

# EWRemoveAttached

---

Removes the attached file specified by a position in the named field of the specified table record.

## Syntax

```
int n = ew.EWRemoveAttached(String sessionId, String  
                             tableName, long id, String fieldName, int position);
```

## Usage

Use the EWRemoveAttached call to remove an attached file from a File or Image field in a record in the table.

## Rules and Guidelines

When removing attached files, consider the following rules and guidelines:

- The username that was used to obtain the specified session token must have sufficient access rights to modify individual records within the specified table. Please verify specific permissions via **Setup > Access > Manage Groups > (Edit Group) > Table > (Edit Table) > Permissions**.
- Agiloft allows specifying fine-grained access permissions on the field level. The username that was used to obtain the specified session token must have sufficient access rights to be able to update field content. Please verify specific permissions via **Setup > Access > Manage Groups > (Edit Group) > Table > (Edit Table) > Field Permissions**.
- This call requires the position of the attached file in the specified field. Position numbering starts from 0.
- One can use the EWRead method to obtain an array of file names for the attached field in the specified field. The sequence of file names in the array will correspond to the positions of the files.
- Only one file per call is removed.
- In general, you use EWRemoveAttached when you know in advance the IDs of the records to retrieve. The client application may use the likes of EWSelectFromTable call to obtain record identifiers beforehand or take the identifiers from the id field of the data structures and linking classes.

# Basic Steps for Removing Attached Files

Removing files from records involves the following basic steps:

1. Determine the id of the record you want to update. You may want to use the EWSelectFromTable or EWSearchTable calls to get the identifier of the record based on some search condition or get the identifier from a previous EWCreate call, or get the id of a linked record from the linking class after performing EWRead or EWUpdate.
2. Determine the position of the file in the field.
3. Call EWRemoveAttached.
4. Process the results.

## Example Task

In MyKB knowledgebase, as user A, remove all files named B.txt currently attached in the field Additional Files of case #456 and return the total number of files currently attached.

The task is completed by performing the following steps:

1. Login to MyKB with "A" and "password" and English as the local language.
2. Read the record to get the list of the files.
3. Find the position of the file.
4. Invoke the remove file call.
5. Get the total number of files currently attached.
6. Logout

## Sample Code - Java

You can generate sample Web Services code for any table by selecting **Setup > Tables > (*Edit Table*) > API > Download Sample**.

```

public int remove() throws Exception {
    EWServiceAPI binding = new EWServiceAPIServiceLocator().getDemo();
    try {
        String sessionId = binding.EWLogin("MyKB", "A", "password", "en");
        WSCase wsCase = (WSCase) binding.EWRead(sessionId, "case", 456);
        String[] fileNames = wsCase.getAdditional_Files();
        int n = 0;
        if (fileNames!=null) {
            for (int i = 0; i < fileNames.length; i++) {
                if ("B.txt".equals(fileNames[i])) {
                    n = binding.EWRemoveAttached("case", 456,
                                                "additional_files", i);
                }
            }
        }
        return n;
    } finally {
        binding.EWLogout(sessionId);
    }
}

```

## Arguments

Name	Type	Description
sessionId	String	Session token
tableName	String	The name of the table where the record is.
id	long	The identifier of the record to attach files to.
fieldName	String	The name of the field to attach files to.
position	int	The position of the file to be removed.

## Response

The current number of files attached in the specified field.

# Faults

**EWSessionException** - client not logged in or session has expired; client should re-login.

**EWPermissionException** - user used to create the session lacks sufficient privileges to read the record.

**EWWrongDataException** - client has supplied wrong data, for instance ID cannot be found.

**EWOperationException** - the operation has been blocked by and Agiloft function, for example a table-level lock.

**EWIntegrityException** - specified table cannot be found or its primary key cannot be identified.

**EWUnexpectedException** - an unexpected exception has occurred; the admin user should report this for investigation.