

Following is a brief overview and description of the LISP routines available for accessing Architectural Desktop Release 2004 Layer functionality.

For these to be operational, AecLMgrLISP40.arx must be loaded. Note that this ARX is never demand loaded out of the box.

The first listing is an list of the LISP function names, followed by a description of each function in turn. This will consist of function name, a list of parameters, and the return type. An example, a short description and a discussion of any known problems will immediately follow.

a) Functions available

```
AecLayerResetLayersInfo
AecLayerSaveStdAsText
AecLayerSaveStdAsLDF
AecLayerLoadStdAsText
AecLayerLoadStdAsLDF
AecLayerLoadAllLDFFiles
AecLayerAddToSnapshot
AecLayerGetSnapshot
AecLayerRestoreSnapshot
AecLayerImportSnapshot
AecLayerExportSnapshot
AecLKeyOverrides
AecToggleLKeyOverrides
AecImportLayerKeyingStyle
AecSetCurrentLayerKeying
AecSetLayerKeyOverride
AecExpandLayerKey
AecGenerateLayerKey
AecLayerKeyList
AecSetLegacyLayerKey
```

Function: AecLayerResetLayersInfo

Parameters: <none>

Returns: void

Example: (AecLayerResetLayersInfo)

This function removes user defined Layer Standard assignments on all the layers in the drawing, and reassigns according to ADT's calculations.

Function: AecLayerSaveStdAsText

Parameters: LayerStd: String, FileName: String

Returns: void (nil if layer standard not found)

Example: (AecLayerSaveStdAsText "My Standard" "C:\\test.txt")

This function saves the named Layer Standard as a structured text file.

Function: `AecLayerSaveStdAsLDF`
Parameters: `LayerStd: String, FileName: String`
Returns: `void (nil if standard not found)`

Example: `(AecLayerSaveStdAsLDF "My Standard" "C:\\test.ldf")`

This function saves the named Layer Standard as a proprietary Layer Definition File. This format is more robust than the text format and is recommended when simply sharing layer standard definitions.

Function: `AecLayerLoadStdAsText`
Parameters: `FileName: String`
Returns: `void`

Example: `(AecLayerLoadStdAsText "c:\\test.txt")`

This function loads layer standards from the supplied file.

Function: `AecLayerLoadStdAsLDF`
Parameters: `FileName : String`
Returns: `void`

Example: `(AecLayerLoadStdAsLDF "c:\\test.ldf")`

This function loads layer standards from the supplied file

Function: `AecLayerLoadAllLDFFiles`
Parameters: `<none>`
Returns: `void`

Example: `(AecLayerLoadAllLDFFiles)`

This function loads the layer standards from all LDF files in the current and AutoCAD search path

Function: `AecLayerAddToSnapshot`
Parameters: `Snapshot: String, (LayerName: String, LayerState: Short, ACIColor: Short, HasUserDesc: Short, UserDesc: String, LineType: String, LTypeFile: String)...`
Returns: `void`

Example: `(AecLayerAddToSnapshot "SnapshotName" ("Layer1" 1 152 0 "" "Continuous" "")) ("Layer2" 3 140 1 "UserDesc" "Continuous" "")`

This function allows a series of layers (not necessarily existing in the current drawing) to a snapshot. The Layer State is the sum of 1= Visible, 2=Lock, 4=Frozen, 8=Frozen in new viewports

Function: `AecLayerGetSnapshot`
Parameters: `Snapshot: String`
Returns: Resbuf of all layers in snapshot

This functions returns a resbuf list of all layers in the specified snapshot, or void if the snapshot does not exist

Function: `AecLayerRestoreSnapshot`
Parameters: `SnapshotName: String`
Returns: void

Example: `(AecLayerRestoreSnapshot "MySnapshot")`

The method restores the properties of all layers in a snapshot.

Function: `AecLayerImportSnapshot`
Parameters: `Filename: String`
Returns: void (nil if error)

Example: `(AecLayerImportSnapshot "mysnaps.ssl")`

This method imports snapshot(s) from the following file types

- Bonus tools files (*.lay)
- Comma delimited files (*.cdf, *.csv)
- Proprietary snapshot files (*.ssl)

Function: `AecLayerExportSnapshot`
Parameters: `SnapshotName: String, Filename : String`
Returns: void (nil if error)

Example: `(AecLayerExportSnapshot "MySnapshot" "C:\\mysnaps.ssl")`

This function exports a specific snapshot to an external file in either of the following file types (determined by the extension of the filename supplied)

- Bonus tools files (*.lay)
- Comma delimited files (*.cdf, *.csv)
- Proprietary snapshot files (*.ssl)

Function `AecLKeyOverrides`
Parameters: <none>
Returns: 0 if overrides are not enabled or available, 1 otherwise

Example: `(setq useril (AecLKeyOverrides))`

Function: `AecToggleLKeyOverrides`
Parameters: <none>
Returns: 1 if overrides are subsequently enabled, 0 otherwise

Example: `(setq useri2 (AecToggleLKeyOverrides))`

`AecToggleLKeyOverrides` is used to invert the status of the Overrides Enabled flag. A short message is displayed on the command line to indicate the new status of the global override enabled flag. Can be used in conjunction with the `AecLKeyOverrides` function.

Function: `AecImportLayerKeyingStyle`
Parameters: `FileName` : String
`LayerKeyStyle` : String
Returns: T if success, Nil otherwise

Example: `(AecImportLayerKeyingStyle "c:\\MyStds.dwg" "MyLayering")`

`AecImportLayerKeyingStyle` is used to force import of a layer key style from an external drawing into the current one. Note the requirement for two backslash as path directory separators.

As seen in the example it requires a full path name or for the file to be in the AutoCAD search path, followed by the textual name of the key style.

Function: `AecSetCurrentLayerKeying`
Parameters: `KeyStyle` : String
Returns: T if success, Nil otherwise

Example: `(AecSetCurrentLayerKeying "AIA (256 Color)")`

`AecSetCurrentLayerKeying` switch the current key style to the one supplied, but if and only if that style exists in the drawing. Can be used in conjunction with `AecImportLayerKeyingStyle` to import and the activate a key style.

Function: `AecSetLayerKeyOverride`
Parameters: `OverrideName` : String
`OverrideValue` : String
Returns: T if successful, Nil otherwise

Example: `(AecSetLayerKeyOverride "Status" "Demo")`

`AecSetLayerKeyOverride` is used to apply a new override value. The example above applies the demolition override on AIA (256 Color)'s status field.

Known problems: Calling this with `OverrideValue` set to "" will *not* clear the override value, and this will remain in the final release version. A workaround is to supply a value which is invalid for the format of the

field. In the example above, you could supply the string "<none>". This is not valid for the status field, and so this override will be subsequently ignored. Alternatively, you could always supply a string that is invalid for all layers, such as "*"

Function: AecExpandLayerKey
Parameter: LayerKey: String
Returns: LayerName: String

Example: (setq USERS1 (AecExpandLayerKey "WALL"))

AecExpandLayerKey converts a supplied key name to the layer name that would be generated (including current override settings) using the current key style. Note that this function does not actually generate the layer, it is for information purposes only.

Function: AecGenerateLayerKey
Parameters: LayerKey : String
Returns: LayerName: String

Example: (setq USERS1 (AecGenerateLayerKey "WALL"))

AecGenerateLayerKey is identical to AecExpandLayerKey *except* that it also generates the layer if not already in existence in the drawing.

Function: AecLayerKeyList
Parameters: <none>
Returns: DefinedKeys : Resbuf

AecLayerKeyList returns a resbuf list of strings, defining all of the currently defined keys for the active layer key style.

Function: AecSetLegacyLayerKey
Parameters: LegacyKey : Short
Returns: T if successful, Nil otherwise

Example: (AecSetLegacyLayerKey 1)

AecSetLegacyLayerKey can be used to flag the key style as being equivalent to a Release 1 keying system, so that when a drawing is saved as R14 layer keying will still work in Architectural Desktop Release 1. The numerical value is identical to one of the values available in the Release 1 drawing setup routine, i.e.

1 = AD 98 - 256 Color
2 = AD 98 - 16 Color
3 = AIA
4 = BS1192

5 = SDESK

6 = ASG

7 = None (i.e. current layer)

For this to work, the drawing in which the key style is stored obviously needs to be saved.

--

Mark Webb, Building Solutions Division
Autodesk., Inc

<http://www.autodesk.com/archdesktop>