User Defined MEL Procedures

The following procedures can be defined to customize *3Delight for Maya*'s behavior. Since calling the `exists` MEL command repeatedly is costly, a cache of this command's results is kept. This cache is flushed before every render. It can also be flushed manually by calling:

```
delightExists -user -flush;
```

`global proc` **`DL_userStartup`** `()`

This procedure is called as the last step of the initialization procedure that is run when the *3Delight for Maya* plugin is loaded.

`global proc` **`DL_userGetExpandedString`** `(string $str, string $shape_name, string $pass_name)`

This procedure is called when a string attribute is parsed, offering the opportunity to completely replace the default string expansion in *3Delight for Maya*. `DL_userGetExpandedString` is passed the string to expand, a string that contains the shape name (or an empty string), and a string containing the render pass name (or an empty string). The last two parameters are usually used to replace the `<shape>` and `<pass>` tokens, respectively. The procedure should return a string that requires no further expansion, as there will be no further substitution attempts done on it.

`global proc` **`DL_userGetStringTokens`** `(string $tokens[], string $values[])`

This procedure is called when a string attribute is parsed by `DL_expandString()` to expand the tokens and environment variables the attribute may contain. `DL_userGetString` is passed two empty string arrays and it is expected to fill the first one with the tokens to look for and the second one with their replacement strings. If a user-defined token is identical to a built-in token, the user-defined one will be used.

`global proc string` **`DL_userConvertTexture`** `(string $src, string $dst, string $color_profile)`

This procedure is called when a texture file, such as one referenced by the *`file`* 2D texture node, needs to be converted to a `tdl` texture file. The procedure is passed the full path to the source file and the full path to the desired tdl file. The `$color_profile` parameter will contain the texture color space name, in the same format as what is accepted by *`tdlmake`*'s *`-colorspace`* option. This parameter can also be set to an empty string, meaning that no color profile was specified for the texture. The returned string is expected to contained an error message, or an empty string if the conversion completed successfully. The default behavior is to call `DL_defaultConvertTexture`.

`global proc string[]` **`DL_userGetNodesForRibArchiveBBox`** `(string $nodes[], string $pass_name)`

This procedure is called when computing the bounding box of a RIB archive being written. It will receive the list of nodes being considered for bounding box evaluation, and the name of the render pass being rendered. The procedure returns the objects that must be used for bounding box evaluation, so it can arbitrarily add or exclude specific nodes from that list.

Here is an example procedure that excludes objects from the bounding box evaluation according to the value of the *ignoreBBoxContribution* attribute in an assigned geometry attributes node:

```
global proc string[]
DL_userGetNodesForRibArchiveBBox(string $nodes[], string $pass_name)
{
  string $nodes_for_rib_archive_bbox[];
  clear($nodes_for_rib_archive_bbox);

  // Retreive the collection used for rendering
  //
  string $all_related_passes[] = DRP_getAllRelatedPasses($pass_name);
  string $shader_collection =
    DRP_getHighestAttrNode("shaderCollection", $all_related_passes);

  // Construct the list of objects to consider for bounding box evaluation.
  // Exclude from the list the objects that have a "ignoreBBoxContribution"
  // attribute set to "true" in an assigned geo attribs node.
  //
  string $all_attribs_nodes[];
  string $attribs_node;

  for($curr_node in $nodes)
  {
    // Get all geo attribs nodes that affect this object
    //
    $all_attribs_nodes =
      DL_getAllGeoAttribsNodesFromShape($curr_node, $shader_collection);

    // Find the geo attribs node that contains the "ignoreBBoxContribution"
    // attribute that we must consider (needed because of the various ways the
    // attributes of a geo attribs node can be inherited).
    //
    // Note: this "ignoreBBoxContribution" would be a user-defined attribute
    // and not something predefined by 3Delight for Maya.
    //
    $attribs_node =
      DL_highestNodeWithAttrib("ignoreBBoxContribution", $all_attribs_nodes);

    // Exclude this object from the list if relevant
    //
    if($attribs_node != "" &&
      getAttr($attribs_node + ".ignoreBBoxContribution"))
    {
      continue;
    }

    // Object was not excluded, so add if to the list
    //
    $nodes_for_rib_archive_bbox[size($nodes_for_rib_archive_bbox)] = $curr_node;
  }

  return $nodes_for_rib_archive_bbox;
}
```