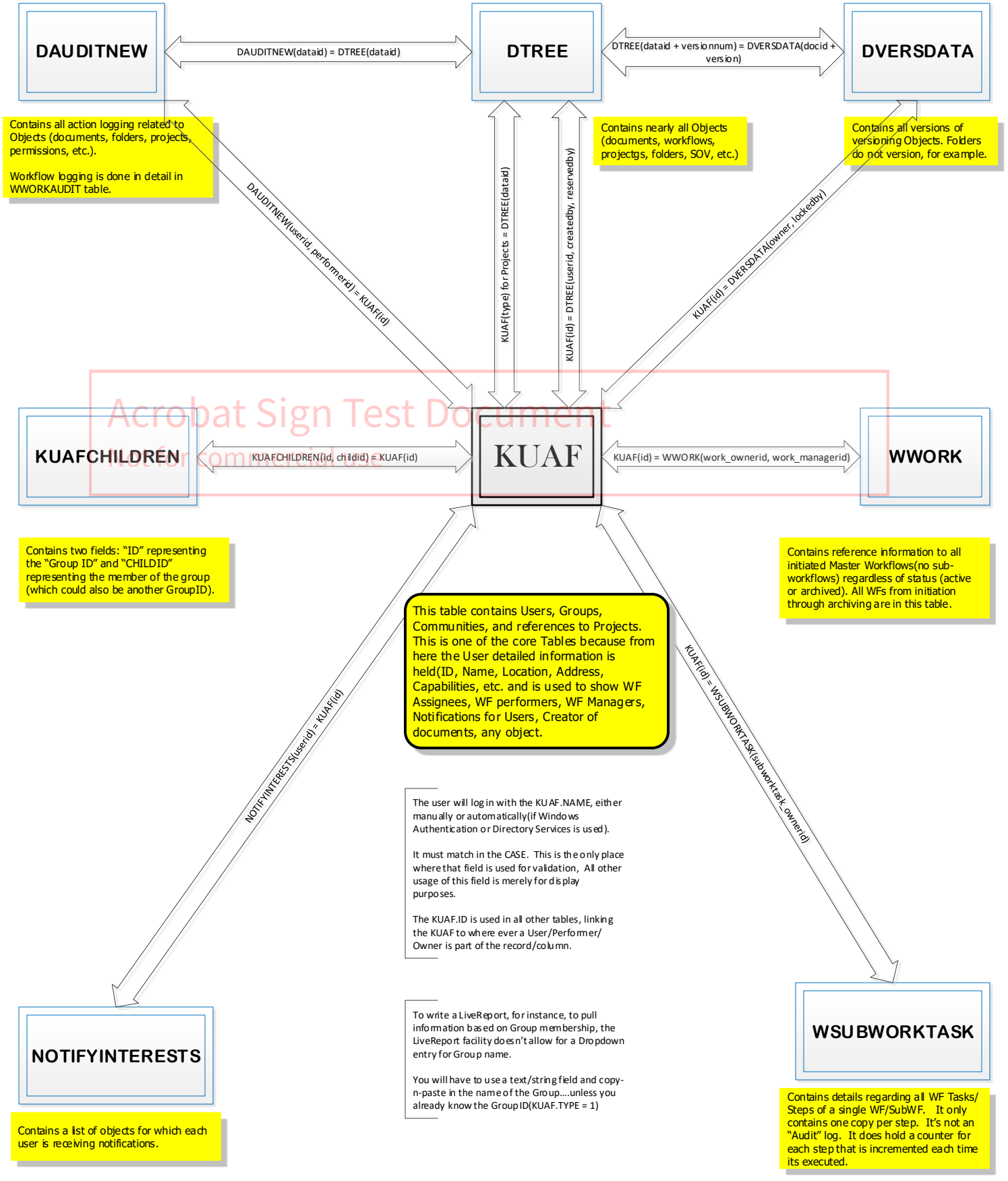
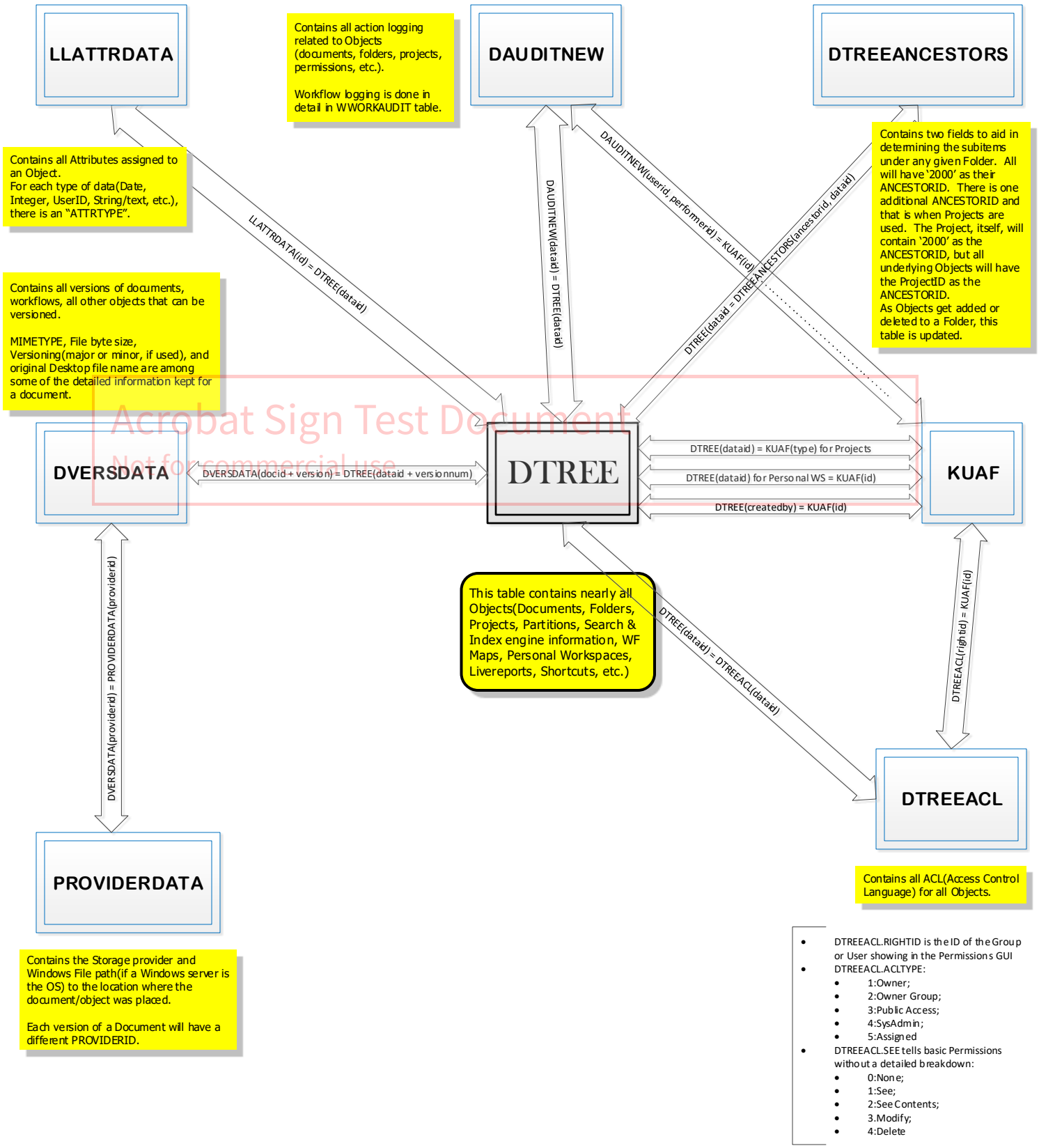
KUAF – User/Group

Table Relationships



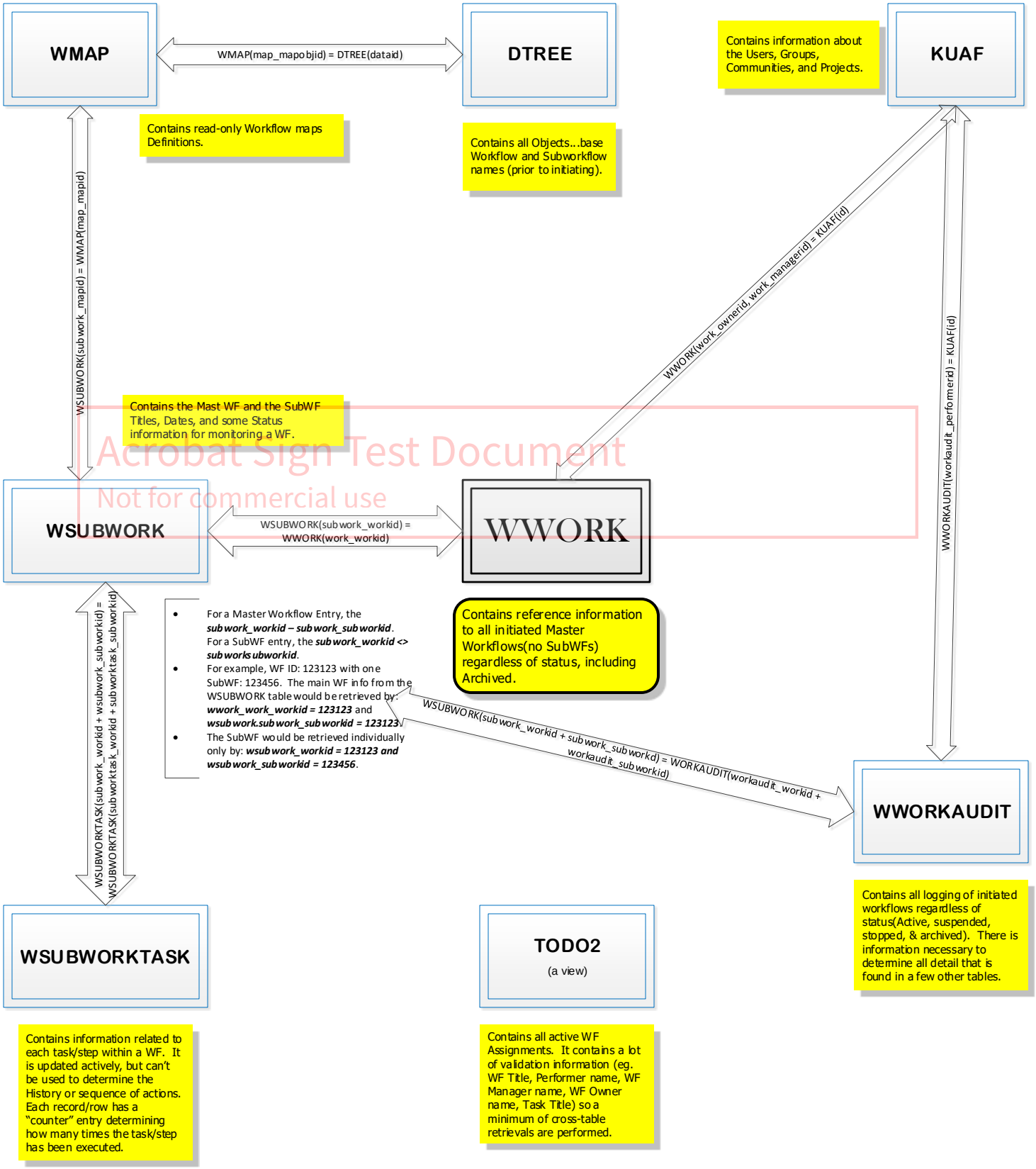
DTREE – All Objects

Table Relationships

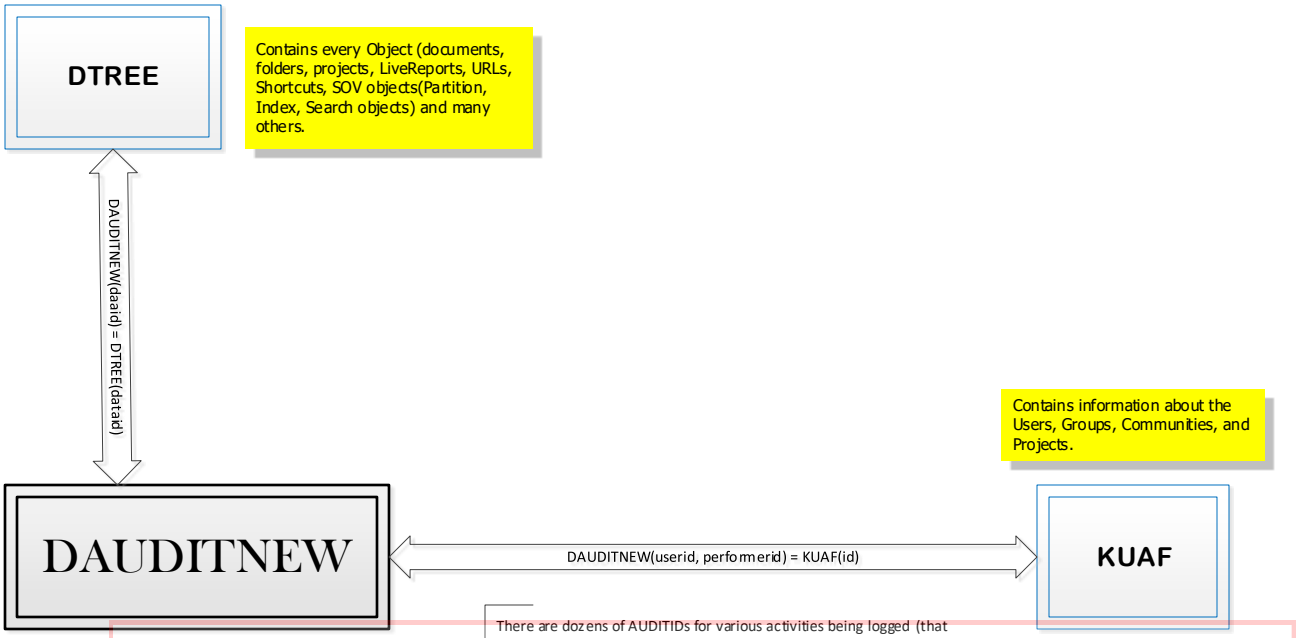


WWORK – All Workflows

Table Relationships



DAUDITNEW – Action logging



There are dozens of AUDITIDs for various activities being logged (that can be turned on or off in this table via the Admin pages.)

There are AUDITIDs that are for Groups-only, Users-only, and a couple are shared with 'normal' Objects: Create and Delete.

To retrieve information for a UserID v. an Object, and a couple of differences exist. For a User, the DATAID and the SUBTYPE are NULL. Check for these to be NULL or you might get other Objects. Nearly all others will have a DATAID and a SUBTYPE.

A list of some of the values can be found in the Champion Toolkit document:
Cntent_Server_Schema_Companion_Reference_Guide.pdf found in the OTKC. In Oracle, you can get a list by running this SQL:

```
select distinct da.auditid, da.auditstr
from dauditnew da
order by da.auditid
```

The EVENTID is non-repeating sequencing number. This field can be used to limit the amount or scope of a query into this table. Limiting a search by the AUDITDATE will cause a prolonged query process. EVENTID is indexed and never repeats. The 1st event of each month and year can be determined and used as a starting/ending point and thereby limiting the scope.

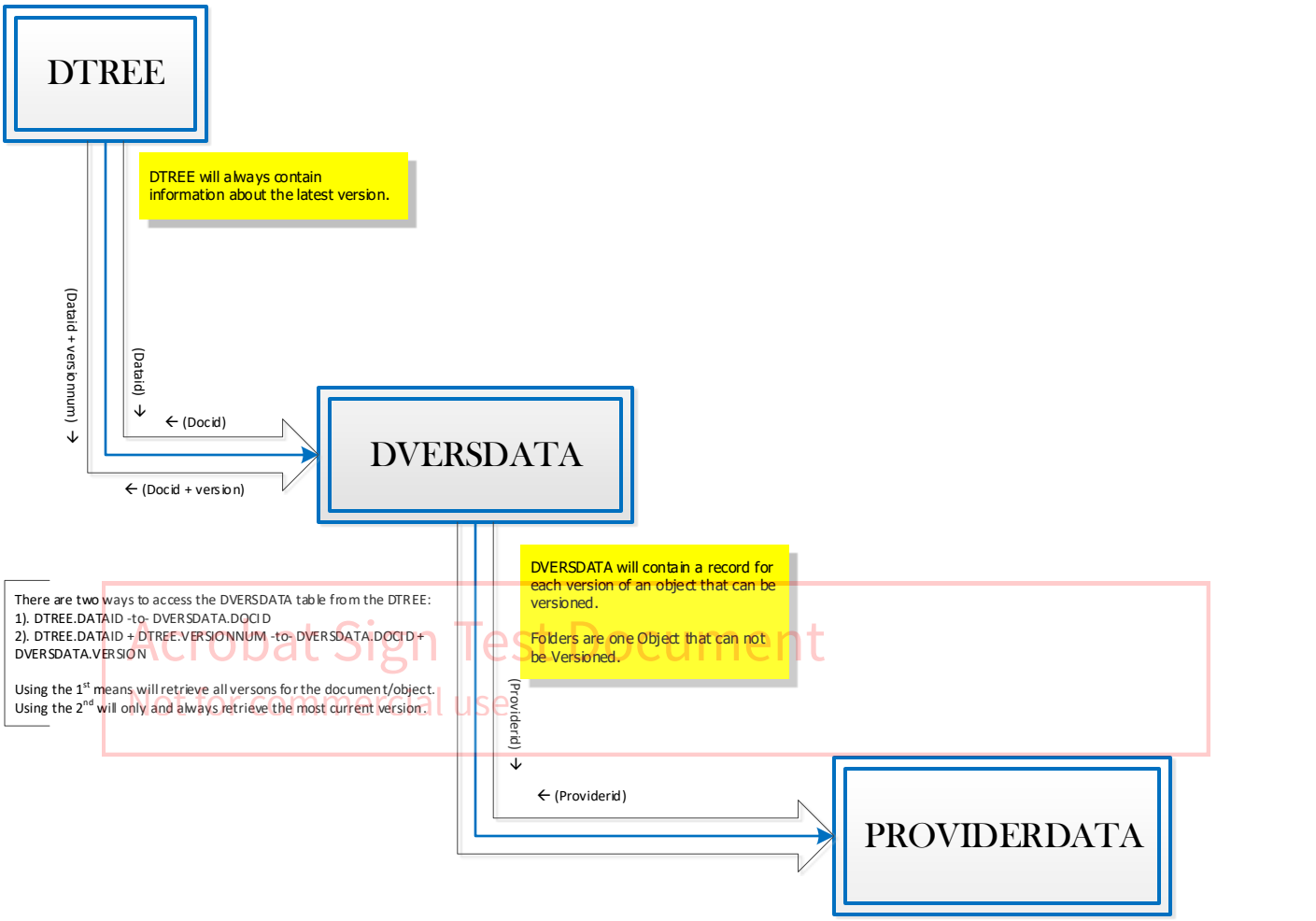
For example, if the 1st entry for 2015 is '1382313' and '8455798' for 2016 for their EVENTIDs, you can use it to limit your search within the Year 2015 by specifying: "EVENTID between 1382313 and 8455798" instead of: "to_char(AUDITDATE, 'yyyy') = '2015'". A function in a SQL query adds significant processing time and this would have to include the entire table since it's not an indexed column.

There are three fields that contain different values depending upon the AUDITID value: VALUEKEY, VALUE1, & VALUE2.

The complexity of all of the combinations would take more than this page. Rather than detail it here, I'd recommend looking at the LRs that I have for various searches/reports. Some report activity without interpreting the values in VALUEKEY, VALUE1, & VALUE2.

The one listing Permissions changes is no more complex than when a document is added. It has to hold previous and new values. It also has to show whether it as done to one item or to sub-items, as well.

Document Table Relationships



There are two ways to access the DVERSDATA table from the DTREE:
1). DTREE.DATAID -to- DVERSDATA.DOCID
2). DTREE.DATAID + DTREE.VERSIONNUM -to- DVERSDATA.DOCID + DVERSDATA.VERSION

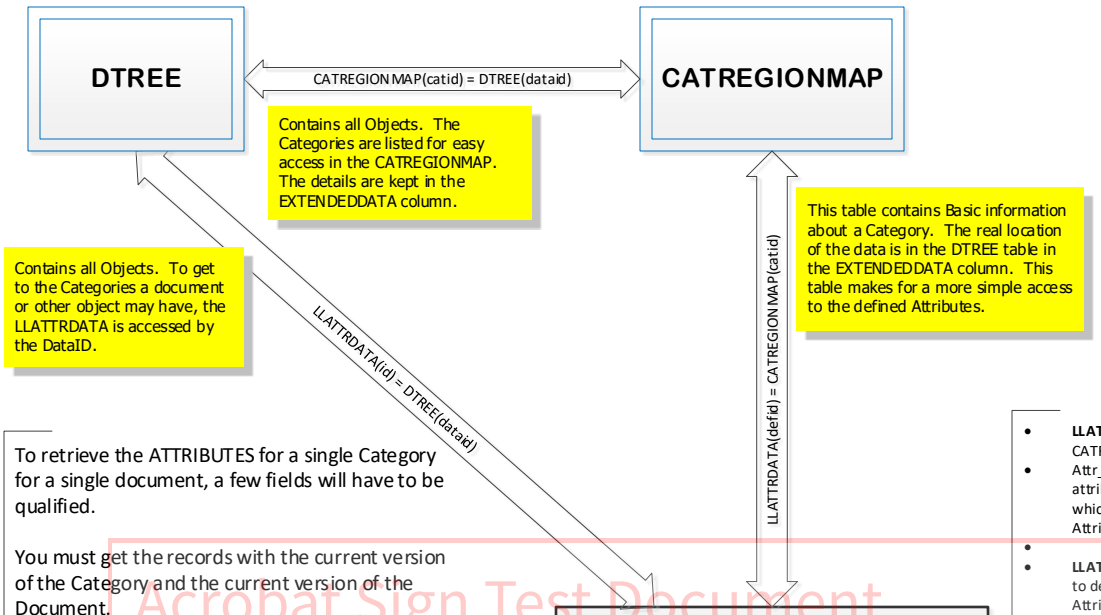
Using the 1st means will retrieve all versons for the document/object.
Using the 2nd will only and always retrieve the most current version.

PROVIDERDATA will contain a record with the file path of where the document/version was placed by the server when it was added to the system.
(eg: 0002/008/20089321.dat)

The document is not encrypted when placed out in the EFS. It is in its native format, but it's renamed to be more easily retrieved.

LLATTRDATA – Category Attribute values

Table Relationships



To retrieve the ATTRIBUTES for a single Category for a single document, a few fields will have to be qualified.

You must get the records with the current version of the Category and the current version of the Document.

DTREE.dataid = LLATTRDATA.id
DTREE.versionum = LLATTRDATA.vernum
CATREGIONMAP.catid = DTREE.dataid and
DTREE.versionnum = LLATTRDATA.defvern for the latest version of the Category

Following the retrieval, the various fields must be interpreted because as the table to the right indicates, there are a variety of types and these will use one of several available fields that are designed to hold these varying types.

Contains all Attributes assigned to an Object.
For each type of data(Date, Integer, UserID, String/text, etc.), there is an "ATTRTYPE".

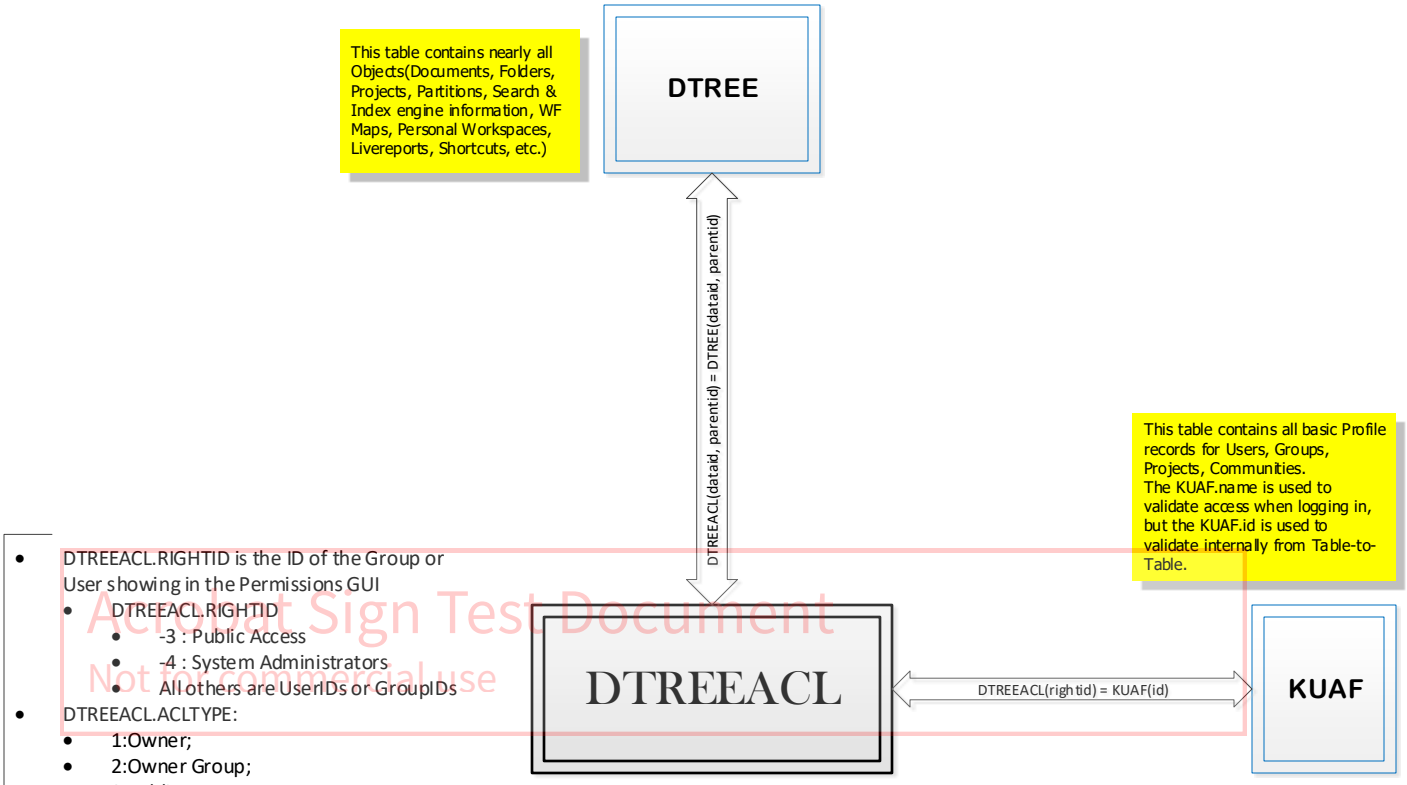
- **LLATTRADATA.ATTRID** is the trailing Digit in the CATREGIONMAP.regionname
- Attr_46018068_8 '8' is a sequencing number for each attribute added to a Category. This is used to identify which Attribute by also getting the name of the Attribute from CATREGRIONMAP.attrname.
-
- **LLATTRDATA.ATTRTYPE** is the value that must be used to determine what type of field was chosen when the Attribute was added to the Category.
- -18 is the "header". It contains the name of the Category.
- -7 – a DATE field is used.
- -4 – a REAL number is used.
- -1 – a STRING/TEXT field is used
- 2 – an INTEGER is used.
- 5 – a BOOLEAN/FLAG field is used(Yes/No, On/Off)
- 10 – a STRING POPUP field is used.
- 11 – a STRING MULTILINE field is used.
- 12 – an INTEGER POPUP field is used.
- 13 – a DATA POPUP field is used.
- 14 – a USER ID field is used.
- 20 – a REAL POPUP field is used.
- 101 – a GROUP ID field is used.
- 305 – an ITEM REFERENCE (DataID) is used.
- 10013 – a TABLE Key is used.
- 10131 – ???

Below is a CASE statement in MS SQL SERVER syntax to determine, parse out, and provide the interpretive data

```
case
    when llad2.attrtype=-18 then 'Category: ['+cast(llad2.valstr as varchar)+']'  /* Category (Parent) */
    when llad2.attrtype=-7 then '{Date}'+DATENAME(YEAR, llad2.valdate)+'/'+DATENAME(MONTH, llad2.valdate)+'/'+DATENAME(DAY, llad2.valdate)
    when llad2.attrtype=-4 then '{'+cast(llad2.attrtype as varchar)+'}'+cast(llad2.valreal as varchar)
    when llad2.attrtype=-1 then '{String/Text}'+cast(llad2.valstr as varchar)
    when llad2.attrtype=2 then '{'+cast(llad2.attrtype as varchar)+'}'+cast(llad2.valint as varchar)
    when llad2.attrtype=5 and llad2.valint = 0 then '{Boolean}'+'False'
    when llad2.attrtype=5 and llad2.valint = 1 then '{Boolean}'+'True'
    when llad2.attrtype=5 and llad2.valint is NULL then '{Boolean}'+'n/a'
    when llad2.attrtype=10 then '{String Popup}'+cast(llad2.valstr as varchar)
    when llad2.attrtype=11 then '{String Multiline}'+cast(llad2.valstr as varchar)
    when llad2.attrtype=12 then '{Integer Popup}'+cast(llad2.valint as nvarchar)
    when llad2.attrtype=14 then '{UserID} User ['+cast(llad2.valint as nvarchar)+' ]'+k.name
    when llad2.attrtype=10013 then '{Table Key}'+cast(llad2.valstr as varchar)
    when llad2.attrtype=10131 then '{GroupID}'+cast(llad2.valstr as varchar)
else 'Other AttrType ['+cast(llad2.attrtype as varchar)+']' end AttributeValue
```

DTREEACL – Permissions for All Objects

Table Relationships



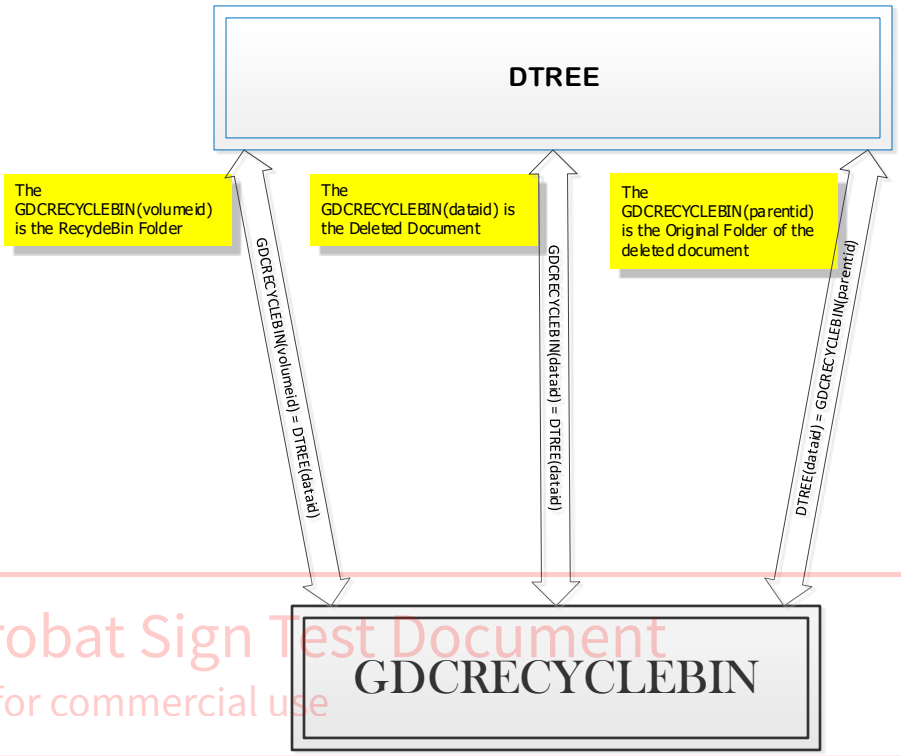
- DTREEACL.RIGHTID is the ID of the Group or User showing in the Permissions GUI
  - DTREEACL.RIGHTID
    - -3 : Public Access
    - -4 : System Administrators
    - All others are UserIDs or GroupIDs
- DTREEACL.ACLTYPE:
  - 1:Owner;
  - 2:Owner Group;
  - 3:Public Access;
  - 4:SysAdmin;
  - 5:Assigned
- DTREEACL.SEE tells basic Permissions without a detailed breakdown:
  - 0:None;
  - 1:See;
  - 2:See Contents;
  - 3.Modify;
  - 4:Delete

ORACLE SQL to extract by BIT value the correct Permissions

```
substr(decode(bitand(dta.permissions,2),0,'None, ')||
decode(bitand(dta.permissions,2),2,'See | ')||
decode(bitand(dta.permissions,36865),36865,'See Contents | ')||
decode(bitand(dta.permissions,65536),65536,'Modify | ')||
decode(bitand(dta.permissions,16),16,'Edit Permissions | ')||
decode(bitand(dta.permissions,131072),131072,'Edit Attributes | ')||
decode(bitand(dta.permissions,4),4,'Add Items | ')||
decode(bitand(dta.permissions,16384),16384,'Delete Versions | ')||
decode(bitand(dta.permissions,8),8,'Delete | ')||
decode(bitand(dta.permissions,8192),8192,'Reserve, '),
1,length(decode(bitand(dta.permissions,2),0,'None, ')||
decode(bitand(dta.permissions,2),2,'See | ')||
decode(bitand(dta.permissions,36865),36865,'See Contents | ')||
decode(bitand(dta.permissions,65536),65536,'Modify | ')||
decode(bitand(dta.permissions,16),16,'Edit Permissions | ')||
decode(bitand(dta.permissions,131072),131072,'Edit Attributes | ')||
decode(bitand(dta.permissions,4),4,'Add Items | ')||
decode(bitand(dta.permissions,16384),16384,'Delete Versions | ')||
decode(bitand(dta.permissions,8),8,'Delete | ')||
decode(bitand(dta.permissions,8192),8192,'Reserve, ')) -2)
```

RECYCLEBIN

Table Relationships



This table contains the entries when a Document or even a Folder is deleted and the RecycleBin is installed. As the path indicates, all three columns point back to the DTREE to obtain the names and locations of the various elements.

A report can written to show the path of both the RecycleBin and the original location in the Enterprise Workspace.

Signature: Augusto
Augusto (May 21, 2025 14:46 ADT)

Email: augusto.barone@outlook.com