Handbuch AlarmWorkflow

Stand: Sonntag, 14. Dezember 2014

Inhaltsverzeichnis

[1 Entwicklungsstadium 4](#_Toc406767044)

[2 Installation 4](#_Toc406767045)

[2.1 Installation MySQL Server 4](#_Toc406767046)

[2.1.1 Installation des MySQL Servers 4](#_Toc406767047)

[2.1.2 Einspielen der Datenbank. 8](#_Toc406767048)

[2.2 Faxempfang 10](#_Toc406767049)

[2.2.1 Installation der Funktion „Windows-Fax und -Scan“ 10](#_Toc406767050)

[2.2.2 Konfiguration der Funktion „Windows-Fax und -Scan“ 11](#_Toc406767051)

[2.3 Installation von Updates 15](#_Toc406767052)

[2.4 Installation von Github 15](#_Toc406767053)

[3 Allgemeine Konfiguration 18](#_Toc406767054)

[3.1 Backend/Service Konfiguration 18](#_Toc406767055)

[3.2 Konfigurationseditor 20](#_Toc406767056)

[3.3 Benutzerdatenverzeichnis 21](#_Toc406767057)

[3.4 Generell 21](#_Toc406767058)

[3.4.1 Feuerwehr 21](#_Toc406767059)

[3.4.2 Auswertung 21](#_Toc406767060)

[3.4.3 Globale Druckkonfiguration 21](#_Toc406767061)

[3.5 Adressbuch 21](#_Toc406767062)

[4 Alarmquellen 22](#_Toc406767063)

[4.1 Fax 22](#_Toc406767064)

[4.2 E-Mail 23](#_Toc406767065)

[4.3 SMS 23](#_Toc406767066)

[5 Backend 24](#_Toc406767067)

[5.1 Engine 24](#_Toc406767068)

[5.2 Verwaltung 24](#_Toc406767069)

[5.2.1 Einsatzmittelkonfiguration 24](#_Toc406767070)

[6 Aufgaben 25](#_Toc406767071)

[6.1 Alarmdetails-Druck 25](#_Toc406767072)

[6.2 Alarmquellendruck 25](#_Toc406767073)

[6.3 Display Wake-Up 26](#_Toc406767074)

[6.4 eAlarm 26](#_Toc406767075)

[6.5 Einsatzdateiexporter 26](#_Toc406767076)

[6.6 E-Mail-Versand 26](#_Toc406767077)

[6.7 Externe Schleifeninfos 27](#_Toc406767078)

[6.7.1 Integration mit Drittprogrammen 27](#_Toc406767079)

[6.8 Push-Versand 27](#_Toc406767080)

[6.9 SMS-Versand 28](#_Toc406767081)

[7 ASP.Net Website 28](#_Toc406767082)

[7.1 Installation 28](#_Toc406767083)

[7.1.1 Installation des IIS - Servers 28](#_Toc406767084)

[7.1.2 Installation der Website 31](#_Toc406767085)

[7.2 Konfiguration 32](#_Toc406767086)

[7.2.1 32-bit Modus aktivieren 32](#_Toc406767087)

[7.3 Modifikation 33](#_Toc406767088)

[8 Anwendungen 34](#_Toc406767089)

[8.1 eAlarm 34](#_Toc406767090)

[8.1.1 App herunterladen und installieren 34](#_Toc406767091)

[8.1.2 Einrichtung 34](#_Toc406767092)

[8.2 Dispositions-tool 34](#_Toc406767093)

[8.3 Windows UI 34](#_Toc406767094)

[9 Glossar 35](#_Toc406767095)

[9.1 Definitionen 35](#_Toc406767096)

[9.2 Objekt-Ausdruck 35](#_Toc406767097)

[9.2.1 Eigene Skriptlogik 35](#_Toc406767098)

[9.2.2 Testen des Ausdrucks 37](#_Toc406767099)

# Entwicklungsstadium

**AlarmWorkflow** befindet sich aktuell im **Beta-Stadium**.

Dies bedeutet v.a., dass sich die Software in einem Zustand steter Entwicklung befindet, nicht in allen Fällen in unter allen Umständen reibungslos funktioniert und sich Features bzw. die Architektur des Systems von Version zu Version teils drastisch verändern können.

Um eine gute und erfolgreiche Entwicklung des Programms zu ermöglichen bitten wir Sie, dies zu verstehen und bei Fehlern sofort die Entwickler zu benachrichtigen, damit sich diese um die Fehlerbehebung bemühen können.

Weiterhin bitten wir Sie inständig, die **Release notes**, die bei jeder Veröffentlichung mit angehängt werden, zu beachten! Fehler, die aus Nichtbeachtung der   
Release notes entstehen können wir nicht in jedem Fall betreuen.

Vielen Dank und viel Freude mit der Software wünscht Ihnen

Das AlarmWorkflow-Team

# Installation

## Installation MySQL Server

### Installation des MySQL Servers

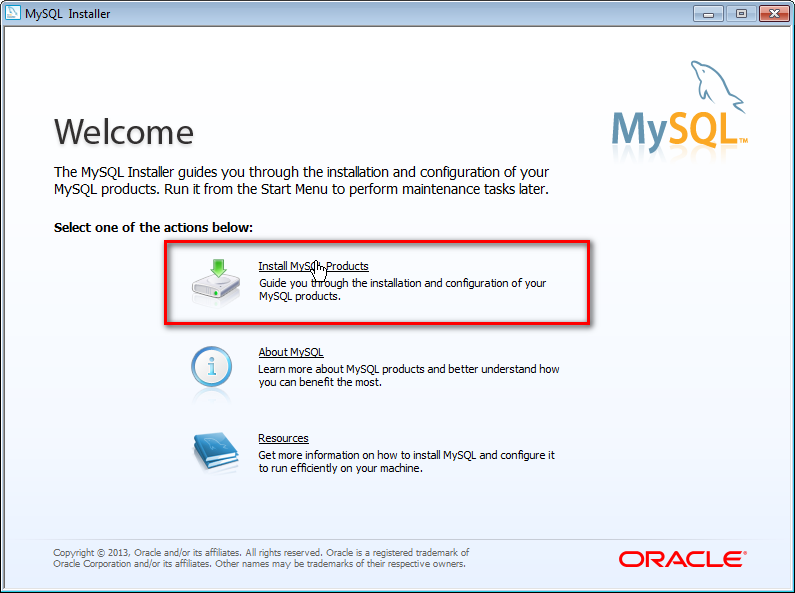
Im Folgenden wird für die Installation der MySQL Datenbank ein   
Windows 7 Professional 32bit verwendet. Die Installation unter Windows XP oder Windows 8 verläuft identisch.

Zu Beginn beziehen Sie bitte den „**MySQL Installer for Windows“** von [MySQL Server Download](http://dev.mysql.com/downloads/installer/).

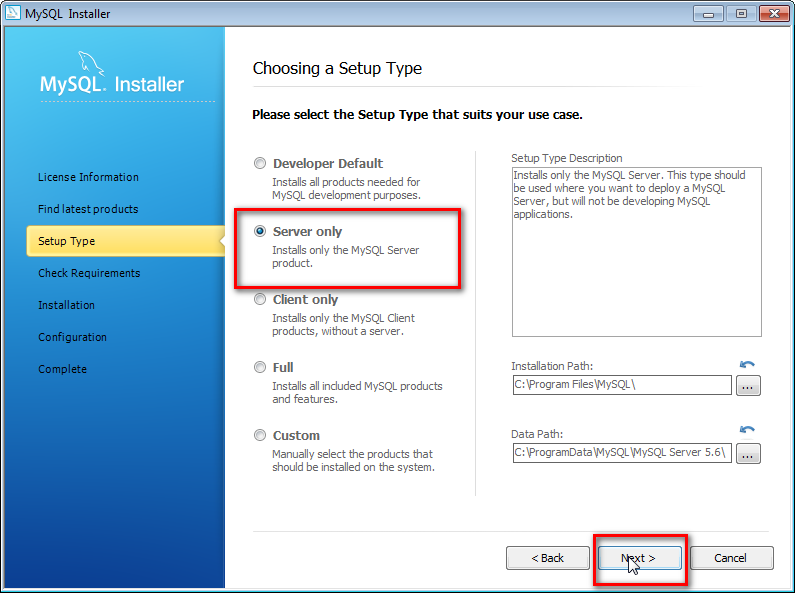
Die heruntergeladene **Datei** bitte nun **ausführen**.

Nun **folgen** Sie dem **Installationsassistenten**.

Zuerst wählen Sie „**Install MySQL Products“**aus.

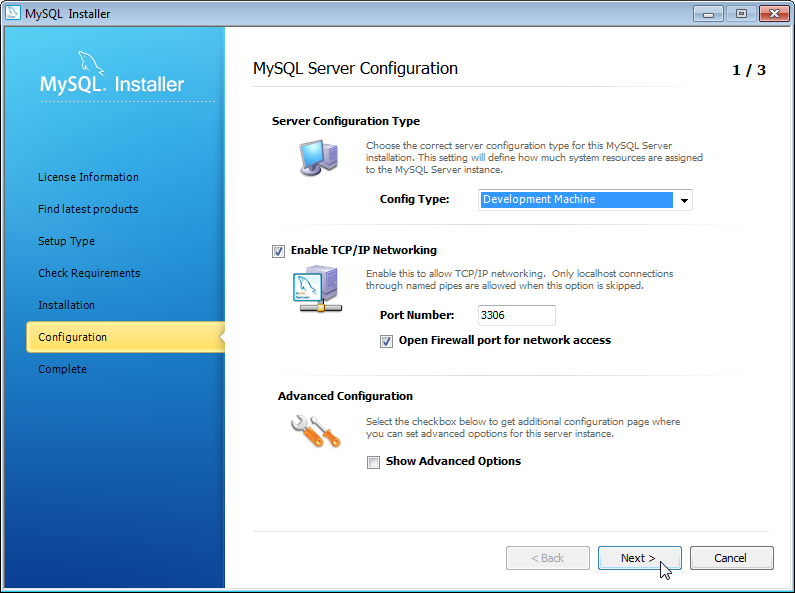


Danach **akzeptieren** Sie die **Lizenzbedingungen** und lassen den Installer im Internet nach Updates suchen. Als „**Setup Type“** wird zu Gunsten des Speicherplatzes und der Downloadgröße „**Server only“** empfohlen.

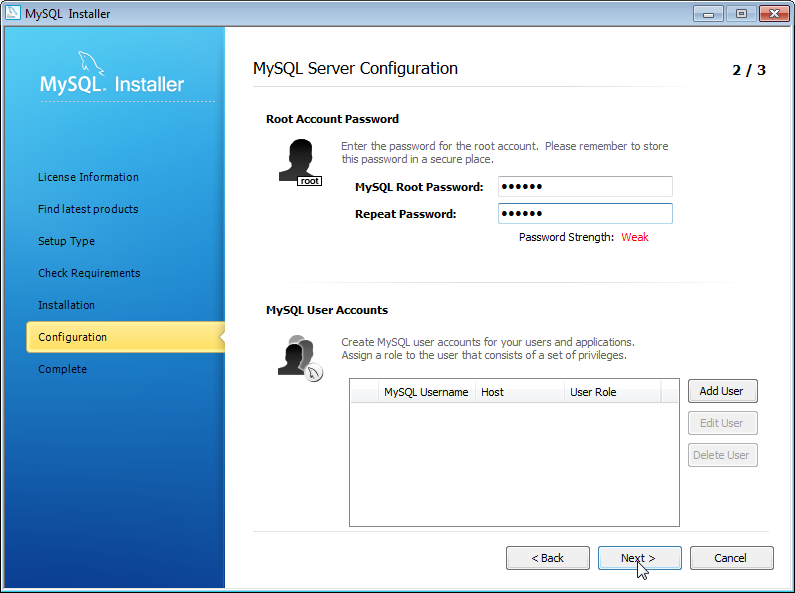


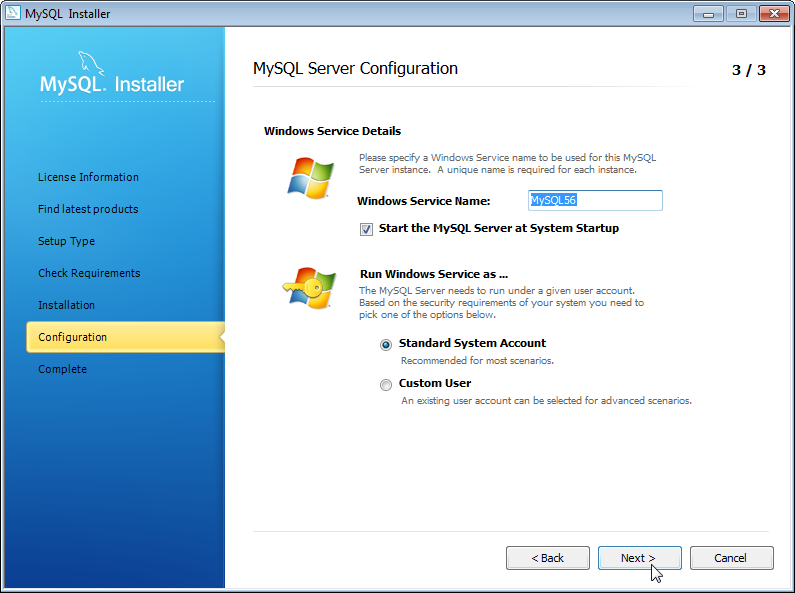
Nun lädt der Installer die notwendigen Daten herunter und installiert diese.

Anschließend werden Sie zur Konfiguration des Servers aufgefordert. Im Grunde können überall die **Standardeinstellungen** belassen werden.



Im folgenden Fenster ist **standardmäßig** **kein** **Passwort** vergeben. Es wird jedoch dringend **empfohlen** ein **Passwort** zu **vergeben**.

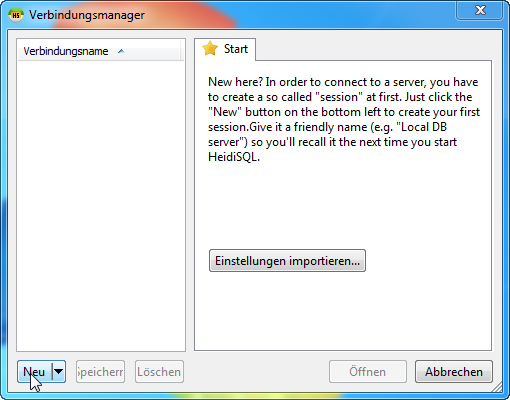




Nach Abschluss der Installation ist der **MySQL** Server **einsatzbereit**, bedarf für **AlarmWorkflow** jedoch noch **einiger Schritte**.

### Einspielen der Datenbank.

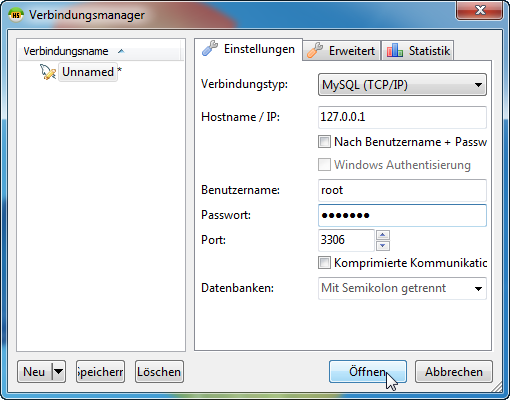
Sie müssen, bevor AlarmWorkflow in die Datenbank schreiben kann, zuerst die notwendige Datenbank und die notwendigen Tabellen erstellen. Dies geht am einfachsten über [HeidiSQL](http://www.heidisql.com/download.php) und mithilfe eines Scripts welches Sie im [GitHub Repository](https://github.com/OpenFireSource/AlarmWorkflow/blob/master/Database/alarmworkflow.sql) finden.



Mit dem soeben heruntergeladenen Programm HeidiSQL **verbinden** Sie sich nun auf die installierte **MySQL** Datenbank. Hier zu geben Sie als:

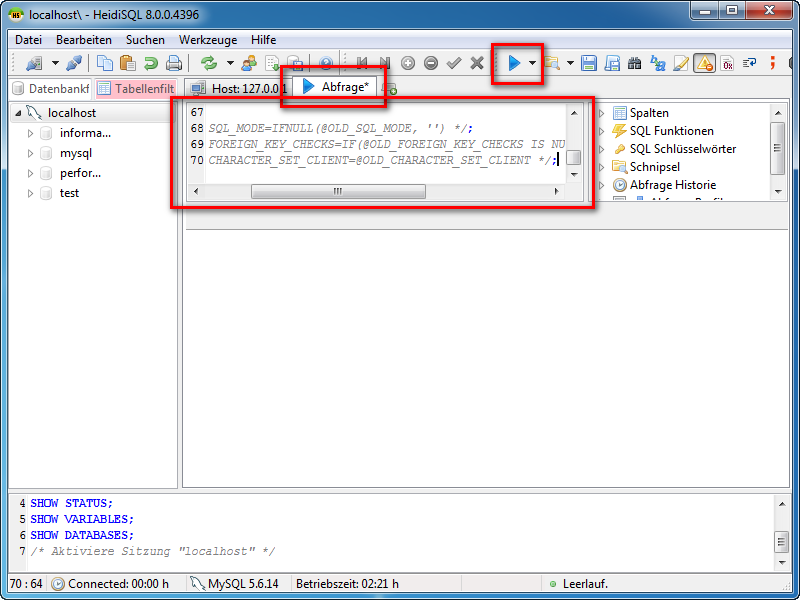
* **IP** 127.0.0.1
* **Benutzername**: root
* **Passwort**: *Das von Ihnen definierte Passwort*

an.



Nach erfolgreichem Verbinden sollten Sie nun die Möglichkeit haben das vom GitHub Repository bezogene Script **auszuführen**.

Hierzu wechseln Sie in den Reiter „**Abfrage“** und **fügen** dort den Inhalt des **Scriptes** **ein**. Das Script können Sie nun mittels **F9** oder dem „**Play-Button**“ **ausführen**.



Gegebenenfalls meldet das Programm, das die Abfrage **Warnungen** liefert. Dies können Sie **ignorieren**.

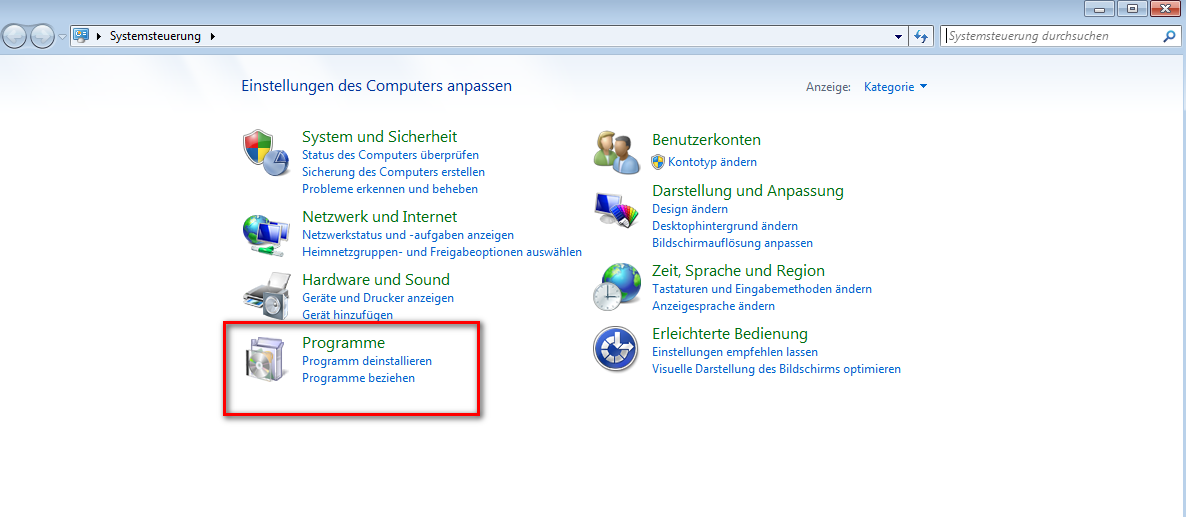
Nun ist die MySQL für AlarmWorkflow einsatzbereit.

## Faxempfang

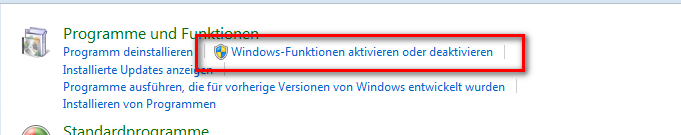
### Installation der Funktion „Windows-Fax und -Scan“

Im Folgenden wird für die Installation der Windows-Fax und -Scan Funktion ein Windows 7 Professional 32bit verwendet. Für Unterstützung unter anderen Betriebssystemen wenden Sie sich bitte an das Team.

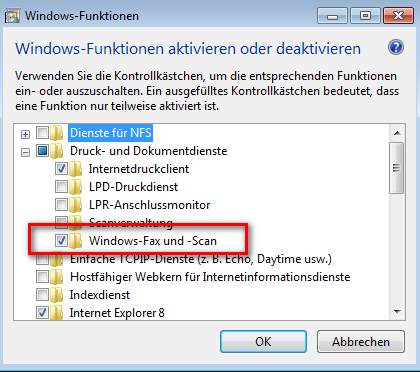
Öffnen Sie die „**Systemsteuerung“.**



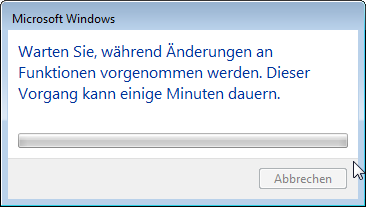
Dort wählen Sie den Menüpunkt „**Programme**“ und gehen nun auf „**Windows-Funktionen**“ aktivieren.



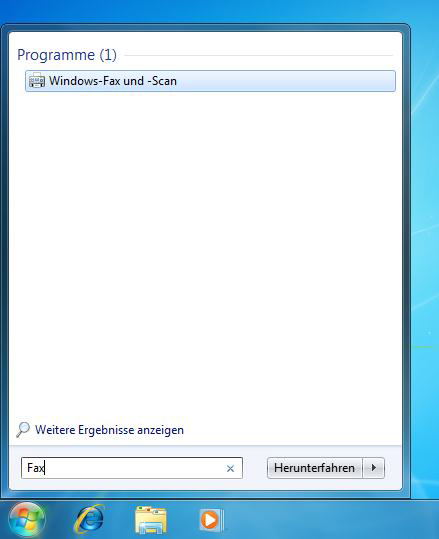
Nun wählen Sie die Funktion „**Windows-Fax und -Scan**“ aus.

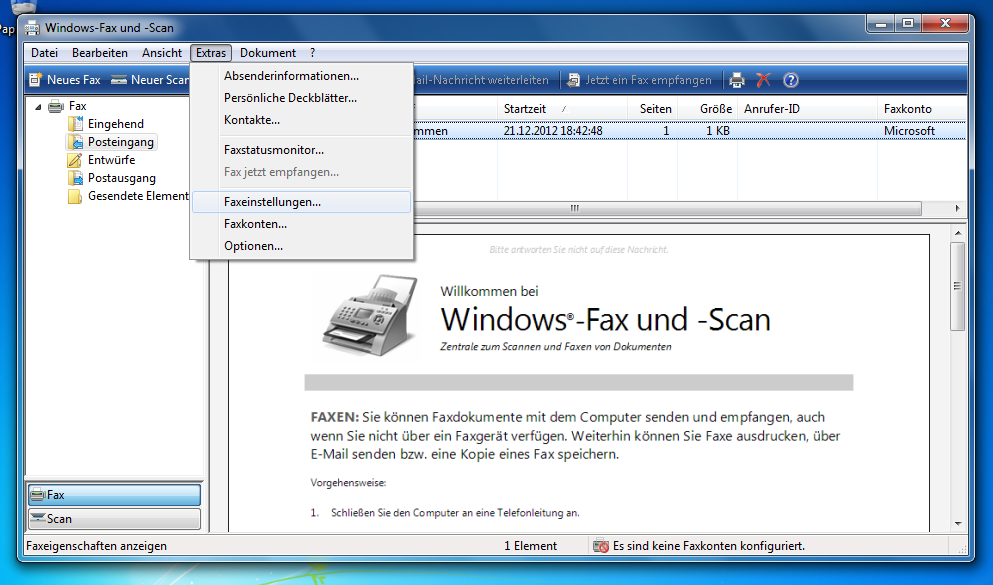


Nach dem Bestätigen mit „**OK**“ haben Sie die Funktion installiert und können diese nun **konfigurieren**.



### Konfiguration der Funktion „Windows-Fax und -Scan“

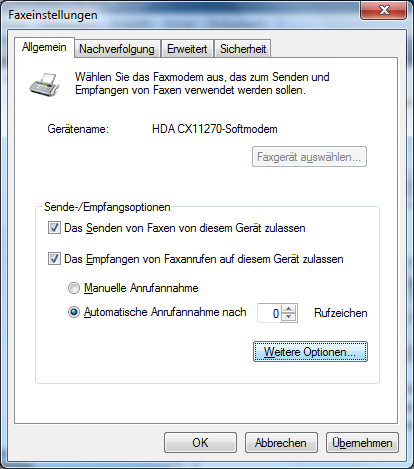




Öffnen Sie die Windows-Fax und -Scan Funktion.

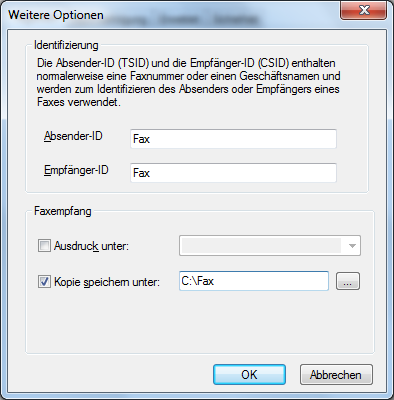
Wählen sie unter „**Extras**“ „**Faxeinstellungen**“ aus.

Aktivieren Sie im darauf folgenden Dialog „**Das Empfangen von Faxanrufen auf diesem Gerät zulassen**“. Außerdem wird empfohlen „**Automatische Anrufannahme nach 0 Rufzeichen**“ zu aktivieren



Des Weiteren ist unter „**Weitere Optionen**“ einzustellen, wo das empfangene Fax **abzuspeichern** ist.

In diesem Fall wurde „**C:\Fax**“ gewählt, was auch unserem Standard entspricht.



Der **Faxempfang** ist hiermit für **AlarmWorkflow** vorbereitet.

## Installation von Updates

Es wird empfohlen vor der Installation eines Updates die alte Version zu deinstallieren bzw. die alten Dateien zu löschen und im Anschluss die neue Version zu installieren.

Die *Backend.config* und die OCR Software können bislang beibehalten werden.

TODO Sicherung erklären!

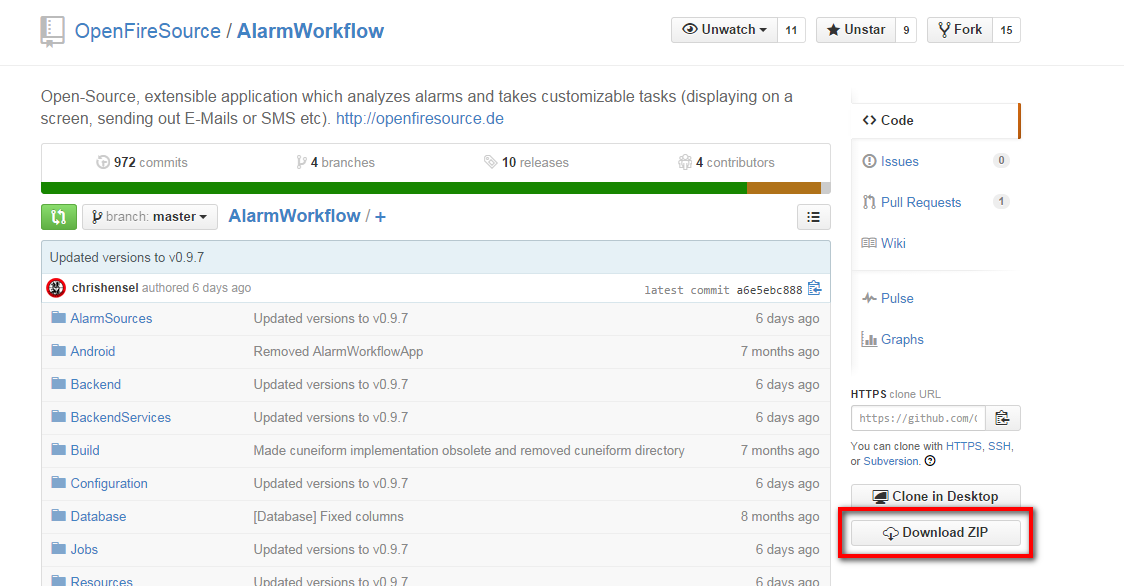
## Installation von Github

Die im Folgenden beschrieben Schritte sind nicht für den normalen Anwender gedacht. Diese sind ausschließlich für Benutzer gedacht welche eine noch nicht veröffentlichte – in der Entwicklung befindliche - Version verwenden möchten! Der Support zu diesen Versionen findet nur eingeschränkt statt.

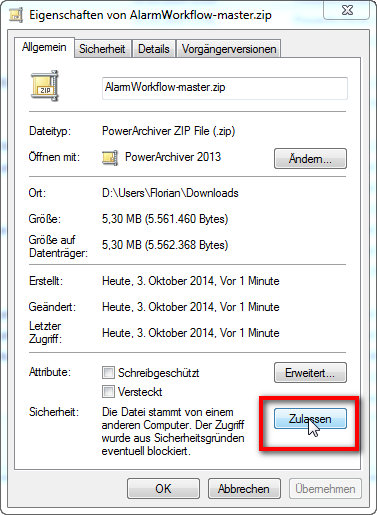
Zur Installation der aktuellen Entwicklungsversion ist das manuelle Erstellen des Sourcecodes notwendig.

Dieser findet sich unter <https://github.com/OpenFireSource/AlarmWorkflow>.

Zum Herunterladen empfiehlt es sich die von Github zur Verfügung gestellte Zip Funktion zu verwenden. Diese findet sich hier:



Die nun heruntergeladene Datei müssen Sie vor dem Entpacken erst „Zulassen“ da andernfalls Probleme beim Starten des Services auftreten können.

Hierzu müssen Sie die Eigenschaften des Zip-Archives über das Kontextmenü der Datei aufrufen und unter Allgemein den Button Zulassen klicken.

Nach dem Zulassen können Sie das Fenster mit Ok schließen und das Archiv entpacken.

In dem entpackten Ordner findet sich eine Datei

„BuildWIndows.bat“

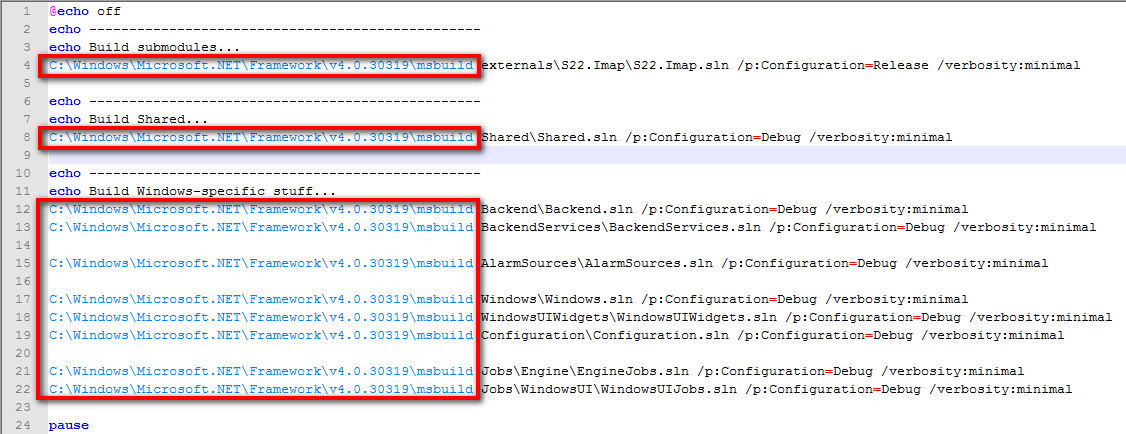
Vor Ausführen der Datei ist in der Regel eine weitere Anpassung von Nöten. Dazu öffnen Sie die Datei bitte mit einem Editor Ihrer Wahl (Editor, Notepad++,…).

Die „Einträge“ msbuild bitte durch

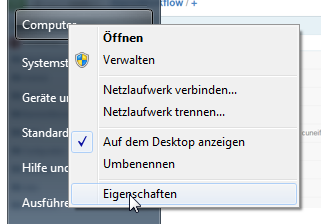
C:\Windows\Microsoft.NET\Framework\v4.0.30319\msbuild

ersetzen und die Datei abspeichern.

Nach Bearbeiten der Datei sollte der Inhalt dieser so aussehen:



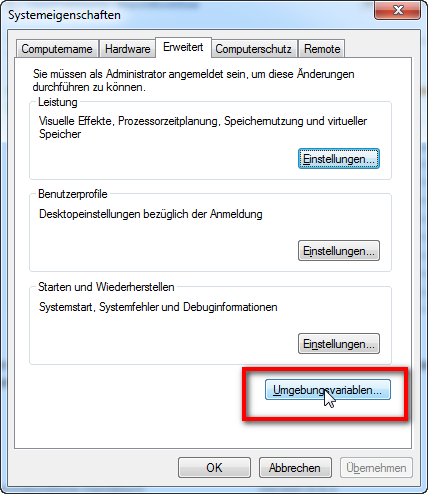
Nun können Sie die Datei ausführen. Im Anschluss daran finden sich im Ordner Build die gewohnten exe-Dateien.

Falls Sie regelmäßig eine Version aus Github erstellen möchten empfiehlt es sich die sog. Path Variable anzupassen.

Öfffnen Sie hierzu die Eigenschaften des Computers:

Hier findet sich an der linken Seite der Eintrag „Erweiterte Systemeinstellungen“

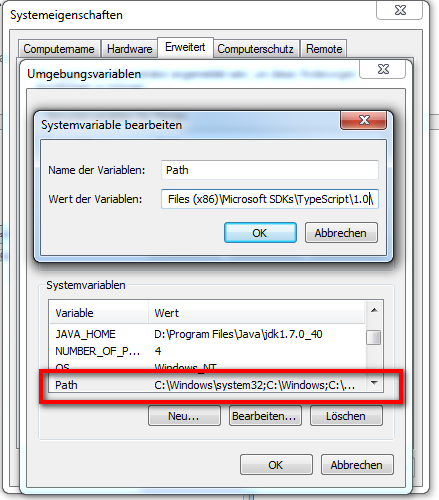
Unter Erweitert findet sich der Eintrag Umgebungsvariablen.



In dem geöffneten Fenster suchen Sie nun unter Systemvariablen den Eintrag “Path”. Diesen Eintrag bearbeiten Sie nun in dem Sie den Wert der Variable um

„;C:\Windows\Microsoft.NET\Framework\v4.0.30319\“

Bitte beachten Sie, dass zwischen dem vorhergehenden Eintrag und dem neuen Eintrag ein Semikolon (;) ist. Nun können Sie alle Fenster mit Ok schließen und den PC neustarten. Danach ist eine Ausführung der BuildWindows.bat in Zukunft ohne Anpassung möglich.



# Allgemeine Konfiguration

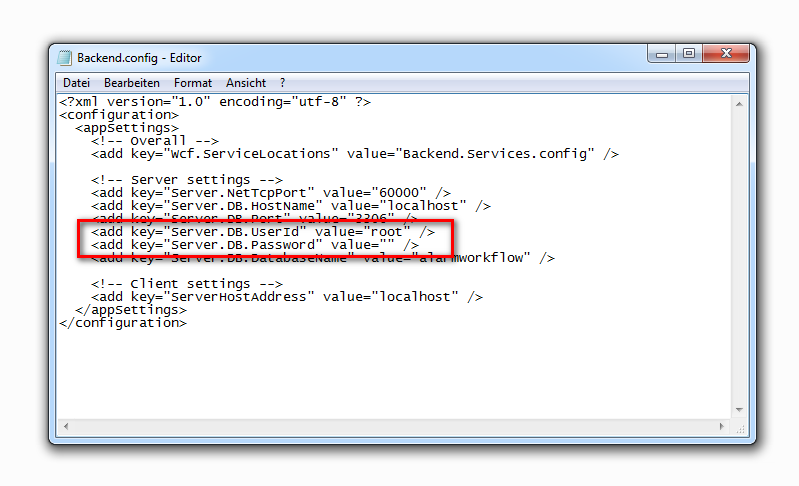
## Backend/Service Konfiguration

Zum Starten bzw. Einrichten des Services ist es notwendig in der

Backend.config

Den MySql Benutzernamen und das dazugehörige Passwort zu hinterlegen.

Hierzu öffnen Sie bitte diese Datei mit einem Texteditor ihrer Wahl und bearbeiten die im folgenden Bild markierten Zeilen.

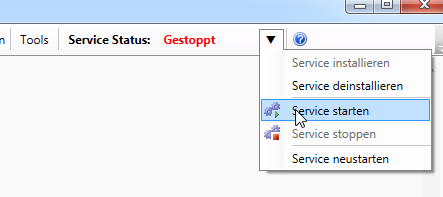


Um weitere Konfigurationen vornehmen zu können ist die **Installation** des **Services** von Nöten.

Hierzu starten Sie bitte den Konfigurationseditor erstmals und **installieren** über die Menüleiste den **Service** mit einem Klick auf **„Nicht installiert“**



Nach erfolgreicher Installation sollte in dem Dropdown Menu nun die Möglichkeit bestehen den **Service** zu **starten**.

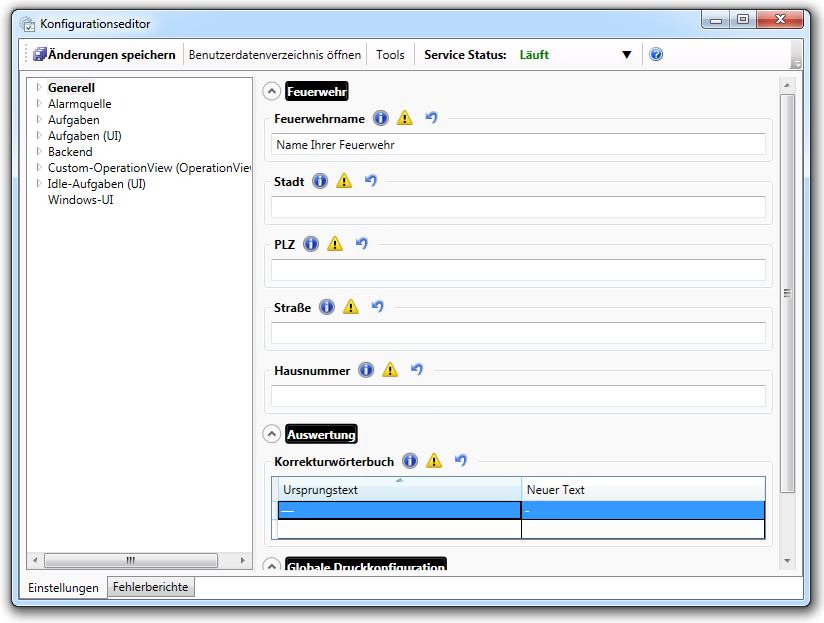


Im Anschluss hieran **starten** Sie bitte den **Konfigurationseditor** **neu**.

Nun sind alle Schritte erledigt um **AlarmWorkflow** **funktionsbereit** zu machen.

## Konfigurationseditor

Um Änderungen an den Einstellungen von AlarmWorkflow vorzunehmen, müssen Sie den Konfigurationseditor benutzen.

Das Programm reagiert auf Änderungen der Einstellungen in unterschiedlicher Weise. Manche Änderungen erfordern es, dass der Dienst bzw. die Clients neugestartet werden müssen.

Stand 06.03.2014: Nicht alle Einstellungen reagieren sofort auf eine Änderung. Es wird daran gearbeitet, weitestgehend alle Einstellungen sensibel auf Änderungen zu machen, soweit möglich.

Der Konfigurationseditor stellt neben den Namen der Einstellungen grafisch dar, ob es sich bei der angegebenen Einstellung um eine „dynamische“ oder eine „statische“ Einstellung handelt:

|  |  |  |
| --- | --- | --- |
| Piktogramm | Typ | Beschreibung |
| D:\Users\Florian\Entwicklung\AlarmWorkflow_Proxy\Windows\Configuration\AlarmWorkflow.Windows.Configuration\Images\dynamicsetting_16.png | Dynamisch | Änderungen werden sofort oder mindestens bei der nächstbesten Gelegenheit übernommen, ohne einen Dienstneustart zu erfordern. |
| D:\Users\Florian\Entwicklung\AlarmWorkflow_Proxy\Windows\Configuration\AlarmWorkflow.Windows.Configuration\Images\staticsetting_16.png | Statisch | Änderungen erfordern zwingend einen Dienstneustart. |

## Benutzerdatenverzeichnis

Hier im Dokument und auch im Forum werden Sie öfters das Wort „Benutzerdatenverzeichnis“ hören. Dieses Verzeichnis enthält z. B. diverse Einstellungen, temporäre Dateien und Logdateien, die u. a. für das Team im Rahmen der Fehlersuche von Interesse sein können. Sie müssen prinzipiell nur wissen, wie Sie dorthin kommen; der tatsächliche Pfad ist zuerst nicht von Relevanz.

Um zu diesem Verzeichnis zu gelangen, müssen Sie einfach nur den Konfigurationseditor öffnen und in der Toolbar auf „**Benutzerdatenverzeichnis öffnen**“ klicken, und schon sind Sie da.

## Generell

### Feuerwehr

Unter dieser Kategorie finden sich Einstellungen, welche zum Beispiel von der Routenplanung in der Webseite oder in der UI verwendet werden.

### Auswertung

Hier besteht die Möglichkeit, Zeichenketten vor der Auswertung durch definierte andere zu ersetzen. Dies wird häufig notwendig, um die natürlichen Falscherkennungen der OCR-Software auszugleichen.

Für erfahrene Benutzer besteht die Möglichkeit im Kontextmenü der Tabelle eine Ersetzung mittels *Regulären Ausdrücken[[1]](#footnote-2)* zu aktvieren. Diese Einstellung wird nur erfahrenen Benutzern empfohlen. Siehe auch unter 9.1.

### Globale Druckkonfiguration

An dieser Stelle müssen die Drucker eingerichtet werden. Im Anschluss an die Konfiguration ist diese nochmals separat im Konfigurations-Editor zu speichern und der Dienst neu zu starten. Die geänderten Werte werden erst nach einem erneuten Start des Konfigurationseditors sichtbar.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Aktiv | Hiermit kann die Konfiguration schnell deaktiviert werden. |
| Name | Hier kann der im Konfigurationseditor verwendete Name definiert werden. |
| Serverpfad | Nur relevant, wenn es sich im einen Netzwerkdrucker handelt. |
| Druckername | Hier ist der Name des Druckers, wie er in Windows angezeigt wird, zu verwenden. |
| Kopien | Anzahl der zu erstellenden Ausdrucke. |
| Sequentiell drucken | Diese Funktion ist zu aktivieren, sollte der Drucker trotz einer Änderung der Anzahl der Kopien nicht die gewünschte Anzahl drucken. I.d.R. aber ist es nicht erforderlich, diesen Wert zu ändern. |

## Adressbuch

Über den Menüpunkt Generell -> Adressbuch können Sie das Adressbuch für die zu alarmierenden Empfänger auswählen.

Nachdem der Adressbuch-Editor geschlossen wird müssen Sie die Änderungen speichern.

Im Konfigurationseditor ist erneut speichern zu wählen und der Dienst neu zu starten damit die Änderungen greifen.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Telefonnummer | Hier ist die Rufnummer welche für den SMS Versand einzutragen. Es kann das nationale oder internationale (z.B. +49) Format verwendet werden.  Für Groupalarm und Rettalarm ist hier die zu alarmierende Gruppe oder Liste einzutragen.  Beispiel:  Alarm an Liste 16, dann bitte eine 16 eintragenAlarm an Gruppe 25, dann bitte +25 eintragen |
| E-Mail Adresse | E-Mailadresse des Empfängers |
| Schleifenempfänger | Hier den Schleifencode eintragen der in der Aufgabe Externe Schleifeninfos verwendet wird |
| Push-Empfänger | Hier ist der zu verwendende Push Job auszuwählen und die jeweilige Dienste/Geräte ID einzutragen |

# Alarmquellen

## Fax

Konfiguration der Fax-Alarmquelle. Diese Alarmquelle wandelt mittels einer eingestellten OCR-Software empfangene Faxe in Text-Dateien um, welche von speziell für das Fax-Layout angepasste Parser analysiert und interpretiert werden können.

Bitte beachten Sie das nur nativ - als **tif**-Datei - empfange Faxe vom   
AlarmWorkflow-Team angestrebte Ergebnisse erzielen können. Eingescannte,   
Abfotografierte oder aus anderen Dateien konvertierte Faxe können die Ergebnisse verfälschen oder eine Auswertung unmöglich machen.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Faxeingang-Verzeichnis | Verzeichnis, in das der Faxdienst seine Faxe ablegt bzw. das von AlarmWorkflow überwacht werden soll. |
| Archiv-Verzeichnis | Hier werden die verarbeiteten Faxe abgelegt. Die Dateien werden nach dem Verarbeitungszeitpunkt benannt. |
| Analyse-Verzeichnis | Hier befinden sich die ausgewerteten Dateien. |
| Pfad | Dies kann leer bleiben, wenn die OCR Software sich im Anwendungsverzeichnis von Alarmworkflow befindet.  Standardmäßig |
| Parser-Kennung | Der zur ILS passende Parser ist auszuwählen. |
| Whitelist | Das Erkennen eines der Wörter ist notwendig, sodass ein Alarm ausgelöst wird. Sollte kein Wort vorhanden sein wird die Einstellung ignoriert. |
| Blacklist | Beim Erkennen eines dieser Wörter löst das Fax keinen Alarm aus. |

## E-Mail

Diese Alarmquelle bietet die Möglichkeit Alarm-Emails von der Leistelle zu empfangen und den Inhalt dieser, bzw. die Anhänge dieser auszuwerten und zu verarbeiten.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Servername | Die Adresse des Mailservers, der zum Empfangen der Mails dient.  Der Mailserver muss das IMAP-Protokoll unterstützten! |
| Port | Die Portnummer, auf dem der Mailserver Mails zur Verfügung stellt. |
| SSL | Bestimmt, ob eine sichere Verbindung mit dem Server hergestellt werden soll. Ggf. weicht dann die Portnummer ab! |
| Zeichensatztabelle | Die Nummer der Codepage, die für eingehende Mails verwendet werden soll. I.d.R. ist keine Änderung erforderlich. |
| Benutzername | Der für die Authentifizierung verwendete Benutzername. |
| Passwort | Das für die Authentifizierung verwendete Passwort. |
| Betreff | Die Zeichenkette welche im Betreff enthalten sein muss um die Mail als Alarm-Email zu verarbeiten. |
| Absender (Mailadresse) | Die Zeichenkette welche im Absender enthalten sein muss um die Mail als Alarm-Email zu verarbeiten. |
| Parser-Kennung | Der für die Textanalyse zu verwendende Parser. |
| Anhänge auswerten | Gibt an, ob statt des Mailkörpers ein spezieller Anhang ausgewertet werden soll. |
| Anhangname | Der Name des Anhangs, der ausgewertet werden soll. |

## SMS

TODO: Zusammenhang mit SMS-App (siehe Abschnitt 8.1.1) erklären!!!

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Parser | Der für die Textanalyse zu verwendende Parser. |

# Backend

## Engine

Hier finden Sie die Möglichkeit die Aufgaben und die Alarmquellen zu (de)aktiveren.

Des Weiteren findet sich hier eine Einstellung um doppelte Alarme zu unterbinden. Dies wird anhand der Einsatznummer erkannt.

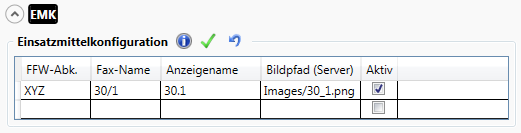
## Verwaltung

### Einsatzmittelkonfiguration

Die *Einsatzmittelkonfiguration* (abgekürzt *EMK*) dient vereinfacht gesagt dazu, bei einem Alarm nur die Informationen über die angeforderten Einsatzmittel so anzuzeigen, wie dies gewünscht ist. In der Regel werden bei einem Alarm nicht nur die eigene(n) Feuerwehr(en) alarmiert, sondern oft zusätzlich auch Wehren aus dem Nachbarort (oder weiter entfernt). Die EMK filtert nun aus diesen Informationen die „eigenen“ Fahrzeuge heraus und stellt sie übersichtlich dar, sodass es leicht erkennbar ist, welche Einsatzmittel disponiert werden *sollten*[[2]](#footnote-3).

Im Folgenden werden die Einstellungen beschrieben.

#### Editor



Der Editor stellt sich als einfache Tabelle dar, in der pro Zeile folgende Einstellungen vorgenommen werden können. Eine Zeile repräsentiert hierbei ein bestimmtes Einsatzmittel.

|  |  |
| --- | --- |
| Spalte | Beschreibung |
| FFW-Abk. | Der erste Teil des Einsatzmittels. Der erste Teil des Einsatzmittelnamens. |
| Fax-Name | Der Name des Einsatzmittels. Der zweite Teil des Einsatzmittelnamens.  Ein Einsatzmittel wird vollständig z. B. „FFW ABC 30/1“ genannt. Die Abkürzung bezieht sich also auf den Text, der Ihre Feuerwehr repräsentiert. Der Fax-Name bezieht sich folglich auf den Text, der das Einsatzmittel repräsentiert.  Eine Aufteilung ist notwendig, um die Erkennungssicherheit im Programm zu steigern. |
| Anzeigename | Der Text, der angezeigt werden soll. |
| Bildpfad (Server) | Der Pfad zur Bilddatei, die das Einsatzmittel repräsentieren soll.  Hierbei ist wichtig zu beachten, dass der Pfad relativ zum „Benutzerdatenverzeichnis“ auf dem **Server** interpretiert wird. |
| Aktiv | Markiert den Eintrag als „Aktiv“. Inaktive Einträge werden vom Programm ignoriert. |

**Hinweis:** Sind keine Einträge vorhanden oder sind alle Einträge inaktiv, so werden die Einstellungen ignoriert und kein besonderes Verhalten angewandt.

# Aufgaben

AlarmWorkflow verfügt über eine Vielzahl von verschiedenen Aufgaben, welche nach Alarmeingang ausgeführt werden können.

Diese Aufgaben sollen im Folgenden kurz beschrieben und erklärt werden.

## Alarmdetails-Druck

Der Alarmdetails-Druck ist eine Aufgabe, welche aus einem eingehenden Alarm eine benutzerdefinierte Zusammenfassung generiert und ausdruckt. Diese kann dann z. B. auf den einzelnen Fahrzeugen mitgenommen und u. a. zur Orientierung verwendet werden.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Druckvorlagendatei | Die HTML-Datei, welche das Layout des Drucks enthält. Die Datei kann *Objektausdrücke* enthalten. |
| Druckwege | Die Druckerwege, auf denen der generierte Bericht gedruckt werden soll. |

## Alarmquellendruck

Der Alarmquellendruck ist eine Aufgabe, welche es ermöglicht, das empfange Fax auf definierten Druckwegen auszudrucken. Das Fax wird auf das Standardpapierformat des Druckers skaliert um das Fehlen von Informationen durch abgeschnittene Texte zu verhindern.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Druckwege | Die Druckerwege, auf denen das empfangene Fax ausgedruckt werden soll. |

## Display Wake-Up

Der Display Wake-Up fungiert als eine Art „Fernsteuerung“ für eine Netzwerksteckdose, welche dazu dient, die Monitore nur dann einzuschalten, wenn auch tatsächlich ein Alarm vorliegt, bzw. diese nach einer gewissen Zeit wieder auszuschalten, um Strom zu sparen.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Ausschalten nach (Minuten) | Nach wie vielen Minuten die Bildschirme wieder ausgeschaltet werden sollen, wenn kein neuer Alarm kommt (0 für kein Timeout). |
| Anzeigenkonfiguration | Die Konfiguration der Anzeigen (ein Eintrag pro Zeile). In der Regel sind dies HTTP-Befehle. Für die genauen Befehle schlagen Sie bitte in der Anleitung Ihrer Steckdose nach. |
| Fehler in Geräteantwort ignorieren | Gibt an, ob etwaige Fehler in der Antwort ignoriert werden sollen und die Aktion als erfolgreich gemeldet werden soll. |

## eAlarm

Die eAlarm Aufgabe dient der Benachrichtigung von den eAlarm Benutzern.

TODO: Zusammenhang mit eAlarm-App (siehe Abschnitt 8.1) erklären!!!

Zum Schutz der Daten wird eine **Verschlüsselung** angeboten. Hierzu ist die Aktivierung der Funktion in der Android Anwendung notwendig.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Header | Titel der Einsatzbenachrichtigung an. Eine Verwendung von Operation-Ausdrücken ist möglich. |
| Text | Textinhalt der Benachrichtigung an. Eine Verwendung von Operation-Ausdrücken ist möglich. |
| Verschlüsselung | Gibt an ob die übertragenen Daten verschlüsselt werden sollen. |
| Passwort | Definiert das von der Verschlüsselung zu verwendende Passwort |

## Einsatzdateiexporter

Dieser Job ermöglicht den Export der Einsatzinformationen für Drittprogramme wie z.B. den AlarmMonitor oder das Programm EVA.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Benutzerdefiniertes Format | Exportiert in eine einfache Textdatei, deren Inhalt sich vollständig anpassen lässt. |
| EVA-Datei Export | Exportiert den Alarm in eine XML-Datei, die von EVA lesbar ist. |

## E-Mail-Versand

Der E-Mail-Versand Job ermöglicht den Versand von E-Mails an die Mitglieder.

Neben der Übermittlung der analysierten Daten als Text wird die Möglichkeit angeboten, das empfangene Fax als JPEG-Bild im Anhang zu versenden.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Hostname | Der Hostname des Servers von ihrem E-Mail-Anbieter. |
| SSL benutzen | Gibt an ob für die Verbindung zum Server des E-Mail-Anbieters eine SSL Verschlüsselung genutzt werden soll. Je nach Anbieter optional oder erforderlich. |
| Benutzername | Ihr Benutzername zur Authentifizierung am Server. (Anzugeben im vom E-Mail-Anbieter definierten Format) |
| Passwort | Ihr Passwort zur Authentifizierung am Server. |
| Authentifizierung benutzen | Gibt an ob für den E-Mail-Versand eine Authentifizierung am von Ihnen definierten Server notwendig ist. Meistens erforderlich. |
| Port | Gibt den Port an, welcher für die Kommunikation mit dem E-Mail-Server zu verwenden ist. |
| Absenderadresse | Gibt die anzuzeigende Absenderadresse an. Entspricht meistens ihrer normalen E-Mail-Adresse. |
| Betreff | Der Betreff unter dem die Mail versendet werden soll. Eine Verwendung von Operation-Ausdrücken ist möglich. |
| Nachrichtentext | Der Inhalt der Mail. Eine Verwendung von Operation-Ausdrücken ist möglich. |
| Fax anhängen | Aktiviert den Versand des Faxes als JPEG-Bild im Anhang. |

## Externe Schleifeninfos

Der Externe Schleifeninfos Job dient der Beschaffung der Schleifeninfos, damit diese an den gerade verarbeitenden Alarm angehängt werden. Diese Information allein kann später bei der Auswertung von Interesse sein, oder aber gleich verwendet werden, um z. B. mittels der Adressbuch-Eintragsfilterung.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Schleifeninfo-Datei | Name der Datei, welche die Schleifeninfos enthält. |
| Maximales Eintragsalter | Die maximale Zeitdauer ab dem Alarmeingang, ab welcher ein Eintrag in der Datei als "aktuell" gilt. |
| Datumsformat | Bestimmt das Datumsformat, welches sich in der Datei vorfindet. Wird bestimmt durch die Sprache des verwendeten Betriebssystems. Es ist in der Regel nicht notwendig, diesen Wert zu ändern. |

### Integration mit Drittprogrammen

Es existieren zahlreiche Programme, welche ein FMS überwachen und bei Eingang diese Information mittels einer Batchdatei in eine Datei schreiben können. Für diese Programme kann die folgende Datei verwendet werden.

Sie müssen *{FILE\_PATH}* durch den Dateinamen ersetzen, in welche das von Ihnen verwendete Programm schreibt.

@echo off

echo %1;%date% %time% >> {FILE\_PATH}

## Push-Versand

Diese Aufgabe ermöglicht den Versand der Einsatzdaten über die Push-Anbieter Prowl und NotifyMyAndroid. Die Empfänger sind im Adressbuch einzutragen.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Nachricht | Gibt den Inhalt der Push-Nachricht an. |

## SMS-Versand

Dieser Job bietet die Möglichkeit, SMS-Nachrichten mittels einem der angebotenen Anbieter zu versenden.

Derzeit stehen die Anbieter smstrade, sms77, RettAlarm und GroupAlarm zur Verfügung.

Der Job ist für *Groupalarm* und *Rettalarm* so programmiert das im Adressbuch als Rufnummer die zu alarmierende Listen oder Gruppennummer einzutragen ist. Siehe Adressbuch!

Der Inhalt der SMS ist frei konfigurierbar.

|  |  |
| --- | --- |
| Einstellung | Bemerkung |
| Benutzername | Gibt den Benutzernamen für den gewählten Provider an.  Für smstrade kann dieses Feld leer bleiben. |
| Passwort | Ihr Passwort für den SMS Provider.  Für Groupalarm und Rettalarm ist an dieser Stelle das „Quick-Login“ Passwort zu verwenden.  Für smstrade ist hier der API-Key einzutragen. |
| Format | Hier definieren Sie den Inhalt der SMS. |
| SMS-Anbieter | Wählen Sie hier aus den vorhandenen SMS Providern aus. |

# ASP.Net Website

## Installation

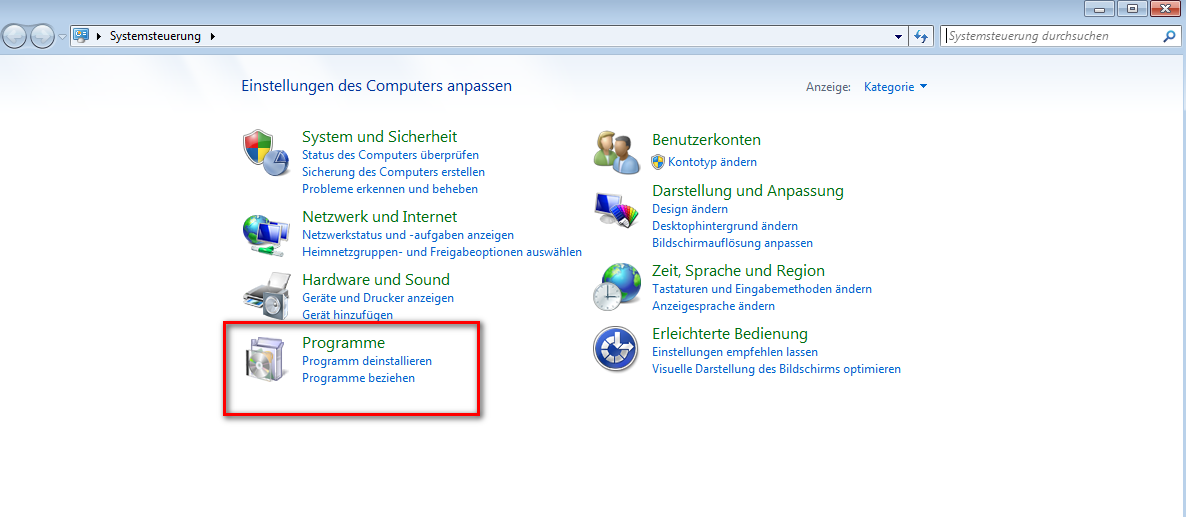
### Installation des IIS - Servers

Im Folgenden wird ausschließlich die Installation des IIS – Servers auf Windows 7 beschrieben. Sollten Sie ein anderes Betriebssystem einsetzen wenden Sie sich   
bitte an das Forum.

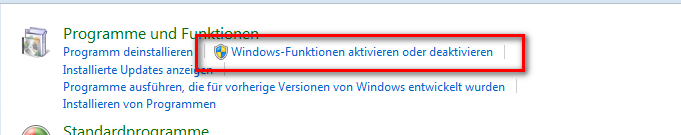
Zuerst benötigen Sie die sogenannten Microsoft Internet Information Services (kurz IIS). Dieses Produkt wird standardmäßig mit Windows 7 ausgeliefert ist jedoch im Regelfall deaktiviert.

Zur Aktivierung folgen Sie bitte den folgenden Schritten.

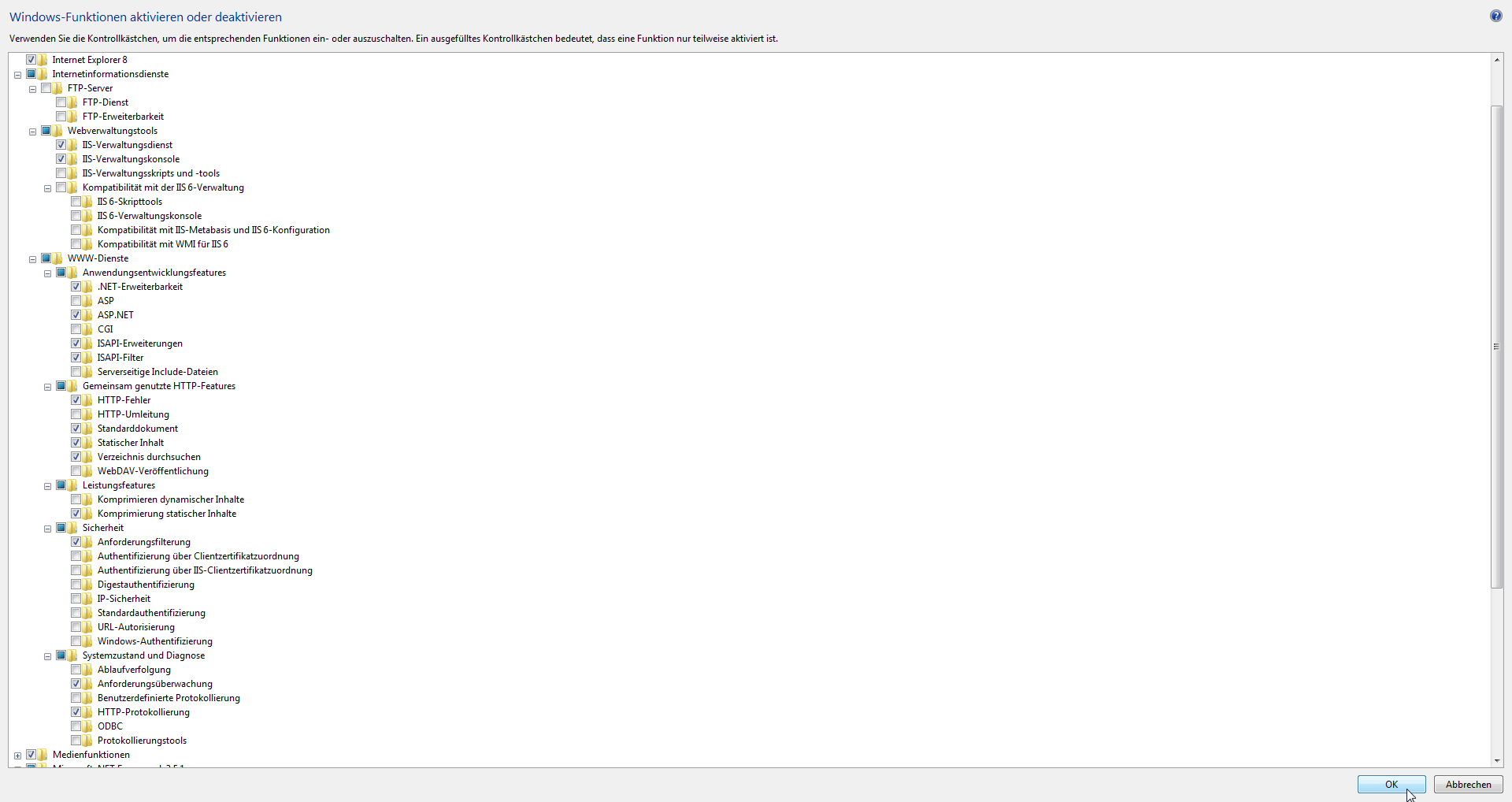
Öffnen Sie die „**Systemsteuerung“.**



Dort wählen Sie den Menüpunkt „**Programme**“ und gehen nun auf „**Windows-Funktionen**“ aktivieren.



Nun wählen Sie die im Folgenden **dargestellten Funktionen** aus.



Gegebenenfalls kann es von Nöten sein, dass ASP.Net im IIS registriert werden muss. Dazu muss mithilfe der Windowskonsole (cmd.exe) dies erfolgen.

Das Programm (aspnet\_regiis.exe) findet sich unter: C:\Windows\Microsoft.NET\Framework\<.Net Version>\ (normalerweise ist die Version v4.0.30319). Dieses ist mit dem Parameter '-i' (installieren) zu starten.

Somit ergibt sich folgender Aufruf:

C:\Windows\Microsoft.NET\Framework\v4.0.30319\aspnet\_regiis.exe –i

### Installation der Website

Für die Installation der Website laden Sie bitte das selbstextrahierende Archiv mit den Dateien für die Website von der OpenFireSource Website herunter. Die Dateien finden sie hier:

<http://www.openfiresource.de/packages/ASP_Website.exe>

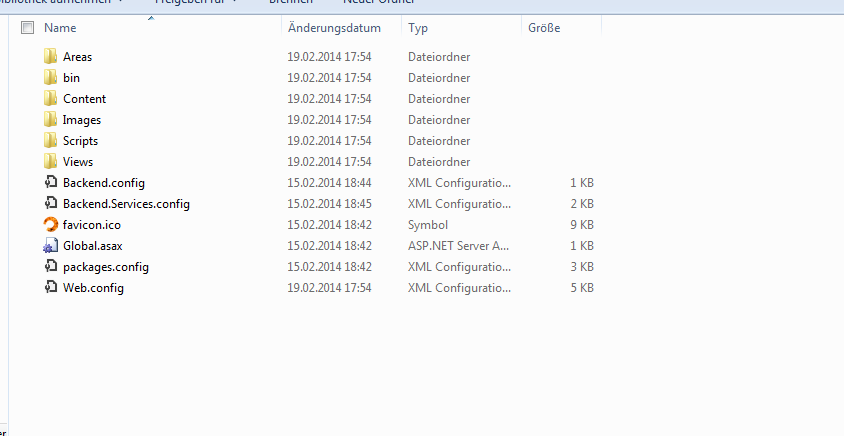
Als nächstes löschen Sie bitte alle Dateien in dem Ordner:

C:\inetpub\wwwroot

Die soeben heruntergeladene Datei entpacken Sie bitte und kopieren alle entpackten Dateien nach:

C:\inetpub\wwwroot

Der Inhalt des Ordners sollte danach so aussehen:



## Konfiguration

### 32-bit Modus aktivieren

Sollten Sie ein Windows 7 64bit o.ä. installiert haben ist der folgende Schritt für Sie auszuführen. Andere User können diesen Schritt überspringen.

Aufgrund der Tatsache, dass AlarmWorkflow al seine 32bit Anwendung ausgeliefert wird muss in den Einstellungen des IIS der 32bit Modus.

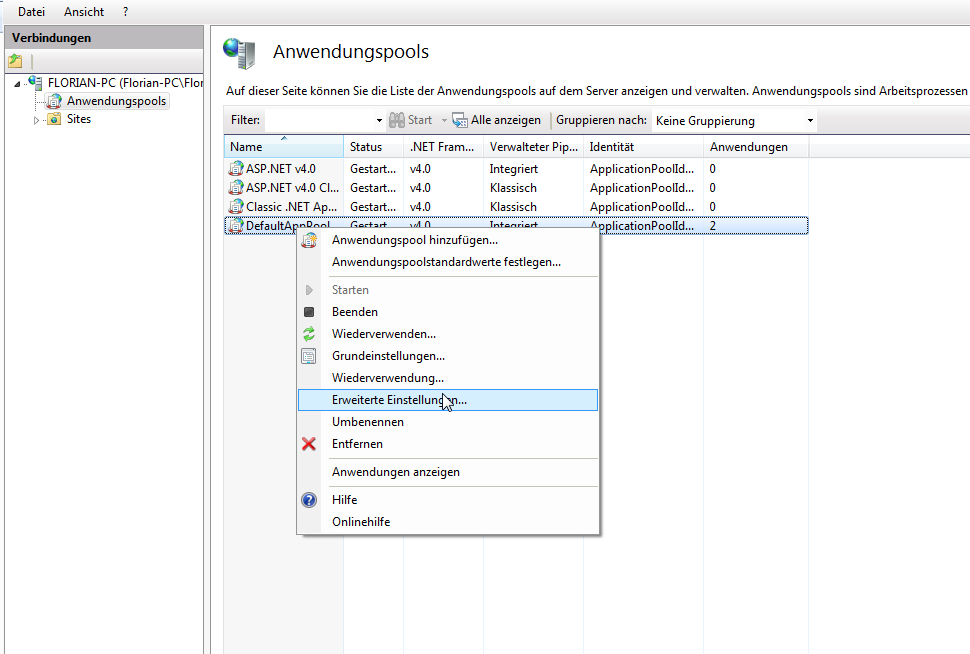
Dazu öffnen Sie die den **Internetinformationsdienste (IIS) –Manager.** Dieskann wahlweise über das Startmenü erfolgen oder über die Eingabeaufforderung mit dem Befehl:

mmc %systemroot%\system32\inetsrv\iis.msc

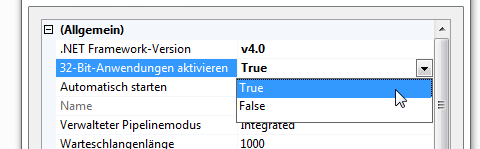
oder mittels

inetmgr

In dem nun geöffneten Fenster navigieren Sie bitte zu den **Anwendungspools** und wählen im **Kontextmenu** des “**DefaultAppPool**“ **Erweiterte Einstellungen** aus.



Nun setzen Sie bitte unter **Allgemein** den Wert „**32-Bit-Anwendungen aktivieren**“ auf **True**



Das Fenster schließen Sie mit einem Klick auf OK. Das übergeordnete Fenster können Sie ebenfalls schließen.

## Modifikation

Die ASP.Net Website besteht im Grunde aus mehreren verschiedenen kleinen Websites, welche jeweils aus einer HTML ähnlichen Datei bestehen und einem C# - Code im Hintergrund.

Der C# Code stellt alle für die Website relevante Funktionen bereit und sollte auch ohne entsprechendes Wissen nicht verändert werden.

Die HTML Seiten jedoch kann man für seine eigenen Bedürfnisse anpassen, modifizieren und erweitern.

So findet sich zum Beispiel unter:

C:\inetpub\wwwroot\Areas\Display\Views\Alarm\Index.cshtml

die Datei welcher zur Anzeige der Alarmmeldungen verantwortlich ist.

Diese können Sie mit einem normalen Texteditor wie zum Beispiel dem Windows Editor, Notepad++, PSPad, … öffnen und bearbeiten.

Wichtig ist jedoch das sie den folgenden Block stehen lassen:

@{

Layout = "IndexLayout.cshtml";

ViewBag.Title = "Alarmanzeige";

var config = AlarmWorkflow.Website.Reports.Areas.Display.Models.WebsiteConfiguration.Instance;

}

Alles weitere ist nahezu komplett HTML bzw. JavaScript Quellcode welcher auch so bearbeitet werden kann.

Für weitere Fragen wenden Sie sich bitte an das Forum um für Ihre Idee/Ihr Problem spezifische Hilfe zu erhalten.

Des Weiteren ist es ab der Version 0.9.8 möglich mehrere getrennte Ansichten zu erstellen. Hierzu kopieren Sie am einfachsten die Datei Index.cshtml und benennen die Kopie ein z.B. HLF.cshtml um.  
In der erstellten Kopie können Sie nun die gewünschten Änderungen vornehmen.

Für den Aufruf der neuen Seite verwenden Sie den neuen Dateinamen. Anstelle von z.B.:

<http://localhost/Display/Alarm/> verwenden Sie nun [http://localhost/Display/Alarm/**Index/HLF**](http://localhost/Display/Alarm/Index/HLF). Entsprechend ist der Aufruf mit weiteren Unterseiten möglich.

# Anwendungen

## eAlarm

### App herunterladen und installieren

Die App unter <http://www.openfiresource.de/packages/> herunterladen oder von KameradIn schicken lassen (z.B. via Mail). Die soeben heruntergeladene Datei Ealarm.apk mit einem Dateimanager(z.B. ES Datei Explorer) öffnen und installieren. Falls eine Sicherheitswarnung kommt zuerst in den Einstellungen unsichere Quellen zulassen. (Einstellungen 🡪 Sicherheit 🡪 „Installation von Apps aus unbekannten Quellen zulassen“)

### Einrichtung

Eigene ID dem für die Alarmierung zuständigen Kamerad schicken. Diese ist in der App unter „Einstellungen“ -> GCM-ID zu finden, mit einem Klick wird diese in die Zwischenablage kopiert.

Es besteht die Möglichkeit mehrere Regeln anzulegen welche unterschiedliche Berichtigungen realisieren. Einzelne Regeln können unterschiedlich priorisiert werden und somit z.B. zu gleichen Zeiten zugewiesen werden.

## Dispositions-tool

Dieses Tool bietet die Möglichkeit händisch weitere Einsatzmittel intern zu disponieren. Gerade bei internen Absprachen, Vorgehen,.. ist dies praktisch um nachkommenden Kameraden zu zeigen welche Einsatzmittel noch gefordert werden. Entgegen aller Jobs und anderer Funktionalitäten ist diese Anwendung nicht weiter zu konfigurieren.

Die Anwendung zeigt den letzten Alarm an, sowie die von der Leistelle disponieren Einsatzmittel. Diese werden rot hinterlegt und können nicht händisch „abbestellt“.

Nicht für diesen Einsatz disponierte Fahrzeuge/Anhänger werden grün hinterlegt und können durch einen Klick disponiert werden.

Händisch disponierte Einheiten werden orange dargestellt und können im Gegensatz zu den festdisponierten Einheiten auch widerrufen werden.

Die nachträglich disponierten Einheiten erscheinen genauso wie fest von der Leistelle disponierte Einheiten auf den Anzeigen in der UI und auf der Website.

## Windows UI

**Todo!**

# Glossar

## Definitionen

|  |  |
| --- | --- |
| Definition | Erklärung |
| Client | I.d.R. ein PC, welcher auf den *Dienst* zugreift und die Daten verwendet um z. B. die Auswertung grafisch aufbereitet darzustellen. |
| Dienst | Das Programm, das auf dem Server läuft und die Abläufe steuert sowie zentralisiert Zugriff auf Daten erlaubt. |
| EMK | Abkürzung für [*Einsatzmittelkonfiguration*](#_Einsatzmittelkonfiguration). |
| OCR | [Optical Character Recognition](http://de.wikipedia.org/wiki/Texterkennung) |
| Regulärer Ausdruck | Eine Art “Schablone”, um Texte zu erkennen und zu (in AlarmWorkflow) zu filtern. Siehe auch [Wikipedia](http://de.wikipedia.org/wiki/Regul%C3%A4rer_Ausdruck). |
| Service | Siehe *Dienst*. |

## Objekt-Ausdruck

Ein Objekt-Ausdruck dient zur Übersetzung von Eigenschaften eines Objekts (z. B. der *Operation*-Klasse) in einen Text, der dann z. B. in Jobs weiterverwendet werden kann.

Ein Objekt-Ausdruck besteht grundsätzlich aus einer öffnenden und einer schließenden, geschweiften Klammer, z. B. wie folgt:

*{<Inhalt>}*

Viele Jobs (z. B. E-Mail, Druck) verwenden von Ihnen frei konfigurierbare Texte, in welchen Sie Objektausdrücke verwenden können, um den ausgegebenen Text nach Ihren Wünschen auszugeben. Sie erkennen diese Möglichkeiten daran, dass Sie im Konfigurationseditor eine Schaltfläche "Verfügbare Eigenschaften" sehen.

Wenn Sie auf diese Schaltfläche klicken, erscheint eine Liste aller zur Verfügung stehenden Eigenschaften, die in Objektausdrücken verwendet werden können. Um sie zu verwenden, können Sie entweder auf den entsprechenden Ausdruck klicken, oder ihn manuell eingeben:

*{NameDerEigenschaft}*

Wenn der Text dann vom Programm verarbeitet wird, findet sich an Stelle des Ausdrucks der tatsächliche Wert wieder. Sollte eine Eigenschaft nicht existieren, wird je nach Objekttyp entweder eine Leerstelle eingefügt (der Ausdruck verschwindet und lässt nichts zurück), oder es wird in sog. "Custom data" des Objekts gesucht (z. B. bei dem Typ *Operation*), und falls dort die Eigenschaft gefunden wurde, der Wert hiervon eingefügt. Dies ist vor allem hilfreich bei manchem Parsern, die gewisse Informationen in die Eigenschaft *CustomData* schreiben.

### Eigene Skriptlogik

Das Objektausdruckssystem unterstützt eine mächtige Möglichkeit, die Ausgabe nahezu unbegrenzt selbst zu verwirklichen. Hierzu muss analog zu einer normalen Eigenschaft folgende Syntax eingefügt werden:

{**$cs=**Pfad\_zu\_einer\_C-Sharp-Codedatei.cs}

Diese Syntax bindet eine benutzerdefinierte **C#**-Codedatei ein, welche zur Laufzeit übersetzt wird und dessen Ergebnis dann an die Stelle des Ausdrucks eingefügt wird.

Es spielt keine Rolle, ob Leerzeichen im Pfad enthalten sind. Der gesamte Inhalt zwischen dem beginnenden ***{$cs=*** und dem schließenden ***}*** wird als ein Pfad behandelt. Anführungs- und Schlusszeichen sind nicht notwendig.

#### Beispiel

Im Repository [findet sich hier](https://github.com/OpenFireSource/AlarmWorkflow/tree/master/Resources/Documentation/Samples/ExpressionScripts) eine beispielhafte Implementierung für den *Operation Printer*.

**Hinweis**: Dieses Beispiel bezieht sich explizit auf den *Operation Printer*. Die generelle Herangehensweise ist aber ähnlich für sämtliche Einstellungen, die *Objektausdrücke* verwenden.

Das Beispiel besteht aus zwei Dateien:

##### HTML-Vorlagendatei

Da sich dieses Beispiel auf den *Alarmdetails-Druck* bezieht, ist in diesem konkreten Beispiel eine *HTML-Datei* notwendig. Diese kann beispielsweise wie folgt aufgebaut sein, um eigene Skriptlogik zu enthalten (für die vollständige Implementierung sehen Sie bitte im o. g. Repository nach):

<tr>

<td>

<b>Fahrzeuge:</b>

</td>

<td>

{$cs=Insert resources.cs}

</td>

</tr>

Es wird hier in einer Zeile und Spalte der Tabelle das Ergebnis aus folgendem Skript ausgegeben (in diesem konkreten Fall kann der Skriptcode HTML und JavaScript-Code enthalten).

**Warnung:** Sie können im konkreten Fall zwar JavaScript-Code einbetten, allerdings wird dies aus Sicherheitsgründen nicht empfohlen.   
Benutzung geschieht ausdrücklich auf eigene Gefahr und ohne Anspruch auf Support durch die Entwickler!

Die **C#-Datei** ist wie folgt implementiert:

public class Script

{

public static string Function(object graph)

{

return string.Join(YourResourceDelimiter, names);

}

}

Die Datei wurde zur besseren Lesbarkeit gekürzt. Für die vollständige Implementierung sehen Sie bitte im o. g. Repository nach.

Es gibt mehrere Regeln, die für die korrekte Ausführung des Skripts befolgt werden müssen:

* Es dürfen keine Namespaces verwendet werden.
* Es darf nur exakt eine **public**-Klasse geben, die „*Script*“ heißen **muss**.
* In dieser Klasse **muss** die Methode „*Function*“die Signatur wie im obigen Beispiel haben.

Sie können Ihre Skriptlogik auch testen. Hierzu verwenden Sie bitte den *Objektausdruckstester*, dessen Beschreibung Sie im nächsten Kapitel finden.

### Testen des Ausdrucks

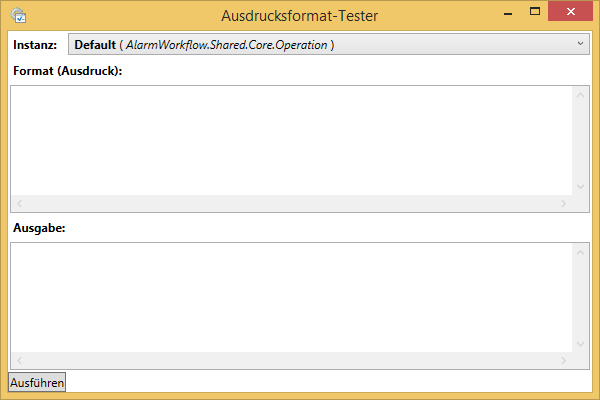
Zum Testen des Ausdrucks können Sie den *Objektausdruckstester* im Konfigurationseditor verwenden. Sie finden das Tool in der Menüleiste unter „Tools 🡪 Objektausdruckstester“.

**Beachten Sie bitte**, dass der O. derzeit nicht sämtliche Features der Objektausdrücke abdecken kann. Er soll in erster Linie dazu dienen, Grundlegende Ausdrücke testen zu können, welche sich auf die vordefinierten Eigenschaften eines Typs beziehen.

Folgende Features werden derzeit vom Objektausdruckstester unterstützt:

|  |  |  |
| --- | --- | --- |
| Feature | Beschreibung | Unterstützt? |
| {Einsatzort} | Eine Eigenschaft, die existiert | Ja |
| {CUSTOMData-Eig.} | Eigenschaft, die sich in der CustomData befindet | Nein |
| {$cs=*Pfad\_Zu\_CS.CS*} | Eigene Skriptlogik | Ja |

Der Tester stellt sich wie folgt dar:



Beschreibung der Fensteraufteilung:

|  |  |
| --- | --- |
| Segment | Beschreibung |
| Instanz | Welche Klasse getestet werden soll. |
| Format (Ausdruck) | Den *Objektausdruck*, den Sie testen möchten. Beachten Sie hierbei auch die Hinweise am Anfang des Kapitels. |
| Ausgabe | Wenn der Ausdruck syntaktisch korrekt war und erfolgreich verarbeitet werden konnte, erscheint hier das Ergebnis, das später im Programm zu erwarten ist.  Sollten bei der Verarbeitung Fehler aufgetreten sein, werden sie hier angezeigt. Beachten Sie dann ggf. auch die Logeinträge, die angelegt werden. Diese sind zu finden im Log-Ordner „Configuration“. |

1. Für Erklärungen zu gewissen Fachwörtern sehen Sie bitte im Glossar nach. [↑](#footnote-ref-2)
2. Dies wird ausdrücklich *unter Vorbehalt* gesagt. Es gilt wie immer die eigene AAO bzw. Regeln! [↑](#footnote-ref-3)