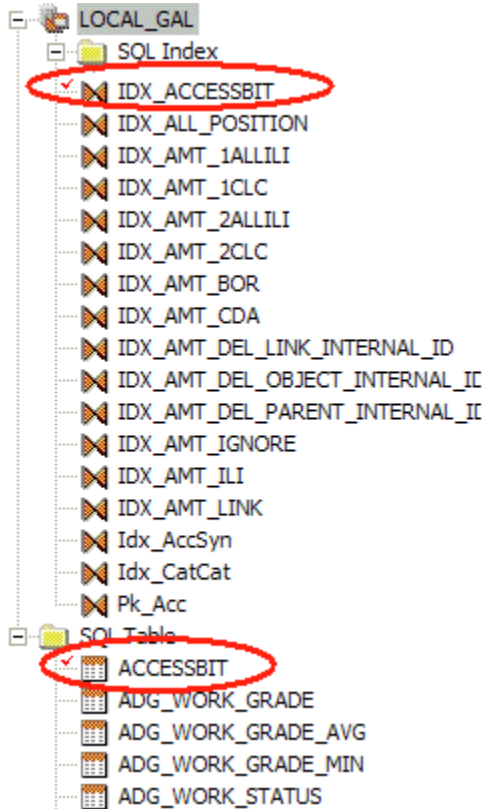


Modifying links between existing linked objects using CI_LINKS

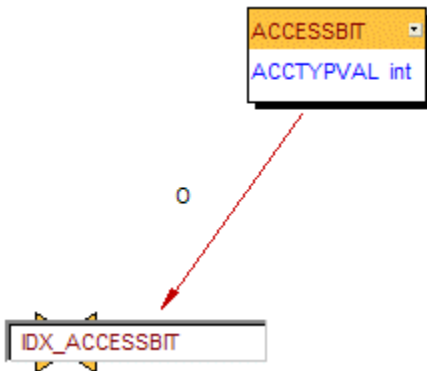
- Neither objects have an OBJECT_GUID

Neither objects have an OBJECT_GUID

This example will explain how to modify an existing "Rely On" link between linked objects ("ACCESSBIT" (caller) and "IDX_ACCESSBIT" (called)) that already exist in the CAST Analysis Service. Neither object has an OBJECT_GUID (they are T-SQL objects), and both belong to the same project. The link will be merged from a "Rely On" link into a "Call Rely On" link:



Current Rely On link is shown as follows:



The first step is to check the ID number of the existing link between the two objects by executing the following query against the CAST Analysis Service:

```

select
cgl.CALLER_GUID,
cgl.CALLED_GUID,
cl.LINK_TYPE_LO,
cl.LINK_TYPE_HI,
cl.LINK_TYPE_LO2,
cl.LINK_TYPE_HI2
from
CTV_GUID_LINKS cgl join CTV_LINKS cl
on cgl.LINK_ID = cl.LINK_ID
where
upper(cgl.CALLER_NAME) = 'ACCESSBIT'
and upper(cgl.CALLED_NAME) = 'IDX_ACCESSBIT'

```

The result confirms that the two objects are indeed linked and that the existing Rely On link has an ID of 32768:

CALLER_GUID	CALLED_GUID	LINK_TYPE_LO	LINK_TYPE_HI	LINK_TYPE_LO2	LINK_TYPE_HI2
[char]	[char]	[int]	[int]	[int]	[int]
NULL	NULL	32768	0	0	0

The next step (after having identified the LINK_TYPE_NAME of the Call link) is to confirm the ID numbers of the two links we are going to merge (one should be 32768 as identified above). Execute the following query against the CAST Analysis Service:

```

select
LINK_TYPE_NAME,
LINK_TYPE_LO,
LINK_TYPE_HI,
LINK_TYPE_LO2,
LINK_TYPE_HI2
from
CTV_LINK_TYPES
where
LINK_TYPE_NAME in ('relyonLink', 'callLink')

```

The result shows that the Rely On link has an ID of 32768 and the Call type link has an ID of 2048:

LINK_TYPE_NAME	LINK_TYPE_LO	LINK_TYPE_HI	LINK_TYPE_LO2	LINK_TYPE_HI2
[int]	[int]	[int]	[int]	[int]
callLink	2048	0	0	0
relyonLink	32768	0	0	0

So at the end of the merge process, the newly created Call Rely On link will have an ID of **34816** (i.e. 32768+2048).

The next step is to insert the data into the CAST entry tables that will cause the links between the T-SQL table and index to merged when the tool is run. Add a new **Update CAST Knowledge Base Tool** and enter the following query:

Microsoft SQL Server

```

insert into <KB_name>.dbo.GUID_OBJECTS (OBJECT_ID, OBJECT_GUID, ERROR_ID)
select OBJECT_ID, OBJECT_FULLNAME, 0
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'ACCESSBIT'
go
insert into <KB_name>.dbo.GUID_OBJECTS (OBJECT_ID, OBJECT_GUID, ERROR_ID)
select OBJECT_ID, OBJECT_FULLNAME, 0
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'IDX_ACCESSBIT'
go
insert into <KB_name>.dbo.CI_LINKS (CALLER_GUID, CALLED_GUID, LINK_TYPE, ERROR_ID)
select
(select OBJECT_FULLNAME from <KB_name>.dbo.CTV_GUID_OBJECTS where OBJECT_NAME = 'ACCESSBIT') CALLER_GUID,
(select OBJECT_FULLNAME from <KB_name>.dbo.CTV_GUID_OBJECTS where OBJECT_NAME = 'IDX_ACCESSBIT') CALLED_GUID,
'callLink',
0
Go

```

CAST Storage Service

```

insert into <KB_name>.GUID_OBJECTS (OBJECT_ID, OBJECT_GUID, ERROR_ID)
select OBJECT_ID, OBJECT_FULLNAME, 0
from <KB_name>.CTV_GUID_OBJECTS
where OBJECT_NAME = 'ACCESSBIT'
/
insert into <KB_name>.GUID_OBJECTS (OBJECT_ID, OBJECT_GUID, ERROR_ID)
select OBJECT_ID, OBJECT_FULLNAME, 0
from <KB_name>.CTV_GUID_OBJECTS
where OBJECT_NAME = 'IDX_ACCESSBIT'
/
insert into <KB_name>.CI_LINKS (CALLER_GUID, CALLED_GUID, LINK_TYPE, ERROR_ID)
select
(select OBJECT_FULLNAME from <KB_name>.CTV_GUID_OBJECTS where OBJECT_NAME = 'ACCESSBIT') CALLER_GUID,
(select OBJECT_FULLNAME from <KB_name>.CTV_GUID_OBJECTS where OBJECT_NAME = 'IDX_ACCESSBIT') CALLED_GUID,
'callLink',
0
/

```

Oracle

```

insert into <KB_name>.CI_LINKS (CALLER_ID, CALLED_ID, LINK_TYPE, ERROR_ID)
select
(select OBJECT_ID from <KB_name>.CTV_GUID_OBJECTS where OBJECT_NAME = 'ACCESSBIT') CALLER_ID,
(select OBJECT_ID from <KB_name>.CTV_GUID_OBJECTS where OBJECT_NAME = 'IDX_ACCESSBIT') CALLED_ID,
'callLink',
0 from DUAL

```

For **Microsoft SQL Server** and **CAST Storage Service**, this is in effect three queries:

- The first two create OBJECT_GUIDs for the two T-SQL objects using the objects' full name.
- The third inserts the data in the CI_LINKS table to modify the link.

For **Oracle**:

- The query inserts the data in the CI_LINKS table to modify the link.



- The 0 parameter will enter 0 in the ERROR_ID column of the CI_LINKS table.
- You can also use variables to replace the Analysis Service name if required. See the on-line Help for the CAST Management Studio for more information.

Then:

- Complete the configuration of the tool and run it as outlined in [What is the Update CAST Knowledge Base Tool](#).
- Make sure you then update the CAST System Views.

If you want to check that the new link has been created successfully, you can use the following query in an SQL IDE against your Analysis Service (adapting it for your environment):

```
select
cgl.CALLER_GUID,
cgl.CALLED_GUID,
cl.LINK_TYPE_LO,
cl.LINK_TYPE_HI,
cl.LINK_TYPE_LO2,
cl.LINK_TYPE_HI2
from
CTV_GUID_LINKS cgl join CTV_LINKS cl
on cgl.LINK_ID = cl.LINK_ID
where
upper(cgl.CALLER_NAME) = 'ACCESSBIT'
and upper(cgl.CALLED_NAME) = 'IDX_ACCESSBIT'
```

The result shows a successful link creation:

CALLER_GUID	CALLER_GUID	CALLER_GUID	CALLER_GUID
[char]	[char]	[char]	[char]
WESLEY\WESLEY.LOCAL_GAL..ACCESSBIT	WESLEY\WESLEY.LOCAL_GAL..IDX_ACCESSBIT	WESLEY\WESLEY.LOCAL_GAL..ACCESSBIT	WESLEY\WESLEY.LOCAL_GAL..IDX_ACCESSBIT

LINK_TYPE_LO	LINK_TYPE_HI	LINK_TYPE_LO2	LINK_TYPE_HI2
[int]	[int]	[int]	[int]
34816	0	0	0

The result of the query shows that the LINK_TYPE_LO ID between the two objects is now 34816, as expected. You can also check in CAST Enlighten - F5 to refresh the Object Browser - new Call Rely On link (CO) between the two objects:

