

# What's in a Name?

## SUMMARY

Entities in Cruiser are identified internally by an unchangeable ID. The entity's name and group are provided for the user's convenience. Therefore, if an entity is renamed or moved it is still regarded as the same entity by Cruiser because it has the same ID.

When Cruiser loads data from a CDI file it tries to match entities in the file with entities in the database by using the ID. If the CDI file originated from another database, Cruiser can still find a match in the database by using the full name of the entity. Any unmatched entity is copied into the database and given a new ID.

FieldMan uses the simple name of User Variable Definitions (UVDs) to create columns in the data editing tables. This means that if two UVDs with the same simple names are used within an assessment, only the first UVD will be shown and measured.

It is recommended that all entities have unique and descriptive simple names, and if the content of an entity is changed that a copy is made and the necessary changes made to the new entity. The new entity should have a name that makes it clear that it is a new version of the entity, such as "Resin Bleeding 2005".

## HOW ENTITIES ARE REALLY DEFINED IN CRUISER

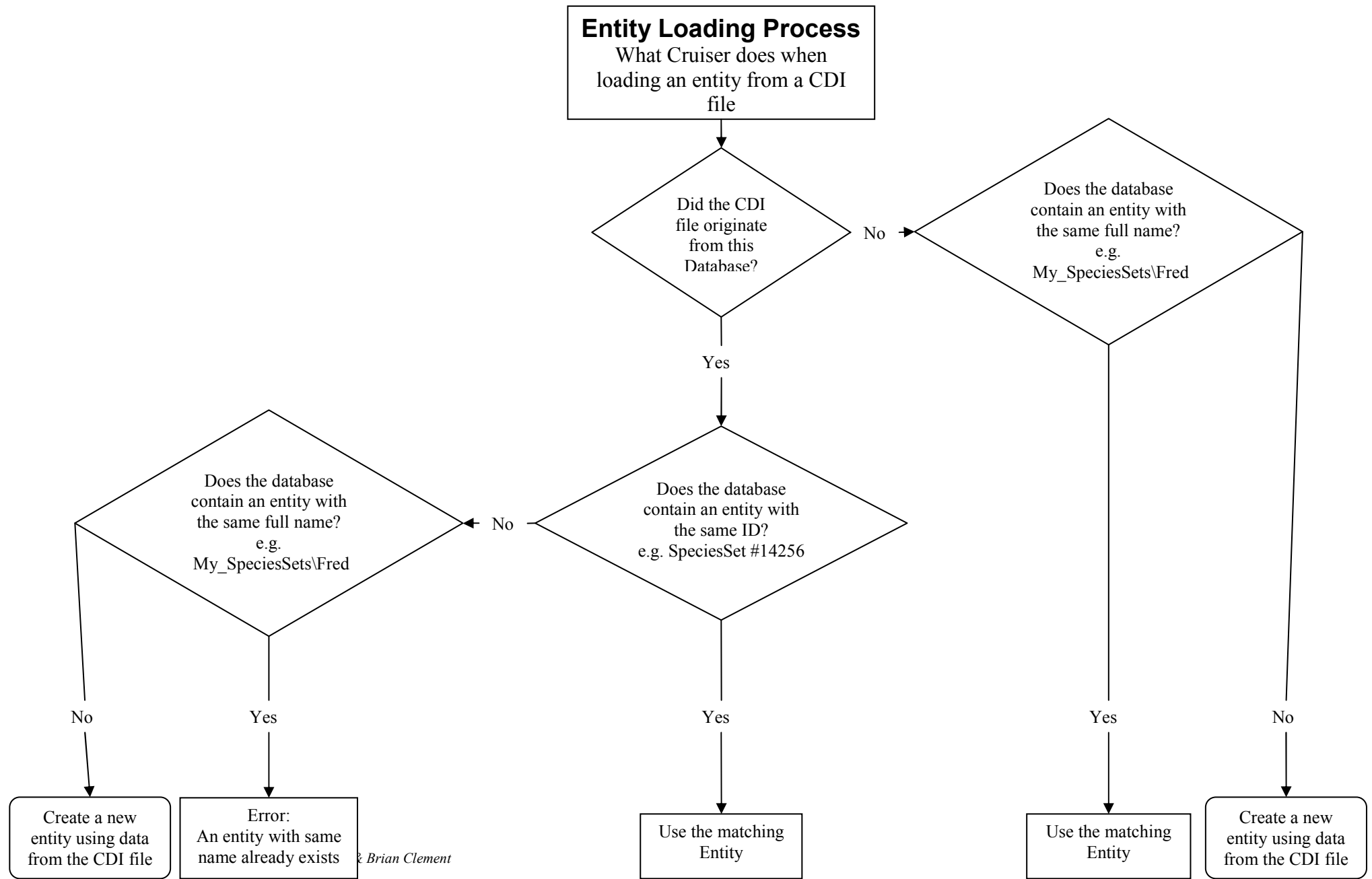
There are three attributes that are inherent to any entity in Cruiser, such as an Assessment, Species Set, Feature Dictionary, or User Variable Definition. They are:

1. **Name.** This is more correctly referred to as the *Simple Name* of the object, such as "Resin Bleeding".
2. **Group.** This is the folder or directory within the Cruiser display containing the entity. The combination of the group and the simple name is the *Full Name* of an entity, such as "\User Variable Definitions\MyUVs\Resin Bleeding". Entities can be moved between groups without affecting links with other entities.
3. **ID.** This is the index in the database of the entity itself. It is stamped on the entity when the entity is created, is unique throughout the database, and it can never be changed while the entity exists. This ID is not shown to the user, but it is the mechanism that Cruiser uses to find an entity in the database, so that links are not broken when an entity is renamed or moved to another group.

The name and group are for the benefit of the user to make sense of what is shown, but the ID is what Cruiser uses to identify the entity. To the computer the name and group are simply values used for display.

## WHAT CRUISER DOES WHEN LOADING ENTITIES FROM A CDI FILE

CDI files contain data representations of entities in Cruiser, and so have a full name that defines the entity's name and group, and an ID field that identifies the entity in the database. Also, the CDI file is stamped with another ID that identifies which database the CDI file was created from. The following diagram describes how Cruiser matches entities in the database with entities in a CDI file.





## ATLAS

When loading an entity from a CDI file, Cruiser first tries to find in its database the entity that was originally copied into the CDI file. If Cruiser can find the entity, it simply uses it instead of re-loading the entity from the file into the database again.

So, first of all, Cruiser checks if the database ID stamped on the file matches its own database, and if so it can use the entity IDs in the file to search for matching entities. This means that if the user has changed an entity's name or group in the database in the meantime, it doesn't matter – the entity is found in the database by its unchangeable ID.

However, if the CDI file's database ID stamp doesn't match the current database, then the file must have come from some other database, and so the entity IDs in the file don't relate to IDs in this database. In this case, the only way of possibly matching an entity in the file with one in the database is if they have the same name and group. Cruiser assumes that if the name and group match then the entity in the database must have been copied from the other database at some previous time, and is probably a copy of the same entity.

If Cruiser can't find a match, either through the ID method or the full name method, it creates the entity in the database, and gives it a brand new ID. However, there is a 'gotcha' here: if the entity can't be found by the ID method, but there *is* an entity in the database with the same full name as the one in the CDI file (e.g. It has been created in the meantime), Cruiser will fail complaining that when it tries to create a new entity (because it can't be found by ID) it can't because there is already an entity with that full name there (but with a different ID).

### **USER VARIABLES, SIMPLE NAMES AND FIELDMAN**

When creating a CDI file for use in FieldMan, there are a couple of things to keep in mind about how FieldMan shows the User Variables to the user when editing the data.

When FieldMan reads a CDI file from Cruiser it copies all of the context data (User Variable Definitions, Dictionaries, Species Sets) into its *context library* and names the User Variable Definitions (UVDs) after the simple name of the UVD. The simple name of the UVD is used instead of the full name because when the data is edited FieldMan adds columns to the table that the UVD is related to, for each UVD. For instance, a plot level UVD called “\User Variable Definitions\Pine UVs\Resin Bleeding” will cause an extra column to be added to the Plot table called simply “Resin Bleeding”, which is a lot easier to read than the full name when space on a handheld computer is tight.

The ramification of using simple names to identify UVDs in FieldMan is that if an assessment uses two UVDs at the same level with the same simple names (but in different groups) then FieldMan will only show the first one of them in the data, and so only that one will be assigned any data. When the completed CDI file is loaded back into Cruiser, it will complain that one of the UVDs has no measured data against it, and won't load the file. This is a very difficult bind to get out of without losing the captured data.

A better approach is to name all UVDs with sensible and unique simple names, even if they are in different groups. After all, is it really sensible to have two UVDs with the same simple name, but meaning something different enough to want to measure data for both of them within the same assessment?

## ALTERING ENTITIES THAT ARE ALREADY IN USE

At some time a user may want to just tweak an entity, such as adding another option to the list of values in a UVD, or removing a feature from a dictionary that is no longer used. Simply changing the existing entity can have some unforeseen consequences, and in some cases is disallowed by Cruiser.

One possible consequence is that it could change how existing data in assessments in the database is interpreted if it was ever re-analysed. If you delete an obsolete feature from a dictionary, or tighten the numeric constraints on a numeric UVD, then existing data may be using the feature or lie outside of the new range, and would thus be rendered invalid.

Another consequence relates to communications with FieldMan and the context library of such entities that FieldMan holds. FieldMan is quite strict about changing context library entries; in fact it simply does not allow it. If something is specified in the context library under a particular name, then the user is not allowed to use that name for a definition that has different values.

It is highly recommended that when an some change to an entity is required that the user creates a new copy of the entity and gives the new entity a new simple name, such as “Resin Bleeding 2005”, and then makes any required changes to the existing entity while there are still no other entities referring to it. When a CDI file that contains the new entity gets to FieldMan it is treated as a completely different entity (which it is) and allows jobs that are currently still in progress in FieldMan to continue using the older definition while the new job uses the new one. An added advantage of using entities with a new name is that it will be obvious to all users that there has been a change made, and which entity they should be using. Remember that the older entity can be moved into another group, such as “Old Dictionaries”, without affecting Cruiser’s ability to find it for use by older assessments. This will also make selecting the right entity even more obvious to the users.

## REFERENCES

Other white papers in the series:

- Controlling the data collection process with ATLAS FieldMan, Sep 2004.
- Separating Context from Data, Dec 2004.
- Division of Responsibility, Mar 2005.