

Allegro CL Runtime

This document contains the following sections:

[1.0 Allegro Runtime introduction](#)

[1.1 Standard Runtime](#)

[1.2 Dynamic Runtime](#)

[1.3 Partner's Runtime](#)

[2.0 Creating runtime applications](#)

[3.0 Conditions on distributing Allegro Runtime created applications](#)

1.0 Allegro Runtime introduction

Runtime is a way to license the distribution of applications written in Allegro CL. It is designed to offer VARs using Allegro CL for product development a way to deliver or deploy their products without requiring their users to have fully-paid licenses for Allegro CL. VARs (Value Added Resellers) are users of Allegro CL who develop applications intended for delivery to their customers.

Allegro CL will build, using (on Unix and Windows) the **generate-application** function (called with argument *image-only* nil) or (on Windows only) the **File | Build Project Distribution** menu command or the **build-project** function (called with *exe-only* specified nil), an application directory containing everything necessary to distribute or deploy to customers.

A *runtime application* is defined to be such a directory. One of the files in that directory is a license file (extension *.lic*). That license file will permit the application to start up. If that license file is removed, the application will not work.

As with all Allegro CL features and add-ons, the Allegro CL installed on your machine includes everything needed to create runtime applications, but in order to do so, you must have a proper license in your license file (*devel.lic* in the Allegro Directory). If you are not licensed to create runtime applications, attempts to do so will fail.

There are three aspects of runtime applications which will be discussed in this document:

- What Runtime is, discussed just below and in the subsections immediately following.
- How a runtime application is created, which is touched on above and considered in more detail in

[Section 2.0 Creating runtime applications.](#)

- What you can do with a runtime application once you have created one, discussed in [Section 3.0 Conditions on distributing Allegro Runtime created applications.](#)

There are three flavors of *Runtime*:

- Allegro CL Standard Runtime (described in [Section 1.1 Standard Runtime](#) below)
 - Allegro CL Dynamic Runtime (described in [Section 1.2 Dynamic Runtime](#) below)
 - Allegro CL Partner's Runtime (described in [Section 1.3 Partner's Runtime](#) below)
-

1.1 Standard Runtime

Allegro CL Standard Runtime applications differ from the Allegro CL Development image in that they necessarily lack certain features and capabilities (other capabilities may be optionally left out):

- No cross referencer (see *cross-reference.htm*).
- No runtime analyzer (see *runtime-analyzer.htm*).
- No stepper (see *The stepper*).
- No disassembler (**disassemble** does not work).
- No compiler (so **compile** and **compile-file** are unavailable; **fasl-read** and **fasl-write** also do not work as they require the compiler).
- No **:help**, **:history**, and *redo* (see *Command and expression history* in *top-level.htm*) top-level commands.

Allegro Enterprise and Enterprise Platinum licenses include a license to make standard runtime applications. The Allegro Professional license does not include a license to make any runtime application. *Contact your Account Manager* for information on purchasing runtime licenses and upgrading your version of Allegro CL.

Please read [Section 3.0 Conditions on distributing Allegro Runtime created applications](#) for information on who you may distribute runtime applications to and how you may deploy them.

1.2 Dynamic Runtime

Allegro CL Dynamic Runtime applications differ from the Allegro CL Development image in that they necessarily lack certain features and capabilities (other capabilities may be optionally left out):

- No cross referencer (see *cross-reference.htm*).
- No runtime analyzer (see *runtime-analyzer.htm*).
- No stepper (see *The stepper*).
- No disassembler (**disassemble** does not work).

Dynamic runtime applications, however, can include the compiler, so **compile** and **compile-file** can be available. (We say *can be* because you can choose to leave the compiler out.) They also allow use of the **:help**, **:history**, and *redo* (see *Command and expression history* in *top-level.htm*) top-level commands. (These additional features differentiate dynamic runtime from [standard runtime](#).)

No Allegro CL license includes (by default) a license to make dynamic runtime applications. A license to create such applications must be purchased (*contact your Account Manager* for information).

Please read [Section 3.0 Conditions on distributing Allegro Runtime created applications](#) for information on who you may distribute runtime applications to and how you may deploy them.

1.3 Partner's Runtime

Allegro CL Partner's Runtime is a superset of [Dynamic Runtime](#), designed for VARs who want to give their customers the ability to program in Lisp. That is, users of a Partner's application will be able to use Allegro CL including the development environment.

Basically, your runtime customers have complete Allegro CL development images, just like you do. Note, however, that support must go through you (your customers should not contact Franz Inc. directly for support).

No Allegro CL license includes (by default) a license to make partner's runtime applications. A license to create such applications must be purchased (*contact your Account Manager* for information).

Please read [Section 3.0 Conditions on distributing Allegro Runtime created applications](#) for information on who you may distribute runtime applications to and how you may deploy them.

2.0 Creating runtime applications

Creating runtime applications for distribution or deployment under your license agreement is done with

(on Unix and Windows) **generate-application** (called with *image-only* nil) or (on Windows only) the **File | Build Project Distribution** menu command or the **build-project** function (called with *exe-only* specified nil).

To use **generate-application** with *image-only* nil, you must supply a value for the *runtime* keyword argument, and this value must be one of the keywords `:standard`, `:dynamic`, or `:partners`. If a different value (including nil) is specified, the call to **generate-application** will fail.

The **File | Build Project Distribution** builds an application from the current project. **build-project** builds an application from any loaded project. On the **Project Manager dialog Build tab**, a control specifies the type of runtime. You should specify the type desired (the initial value is `:standard`).

You must be licensed to build a runtime application of the specified type. If you are not licensed (or if your *devel.lic* license file does not include the correct license), *contact your Account Manager* for assistance (you will have to get a new license file).

If you are licensed to create the specified type of runtime application (and assuming there are no other problems, a directory containing all files necessary to distribute or deploy your application will be created. This directory comprises the runtime application. You must not (it violates your license to do so) copy files from elsewhere in the Allegro CL directory into the application directory. Note that it is possible to tell the system to use an existing directory. The purpose of doing so is to avoid having to delete directories as attempts to build the application fail. Do not use a directory that contains files from elsewhere in the Allegro CL directory.

Creation of applications using **generate-application** is described in *delivery.htm*. Creation using **File | Build Project Distribution** or **build-project** is described in the Common Graphics/IDE documentation, which is described in *About Common Graphics and IDE documentation*. Look at the *The IDE User Guide*, particularly *Chapter 4*, which discusses projects.

The following table indicates what values certain keyword arguments to **generate-application** should have.

Arguments:	Specify this value:	Notes:

<pre>:include-devel-env</pre>	<pre>nil in :standard and :dynamic. Any value in :partners.</pre>	<p>When <code>:include-devel-env</code> is true, the file <code>develenv.cl</code> is loaded into the image being built. That file loads many modules using cl:require, including some that are not permitted in a standard or dynamic runtime image. For those modes, you can copy that file and edit to remove the require forms for things not allowed in runtime, specifying the edited file as one to be loaded (with the <code>input-files</code> argument to generate-application). Alternatively, you can specify the <code>:runtime-bundle</code> argument true. (See <code>delivery.htm</code> for a full discussion of these arguments.)</p>
<pre>:include-compiler</pre> <pre>:discard-compiler</pre>	<p>In <code>:standard</code> runtime, <i>both</i> must be <code>nil</code> or <i>both</i> must be <code>t</code>.</p> <p>In <code>:dynamic</code> and <code>:partners</code> runtime (where the compiler is allowed) all (consistent) values are allowed.</p>	<p>If both have value <code>t</code>, the compiler is available while the image is being built and discarded at the end of the build process.</p>
<pre>:load-xref-info</pre> <pre>:record-xref-info</pre> <pre>:discard-xref-info</pre>	<pre>nil in :standard and :dynamic. Any value in :partners.</pre>	<p>The cross reference facility cannot be part of standard or dynamic runtime. Specifying <code>:discard-xref-info</code> to be <code>nil</code> for those modes seems counter-intuitive but since no xref info is loaded, none needs to be discarded. Specifying <code>:discard-xref-info</code> true causes the xref module to be loaded, and that is not allowed in a standard or dynamic runtime image.</p>

A standard Allegro Runtime image will have `:runtime-standard` on the `*features*` list. A dynamic Allegro Runtime image will have `:runtime-dynamic`. A partner's Allegro Runtime image will have `:runtime-partners`.

3.0 Conditions on distributing Allegro Runtime created applications

The license agreement governing the use of Allegro CL restricts your ability to distribute files associated with Allegro CL and to distribute or deploy applications created with Allegro CL. The license agreement is included with the Allegro CL distribution. If you have any question about what you are licensed to distribute or deploy, please *contact your Account Manager* for assistance. The license agreement is the governing document, and if there is any conflict between what is said in the Allegro CL documentation and the license agreement, you are bound by the terms in the license agreement.

But here are some general principles about distributing files and distributing or deploying applications (but again, please refer to your license agreement for specific details):

- You may distribute source files you wrote yourself and compiled versions of such files without restriction. You may include in your source files any code in the *examples/* directory of the Allegro CL directory (but you must copy that code into your own files) and any sample code found in the Allegro CL documentation.
- You may not distribute Allegro-CL-supplied files other than those placed in the runtime application directory by **generate-application**, the **File | Build Project Distribution** menu command, or **build-project**.
- Runtime applications (created with **generate-application**, or the **File | Build Project Distribution** menu command, or **build-project**) may be distributed or deployed only under the terms of the Allegro CL License Agreement (entitled 'FRANZ SOFTWARE LICENSE AGREEMENT') and any additions or supplements to it. **Note that being licensed to create a runtime application does not mean you are licensed to sell or deploy or even distribute or deploy for free that application without restriction.** Your rights to distribute and deploy applications and the conditions you must impose on recipients are controlled by the Allegro CL License Agreement.

If you are licensed to distribute or deploy a runtime application, you may only distribute the contents of the directory created by **generate-application**, or the **File | Build Project Distribution** menu command, or **build-project** (called with *exe-only* specified `nil`). No other files (except those you wrote yourself and compiled versions of such files) can be included.

Distributing applications to other licensed Allegro CL users

Note that because your source files and compiled versions of those files can be distributed without restriction, the way to distribute an application to another licensed Allegro CL customer without worrying about license agreement restrictions is to distribute your source files (and/or compiled versions of your source files), along with a file which creates the application (perhaps with a call to **generate-application**, or the **File | Build Project Distribution** menu command, or **build-project**, or perhaps in some other fashion).

The other licensed Allegro CL user can then use your files to create the application. This ensures that the recipient is in fact licensed to use the application (that the necessary add-on products are available, for example). Just because you believe another user is properly licensed, you may not distribute a runtime application (the contents of the directory created by **generate-application**, or the **File | Build Project Distribution** menu command, or **build-project**) except under the terms of your (not the recipient's) license agreement.

Again, if you have any questions about licensing, *contact your Account Manager* for assistance.