

NAME

ADMISSIONS, ADMISSIONS_get0_admissionAuthority, ADMISSIONS_get0_namingAuthority,
 ADMISSIONS_get0_professionInfos, ADMISSIONS_set0_admissionAuthority,
 ADMISSIONS_set0_namingAuthority, ADMISSIONS_set0_professionInfos, ADMISSION_SYNTAX,
 ADMISSION_SYNTAX_get0_admissionAuthority,
 ADMISSION_SYNTAX_get0_contentsOfAdmissions, ADMISSION_SYNTAX_set0_admissionAuthority,
 ADMISSION_SYNTAX_set0_contentsOfAdmissions, NAMING_AUTHORITY,
 NAMING_AUTHORITY_get0_authorityId, NAMING_AUTHORITY_get0_authorityURL,
 NAMING_AUTHORITY_get0_authorityText, NAMING_AUTHORITY_set0_authorityId,
 NAMING_AUTHORITY_set0_authorityURL, NAMING_AUTHORITY_set0_authorityText,
 PROFESSION_INFO, PROFESSION_INFOS, PROFESSION_INFO_get0_addProfessionInfo,
 PROFESSION_INFO_get0_namingAuthority, PROFESSION_INFO_get0_professionItems,
 PROFESSION_INFO_get0_professionOIDs, PROFESSION_INFO_get0_registrationNumber,
 PROFESSION_INFO_set0_addProfessionInfo, PROFESSION_INFO_set0_namingAuthority,
 PROFESSION_INFO_set0_professionItems, PROFESSION_INFO_set0_professionOIDs,
 PROFESSION_INFO_set0_registrationNumber – Accessors and setters for ADMISSION_SYNTAX

SYNOPSIS

```
typedef struct NamingAuthority_st NAMING_AUTHORITY;
typedef struct ProfessionInfo_st PROFESSION_INFO;
typedef STACK_OF(PROFESSION_INFO) PROFESSION_INFOS;
typedef struct Admissions_st ADMISSIONS;
typedef struct AdmissionSyntax_st ADMISSION_SYNTAX;

const ASN1_OBJECT *NAMING_AUTHORITY_get0_authorityId(
    const NAMING_AUTHORITY *n);
void NAMING_AUTHORITY_set0_authorityId(NAMING_AUTHORITY *n,
    ASN1_OBJECT* namingAuthorityId);
const ASN1_IA5STRING *NAMING_AUTHORITY_get0_authorityURL(
    const NAMING_AUTHORITY *n);
void NAMING_AUTHORITY_set0_authorityURL(NAMING_AUTHORITY *n,
    ASN1_IA5STRING* namingAuthorityUrl);
const ASN1_STRING *NAMING_AUTHORITY_get0_authorityText(
    const NAMING_AUTHORITY *n);
void NAMING_AUTHORITY_set0_authorityText(NAMING_AUTHORITY *n,
    ASN1_STRING* namingAuthorityText);

const GENERAL_NAME *ADMISSION_SYNTAX_get0_admissionAuthority(
    const ADMISSION_SYNTAX *as);
void ADMISSION_SYNTAX_set0_admissionAuthority(
    ADMISSION_SYNTAX *as, GENERAL_NAME *aa);
const STACK_OF(ADMISSIONS) *ADMISSION_SYNTAX_get0_contentsOfAdmissions(
    const ADMISSION_SYNTAX *as);
void ADMISSION_SYNTAX_set0_contentsOfAdmissions(
    ADMISSION_SYNTAX *as, STACK_OF(ADMISSIONS) *a);

const GENERAL_NAME *ADMISSIONS_get0_admissionAuthority(const ADMISSIONS *a);
void ADMISSIONS_set0_admissionAuthority(ADMISSIONS *a, GENERAL_NAME *aa);
const NAMING_AUTHORITY *ADMISSIONS_get0_namingAuthority(const ADMISSIONS *a);
void ADMISSIONS_set0_namingAuthority(ADMISSIONS *a, NAMING_AUTHORITY *na);
const PROFESSION_INFOS *ADMISSIONS_get0_professionInfos(const ADMISSIONS *a);
void ADMISSIONS_set0_professionInfos(ADMISSIONS *a, PROFESSION_INFOS *pi);

const ASN1_OCTET_STRING *PROFESSION_INFO_get0_addProfessionInfo(
    const PROFESSION_INFO *pi);
```

```

void PROFESSION_INFO_set0_addProfessionInfo(
    PROFESSION_INFO *pi, ASN1_OCTET_STRING *aos);
const NAMING_AUTHORITY *PROFESSION_INFO_get0_namingAuthority(
    const PROFESSION_INFO *pi);
void PROFESSION_INFO_set0_namingAuthority(
    PROFESSION_INFO *pi, NAMING_AUTHORITY *na);
const STACK_OF(ASN1_STRING) *PROFESSION_INFO_get0_professionItems(
    const PROFESSION_INFO *pi);
void PROFESSION_INFO_set0_professionItems(
    PROFESSION_INFO *pi, STACK_OF(ASN1_STRING) *as);
const STACK_OF(ASN1_OBJECT) *PROFESSION_INFO_get0_professionOIDs(
    const PROFESSION_INFO *pi);
void PROFESSION_INFO_set0_professionOIDs(
    PROFESSION_INFO *pi, STACK_OF(ASN1_OBJECT) *po);
const ASN1_PRINTABLESTRING *PROFESSION_INFO_get0_registrationNumber(
    const PROFESSION_INFO *pi);
void PROFESSION_INFO_set0_registrationNumber(
    PROFESSION_INFO *pi, ASN1_PRINTABLESTRING *rn);

```

DESCRIPTION

The **PROFESSION_INFOS**, **ADMISSION_SYNTAX**, **ADMISSIONS**, and **PROFESSION_INFO** types are opaque structures representing the analogous types defined in the Common PKI Specification published by T7 & TELETRUST <[https://www.bundesnetzagentur.de/EVD/DE/SharedDocuments/Downloads/Anbieter_Infothek/Common_PKI_v2.0_02.pdf?Knowledge of those structures and their semantics is assumed](https://www.bundesnetzagentur.de/EVD/DE/SharedDocuments/Downloads/Anbieter_Infothek/Common_PKI_v2.0_02.pdf?Knowledge%20of%20those%20structures%20and%20their%20semantics%20is%20assumed)>.

The conventional routines to convert between DER and the local format are described in **d2i_X509**(3). The conventional routines to allocate and free the types are defined in **X509_dup**(3).

The **PROFESSION_INFOS** type is a stack of **PROFESSION_INFO**; see **DEFINE_STACK_OF**(3) for details.

The **NAMING_AUTHORITY** type has an authority ID and URL, and text fields. The **NAMING_AUTHORITY_get0_authorityId()**, **NAMING_AUTHORITY_get0_get0_authorityURL()**, and **NAMING_AUTHORITY_get0_get0_authorityText()**, functions return pointers to those values within the object. The **NAMING_AUTHORITY_set0_authorityId()**, **NAMING_AUTHORITY_set0_get0_authorityURL()**, and **NAMING_AUTHORITY_set0_get0_authorityText()**, functions free any existing value and set the pointer to the specified value.

The **ADMISSION_SYNTAX** type has an authority name and a stack of **ADMISSION** objects. The **ADMISSION_SYNTAX_get0_admissionAuthority()** and **ADMISSION_SYNTAX_get0_contentsOfAdmissions()** functions return pointers to those values within the object. The **ADMISSION_SYNTAX_set0_admissionAuthority()** and **ADMISSION_SYNTAX_set0_contentsOfAdmissions()** functions free any existing value and set the pointer to the specified value.

The **ADMISSION** type has an authority name, authority object, and a stack of **PROFESSION_INFO** items. The **ADMISSIONS_get0_admissionAuthority()**, **ADMISSIONS_get0_namingAuthority()**, and **ADMISSIONS_get0_professionInfos()** functions return pointers to those values within the object. The **ADMISSIONS_set0_admissionAuthority()**, **ADMISSIONS_set0_namingAuthority()**, and **ADMISSIONS_set0_professionInfos()** functions free any existing value and set the pointer to the specified value.

The **PROFESSION_INFO** type has a name authority, stacks of profession Items and OIDs, a registration number, and additional profession info. The functions **PROFESSION_INFO_get0_addProfessionInfo()**, **PROFESSION_INFO_get0_namingAuthority()**, **PROFESSION_INFO_get0_professionItems()**, **PROFESSION_INFO_get0_professionOIDs()**, and **PROFESSION_INFO_get0_registrationNumber()** functions return pointers to those values within the object. The

PROFESSION_INFO_set0_addProfessionInfo(), **PROFESSION_INFO_set0_namingAuthority()**, **PROFESSION_INFO_set0_professionItems()**, **PROFESSION_INFO_set0_professionOIDs()**, and **PROFESSION_INFO_set0_registrationNumber()** functions free any existing value and set the pointer to the specified value.

RETURN VALUES

Described above. Note that all of the *et0* functions return a pointer to the internal data structure and must not be freed.

SEE ALSO

X509_dup (3), **d2i_X509** (3),

COPYRIGHT

Copyright 2017–2026 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at [<https://www.openssl.org/source/license.html>](https://www.openssl.org/source/license.html).