

Unique References

References are important to prevent a lack of clarity and unambiguousness within the [IT documentation](#). If you want to reference [objects](#) for third-party systems, for example, various [attributes](#) can be used for this task.



Data imports

In order to detect already existing data, there are relevant strategies for [data imports](#). For the most part, they refer to the attributes mentioned below. Details can be found in the corresponding articles.

Object ID

Contents

Each object in i-doit receives an identifier (ID). This identifier is always unique for each i-doit tenant. The ID is a positive number greater than 0 and can have a length of up to 11 digits. It is automatically assigned by MySQL/MariaDB by incrementing the number by 1. A retroactive change of the ID by the user is not possible in order to prevent inconsistencies. When you [delete objects \(Purge\)](#), the IDs are free again but they are not reusable.

The object ID is represented as the `Object ID` attribute in the [General category](#).

Object ID

1459

The number from which the counter for the object IDs starts is determined during the [setup of i-doit](#). The default starting number is 1.

The object ID is very important for the generation of the URIs. By means of the ID you can access objects in an unambiguous way via a link. The name of the variable is `objID`. The naming structure is:

```
http://i-doit.example.net/i-doit/?objID=1000
```

Object IDs are especially used for internal purposes, but they can also be helpful for the user or for third-party systems. The IDs are essential for [API access](#), for example.

SYS-ID

The `sys-ID` attribute is unique within all tenants and installations. For this purpose, a 10-digits (being the minimum) positive number is generated from the [UNIX timestamp](#). This number is displayed in the [General category](#). It is synonymously used unhyphenated as `SYSID`.

SYSID

SYSID_1375970889

You can specify a prefix for each [object type](#), which precedes the automatically generated number. You can set the prefix at [Administration CMDB settings Object type configuration \[Object type group\] \[Object type\] SYSID prefix](#). If you do not specify your own prefix, the SYSID will begin with the `SYSID_` prefix. When you set an alternative prefix, the generation of the number changes: The object ID is used instead of the UNIX timestamp.

SYSID prefix

SERVER_

By default, the SYSID cannot be changed retroactively by the user. However, you can enable changes by activating the corresponding function at [Administration CMDB settings General settings SYS-ID readonly](#).

Object Title

It is absolutely essential that each object receives a title. This title is documented as an attribute in the [General category](#). Synonymously this attribute is also called `Name` or `Object link`.

Fileserver01 

Title

- %OBJID%** = 4124
- %OBJTYPEID%** = 5
- %TIMESTAMP%** = Timestamp (1464773815)
- %Y%** = Full year (2016)
- %y%** = Abbreviated year (16)

 **Allocate object titles automatically**

Object titles can also be assigned automatically. This is especially helpful for [templates](#):

1. Create template
2. Use the available placeholders as object title and save
3. Assign the template as default template for the object type
4. Create new object, the object title is already set

IP Address

You can assign as many IP addresses to each object as you wish. This is done in the **Hostaddress** category by creating a category entry for each IP address.

IP addresses are not unique per se. For example, in private networks the same addresses (192.168.1.1 etc.) recur again and again. When [Layer-3 networks](#) with overlapping network areas are documented in i-doit, there often will be objects with ambiguous IP addresses. Due to technologies like DHCP it becomes even harder to identify objects by using dynamically assigned IP addresses. Hence, this attribute is **not suited** for unambiguous referencing in most cases.

Hostname and FQDN

The **Hostname** is assigned in the **Hostaddress** category. For each object any number of host names can be documented, even if it is in many cases assigned unambiguously for each device (for example servers). As with the IP address (see above), a category entry is generated for each host name.

When you also specify the **DNS domain** attribute per category entry, this automatically results in the Fully Qualified Domain Name (**FQDN**).

Just like IP addresses, the name should be unambiguous but due to various reasons it can also be a recurring name. Oftentimes the FQDN is equated to the object title. The usage of host name or FQDN for referencing should be considered carefully.

IPv4 address	<input type="text" value="192.168.0.40"/>	 
Netmask	<input type="text" value="255.255.0.0"/> / <input type="text" value="16"/>	
Default gateway for the net	<input type="text" value="No"/>	
Hostname	<input type="text" value="zn-srv01.i-doit.int"/>	
DNS server	<input type="text" value=""/>	 
DNS domain	<input type="text" value="f x"/>	
FQDN	zn-srv01.i-doit.int.f	

MAC Address

As a rule, MAC addresses are unique worldwide since they are set by the manufacturers of the network components. The low risk of recurring MAC addresses is however only right in theory. Due to the increasing virtualization and the resulting change of MAC addresses by users, this attribute is not always the first choice for unambiguous referencing. As many MAC addresses as desired can be set by entering a MAC address for each port in the **Port (Network)** category. When using technologies like bonding or trunking, the MAC address is documented in the **Logical port (Network)** category.

MAC-address	00:80:42:23:ab:01
-------------	-------------------

Inventory Number

In most cases the inventory of property is carried out across the entire organization and is therefore not only an IT-related topic (Asset Management). Each material property which is to be inventoried receives a unique number. In i-doit this number can be documented for each object in the **Inventory number** attribute within the **Accounting** category.

At **Administration** **CMDB settings** **Object type configuration** [**Object type group**] [**Object type**] the automated generation of inventory numbers can be configured, if desired. Here various placeholders are available for creating a naming structure per object type.

Automatic inventory number	<div style="border: 1px solid #ccc; padding: 5px;"><p>%OBJID% = 5947</p><p>%OBJTYPEID% = 5</p><p>%OBJTITLE% = Objekt-Titel</p><p>%SYSID% = SYSID_1464774274</p><p>%TIMESTAMP% = Timestamp (1464774274)</p></div>
----------------------------	---

✓ Labels

Inventory numbers and QR codes offer an ideal combination and can be joined to labels. QR codes are generated for each object by i-doit. Such labels can be printed using the i-doit [QR-code printer](#), for example.

Identifier of Third-Party Systems

Data from third-party system, for example from the inventory/ discovery area, can be [imported](#) to i-doit. In most cases, these systems also possess unambiguous identifiers or something similar. When these identifiers are imported, the category **Data source** can be used. This way, multiple used systems can be synchronized in order to reference objects on all systems unambiguously.

Workplace: Arbeitsplatz von Adam Riese (Data Source)

SYS-ID: **SYSID_1376042065** Location: Düsseldorf > Zentrale > Raum 00.01

Purpose: **Production** Contact assignment

Relationship: **Implicit (6), Explicit (0)** Primary access URL: -

Key	<input type="text"/>	
Value	<input type="text"/>	
Type	-	<input type="button" value="v"/>
Group	<input type="text"/>	
Last scan	-	01:00:00
Initial Import	-	01:00:00
Last updated	<input type="text"/>	

Description

B I U S I x x = = - - ← → ↻

Constants

Internally, i-doit uses constants that unambiguously reference all kinds of data. A constant consists of capital letters (A-Z) and underscores (_). Such constants are used for some objects that already exist in the default installation of i-doit. The `Person` object "admin", for example, has the constant `C_OBJ_PERSON_ADMIN`.

These constants cannot be viewed or even edited in the web GUI. For this reason, this is done in SQL in the database. Each tenant has a separate database in which the `isys_obj` table exists. All objects are saved there. The column for the constant is called `isys_obj_const`.



Database manipulation

Manipulations of the database endanger the functionality of *i-doit*. For this reason, we cannot provide any warranties for problems resulting from manipulations. Thus any activity should be well considered and agreed with us beforehand, if necessary.

Further Unique Attributes

- Attributes can be [defined as unique](#) globally, per object type or (for list categories) per object. In case of multiple occurrences a notification window by i-doit is displayed when you try to save your entries.
- If the attributes mentioned above do not suffice, additional ones can be [created](#).