

instance

For the instance scenegraph location, in addition to the standard Katana instance attributes and arbitrary user data (`geometry.arbitrary.*`), KtoA recognizes the following extra attribute:

- `instance.inherit_xform`: if this int attribute is present and non-zero, the instance will not inherit and composite the source's transform with the instance's transform.
- `instance.ID`: instead of requiring `geometry.instanceSource` to point to a location of type instance source, any other scenegraph location that also has an `instance.ID` string that matches this one will automatically become instances of each other (and the first one encountered will be the source). Note that this attribute applies to scenegraph locations of any shape type as well, not just `instance:poly mesh, subdmesh, curves, pointcloud, render procedural, volume, implicit, etc.`

Also, for arbitrary user data, you may set arbitrary data that has any scope (`primitive - Arnold constant, point - Arnold varying, face - Arnold uniform, or vertex - Arnold indexed`). KtoA will pass that along to Arnold, which if it matches the instance source in topology (e.g. the number of varying values is the same as the number of vertices in the source mesh) the user data will apply to the instance at render time.

Finally, because `instance source` locations can have various children of varying topologies, if you wish to override or add arbitrary data to a particular `instance source child` location when it is instanced, you should create a child of the `instance` location (child of location type `locator` will do) and set the `geometry.arbitrary.*` data there. If the instance child matches an instance source child by name/path, it will then apply.

For convenience, instancing can be set up more easily using the `ArnoldInstanceSettings` node.