ANSYS ACT 19.1 Known Issues and Limitations

Table of Contents

ANSYS ACT 19.1 Known Issues and Limitations

This document lists known issues and limitations in the ACT 19.1 release.

ACT Debugger is currently supported only on Windows platform

The **ACT Debugger** is currently supported only on the Windows platform from the **Project** page in ANSYS Workbench and from DesignModeler and Mechanical.

Graphics API issues in ANSYS DesignModeler and Mechanical when no extensions are loaded

There are some limitations on the Graphics API from the **ACT Console** in ANSYS DesignModeler and Mechanical when no extensions are loaded. For instance, the Factory2D does not work. Therefore, you should load one or more extensions before using the Graphics API from the **ACT Console**.

Limited ACT localization support

Localization of ACT is limited to the languages currently supported in ANSYS Workbench. This limitation does not apply to the ability to manage various languages within the extension. For example, the property names created by an extension do not have to be in the same language as the current activated language in Workbench.

There is no mechanism to integrate localization for the property names defined by an extension. To manage different languages for your property names, you must develop localization yourself. Both regional settings based on the "." or the "," decimal symbol are available. However, the implementation of the extension should use the "." symbol for any value defined at the XML or IronPython level.

ACT is unable to create a chart from ANSYS Mechanical

When using ACT to create a figure from the chart API, the following error prevents the graphics display in the Mechanical window:

```
Object reference not set to an instance of an object.
```

As a workaround, add the following code to your script to create an empty window in which the chart can display:

```
import clr
clr.AddReference("Ans.UI.Toolkit")
clr.AddReference("Ans.UI.Toolkit.Base")
import Ansys.UI.Toolkit
if Ansys.UI.Toolkit.Window.MainWindow == None:
   Ansys.UI.Toolkit.Window.MainWindow = Ansys.UI.Toolkit.Window()
```

Limitations on ACT postprocessing of ANSYS Mechanical results

If you use ACT for postprocessing Mechanical results, you should be aware of the following limitations:

Scoping for custom results

Custom results do not support using a geometric path as scoping. You can only use a selection of nodes and elements as scoping.

Custom results in projects with models containing remote points

ACT custom results do not work in projects with Mechanical models that contain remote points (such as remote mass, displacement, remote force, spring, and so on).

Compressed result file not supported

The ACT postprocessing API does not support the compressed result file for Mechanical, which is created by the Mechanical APDL command `/FCOMP`.

A node merge action on the mesh is unsupported

The ACT postprocessing API does not support a node merge action on the mesh.

Results availability for certain models

Full results availability through the ACT postprocessing API is not guaranteed for Mechanical models containing:

- Bolt pretension loads
- Beam connections
- Link elements
- Pipe elements
- Elbow elements

Coordinate system for custom results returned for beam element

Custom results returned for the beam element are based on the global coordinate system and not the local coordinate system.

Results not available with ACT postprocessing API

The ACT postprocessing API can access only results stored in the result (RST) file. The non-RST file results are listed in the table below and defined with their result code or component labels.

Result Code	Description
SERR	Elemental structural error energy
TERR	Elemental thermal error energy
BERR	Elemental magnetic error energy

Component Label	Description
SUM	Vector sum
MSUM	Mass sum
SFSUM	Surface force sum
1, 2, 3	Principal
INT	Intensity
EQV	Equivalent
MAXSHEAR	Maximum shear
VECTORS	Vectors
WRAPPING_FACTOR	Wrapping factor

Archived projects containing solved ACT custom results

When you restore an archived project containing previously solved ACT custom results, you might encounter a solve failure after modifying the model data and updating the analysis in ANSYS Mechanical. As a workaround:

1. Right-click the **Solution** object in the Mechanical tree and select **Clear Generated Data** from the context menu.
2. From the **Solution** object, access the solver files and delete all files in the directory.
3. Re-solve the model.

Default location of ACT extensions for SpaceClaim

The default location in which SpaceClaim is to look for ACT extensions is `%ANSYSversion_DIR%\scdm\Addins`. However, SpaceClaim is not currently recognizing this location. Workarounds include either installing extensions in `%APPDATA%\Ansys\v191\ACT\extensions` or using the gear icon on the graphic-based **Extension**

Manager accessed from the **ACT Start Page** to add the folder for the default location or the folder to which you installed the extension.

On Linux, starting the ACT Start Page can cause an unexpected shutdown

When you start the **ACT Start Page** on Linux, an unexpected shutdown can occur. On some Linux operating system variants such as Red Hat, removing the package `totem-mozplugin` resolves the issue:

```
yum remove totem-mozplugin
```

On Linux, some property controls are not supported in ANSYS Mechanical

The following property controls are not supported on Linux in Mechanical:

- `FileOpen`
- `FolderOpen`
- `PropertyTable`

These controls are implemented using the ANSYS UI Toolkit, which is currently not supported on Linux when executed within Mechanical.

On Linux, using the ACT API to access `ElementNode` results always returns 0

On Linux, when the ACT API is used to access `ElementNode` results in Mechanical, **0** is always returned.

Some Linux versions contain a JPEG library that conflicts with the ANSYS package

In some Linux versions, a JPEG library conflicts with the ANSYS package. When using ACT wizards with JPEG images, this conflict can produce an SIGSEV error and an application crash. If this occurs, you can convert your images to other file formats such as GIF or PNG.

Important:

In 19.0, the namespace of the automation API object has changed. Therefore, you should not use the method `GetType()` as a way to verify the nature of the object that you have. Instead, you must use the property `DataModelObjectCategory`, which is available on all objects. It returns an `enum` value indicating the type of the object. For example, the following command returns true:

```
ExtAPI.DataModel.Project.DataModelObjectCategory ==
DataModelObjectCategory.Project
```