Digitale Waldbrandkarte (DWBK) Dokumentation

[**Übersicht**](#_fzi1i2kqyp0) **1**

[**Benutzungs-Voaraussetzungen**](#_opa0uognztlc) **2**

[Administrator](#_v339ds9tb199) 2

[Systemanforderungen](#_gut039qv472) 2

[System-Umgebungsvariable](#_b0jdy1i2534d) 2

[DWBK-Konfigurator](#_dcp5xx5ztdud) **3**

[Administrator-Settings](#_qi2sv2bvgpd) 3

[Konfigurations-Attribute](#_779kv8j24r5d) 3

[allowSavePassword](#_lh3h0f8loh5) 3

[logLevel](#_3wzp5r3iad9e) 3

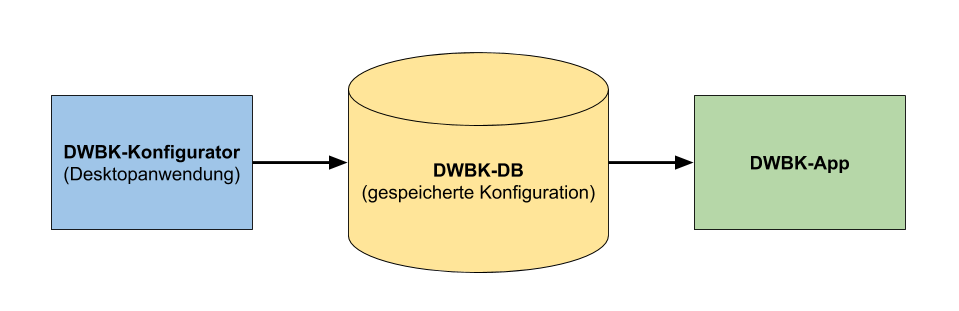
[wmsMaxThreads](#_87vhmcab72dj) 4

[Boundingbox-Konfiguration](#_vj7r3grfv15f) 4

[Layer Konfiguration](#_osq3juxvnsvq) 5

# Übersicht

Das folgende Schaubild stellt die Grundidee des DWBK-Projekts dar:

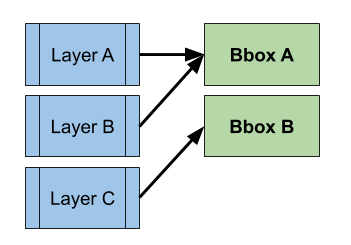


Mit der Desktopanwendung kann ein Fachanwender / Administrator die Datenbank verwalten, welche die Datengrundlage für die App ist.

Mit dem DWBK-Konfigurator werden 2 Typen verwaltet:

* Layer: Ein Layer ist eine Kartenebene, auf welcher Fachinformationen dargestellt werden
* Boundingbox: Eine Boundingbox ist ein Rechteck, welches einen räumlichen Ausschnitt definiert

In der Anwendung hat jeder Layer eine Boundingbox. Eine Boundingbox kann aber für mehrere Layer verwendet werden:



Es könnte Beispielsweise Sinn machen ein Luftbild für das gesamte Stadtgebiet zu definieren. Andere Informationen können wiederum nur für bestimmte Stadtteile oder Außengebiete relevant sein.

# Benutzungs-Voaraussetzungen

## Administrator

Die Datenbank muss über den DWBK-Konfigurator administriert werden. Hierfür sollte es mindestens einen Fachanwender / Administrator geben.

Hilfreich (aber nicht zwingend erforderlich) sind grundlegende GIS-Kenntnisse.

Alle notwendigen Konfigurations-Möglichkeiten werden in diesem Dokument erläutert.

## Systemanforderungen

Um den DWBK-Konfigurator verwenden zu können, muss Java installiert sein.

Zum Speichern der Datenbank sollte ausreichend Speicherplatz zur Verfügung stehen. Der Bedarf an Speicherplatz kann insbesondere schnell steigen, wenn viele Layer in einer hohen Auflösung lokal gespeichert werden sollen. Das betrifft natürlich sowohl den Arbeitsplatz an welchem die Konfiguration ausgeführt wird als auch die Endgeräte, welche die App verwenden.

Zur Verteilung der konfigurierten Datenbank können entsprechende Cloud-Speicher sehr hilfreich sein.

## System-Umgebungsvariable

Um die Anwendung verwenden zu können, muss folgende Umgebungsvariable definiert werden:

**DWBK\_CONFIG\_DIRECTORY**

Die Variable muss auf einen Ordner zeigen. Für den Ordner ist zwingend eine Schreibberechtigung erforderlich.

Folgende Dateien werden in dem Ordner abgelegt:

* dwbk\_db.gpkg (TODO: eigenes Kapitel zur Beschreibung der Datenbank)
* dwbk\_settings.json (siehe TODO: LINK zu SETTINGS)
* dwbk.log (TODO: eigenes Kapitel zum Logging)

# DWBK-Konfigurator

## Administrator-Settings

Es ist möglich eigene Settings für die Konfiguration zu speichern.

Die Settings werden beim Start der Anwendung automatisch generiert und als JSON in dem definierten Konfigurationsordner (DWBK\_CONFIG\_DIRECTORY) abgelegt. Der Dateiname ist immer "dwbk\_settings.json".

Die Anwendung kann nicht gestartet werden, wenn die Settings fehlerhaft sind. Die Anwendung informiert den Benutzer über die Fehlkonfiguration und es wird ein Log-Eintrag erstellt, welcher zur Fehlerbehebung (TODO: siehe LOGGING) genutzt werden kann. Aktuell sieht die Standardkonfiguration wie folgt aus:

{

"allowSavePassword": true,

"logLevel": "INFO",

"wmsMaxThreads": 8

}

### Konfigurations-Attribute

Im Folgenden werden die Attribute und deren Einstellungsmöglichkeiten erläutert.

##### **allowSavePassword**

*Typ: boolean*

Das Attribut legt fest, ob das Speichern von Passwörtern in die lokale Datenbank (TODO: LINK DB.GPKG) erlaubt sein soll.

Für einige Dienste ist die Eingabe von Benutzername und Passwort erforderlich. Das Speichern der Daten kann den Administrator unterstützen.

Wenn das Speichern erlaubt ist, werden die Passwörter immer verschlüsselt. Wenn sich jemand die Datenbank anschaut ist das Passwort also nicht direkt ersichtlich!

##### **logLevel**

*Typ: string*

Gibt das Log-Level an welches nach dem Start der Applikation verwendet werden soll. Bis zum Start der Anwendung wird immer das Level "INFO" (Standardwert) verwendet.

Das Level muss einem der folgenden Standard-Log-Levels entsprechen:

* SEVERE (höchster Wert)
* WARNING
* INFO
* CONFIG
* FINE
* FINER
* FINEST (niedrigster Wert)

##### **wmsMaxThreads**

*Typ: number*

Gibt die Anzahl der maximal zu verwendeten Threads zurück, welche verwendet werden um die einzelnen WMS Bild-Kacheln zu laden.

Der Wert muss eine Ganzzahl > 0 sein. Als grobe Richtzahl kann die Anzahl der Prozessorkerne dienen.

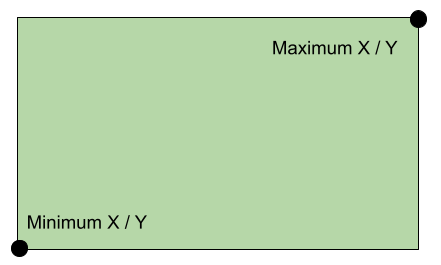
Es kann insbesondere Sinn machen mehrere Anfragen loszuschicken, wenn der angefragte Dienst langsam antwortet.

Ein zu großer Wert hat kein Optimierungspotential, da die Bild-Kacheln nur von einem einzigen Thread geschrieben werden. Dies ist erforderlich, weil die Reihenfolge der Tilematrix ([Geopackage Spefifikation](http://www.geopackage.org/spec/)) gewährleistet werden soll.

## Boundingbox-Konfiguration

Die Boundingbox ist ein Rechteck, welches einen räumlichen Ausschnitt definiert. Im Wesentlichen wird eine Boundingbox durch 2 Koordinatenpaare (bzw. 4 Koordinaten) definiert:

1. Untere linke Ecke: Minimum X / Y
2. Obere rechte Ecke: Maximum X / Y



Zunächst sollte der Gesamtausschnitt definiert werden, auf welchen die App später beschränkt ist. Ohne Boundingbox können keine Layer erstellt werden.

Im Folgenden werden die Konfigurationsmöglichkeiten erläutert.

## Layer Konfiguration

Ein Layer ist eine Kartenebene, auf welcher Fachinformationen dargestellt werden. Die Layer werden über Services angesprochen. Es werden 2 Arten von Services für Layer unterstützt:

1. WMS (Web Map Service): Rasterdaten (Bildkacheln)
2. WFS (Web Feature Service): Vektordaten (Geometrien)

Im Folgenden werden die Konfigurationsmöglichkeiten erläutert.

- Die URI muss ein GetMap-Request sein.

- Die URI muss im Browser das gewünschte Raster darstellen. Konfigurationsfehler sollten hier bereits auffallen.

- Für den kompletten Layer wird die konfigurierte Boundingbox verwendet (und nicht die bbox aus der URI).

- Es sollten alle Parameter angegeben werden, welche berücksichtigt werden sollen (insbesondere die Pflichtparameter...)

- WMS-Layer werden lokal immer in "EPSG:3857" gespeichert.

- Wenn ein Layer nicht in "EPSG:3857" angefragt werden kann, wird das Koordinatensystem der GetMap-URI verwendet und anschließend transformiert

- Zum Erstellen der Raster-Pyramide werden alle Konfigurierten Genauigkeiten berücksichtigt

- Die Genauigkeit wird in ganzen Metern angegeben

- Erlaubte Genauigkeiten sind ausschließlich Teiler von 500 (500 % <Genauigkeit> = 0) ?!

- Mehrere Genauigkeiten werden durch Semikolon (';') getrennt angegeben