# About Catalogs

Catalogs classify standard objects according to user defined keywords. You can create catalogs, browse through catalogs, and query catalogs to find the required items and instantiate those items in the design context. You can modify a catalog in the Design IP Classification app.

**See Also**

Creating a Catalog

A catalog is a tree structure consisting of the following:

- **Chapters:** A chapter references other chapters or items. It is used to classify elements (items).
- **Items:** An item is a reference to an external document or entity. Items are described with keyword values.
- **Keywords:** A keyword is an attribute describing a chapter or item. You can assign attribute value to each item. You can associate keywords to chapters or items and search by attribute value to filter its content.

You can access a catalog entity in many ways as given below:

- Any chapter can be referenced by one or more chapters.
- Any chapter can be considered to be an entry point to a catalog.

You can modify the name and the description of a catalog or a chapter in the properties dialog box. You can select the **Properties** command from the context menu of a catalog or a chapter.

**Important:** A catalog is a short-transaction application, that is, catalog objects are saved in the database at the end of each command or transaction.

# About Removing Part Libraries

Removing a part family from its parent makes it an orphan and could cause problems.

If a part family is removed from its parent classification (for example, removing a sub part family from a part family), logically any attribute groups that this class inherited from any ancestors must be lost. Practically, that would mean that any classified end items that are contained in that class and all subclasses would instantly lose all these attributes. Potentially valuable data would be lost.

To protect your data and to support reclassifying (moving from one parent to another), IP Classification provides choices so you can decide how to remove the part family, and shows the attribute groups that will be lost and how many classified items will be affected:

- **Carry over inherited attribute groups.** All attribute groups that this class inherits from its parents will be associated directly with the soon-to-be-orphaned class, immediately before actually orphaning it. These attribute groups are no longer inherited, but assigned directly to that class. Any non-inherited attribute groups are always included. No attributes or attribute values will be lost.

  If the newly-orphaned class is then moved into a new parent class, the attribute groups that this class carried with it will continue to be associated with this class. If any of these attribute groups would be inherited from the new parent, they are removed from the class, and marked as inherited: there will be no duplication.
- **Lose inherited attribute groups.** The inherited attributes will be removed from the orphaned class, and therefore also from any subclasses that are still attached to that class, and ultimately from any classified items anywhere in that subtree. Many objects may be affected. The values for these attributes on the affected objects are lost.