by Stephan Winter

Inhaltsverzeichnis

[1 Archivar 1](#_Toc135074494)

[2 Vorwort 5](#_Toc135074495)

[3 Konzept 5](#_Toc135074496)

[4 Systemanforderungen 5](#_Toc135074497)

[5 Installation 6](#_Toc135074498)

[6 Setup Server 6](#_Toc135074499)

[7 Vorbereitung 7](#_Toc135074500)

[8 Suche FireBIRD 7](#_Toc135074501)

[9 Server-Informationen 8](#_Toc135074502)

[10 Sicherheit 9](#_Toc135074503)

[11 Mail 10](#_Toc135074504)

[12 Plugins 11](#_Toc135074505)

[13 Basisdaten 12](#_Toc135074506)

[14 Import 13](#_Toc135074507)

[15 Server start 14](#_Toc135074508)

[16 Launcher.ini 14](#_Toc135074509)

[17 Download-Webseite 15](#_Toc135074510)

[18 Launcher 15](#_Toc135074511)

[19 Archivar-Client 16](#_Toc135074512)

[20 Clientinstallation 17](#_Toc135074513)

[21 Erste Konfiguration 18](#_Toc135074514)

[22 Admin 18](#_Toc135074515)

[23 Bilder 19](#_Toc135074516)

[24 Gremium 20](#_Toc135074517)

[24.1 Import 21](#_Toc135074518)

[25 Mitglieder des Gremiums 21](#_Toc135074519)

[26 Mitglieder 22](#_Toc135074520)

[26.1 Import 22](#_Toc135074521)

[27 Datenfelder 22](#_Toc135074522)

[27.1 Globale Datenfelder 23](#_Toc135074523)

[27.2 Lokale Datenfelder 23](#_Toc135074524)

[28 Datenfeldeditor 24](#_Toc135074525)

[29 Neues Datenfeld 24](#_Toc135074526)

[30 Datenfeldtypen 25](#_Toc135074527)

[30.1 Datentyp Table 25](#_Toc135074528)

[31 Aufgaben 26](#_Toc135074529)

[32 Aufgabentypen 26](#_Toc135074530)

[33 Neue Vorlage 27](#_Toc135074531)

[34 Systemvorlage 28](#_Toc135074532)

[35 Vorlage Beschluss 28](#_Toc135074533)

[36 Vorlage bearbeiten 28](#_Toc135074534)

[37 Definition der Datenfelder 29](#_Toc135074535)

[38 Formulare definieren 29](#_Toc135074536)

[38.1 Der Formulareditor 30](#_Toc135074537)

[39 Controls 30](#_Toc135074538)

[40 Eigenschaften 31](#_Toc135074539)

[41 Formularbearbeitungsbereich 32](#_Toc135074540)

[42 Formulare 33](#_Toc135074541)

[42.1 Neues Formular 34](#_Toc135074542)

[43 Struktur 34](#_Toc135074543)

[44 Report 35](#_Toc135074544)

[44.1 Style 35](#_Toc135074545)

[44.2 Dateien 36](#_Toc135074546)

[44.3 Bibliothek 36](#_Toc135074547)

[44.4 Testdaten 36](#_Toc135074548)

[45 DWS 37](#_Toc135074549)

[46 DWS Erweiterungen 37](#_Toc135074550)

[46.1 Funktionen 37](#_Toc135074551)

[46.2 Klassen 37](#_Toc135074552)

[47 Spezielle HTML-Tags 37](#_Toc135074553)

[47.1 Field 38](#_Toc135074554)

[47.2 Table 38](#_Toc135074555)

[47.3 Script 38](#_Toc135074556)

[47.4 System 39](#_Toc135074557)

[48 Textbausteine 39](#_Toc135074558)

[48.1 Variablen 40](#_Toc135074559)

[49 Webseitendateien 41](#_Toc135074560)

[50 ePub Manager 41](#_Toc135074561)

[51 Dateicache 42](#_Toc135074562)

[51.1 dwslib 42](#_Toc135074563)

[51.2 data 42](#_Toc135074564)

[52 Kategorie.JSON 42](#_Toc135074565)

[53 misc.pas 43](#_Toc135074566)

[54 Tools 43](#_Toc135074567)

[55 Open Source 43](#_Toc135074568)

[56 Literaturverzeichnis 44](#_Toc135074569)

# Vorwort

Der Archivar ist ein Programm zur Unterstützung der Arbeit der Betriebsräte. Es bietet umfassende Möglichkeiten zur Unterstützung der Arbeiten.

Es hilft bei:

* Anhörungen
* Sitzungen
* Abstimmungen, persönlich oder online
* Verwalten von relevanten e-Books
* Protokolle
* Dokumentenverwaltung (inkl. Löschfristen)
* Tagebuch

Es ist eine mehrschichtige Anwendung, die aus einem Server mit dem Archivar-Server und einer Datenbank besteht und einem Client.

# Konzept

Das Grundkonzept des Programms ist, das es so flexibel wie möglich ist. Es liegen, außer als Beispiel, keine fertigen [Formulare](#msLHqehRHEJlQbjo) für Anhörungen vor. Die Erfahrung zeigt, das in jeder Firma zwar die [Aufgaben](#RqksctvtyyAgQbUG) des [BetrVG](https://www.gesetze-im-internet.de/betrvg/) natürlich wahrgenommen werden, aber auf andere Art und mit anderen Daten.

Daher ist der erste Schritt bei der Nutzung des Programms auch, sich über die, bei den Anhörungen nötigen Daten, klar zu werden. Wenn diese Erfasst wurden, dann wird im zweiten Schritt mit Hilfe eines Formulareditors ein passendes Eingabeformular erstellt.

Zu den Daten gehört natürlich auch eine Visualisierung (in der Regel als Protokoll), daher wird dann mit Hilfe von HTML (und notfalls auch Skripten), beschrieben, wie aus den Daten Text wird. Auch dafür gibt es Unterstüzung im Editor.

So kann jeder Bearbeiter sich seinen Anhörungstypen auswählen und die Daten über das Formular eingeben.

Zu jeder Sitzung, in der Beschlüsse getroffen werden, gehört auch ein Protokoll. Daher kann jede Anhörung entsprechend einem Protokoll zugeordnet werden.

Das fertige Protokoll ist am Ende ein "Abfallprodukt" einer Sitzung.

# Systemanforderungen

Als Basis dient ein recht einfacher Computer mit Windows. Wie bei allen Servern ist es gut, wenn das System über einen überdurchschnittlichen Speicher und größere Festplatten verfügt. Für die Datensicherheit wäre ein RAID-Festplattensystem eine gute Wahl.

Wenn man sich nicht um die Administration und das Backup des Servers kümmern kann/möchte, dann ist die Wahl eine V-Servers bei einem Hoster (z.b.Strato)

Alternativ kann auch ein normaler Bürorechner zum Einsatz kommen, da die vom [Archivar](#AfKDdHfdvhWwqaaR) erzeugte Last ist in der Regel so klein ist, das auf dem Rechner ganz normale Büro Tätigkeiten auch weiterhin ausgeführt werden können.

Als Datenbank kommt [[Fir21]](#RALsOsQ) zum Einsatz, diese muss vor der Installation des Archivars installiert worden sein. Getestet wurde das Programm mit den Versionen 2.5 und 3.

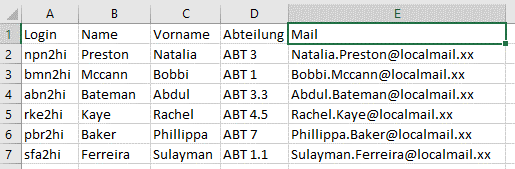
# Installation

Zur Vorbereitung der Installation bietet sich die Windows-Sandbox an, um die Installation zu Testen.

Es werden die Daten für SMTP und IMAP4für den Mails benötigt, sofern das Senden und Empfangen von Mails gewünscht ist.

Um die [Mitglieder](#TbAaxoYxFHqRlrAU) des Gremiums gleich bei der Installation einzufügen, wird eine Excel- oder CSV-Datei mit folgendem Aufbau benötigt. Natürlich können die [Mitglieder](#TbAaxoYxFHqRlrAU) auch später importiert oder manuell eingefügt werden.

Der Aufbau der Excel-Datei:



**Achtung:**

Nutzer, die in keinem [Gremium](#OitUCVCNXjXVOKBv) sind, können sich auch nicht in dem System anmelden. Ausnahme ist natürlich der Benutzer "Admin"

# Setup Server

Nach der [Vorbereitung](#mTMyQAYitRCrkjoO) des Server-PC's sind das Setup-Programm ausgeführt. Dieses Konfiguriert den Archivar-Server entsprechend.

Folgende Schritte sind notwendig:

* [Vorbereitung](#mTMyQAYitRCrkjoO)
* [Suche FireBIRD](#XUiINusGFncHPfcj)
* [Server-Informationen](#dqfsDauJCICqAlld)
* [Sicherheit](#hINivIHcnfAEjiSM)
* [Mail](#fLlIAVojOMdsAKgB)
* [Plugins](#uyeYINUpIRWEbLHn)
* [Basisdaten](#cKHrdBhXulcIxaiH)
* [Import](#FxiivBXXrIDfkbBR)
* [Server start](#wfbqRCqtGXulYXml)
* [Launcher.ini](#gbaUtluLkLMLuPih)

Im Verzeichnis "InitialData" liegen die [Basisdaten](#cKHrdBhXulcIxaiH), in der Regel, als Textdateien vor. Für eine erste Installation sind hier keine Änderungen nötig.

Außerdem kann noch die [Download-Webseite](#wUsCridaofNhcPhu) für den [Launcher](#fGuarKMGyEyOuBKw) geändert werden.

# Vorbereitung

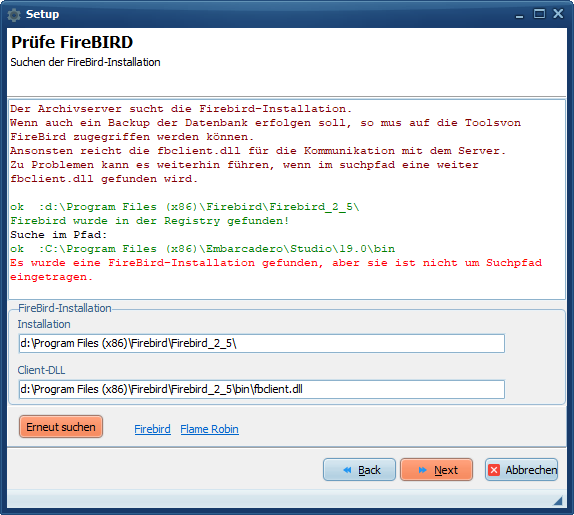
Im ersten Schritt wird das [RDBMS FireBIRD](#RALsOsQ) installiert, aktuell wird die Version 3.x empfohlen. Die Daten des Archivar-Servers werden in einer FireBIRD-Datenbank abgelegt.

Es wird empfohlen auch [FlameRobin](#18019aee-480d-40df-a025-cc576efaebfe) zu installieren, damit kann geprüft werden, ob die [Installation](#WkKNkkQLjquAWtVb) von FireBIRD erfolgreich war.

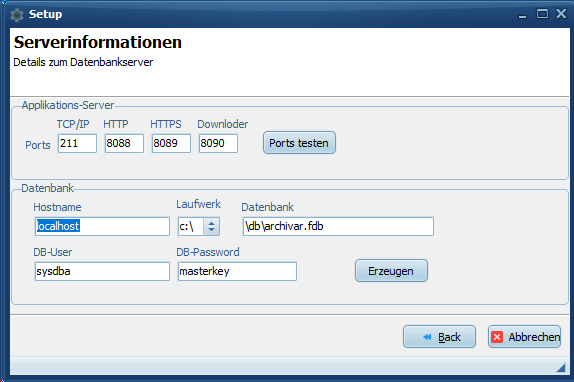
Als nächstes wird das Konfigurationprogramm für den Server ausgeführt. Hierbei wird u.a. die Datenbank angelegt und Initialisiert.

# Suche FireBIRD

Es wird die Installation von FireBIRD gesucht. Diese ist notwendig, da über die fbclient.dll mit dem Server kommuniziert wird und die FireBIRD-Tools (z.B. Backup, Restore, etc).



# Server-Informationen



Die drei Protokolle:

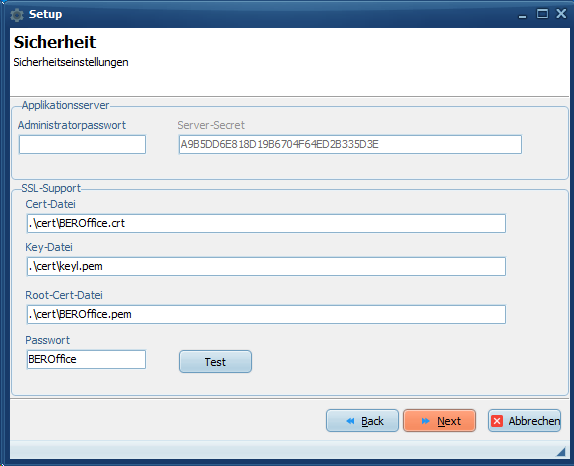
* TCP/IP
* HTTP
* HTTPS

werden direkt für die Kommunikation mit dem Archivar-Server benötigt. TCP/IP ist dabei zu bevorzugen. Über den Button "Ports testen" kann geprüft werden, ob die entsprechenden Ports auf dem Server verfügbar sind.

Der zweite Teil betrifft die Datenbank für die Daten des Archivar-Servers. Die FireBIRD-Datenbank kann auf einem anderen Server erfolgen. Trotzdem ist es nötig, das auch auf dem Archivar-Server eine [Installation](#WkKNkkQLjquAWtVb) des FireBIRD erfolgt. Es werden entsprechende [Tools](#aHwLoGSepTEpnAiG), wie z.B. Backup und Restore der [Installation](#WkKNkkQLjquAWtVb) benötigt.

Der Button "Erzeugen" erstellt die entsprechende Datenbank.

# Sicherheit



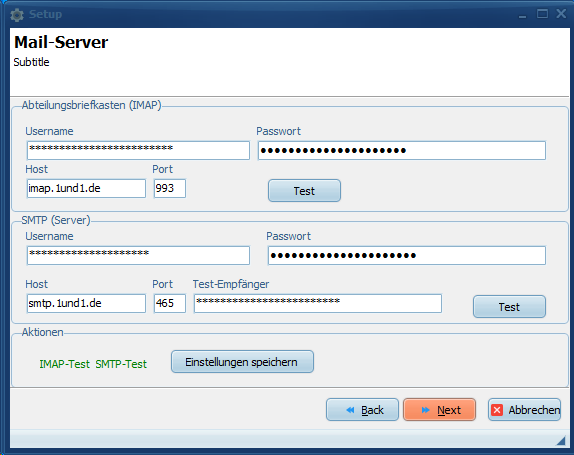
Kennworte werden als MD5 in der Datenbank gespeichert. Damit es nicht so einfach ist, diese zu erkennen, wird das "Server-Secret" zu jedem Kennwort hinzugefügt. Damit ist eine Rückwärtssuche über google nach Kennworten nicht möglich.

Weiterhin kann festgelegt werden, welches Kennwort der [Admin](#uInaYXYwBTXGbJpv) des Servers hat. Es ist nicht empfohlen, das Kennwort leer zu lassen.

Für die HTTPS-Kommunikation werden entsprechende Zertifikate benötigt. Es wird ein Satz selbst erzeugter Zertifikate mitgeliefert. Hier kann entsprechend auf andere, gültige Zertifikate gewechselt werden. Der Button "Test" zeigt, ob die Zertifikate genutzt werden können.

Die Zertifikate liegen in dem Verzeichnis "cert".

# Mail

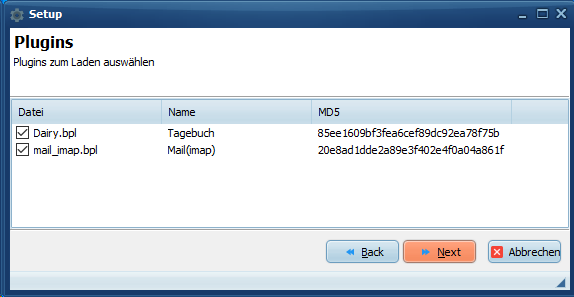


Zum Versenden von eMail wird ein SMTP-Konto benötigt. Der Server kann einige Informationen als Mail an die Benutzer schicken.

Zum Empfangen von Mails wird entsprechend der IMAP4-Zugang benötigt.

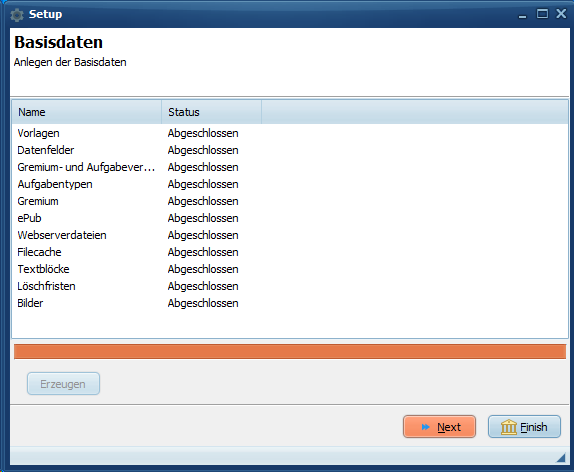
Achtung im Client müssen dann noch die entsprechenden Mail-Ordner ausgewählt werden.

# Plugins



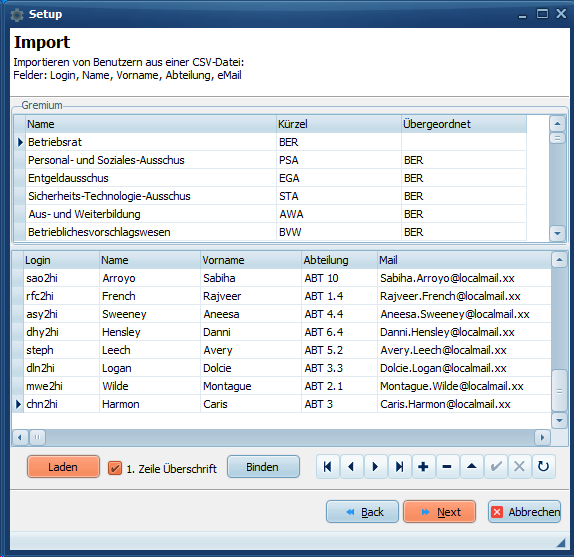
Einige Funktionen werden als Plugins zur Verfügung gestellt, diese Können vom Administrator verwaltet werden.

# Basisdaten



Zu der Grundinitialisierung des Servers gehören einige Daten, die für die Funktionsweise des Programms benötigt wird. Viele Informationen liegen als JSON- oder XML-Dateien vor.

# Import



Dieses Formular ist zweiteilig, im oberen Teil werden die möglichen Gremien angezeigt. in der Regel ist der Betriebsrat das oberste [Gremium](#OitUCVCNXjXVOKBv). Ausschüsse und Arbeitsgruppen setzen sich dann aus den Mitgliedern des Betriebsrates zusammen.

In der unteren Hälfte kommt die Datei mit den Personen des Betriebsrates zum Einsatz. Diese wurde im Rahmen der Vorbereitung auf die [Installation](#WkKNkkQLjquAWtVb) hoffentlich angelegt.

Über den Button "Laden" wird diese Datei geladen, aber noch nicht in den Server eingetragen. Hier sollte als [Gremium](#OitUCVCNXjXVOKBv) der Betriebsrat ausgewählt werden. mit Hilfe des Buttons "Binden" werden dann die Personen dem Betriebsrat zugewiesen.

Die anderen Gremien werden dann vom Administrator im Archtivar-Tool mit den entsprechenden Personen versehen.

**Achtung:**

Wenn eine Person keinem [Gremium](#OitUCVCNXjXVOKBv) zugeordnet ist, so kann sich diese nicht einloggen!

# Server start

Aktuell gibt es 2 Versionen des Servers:

* Windows dienst
* Comandline-Tool

Es kann immer nur eine Version zur Zeit auf einem PC laufen. Der Dienst wird empfohlen , da er automatisch von Windows verwaltet wird. Das Tool hingegen ist zum Ausprobieren des Servers eine gute Wahl.

Um zu Testen, ob der Server läuft kann man im Webbrowser dann folgendes eingeben:

http://localhost:8090

Wenn unter [Server-Informationen](#dqfsDauJCICqAlld) für den "Downloader" ein anderer Port genommen wurde, so ist natürlich dieser zu verwenden.

Ist die Seite zu erreichen, aber nicht aus dem Netz, so ist ggf. der Firewall noch nicht richtig konfiguriert.

**Achtung**:

Einige Informationen werden in der Registry unter HKEY\_CURRENT\_USER gespeichert. Daher sollte der Dienst auch immer mit diesem Account gestartet werden !!

# Launcher.ini

Die Konfigurationsdatei wird mit auf den Client übertragen, es ist eine klassische [INI-Datei](https://de.wikipedia.org/wiki/Initialisierungsdatei) . Der Aufbau der Serveradresse ist wie folgt:

host=<Protokoll>://<ServerName|IP>:<Port>

[server]

host=http://ArchivarServer:8088

Es gibt drei mögliche Protokolle:

1. ds
2. http
3. https

Die möglichen Ports für die Kommunikation wurden bei dem Setup in dem Teil [Server-Informationen](#dqfsDauJCICqAlld) eingetragen und getestet. Ob man besser den Servernamen oder die IP nimmt, das hängt immer von der vorhandenen Infrastruktur ab.

Wenn für das Protokoll "ds", der Port auf 211 bleibt, dann kann man die Adresse wie folgt einkürzen:

host=<ServerName|IP>

# Download-Webseite

Die Webseite für den Download des Launchers ist unter www\_dnl\index.html im Serververzeichnis zu finden.

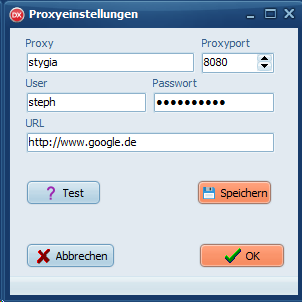
Beim Verändern der Datei ist darauf zu Achten, das einige Informationen aus der [Config.txt](http://Config.txt) stammen und andere, wie z.B. die Ports, vom Server selber eingetragen werden.

# Launcher

Der Launcher wird vom [Archivar](#AfKDdHfdvhWwqaaR) runter geladen, er dient dazu den [Archivar-Client](#amAMqHKHnaDBOImF) runter zu laden und zu starten.



Wenn der Archivar-Server außerhalb des Unternehmens lauft, so ist es möglicherweise nötig, einen Proxy einzurichten.

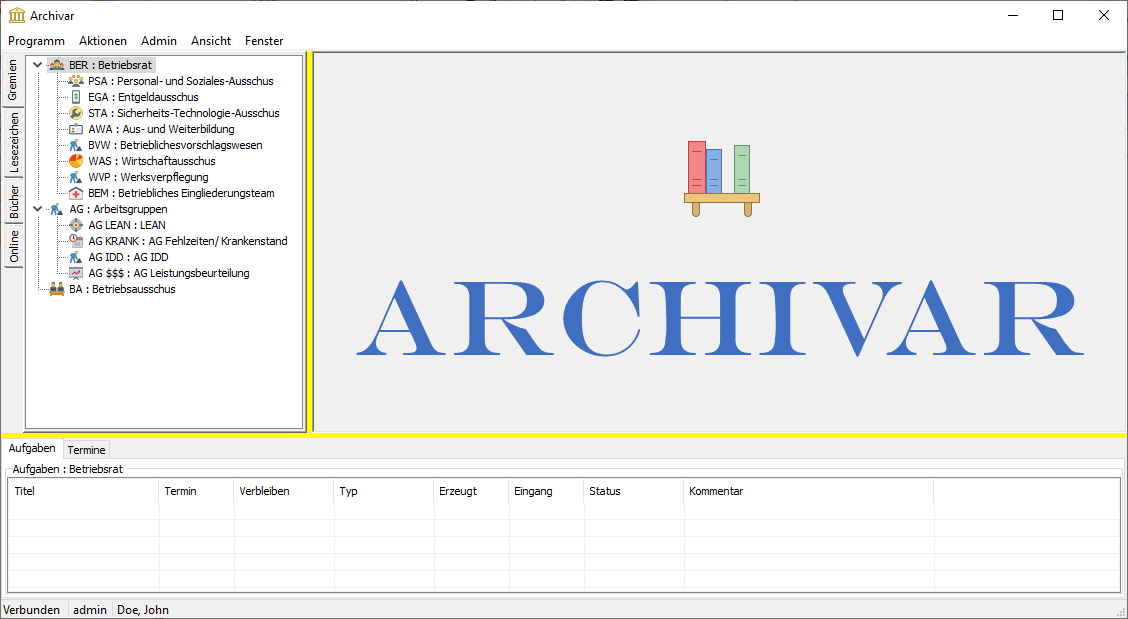


Die notwendigen Einstellungen für den Proxy sollte die Firmen-IT bereit stellen.

Bei dem Start des Archivar-Clients werden entsprechend die Proxy-Einstellungen mit übergeben.

# Archivar-Client

War die Installation erfolgreich sollte sich folgendes Bild bieten:



Die Grundkonfiguration kann nur von dem Administrator gemacht werden. Der Administrator kann, muss aber nicht, ein Mitglied des Betriebsrates sein. Allerdings hat er zu allen Informationen, bis auf den persönlichen Daten der Mitglieder, zugriff.

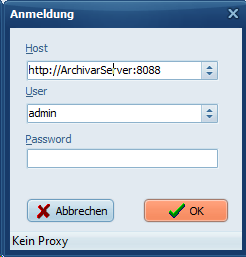
In dem Kapitel [Erste Konfiguration](#aDXEdnjVcONKpPaM) wird genauer beschrieben, was nach der [Installation](#WkKNkkQLjquAWtVb) gemacht werden muss.

# Clientinstallation

Um den Client zu Installieren, muss einfach nur die Webseite des Archivar-Servers aufgerufen werden, die den "[Launcher](#fGuarKMGyEyOuBKw)" zum Download anbietet.

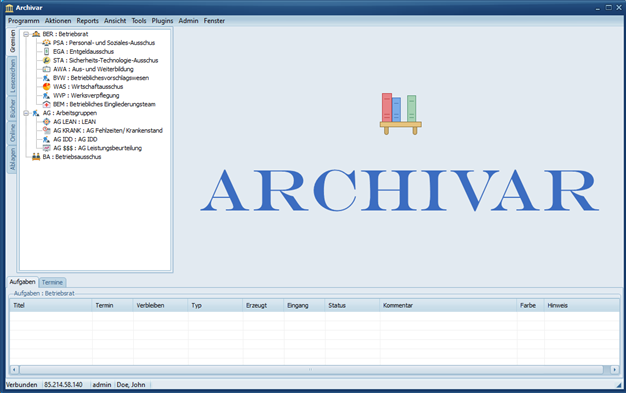
Der Download der nötigen Daten kann je nach Protokoll und Internetverbindung eine Weile dauern. Wenn alles Erfolgreich ist, dann wird die Anmeldung des Clients gezeigt.

Als erster sollte sich der [Admin](#uInaYXYwBTXGbJpv) anmelden und die Einrichtung abschließen.



Auf diesem System hat der [Admin](#uInaYXYwBTXGbJpv) kein Kennwort (das ist eine Sicherheitslücke, die größer ist, als die Entfernung Erde - [Stephenson 2-18](https://de.wikipedia.org/wiki/Stephenson_2-18))

War die Anmeldung erfolgreich, dann sollte der Client ungefähr so aussehen:



# Erste Konfiguration

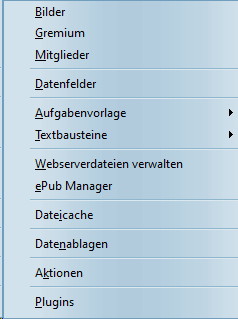
Das wichtigste sind zu erst die [Mitglieder](#TbAaxoYxFHqRlrAU), denn sie haben Zugriff auf das Programm. Unter [Mitglieder](#TbAaxoYxFHqRlrAU) werden mindestens die gewählten Betriebsräte verstanden, sinnvoll ist es aber auch die Ersatzmitglieder einzubinden.

Alleine Mitglied zu sein reicht aber für den Zugriff noch nicht aus. Man muss auch zu mindestens in einem [Gremium](#OitUCVCNXjXVOKBv) sein. In dem Gremium Betriebsrat sind alle [Mitglieder](#TbAaxoYxFHqRlrAU) vertreten.

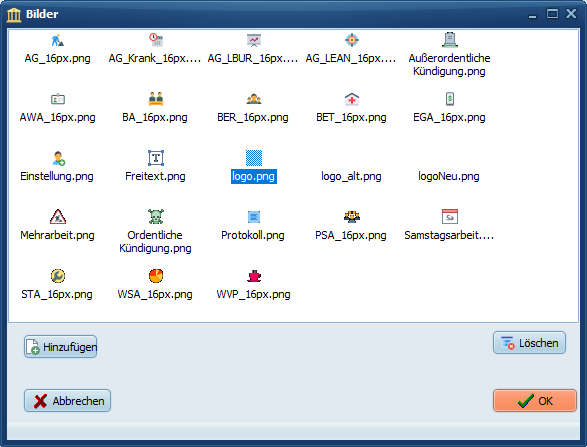
Wenn ein Betriebsrat sein Mandat nieder legt oder nicht mehr gewählt hat, so wird es aus den Gremien entfernt, damit erlischt die Möglichkeit sich anzumelden.

# Admin

Der Admin kann viele Dinge auch über den Client einstellen.



# Bilder



Die Bilder liegen, in der Regel als 16x16 Png vor, die Ausnahme ist das [logo.png](http://logo.png) das kann und sollte größer sein.

Die Bilder werden meist in den Menüs genutzt, um z.B. ein Gremien und Arbeitsgruppen zu visualisieren.

Wenn es Bilder gibt, deren Name einem [Aufgabentypen](#BhSmAtEiXcluuUcO) entspricht, so werden diese in verschiedenen Anzeigen (z.B. den Lesezeichen) genutzt.

Die Bilder werden in der Regel beim Login synchronisiert.

# Gremium

Im Rahmen der Installation wurden schon einige Gremien eingetragen. Mindestens 1 Gremium, nämlich den Betriebsrat, muss es geben. Ob und wie viele Ausschüsse und Arbeitsgruppen gebildet werden, das hängt von der jeweiligen Situation ab.



Über den Menüpunkt Bilder werden die [Bilder](#OjEDskjnUQXEOBGr) für die verschiedenen Gremien angelegt.

Nach dem Anlegen der Gremien, müssen entsprechend die [Mitglieder](#TbAaxoYxFHqRlrAU) zugeordnet werden. Alle Vollmitglieder sind natürlich im Betriebsrat, ob man auch die Ersatzmitglieder mit aufnimmt, hängt von der Situation ab.

Die Farbe des Gremiums wird verwendet um Mails zu kennzeichnen.

Über den Button "Mitglieder" kann man die [Mitglieder des Gremiums](#VsyLjxxRMQywIoGl) bestimmen.

## Import

Es ist auch hier Möglich, die Struktur über eine CSV-Datei zu importieren:

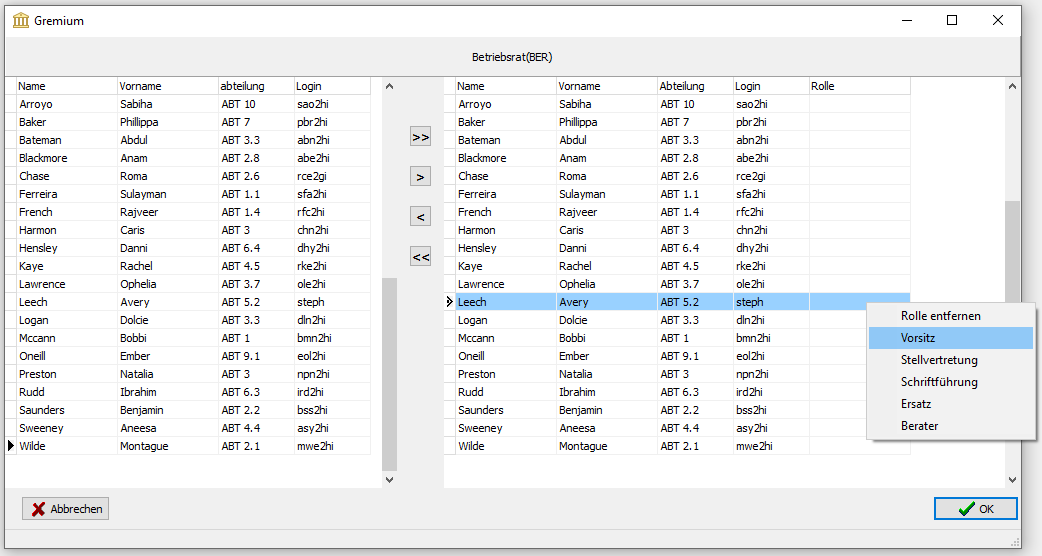
Kurz;Name;Vorgänger;Image

BER;Betriebsrat;;BER\_16px.png

PSA;Personal- und Soziales-Ausschus;BER;PSA\_16px.png

....

# Mitglieder des Gremiums



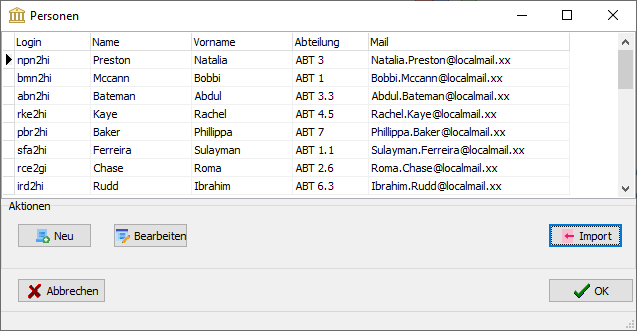
* >> Überträgt alle Namen von links nach rechts
* > Überträgt die ausgewählten Namen von links nach rechts. Mehrfache Auswahl erfolgt mit Strg.
* < Löscht die ausgewählten Namen von der linken Seite
* << Löscht die gesamte linke Seite.

Mit dem Kontextmenü (rechte Maustaste) können noch folgende Rollen vergeben werden:

* Vorsitz
* Stellvertretung
* Schriftführung
* Ersatz
* Berater

Die Rollen haben nur informativen Charakter, dadurch bekommt der entsprechende Benutzer weiteren Rechte in der Software.

# Mitglieder



## Import

Statt der einzelnen Eingabe aller Namen, können die Daten auch über eine CSV-Datei eingelesen werden.

Der Aufbau ist dann wie folgt:

id;Vorname;Name;abteilung;mail

npn2hi;Natalia;Preston;ABT 3;Natalia.Preston@localmail.xx

bmn2hi;Bobbi;Mccann;ABT 1;Bobbi.Mccann@localmail.xx

.....

Die erste Zeile ist die Kopfzeile mit den Spaltennamen, dann kommen entsprechen für jede Person eine Zeile.

Sind alle Personen eingetragen, dann können die Gremien eingetragen werden.

# Datenfelder

Ein Datenfeld beschreibt eine Information. Über die Datenfelder können Suchen und Auswertungen gesteuert werden. Alle Vorgänge beruhen auf Daten mit bestimmten Bedeutungen.

Für eine einfache Versetzung sind z.B. folgende Daten notwendig:

* Name/Vorname des Kollegen
* Alte Abteilung
* Neue Abteilung
* Name/Vorname des neuen Vorgesetzten

Die Idee ist, das man diese Daten mit einem Namen versieht, so würden sich aus den Daten folgende Datenfelder ergeben:

* MA\_NAME
* MA\_VORNAME
* ABTEILUNG
* ABTEILUNG\_NEU
* VG\_NAME
* VG\_NACHNAME

In normalen Textdokumenten stehen diese Informationen auch, aber in der ohne die Metainformationen. Suchen in Textdokumenten liefern daher in der Regel ungenaue Suchergebnisse. Wird ein Name gefunden, so muss der Benutzer feststellen, ob es der Name des Kollegen oder dessen Vorgesetzten ist.

## Globale Datenfelder

Ein globales Datenfeld kann in jedem Vorgang einfach genutzt werden. Es ist vordefiniert und an seiner Definition kann nichts geändert werde. Betrachtet man die Vorgänge in einem Gremium, dann zeigt sich sehr schnell, welche Datenfelder bedeutend sind.

Die Faustregel ist, lieber zu viele globale Datenfelder, als zu wenig.

Die Bedeutung eines globalen Datenfeldes ist allgemein bekannt.

**Aber:**

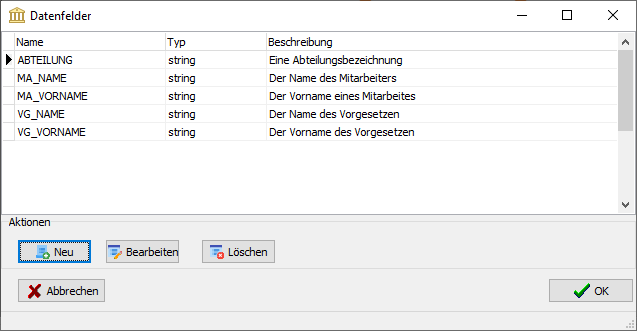
Vermieden werden sollte so etwas, wie das ein Mitarbeitername mal mit 50 und mal mit 100 Zeichen definiert wird. Globale Definitionen sollten alle nötigen Fälle abdecken.

## Lokale Datenfelder

Lokale Datenfelder werden dann benutzt, wenn diese Daten lediglich von sehr begrenztem Nutzen sind. Wenn z.B. für eine Einstellung die Schuhgröße des neuen MA's wichtig ist, aber diese Information keine weitere Bedeutung hat, so wäre ein lokales Datenfeld hier angebracht.

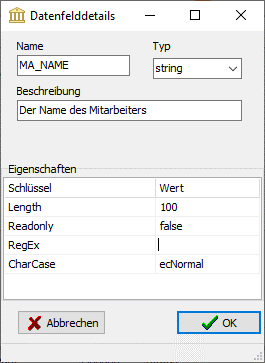
Datenfelder sind die Grundlage für alle Vorgänge im Programm.

# Datenfeldeditor



Wird ein globales Datenfeld gelöscht, so betrifft das nicht die verwendeten Vorlagen. Wird einer Vorlage ein globales Datenfeld hinzugefügt, so werden die Informationen kopiert. Lediglich die Bedeutung des Datenfeldes wird in der Vorlage von global auf lokal geändert.

# Neues Datenfeld



Die Eigenschaften sind für die verschiedenen [Datenfelder](#LmyfynqqqMiliiKb) unterschiedlich.

# Datenfeldtypen

| **Feldname** | **Bedeutung** |
| --- | --- |
| string | Kurze Texte bis 255 Zeichen |
| integer | Positive und negative ganze Zahlen |
| float | Gleitkommazahlen (z.B. 3.14) |
| date | Ein Datum ohne Zeit |
| time | Eine Zeit ohne Datum |
| datetime | Datum und Uhrzeit |
| bool | Ein logischer Wert (true oder false) |
| enum | Ein Aufzählungstyp, z.B. für Lohngruppen |
| text | Ein Text, der länger als 255 Zeichen sein kann. in der Regel enthält so ein Text auch Formatierungsinformationen. |
| table | Eine Tabelle aus Datenfeldern. |
| linktable | Eine Linktable ist eine "Kopie" einer richtigen Tabelle mit einem anderen Namen. Damit müssen inhaltlich gleiche Tabellen nicht mehrfach definiert werden. |

Jeder Datentyp hat darüber hinaus noch Eigenschaften, die man einstellen kann, z.B. bei einem String könnte man definieren, das dort nur Großbuchstaben möglich sind.

## Datentyp Table

Dieser Datentyp ist ein Besonderheit, denn eine Tabelle wird hier eine Liste von Datenfeldern beschrieben.

Dadurch ist wird jedes Feld wieder einem Datenfeld zugeordnet und die Bedeutung des Inhaltes des Feldes ist definiert.

Wenn bei einer ein Kollege versetzt wird, dann wird u.a. das Feld MA\_NAME genutzt. Wenn es Gruppenversetzungen gibt, dann ist eine der Spalten der Tabelle vom Typ MA\_NAME.

# Aufgaben

Ein großer Teil der Betriebsratsarbeit besteht aus dem Bearbeiten von Anhörungen und Anträgen. Für diese Aufgaben werden verschieden Daten benötigt und es gelten auch verschiedene Fristen für die Erledigung.

Die benötigten Informationen für Anhörungen sind, in der Regel, immer die gleichen. Nehmen wir das Beispiel einer Einstellung:

* In welcher Abteilung soll eingestellt werden?
* Wer soll eingestellt werden?
* Wann soll die Einstellung stattfinden?
* Welches Gehalt wird gezahlt?
* Gab es interne Bewerber?

Je nach Firma können noch weitere Informationen von Nutzen für die Beschlussfindung sein.

Auf der Basis dieser Informationen kann nun also ein Formular entwickelt werden, das diese Informationen enthält. Ebenso kann ein passender Textblock daraus erzeugt werden, der in das Protokoll eingefügt werden kann.

Eine Vorlage fasst all dieses Zusammen, so das bei der Bearbeitung einer Aufgabe weder Informationen vergessen werden können, noch das Vergessen wird, sie in das Protokoll einzutragen.

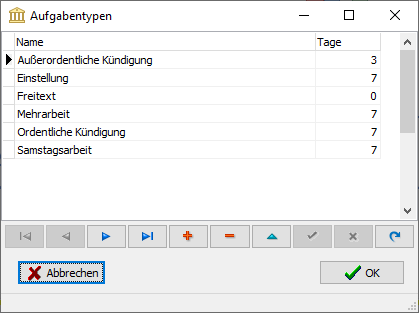
Ebenso wird das Suchen in den Protokollen einfacher, denn dadurch, das jedes Datenfeld einen logischen Namen hat, kann man nun suchen der Art:

Suche in den Einstellungen nach Max Mustermann

problemlos ausführen.

# Aufgabentypen

Für verschieden [Aufgaben](#RqksctvtyyAgQbUG) gibt es auch verschieden Fristen die ggf. einzuhalten sind, z.b. für eine Einstellung sind das aktuell 7 Tage. Das Fristende wird so automatisch berechnet.

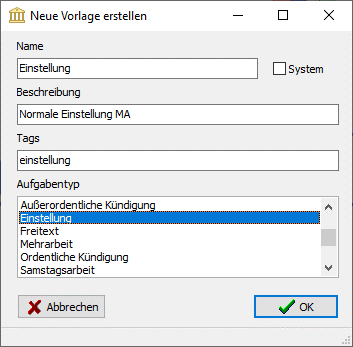


Es gibt auch Aufgaben ohne Frist, dies sind in der Regel interne Aufgaben.

# Neue Vorlage

Jede neue Vorlage braucht einen prägnanten Namen und eine kurze Beschreibung wofür genau die Vorlage gedacht ist.

Über den [Aufgabentypen](#BhSmAtEiXcluuUcO) definiert sich, ob und welche Frist in dieser Vorlage genutzt wird.



Im zweiten Schritt kann man dann die [Vorlage bearbeiten](#lBIWjedwoqrYPPud).

# Systemvorlage

Diese Vorlagen sind für die Funktionsweise des Programms unerläßlich. Es werden mit diesen Vorlagen z.B. der Protokollrahmen oder der Beschluss einer Anhörung beschrieben.

# Vorlage Beschluss

Diese Vorlage hat folgende Daten Felder, deren Namen nicht geändert werden dürfen:

* be\_zustimmung : integer
* be\_ablehnung : integer
* be\_enthaltungen : integer
* be\_text : text
* be\_ergebnis : text
* be\_datum : datum
* be\_zeit : time
* be\_tab\_teilnehmer : table
* Name : string
* Vorname : string
* Abteilung : string
* Rolle : string
* be\_tab\_abwesende : table
* Name : string
* Vorname : string
* Abteilung : string
* Rolle : string
* be\_tab\_nicht\_abgestimmt : table
* Name : string
* Vorname : string
* Abteilung : string
* Rolle : string

Das Programm erzeugt bei einem Beschluss genau diese [Datenfelder](#LmyfynqqqMiliiKb) für die Dokumentation. Über den Vorlageneditor können die Vorlagen entsprechend angepasst werden, bis auf diese Namen.

Diese Namen sind im Programm fest programmiert!

Es können sehr wohl weitere [Datenfelder](#LmyfynqqqMiliiKb) in der Vorlage angelegt werden, diese müssen dann allerdings per Skript gefüllt werden.

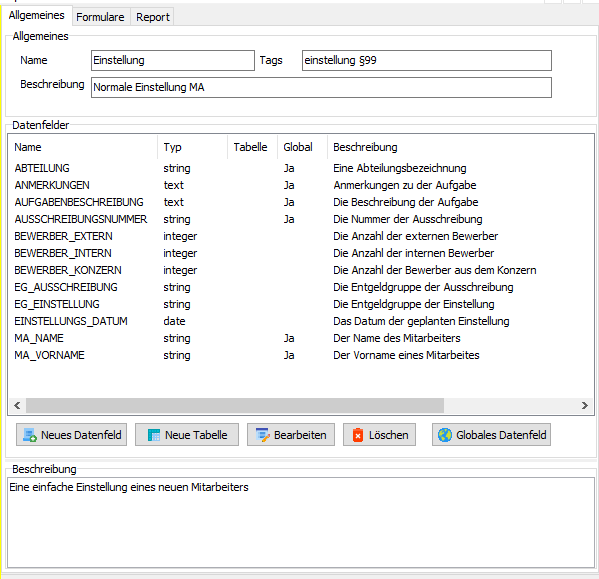
# Vorlage bearbeiten

Für die Bearbeitung einer Vorlage sind folgende Schritte nötig:

* [Definition der Datenfelder](#oVrSubnShcTnOBgP)
* [Formulare definieren](#dQERPgoyFwbpaRxx)
* [Report](#tkTbEpqbExdEkVui)

# Definition der Datenfelder

Im ersten Schritt werden die nötigen [Datenfelder](#LmyfynqqqMiliiKb) eingegeben. Sofern globale [Datenfelder](#LmyfynqqqMiliiKb) existieren, sollten diese Möglichst genutzt werden, statt neue zu definieren, die dann die identische Bedeutung haben.



# Formulare definieren

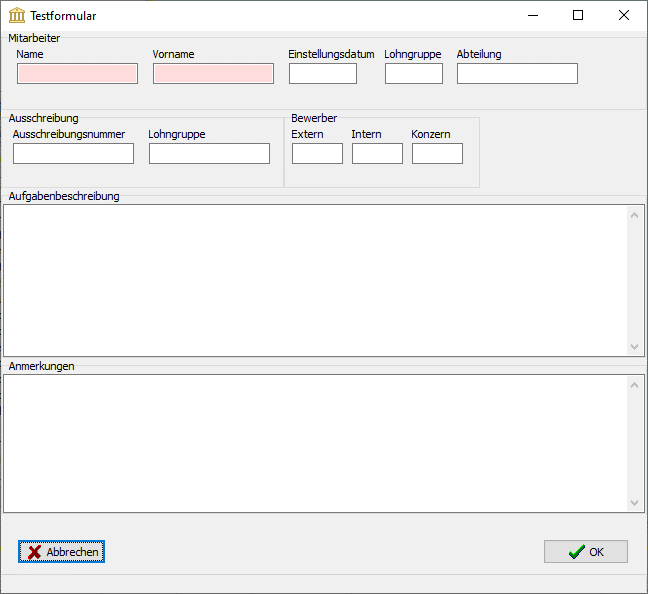
In diesem Schritt wird ein Formular entworfen, mit dem die [Datenfelder](#LmyfynqqqMiliiKb) gefüllt werden können.

## Der Formulareditor

Der Editor besteht aus 5 Teilen:

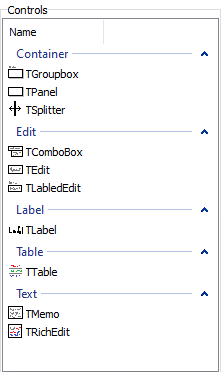
* [Formulare](#msLHqehRHEJlQbjo)
* [Struktur](#QbWLjrteLgoxsWqT)
* [Controls](#sIDcUbwhneTeEmxQ)
* [Eigenschaften](#mpEqfmJmRQEjpNQS)
* [Formularbearbeitungsbereich](#uUOXnrTISMvCsdRK)

Das folgende Formular wird genutzt um eine Einstellung zu erfassen:



# Controls

Mit Hilfe der Controls wird das Formular gebildet. Sie bieten die Möglichkeit eingaben im Formular mit den Datenfeldern zu verbinden.

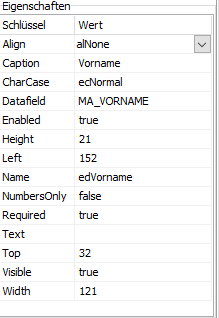


|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| TGroupbox | Ein Container mit einem Rand und einer Überschrift. |
| TPanel | Ein randloser Container. |
| TSplitter | Ein Element um Container zu verschieben. |
| TComboBox | Ein Editfeld mit einer ausklappbaren Liste. |
| TEdit | Ein Editfeld. |
| TLabeledEdit | Ein Editfeld mit einer Überschrift. |
| TLabel | Ein Text. |
| TTable | Eine Tabelle. |
| TMemo | Ein großes Textfeld. |
| TRichEdit | Ein großes Textfeld mit Möglichkeiten für Farbe. |

Genauere Erklärungen zu diesen Elementen kann [hier](#RaWqtKV)  gefunden werden.

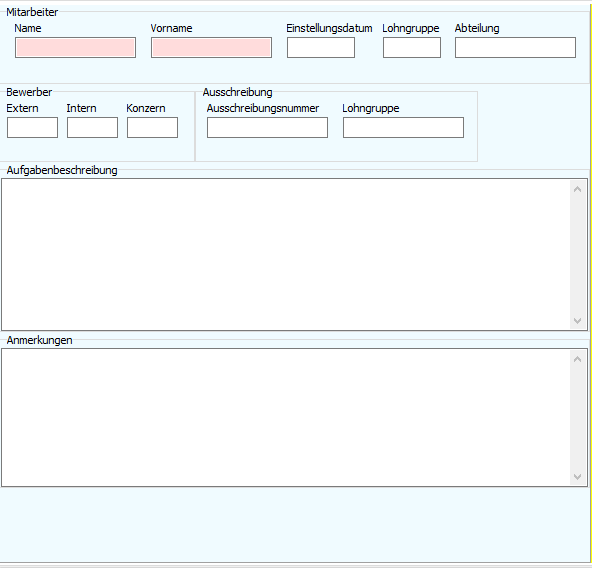
# Eigenschaften

Jedes Control hat einige Eigenschaften, die man verändern kann. Die Wichtigste ist der Name, denn er beschreibt die Aufgabe des [Controls](#sIDcUbwhneTeEmxQ) .

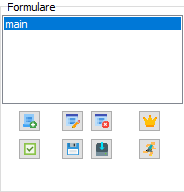


Die Eigenschaft "DataField" sollte ebenfalls gesetzt werden, ansonsten kann man zwar Daten in das Feld eingeben, aber diese werden nirgendwo gespeichert.

# Formularbearbeitungsbereich



# Formulare

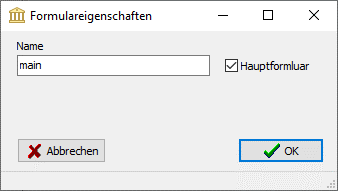


Die möglichen Aktionen für Formulare sind:

|  |  |
| --- | --- |
| **Icon** | **Aktion** |
|  | Neues Formular erstellen |
|  | Formular bearbeiten |
|  | Formular löschen |
|  | Als Hauptformular setzen |
|  | Formular prüfen(Prüft ob jedes Eingabefeld mit einem Datenfeld verbunden ist) |
|  | Testdaten speichern |
|  | Testdaten laden |
|  | Formular testen |

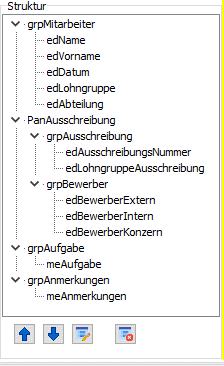
## Neues Formular

Jede Vorlage (außer einige Systemvorlagen) braucht ein Hauptformular:



# Struktur

Die Struktur zeigt die Anordnung der verschieden [Controls](#sIDcUbwhneTeEmxQ) in dem Formular an. Es ist eine gute Idee, zusammengehörige Elemente in Containern wie einer GroupBox oder einem Panel abzulegen.



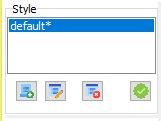
|  |  |
| --- | --- |
| **Icon** | **Aktion** |
|  | Bewegt ein Element in der Hierarchie nach oben. |
|  | Bewegt ein Element in der Hierarchie nach unten. |
|  | Benennt ein Element um. |
|  | Löscht ein Element. |

# Report

Der letzte Teil der Vorlage ist der Report. Hier wird definiert, wie die Daten für das Protokoll aufbereitet werden sollen. Grundsätzlich wird das Protokoll als HTML erzeugt, das dann als PDF ausgedruckt, signiert und archiviert werden kann.

Für die Darstellung der Daten in Texte werden HTML-Templates [[Web21]](#RobuSRI) genutzt. Wenn man es besonders dynamisch gestalten möchte, dann kann auch noch auf Delphi Web Script[[Del21]](#RxuUOWp) zurück gegriffen werden.

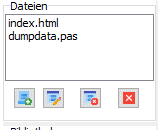
## Style



Per se hat jede Vorlage einen Standart-Style, der zur Darstellung der Daten automatisch gewählt wird. Über den Style kann der Bearbeiter verschieden Darstellungen auswählen, wenn z.B. bei einer Einstellung unterschieden werden soll, ob es externe Bewerber gab oder nicht.

Jeder Style hat immer die HTML-Datei "index.html, sie wird benutzt im den Inhalt der [Datenfelder](#LmyfynqqqMiliiKb) darzustellen. In das HTML werden [spezielle HTML-Tags](#aoKjeMlOsJvkOyLF) eingebettet, die dann im dem Inhalt der [Datenfelder](#LmyfynqqqMiliiKb) ersetzt werden.

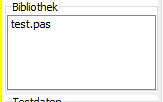
## Dateien



Zu jedem Style können auch weitere Dateien, z.B. Skripte, HTML-Dateien oder Bilder gehören. Bilder sollten einen absolut eindeutigen Namen haben der Art:

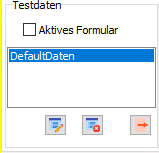
Einstellung\_default\_ok.jpg da es sonst zu Problemen bei der Darstellung kommt.

## Bibliothek



Für Scripte, die öfter genutzt werden können, gibt es die Bibliothek. Dort kann z.B. die Funktion Ausgabe des Abstimmungsergebnisses gespeichert werden.

## Testdaten



Für eine besseren Eindruck, wie das Ergebnis als Text aussieht, können Testdaten genutzt werden. Diese können gespeichert werden oder auch direkt aus dem Eingabeformular genutzt werden.

# DWS

Für die Ausführung von Scripten ist das DWS[[Del21]](#RxuUOWp) in das Programm integriert. Zur Nutzung sind grundlegende Kenntnisse in PSACAL bzw. [Delphi](#RDBhVcP) nötig.

# DWS Erweiterungen

Für die Anwendung wurde das DWS um einige Funktionen und Klassen erweitert.

## Funktionen

* function ScriptParamCount : integer;
* function ScriptParam( index : integer ) : string;
* function hasField( name : string ) : integer;
* fucntion getFieldStr( name : string ) : string;
* function getFieldInt( name : string ) : ínteger;
* function getTable( name : string ) : TTable;
* function printXML : string;

## Klassen

TTable

* property TableHeader : TTableHeader;
* function Rows : integer;
* function Cols : integer;
* function Cell( row : integer; col : integer ): string;

TTableHeader

* function count : integer;
* function Names : array of string;
* function name( index : integer ) : string;
* fucntion Caption( name : string ) : string;
* function Caption( index : integer ) : string;
* function Width( name : string ) : integer;
* function Width( index : integer ) : integer

# Spezielle HTML-Tags

Ein Beispiel für die Verwendung der Tags:

Mitarbeiter: <#field MA\_NAME>, <#field MA\_VORNAME><br>

Durch diese Anweisung wird ein Text der Art:

Mitarbeiter: Doe, John

erzeugt.

Folgende Tags gibt es:

* field
* table
* script
* system

## Field

Dies ist das häufigste Tag, denn es dient zur Ausgabe eines einfachen Datenfelds.

Beispiel:

<#field ma\_name>

## Table

Dieses Tag hat immer mindestens einen Parameter, den Tabellennamen. Es können nach dem Tabellennamen aber auch noch die Namen der darzustellenden [Datenfelder](#LmyfynqqqMiliiKb) angegeben werden.

Beispiele:

<#table Überstunden>

<#table Überstunden ma\_name ma\_vorname stunden>

<#table Überstunden stunden ma\_name ma\_vorname >

Das erste Beispiel gibt die vollständige Tabelle aus, während die anderen Beispiele nur Teile der Tabelle ausgeben und auch die Spalten in verschiedene Reihenfolgen ausgeben.

## Script

Mit diesem Tag wird ein DWS-Script aufgerufen. Es können auch noch Parameter mit an das Script übergeben werden. Scripte werden dann benutzt, wenn auf Grund von Bedingungen Ausgaben erfolgen sollen.

Ein gutes Beispiel ist das Abstimmungsergebnis eines Beschlusses. Einem Beschluss kann zugestimmt werden oder er wird abgelehnt. Entweder Einstimmig oder mit Gegenstimmen/Enthaltungen.

Da könnte man auch über mehrere Styles realisieren, dann hätte man aber viel Pflegeaufwand, hier bietet sich ein Script an, das auf Grund des Abstimmungsergebnisses den richtigen Text erzeugt.

## System

Mit diesem Tag werden spezielle Informationen aus dem Programm ausgegeben.

Mögliche Parameter des Tags:

|  |  |
| --- | --- |
| **Name** | **Bedeutung** |
| title | Der Titel der Aufgabe |
| date | Das aktuelle Datum im Format dd.mm.yyyy |
| time | Die aktuelle Zeit im Format hh:mm |
| user | Der Name des aktuelle Benutzers |
| host | Der Name des Computers |

# Textbausteine

Textbausteine sind immer wieder kehrende Texte, die man in den Beschlüssen oder Protokollen benötigt:

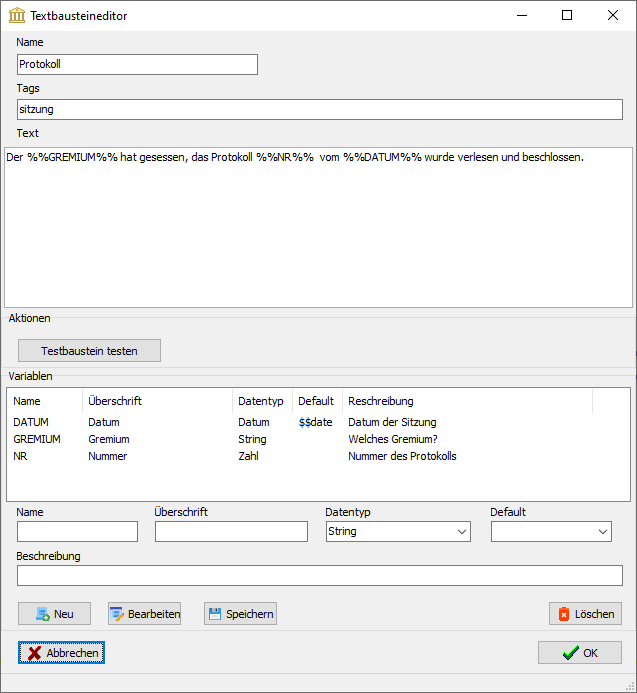
*Der* ***PSA*** *hat gesessen und das Protokoll Nr.* ***0815*** *kann eingesehen werden.*

In diesem Text sind der Name des Ausschusses und die Protokoll Nummer die beiden variablen Anteile.

*Der PSA stimmt der Einstellung zu, widerspricht aber der Eingruppierung.*

In diesem Satz sind keine variablen Anteile vorhanden.

Um solche Sätze nicht jedes mal neu schreiben zu müssen, kann man Textbausteine definieren, die man dann einfach per Drag&Drop in den Beschluss oder die Aufgabe ziehen kann.



Jeder Textbaustein braucht einen möglichst sprechenden, aber auch kurzen Namen. Über die Tags (Stichworte) kann man einen Baustein zu logischen Gruppen hinzufügen.

Der eigentliche Text des Bausteines kann auch Variablen enthalten, diese Informationen werden dann beim Einfügen von dem Benutzer abgefragt und in den Text eingetragen.

## Variablen

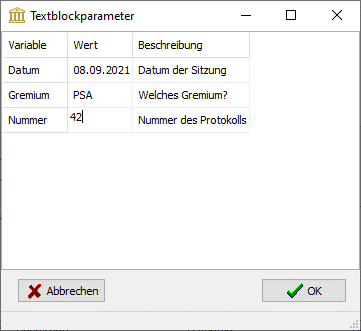
Jede Variable braucht einen, in diesem Baustein, eindeutigen Namen. Dieser Name wird mit jeweils %% eingeschlossen und später ersetzt. Die, meistens, besser lesbare Überschrift wird später im Eingabeformular angezeigt. Über ein Kontextmenü kann man die Variablen auch direkt in den Text an der aktuellen Stelle einfügen lassen.

Folgende Datentypen werden aktuell unterstützt:

* Zahl, eine ganze Zahl.
* Datum, ein Datumswert.
* String, ein freier Text.

Variablen können auch eine Voreinstellung haben, diese wird im Eingabeformular dann voreingestellt.

Weiterhin ist eine Beschreibung der Variable möglich. Sie sollte dem Benutzer als Erklärung dienen, wozu diese Variable gedacht ist.

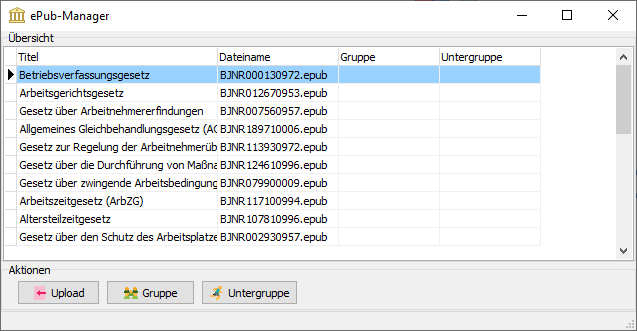


Beim Einfügen des Textbausteines werden dann die Variablen durch die neuen Inhalte ersetzt.

# Webseitendateien

# ePub Manager

Für die tägliche Arbeit sind Gesetze die Arbeitsgrundlage. Über die Seite [Gesetze im Internet](#RQAdMUU) kann man sich Gesetze als ePub runterladen.



# Dateicache

Über den Dateicache werden Daten zur Verfügung gestellt, die gemeinsam genutzt werden. Bei jedem Programmstart wird der lokale Cache geprüft und ggf. mit dem globalen Synchronisiert.

Wird eine Datei aktualisiert, so werden die Clients automatisch informiert und aktualisieren die entsprechende Datei.

Um die Daten besser zu strukturieren sind im Cache auch einfache Unterverzeichnisse erlaubt, bzw. einige vordefiniert.

## dwslib

* [misc.pas](http://misc.pas)

## data

* [Kategorie.JSON](http://Kategorie.JSON)

# Kategorie.JSON

Man kann den [Aufgaben](#RqksctvtyyAgQbUG) eigene Bezeichnungen und Farb-Codes zuweisen, dazu dient diese Datei.

Beispiel:

{

"Kategorie":

[

{

"name":"Unkritisch",

"farbe":"#A9F5A9"

},

{

"name":"Kritisch",

"farbe":"#FF0000"

},

{

"name":"Nachfrage",

"farbe":"#F7D358"

},

{

"name":"Externe",

"farbe":"#FE2EC8"

},

{

"name":"Unbearbeitet",

"farbe":"#FFFFFF"

}

]

}

# misc.pas

# Tools

hallo

# Open Source

|  |  |
| --- | --- |
| **Projekt** | **Quelle** |
| Delphi Web Script | <https://github.com/EricGrange/DWScript> |
| Synedit | <https://github.com/SynEdit/SynEdit> |
| LockBox | <https://github.com/TurboPack/LockBox> |
| Excel4Delphi | <https://github.com/rareMaxim/Excel4Delphi> |
| DragNDrop | https://github.com/landrix/The-Drag-and-Drop-Component-Suite-for-Delphi |

# Literaturverzeichnis

https://de.wikipedia.org/wiki/Liste\_sehr\_gro%C3%9Fer\_Sterne. (n.d.). Liste sehr großer Sterne.

flamerobin.org. (n.d.). flamerobin.org.