

GoldSim 15 : [Advanced Modeling Concepts \(../../Content/GS/advancedmodelingconcepts.htm?\)](#)

[TocPath=Advanced%2520Modeling%2520Concepts%257C 0](#) : [Using External Application Elements](#)

[\(../../Content/GS/usingexternalapplicationelements.htm?\)](#)

[TocPath=Advanced%2520Modeling%2520Concepts%257CUsing%2520External%2520Application%2520Elements%257C 0](#)

:

External (DLL) Elements

In some situations, you may wish to define a complex function which can not be readily implemented using the expression editing features supplied by GoldSim. For example, calculation of an output may require very complex logic which would be cumbersome to represent using a [Selector](#) element, or it may require a numerical solution technique (e.g., iteration).



GoldSim provides two separate ways to deal with such situations:

- You can develop separate program modules (written in C, C++, FORTRAN or other compatible programming languages) which can then be directly coupled with the main GoldSim algorithms. These user-defined modules are referred to here as [external functions](#), and the elements through which they are coupled to GoldSim are called [External \(DLL\) elements](#).
- You can specify a sequence of statements directly within the GoldSim interface using a Script element.

In general, using Script elements is preferred, as this approach has the key advantages that 1) it does not require use of a separate programming language and interface; and 2) it is much more transparent (all of the "code" can be seen directly in GoldSim). However, in some cases (e.g., linking complex existing programs directly into GoldSim), Script elements cannot be used and External (DLL) elements are the only option.

The modules are linked into GoldSim as DLLs (Dynamic Link Libraries) at run time. Integrating these external modules into GoldSim requires that you develop a "wrapper" (or "shell") around your existing function, and compile it into a DLL. In most cases, this will require only a limited number of programming modifications. GoldSim supports both 32-bit and 64-bit DLLs.

An example model which uses an External (DLL) element (External.gsm) can be found in the General Examples/External folder in your GoldSim directory (accessed by selecting **File | Open Example...** from the main menu). This folder also includes the corresponding DLL, as well as the source code (in C++ and Fortran) for the example.

Learn more

[**Implementing the External Module**](#)
[**\(implementingtheexternalmodule.htm\)**](#)

[**Defining the External Element**](#)
[**\(definingtheexternalelement.htm\)**](#)

[**Locking onto a DLL File**](#) ([**lockingontoadllfile.htm**](#))

[**Running the DLL in a Separate Process**](#)
[**\(runningthedllinaseparateprocess.htm\)**](#)

[**Using an External Element to Define Lookup Tables**](#)
[**\(usinganexternalelementtodefinelookuptables.htm\)**](#)

[**Using an External Element to Read and/or Output**](#)

[**Time Series**](#)

[**\(usinganexternalelementtoreadandoroutputtimeseries.htm\)**](#)

[**When is the External Element Called?**](#)
[**\(whenistheexternalelementcalled.htm\)**](#)

[**Exporting the External Element Interface**](#)
[**\(exportingtheexternalelementinterface.htm\)**](#)

[**Saving External Element Outputs**](#)
[**\(savingexternalelementoutputs.htm\)**](#)

Related topics...

