

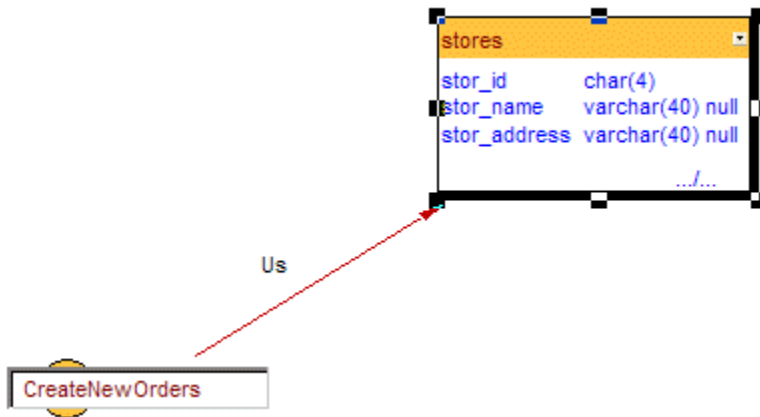
# Removing an existing link using CI\_NO\_LINKS

## On this page:

- [Based on object's ID](#)
- [Based on object's GUID](#)
  - [Between two objects with OBJECT\\_GUIDs](#)
  - [Between two objects that do not have OBJECT\\_GUIDs](#)
  - [Between one object with an OBJECT\\_GUID and one object without an OBJECT\\_GUID](#)

## Based on object's ID

In this example, we are going to remove a link between two SQL objects (a T-SQL procedure called "CreateNewOrders" and a T-SQL table called "stores") from the CAST Analysis Service. The image below show the link in CAST Enlighten:



The first step is to check that the link exists in the CAST Analysis Service using the CAST System Views. Use the following query in an SQL IDE against your CAST Analysis Service:

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'CreateNewOrders'
and CALLED_NAME = 'stores'
```

This query confirms this:

```
Cnt
[int]
----
1
```

The next step is to insert the data into the CAST entry tables that will cause the link in question to be removed from the Analysis Service when the tool is run. Add a new **Update CAST Knowledge Base Tool** and enter the following query:

```
insert into <KB_name>.dbo.CI_NO_LINKS (CALLER_ID, CALLED_ID, ERROR_ID)
select CALLER_ID, CALLED_ID, 0
from <KB_name>.dbo.CTV_GUID_LINKS
where
CALLER_NAME = 'CreateNewOrders'
and CALLED_NAME = 'stores'
go
```



- The 0 parameter will enter 0 in the ERROR\_ID column of the CI\_NO\_LINKS table.
- Note that you need to specify the Analysis Service database and user as above "<KB\_name>.dbo" otherwise the query will fail when the job is run.
- 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 job and run it as outlined in [Using the Update CAST Knowledge Base Tool](#).
- Make sure you then update the CAST System Views.

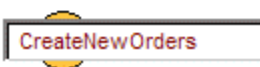
If you want to check that the link has indeed been removed successfully, you can use the following query in an SQL IDE against your Analysis Service (adapting it for your environment):

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'CreateNewOrders'
and CALLED_NAME = 'stores'
```

The result returns 0 for this OBJECT\_NAME:

```
Cnt
[int]
----
0
```

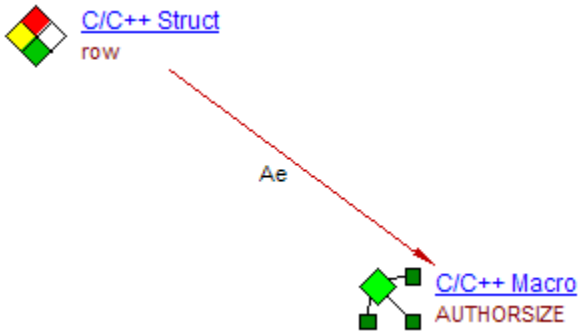
You can also check in CAST Enlighten – F5 to refresh the view - link no longer exists:



Based on object's GUID

### Between two objects with OBJECT\_GUIDs

In this example, we are going to remove a link between two C++ objects (a Structure called "row" and a Macro called "AUTHORSIZE") from the CAST Analysis Service. The image below shows the link in CAST Enlighten:



The first step is to check that the link exists in the CAST Analysis Service using the CAST System Views. Use the following query in an SQL IDE against your CAST Analysis Service:

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'row'
and CALLED_NAME = 'AUTHORSIZE'
```

This query confirms this:

```
Cnt
[int]
----
1
```

The next step is to insert the data into the CAST entry tables that will cause the link in question to be removed from the Analysis Service when the tool is run. Add a new **Update CAST Knowledge Base Tool** and enter the following query:

```
insert into <KB_name>.dbo.CI_NO_LINKS (CALLER_GUID, CALLED_GUID, ERROR_ID)
select CALLER_GUID, CALLED_GUID, 0
From <KB_name>.dbo.CTV_GUID_LINKS
where
CALLER_NAME = 'row'
and CALLED_NAME = 'AUTHORSIZE'
go
```



- The 0 parameter will enter 0 in the ERROR\_ID column of the CI\_NO\_LINKS table.
- Note that you need to specify the Analysis Service database and user as above "<KB\_name>.dbo" otherwise the query will fail when the job is run.
- 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 job and run it as outlined in [Using the Update CAST Knowledge Base Tool](#).
- Make sure you then update the CAST System Views.

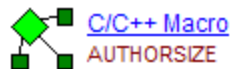
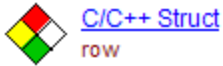
If you want to check that the link has indeed been removed successfully, you can use the following query in an SQL IDE against your Analysis Service (adapting it for your environment):

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'row'
and CALLED_NAME = 'AUTHORSIZE'
```

The result returns 0 for this OBJECT\_NAME:

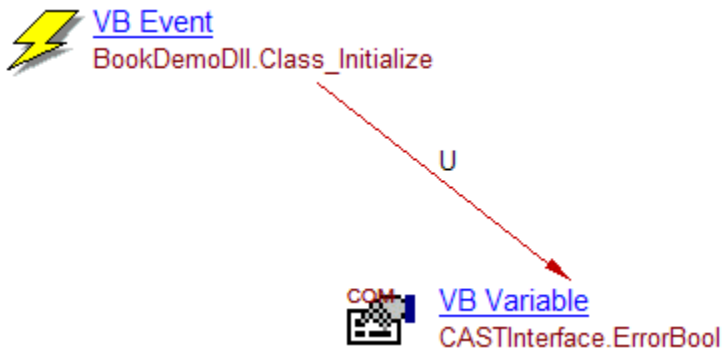
```
Cnt
[int]
----
0
```

You can also check in CAST Enlighten - F5 to refresh the view - link no longer exists:



## Between two objects that do not have OBJECT\_GUIDs

In this example, we are going to remove a link between two VB objects (VB Event called "Class\_Initialize" and a VB Variable called "ErrorBool") from the CAST Analysis Service. The image below shows the link in CAST Enlighten:



The first step is to check that the link exists in the CAST Analysis Service using the CAST System Views. Use the following query in an SQL IDE against your CAST Analysis Service:

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'Class_Initialize'
and CALLED_NAME = 'ErrorBool'
```

This query confirms this:

```
Cnt
[int]
----
1
```

The next step is to insert the data into the CAST entry tables that will cause the link in question to be removed from the Analysis Service and to create the OBJECT\_GUID for the objects in question when the tool is run. Add a new **Update CAST Knowledge Base Tool** and enter the following query:

```

insert into <KB_name>.dbo.GUID_OBJECTS (OBJECT_ID, OBJECT_GUID, ERROR_ID)
select OBJECT_ID, OBJECT_NAME, 0
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'Class_Initialize'
union
select OBJECT_ID, OBJECT_NAME, 0
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'ErrorBool'
go
insert into <KB_name>.dbo.CI_NO_LINKS (CALLER_GUID, CALLED_GUID, ERROR_ID)
select (select OBJECT_NAME
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'Class_Initialize'),
(select OBJECT_NAME
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'ErrorBool'), 0
go

```

This is in effect two queries:

- The first inserts the two object's OBJECT\_ID and OBJECT\_NAME into the GUID\_OBJECTS table to create a correspondence and also creates a data union.
- The second inserts the data in the CI\_NO\_LINKS table to remove the link between the two objects.



- The 0 parameter will enter 0 in the ERROR\_ID column of the tables in question.
- Note that you need to specify the Analysis Service database and user as above "<KB\_name>.dbo" otherwise the query will fail when the job is run.
- 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.
- Please see [Creating OBJECT\\_GUIDs](#) for more information about creating OBJECT\_GUIDs for objects that do not have them.

Then:

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

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

```

select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'Class_Initialize'
and CALLED_NAME = 'ErrorBool'

```

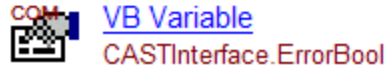
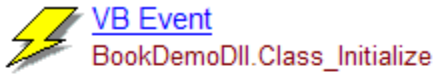
The result returns 0 for this OBJECT\_NAME:

```

Cnt
[int]
----
0

```

You can also check in CAST Enlighten - F5 to refresh the view - link no longer exists:



Finally you can also check that the new OBJECT\_GUID for each object is correct:

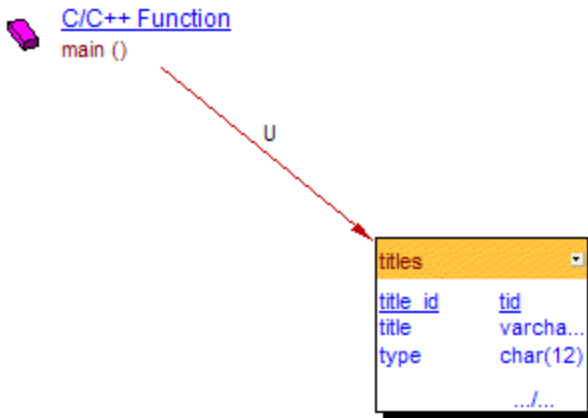
```
select
OBJECT_NAME, OBJECT_GUID
from CTV_GUID_OBJECTS
where OBJECT_NAME
in ('Class_Initialize' , 'ErrorBool')
```

The results show that all is in order:

OBJECT_NAME	OBJECT_GUID
[char]	[char]
Class_Initialize	Class_Initialize
ErrorBool	ErrorBool

## Between one object with an OBJECT\_GUID and one object without an OBJECT\_GUID

In this example, we are going to remove a link between a C++ object (a Function called "main ()") and a T-SQL server object (a Table called "titles") from the CAST Analysis Service. Note that the C++ object has a GUID and the T-SQL server object does not. The image below shows the link in CAST Enlighten:



The first step is to check that the link exists in the CAST Analysis Service using the CAST System Views. Use the following query in an SQL IDE against your CAST Analysis Service:

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'main ()'
and CALLED_NAME = 'titles'
```

This query confirms this:

```
Cnt
[int]
----
1
```

You can also check which object has a GUID and which does not:

```
Select CALLER_GUID, CALLED_GUID
From CTV_GUID_LINKS
Where CALLER_NAME = 'main ()'
and CALLED_NAME = 'titles'
go
```

This confirms that the C++ object has a GUID and that the T-SQL server object does not:

```
CALLER_GUID
[char]
C_Fct.main.C_Fi."D:\TESTAPPLICATIONS\SQL_SERVER\DEMO_C\EXAMPLE..C"
CALLED_GUID
[char]
NULL
```

Thus a GUID must be created for the "titles" object.

The next step is to insert the data into the CAST entry tables that will cause the link in question to be removed from the Analysis Service and to create the OBJECT\_GUID for the object in question when the tool is run. Add a new **Update CAST Knowledge Base Tool** and enter the following query:

```
insert into <KB_name>.dbo.GUID_OBJECTS (OBJECT_ID, OBJECT_GUID, ERROR_ID)
select OBJECT_ID, OBJECT_NAME, 0
from <KB_name>.dbo.CTV_GUID_OBJECTS
where OBJECT_NAME = 'titles'
go
insert into <KB_name>.dbo.CI_NO_LINKS (CALLER_GUID, CALLED_GUID, ERROR_ID)
select CALLER_GUID, CALLED_NAME, 0
from <KB_name>.dbo.CTV_GUID_LINKS
where CALLER_NAME = 'main ()'
and CALLED_NAME = 'titles'
go
```

This is in effect two queries:

- The first inserts the object's OBJECT\_ID and OBJECT\_NAME into the GUID\_OBJECTS table to create a correspondence.
- The second inserts the data in the CI\_NO\_LINKS table to remove the link between the two objects.



- The 0 parameter will enter 0 in the ERROR\_ID column of the tables in question.
- Note that you need to specify the Analysis Service database and user as above "<KB\_name>.dbo" otherwise the query will fail when the job is run.
- 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.
- Please see [Creating OBJECT\\_GUIDs](#) for more information about creating OBJECT\_GUIDs for objects that do not have them.

Then:

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


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

```
select count(1) Cnt
from CTV_GUID_LINKS
where
CALLER_NAME = 'main ()'
and CALLED_NAME = 'titles'
```

The result returns 0 for this OBJECT\_NAME:

```
Cnt
[int]
----
0
```

You can also check in CAST Enlighten – F5 to refresh the view - link no longer exists:

 [C/C++ Function](#)  
main ()



title_id	tid
title	varcha...
type	char(12)
...	...

Finally you can also check that the new OBJECT\_GUID for the object that did not have one is correct:

```
select
OBJECT_NAME,
OBJECT_GUID
from CTV_GUID_OBJECTS
where OBJECT_NAME = 'titles'
```

The results show that all is in order:

```
OBJECT_NAME OBJECT_GUID
[char]       [char]
titles      titles
```