encoding="utf-8"?>
<appSettings>
  <add key="BaseUrl" value="http://webmedia.gorba.com:8080/webmedia/" />
  <add key="BaseWorkingDirectory" value="C:\Temp\Gorba\" />
  <add key="TFTDataDirectory" value="C:\Temp\Gorba\Monitoring Daten\Transfer\" />
</appSettings>
```

3.2.4.3 Konfiguration der Datenbankverbindung

Die Verbindung zur Datenbank wird in der Sektion **<connectionStrings>** definiert. Vorgegeben ist eine Verbindung zu einer Datenbank, die auf demselben Serversystem (localhost) liegt und mit den Benutzerdaten arbeitet, die im Datenbank-Skript erzeugt wurden.

Gegebenenfalls können die Teilbereiche mit server, user id, password und database angepasst werden.

Beispielausschnitt:

```
...server=127.0.0.1;user id=gorba_webaccess;password=gorba;database=gorba_web_media...
```

3.2.4.4 Konfigurationsunterschiede IIS6 zu IIS7

Die Applikation benutzt einen speziellen Kommunikationshandler, der je nach IIS-Version⁸ auf unterschiedliche Art und Weise registriert werden muss.

Für den IIS6 ist der Handler in der Sektion <system.web> zu registrieren:

```
<configuration>

[...]

  <system.web>
    <compilation debug="true" />
    <authentication mode="Windows" />
    <pages validateRequest="false" />
    <httpRuntime maxRequestLength="16192" />

    <httpHandlers>
      <add verb="*" path="*.ashx"
        type="BWT.Providers.AsyncCometHandler, BWT.Providers" />
    </httpHandlers>
  </system.web>
</configuration>
```

Für den IIS7 ist der Handler in der Sektion <system.webServer> zu registrieren:

⁸ Der IIS7.5 des Windows Server 2008 R2 verhält sich wie der IIS 7

```
<configuration>

[...]

    <system.webServer>
        <handlers>
            <add verb="*" path="*.ashx" name="AsyncCometHandler"
                type="BWT.Providers.AsyncCometHandler" />
        </handlers>
    </system.webServer>
</configuration>
```

Beide Varianten sind in der Konfigurationsdatei eingepflegt, wobei der IIS6 spezifische Teil auskommentiert ist.

3.2.5 Mehrsprachigkeit

Die Applikation ist für Mehrsprachigkeit ausgelegt. Alle in der Anwendung sichtbaren Texte sind in der Ressourcedatei [translations.xml](#) im Applikationsverzeichnis abgelegt. In dieser Datei ist für jeden Text jeweils eine Übersetzung in den unterstützten Sprachen abgelegt.

Neue Sprachen lassen sich in der Ressourcedatei einfach ergänzen.

Die Auswahl, welche Sprachen in der DropDown beim Login angezeigt werden, ist in der Datei [\[Applikationsverzeichnis\]\bml\Login.bml](#) festgelegt. Neue Sprachen müssen dort ebenfalls ergänzt werden. Die Standardsprache wird in der [web.config](#) mit dem Eintrag [Application.DefaultLanguage](#) konfiguriert.

3.3 URL-Snapshot Dienst

3.3.1 Installation

Der URL-Snapshot Dienst wird als ZIP-Archiv ausgeliefert und muss an einem geeigneten Ort ausgepackt werden. Dieses Verzeichnis ist das [Dienstverzeichnis](#).

Beispiel Pfad

```
C:\Programme\Gorba URL Snapshot Service
```

Der Dienst kann mit einer Commandshell mit Administrationsrechten über die Kommandozeile installiert und deinstalliert werden

Installation

```
[Dienstverzeichnis]\icenter.web.media.UrlSnapshotService.exe /install
```

Deinstallation

```
[Dienstverzeichnis]\icenter.web.media.UrlSnapshotService.exe /uninstall
```

In der Windows-Systemsicht Dienste kann der Dienst [icenter web.media URL Snapshot Service](#) gestartet und gestoppt werden.

3.3.2 Konfiguration

Im Dienstverzeichnis wird das Logging analog zur Webapplikation über die XML-Datei ***Bluehands.Repository.Diagnostics.Log.Config*** konfiguriert.

Die Betriebsparameter sind in der XML-Datei

icenter.web.media.UrlSnapshotService.exe.config definiert. Die relevanten Parameter für die notwendigen Anpassungen finden sich in der Sektion ***<appSettings>*** und werden im Folgenden erläutert.

- ***BaseWorkingDirectory:** Hier wird das Arbeitsverzeichnis angegeben, unter dem der Dienst seine Inhalte und Updateverzeichnisse ablegt. Dieses Verzeichnis muss Schreibrechte für den Netzwerkdienst⁹ haben.
Es muss das selbe Verzeichnis angegeben werden, wie für die Webapplikation.
Standardwert/Beispiel: *C:\Temp\Gorba*
- **URLSnapshotCheckIntervalSeconds:** Der Wert gibt an, in welchem Intervall (in Sekunden) der Dienst prüfen soll, ob Auto-Content auf Aktualität überprüft werden soll.
Standardwert: *60*
- **Application.SnapshotSavingQuality:** Gibt an, mit welcher Qualität, von 0 (Niedrig) bis 100 (Hoch), das Snapshot einer Webseite abgespeichert wird.
Standardwert: *95*
- **WebService.EndPoint.Address:** Die Adresse des WebServices welcher die Snapshots einer Webseite generiert.
Beispiel *http://webmedia.gorba.com/urlsnapshot.webservice/Service1.asmx*
- **Application.UseUrlSnapshotWebService**
Legt fest, ob die SnapShot Generierung vom Dienst ausgelagert über den Webservice stattfinden soll. Alternativ wird im Dienst generiert.
Standardwert: *false*

Die Datenbankverbindung wird analog zur Webapplikation konfiguriert.

3.3.3 Delegierte URL-Snapshot Generierung über URL-Snapshot Webservice

Die Snapshot Generierung verwendet einen „ferngesteuerten“ Internet Explorer (IE) als Serverkomponente. Unter gewissen Umständen kann es vorkommen, dass durch die Verwendung des IE über die Serverautomatisierung nicht immer alle Ressourcen wieder frei gegeben werden, die für einen Snapshot benötigt werden (Memoryleak).

Um eine zu starke Speicherauslastung des Servers vermeiden zu können, kann die Snapshot Generierung an einen separaten Webservice delegiert werden. Der Webservice wird dann unter die Ressourcenüberwachung das Betriebssystems gestellt und kann bei Bedarf automatisch neu gestartet werden.

⁹ Es muss dasjenige Benutzerkonto Schreibrechte haben, unter dem der Dienst ausgeführt wird.

Die Delegation an einen Snapshot WebService ist optional und sollte nur bei beobachteten Speicherproblemen benutzt werden.

3.3.3.1 Installation

Der URL-Snapshot WebService wird als ZIP-Archiv ausgeliefert und muss an einem geeigneten Ort ausgepackt werden. Dieses Verzeichnis ist das **WebServiceverzeichnis**.

Beispiel Pfad

```
C:\inetpub\wwwroot\gorba_icenter_web_media\webservice
```

Jetzt sollte mit dem Internetinformationsdienste-Manager eine neue Anwendung hinzugefügt werden. Aus FQDN und Alias und dem Servicenamen „Service1.asmx“ ergibt sich der vollständige URL, unter dem der WebService erreichbar sein wird.

Der physikalische Pfad zeigt auf das Applikationsverzeichnis, sprich auf das oben beschriebene Verzeichnis, in das das Zip-Archiv entpackt wurde.

Die URL des WebServices wird bei der Konfiguration des URL-Snapshot Dienstes und der Webanwendung benötigt.

Beispiel FQDN

```
icenter.gorba.com
```

Beispiel Alias

```
urlsnapshot.webservice
```

Resultierender URL für den WebService

```
http://icenter.gorba.com/urlsnapshot.webservice/Service1.asmx
```

Falls in den Konfigurationsdateien der Webanwendung oder des URL-Snapshot Dienstes der Eintrag "WebService.EndPoint.Address" nicht mit der URL des WebServices übereinstimmt sollte dies in der Web~ bzw. App.User.Config angepasst werden.

3.3.3.2 AnwendungsPool

Es empfiehlt sich den Webservice in einen eigenen Anwendungs-Pool zu verlagern (unter Grundeinstellungen) um die Verwendung von Ressourcen anpassen zu können.

Wiederverwendung bei starker Nutzung des Arbeitsspeichers:

Im Dialog „Erweiterte Einstellungen“ des Anwendungspools für den WebService unter der Gruppierung „Wiederverwendung“ setze man den Wert für die Einstellung „Limit für den privaten Speicher (KB)“ auf 200000 (oder einen anderen akzeptablen Wert).

Dies hat zur Folge, dass der speicherlastige Vorgang der Snapshot-Generierung, welcher im WebService ausgelagert ist, nie mehr Speicher blockiert, als eingestellt wurde.

3.4 Übersicht

Zusammenfassend sind die wichtigsten Eckdaten der Standardinstallation festgehalten:

- Serverhardware: 2-CPU Serversystem, 2 GB Ram, 500GB Festplattenkapazität
- Betriebssystem: Windows Server 2003 x86
- Laufzeitumgebung: .NET Framework 2.0
- Datenbank: MySQL 5.1
- Zusätzliche Software:
- MySQL .NET Connector 6.5
- Codec-Pack

Leserechte für die Webapplikation sind auf folgenden Verzeichnissen notwendig:

- [Monitoring Datenverzeichnis]

Schreibrechte für die Webapplikation und den URL Snapshot Dienst sind auf folgenden Verzeichnissen notwendig:

- [Protokollverzeichnis]
- [BaseWorkingDirectory]
- [Applikationsverzeichnis]\VehicleData

Initiale Zugangsdaten:

- root: 1234 (-5@2p%FB%N_x)

4 Einschränkungen

Addblocker blockieren ggf. die Bilder. Man muss die Site manuell vom ABP ausschließen.

5 Ergänzungen

5.1 Gesicherte Übertragung

Wenn die Anwendung nicht nur im Intranet sondern auch öffentlich zugänglich über das Internet genutzt wird, empfehlen wir zum Schutz von Benutzername/Passwort die Verwendung von **HTTPS** als Kommunikationsprotokoll. Die Anwendung selbst implementiert keinen eigenen Verschlüsselungsmechanismus.

5.2 Update von Version 1.0 auf Version 1.1

Beim Update einer bestehenden Version auf den Branch „classic“ in der Version 1.1 gilt es folgende Punkte zu beachten:

- Das .NET Framework 2.0 muss installiert sein, ggf. nachinstallieren.
- Der MySQL Connector für ADO.NET muss aktualisiert werden auf Version 6.5 oder neuer.
- Der bisher verwendete Anwendungspool (Application pool) der Website muss auf das .NET Framework in der Version 2.0 umgestellt werden.