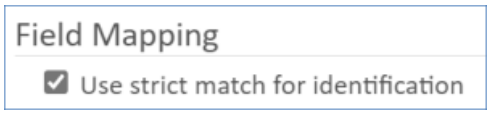
Matching Records with Synchronization

Record synchronization between an Agiloft table and an external system will match and replace fields based on the options that were specified in the Sync Configurations.

Identifying Fields

If the Use strict match for identification check box in the Field Mapping wizard is ticked, at least one of the fields marked as Identifying must have a value unique for each record in the table or structure concerned.



Field Mapping option to Use strict match

These fields determine the 1-1 mapping of the Agiloft table records to the external structure records. Good examples of identifying fields which are likely to create a logical match to sync records in both systems are Email and Full Name when synchronizing Contacts, and Summary when synchronizing Cases.

The Identifying field is not necessarily an ID field. In fact, if there is an auto-incrementing or GUID ID field as there is in Agiloft, it is generally a bad idea to have this field marked as identifying, since it's unlikely that this value would have an ID match in another system. Even worse, it might match directly to a completely unrelated record in the external system.

Record Peering

For every record in an Agiloft table or external system structure subject to a new sync, and for every such record created since the last sync, a corresponding record in the other system is determined by matching the identifying fields from the two systems. If there's no such record, one is created.

When the two matched records have been identified, sync stores them in an Agiloft ID - external system ID pair, peering the Agiloft record with the external record. Logically, sync treats the pair as a single data record, having Agiloft and external views.

If a new record is created manually in both systems, the identification process prevents sync from duplicating them.

Matching

There are two algorithms for matching:

- Best match: If there are three identifying fields, sync first tries to match all three. If there is no match, it tries matching by any two. If there is still no match, it matches by any one of the fields.
- Exact match: If there are three identifying fields, sync first tries to match all three. If there is no match, it creates a new record.

The algorithm is selected in the Field Mapping wizard by selecting the "Use strict match for identification" checkbox. The choice of exact matching may prevent matching if any identifying field has non-unique values.

After the peering is established, it isn't cleared until either one of these conditions is met:

- The records are deleted in both systems
- The peering information is reset by the Reset Records Peering button on the General tab of the Sync Configuration wizard

Conflict Resolution

If both records of a pair have been changed since the last sync to different values, one of them should be updated. The conflict can be resolved by either of two methods:

- Specify one system as the source of truth, whose value should always be preserved
- Preserve the most recent value, according to the timestamps

This choice is made in the Running tab of the Sync Configuration wizard.

Records are never merged; all fields in a record are always updated from a particular source record. However, because you can choose which fields to synchronize, you can control how much of the record is overwritten.

Dual Synchronization Problem

If two or more external systems are kept in sync with a given Agiloft table, which means using two or more different sync configurations for the same record set, and if conflicts are resolved by preserving the most recent value, the results are affected by the order in which the synchronizations are run.

Suppose an updatable field between Agiloft and two external systems A and B has the values and timestamps shown, and we synchronize A with Agiloft, and then B with Agiloft at 4 pm:

```
A p (3 pm) => p (4 pm)
B q (2 pm) => p (4 pm)
<ac:structured-macro ac:name="companyname" ac:schema-version="1" /> r (1 pm) => p
(4 pm)
```

The first update changes Agiloft because 3 pm > 1 pm, and the second changes the external system B because 4 pm > 2 pm.

But if we synchronize B first, and then A, the effect is:

```
XS A p (3 pm) => q (4 pm)
XS B q (2 pm) => q (4 pm)
<ac:structured-macro ac:name="companyname" ac:schema-version="1" /> r (1 pm) => q
(4 pm)
```

The first update changes Agiloft because 2 pm > 1 pm, and the second changes external system A because 4 pm > 3 pm.

This is because there is only one timestamp on each value in a pair, whether it results from sync or user updating.