NAME

pkgdata – package data for use by ICU

SYNOPSIS

pkgdata [**–h**, **–?**, **––help**] [**–v**, **––verbose**] [**–c**, **––copyright** | **–C**, **––comment** *comment*] [**–m**, **––mode** *mode*] **–p**, **––name** *name* **–O**, **––bldopt** *options* [**–e**, **––entrypoint** *name*] [**–r**, **––revision** *version*] [**–F**, **––rebuild**] [**–I**, **––install**] [**–s**, **––sourcedir** *source*] [**–d**, **––destdir** *destination*] [**–T**, **––tempdir** *directory*] [*file* ...]

DESCRIPTION

pkgdata takes a set of data files and packages them for use by ICU or applications that use ICU. The typical reason to package files using **pkgdata** is to make their distribution easier and their loading by ICU faster and less consuming of limited system resources such as file descriptors. Packaged data also allow applications to be distributed with fewer resource files, or even with none at all if they link against the packaged data directly.

pkgdata supports a few different methods of packaging data that serve different purposes.

The default packaging *mode* is **common**, or **archive**. In this mode, the different data files are bundled together as an architecture-dependent file that can later be memory mapped for use by ICU. Data packaged using this mode will be looked up under the ICU data directory. Such packaging is easy to use for applications resource bundles, for example, as long as the application can install the packaged file in the ICU data directory.

Another packaging mode is the **dll**, or **library**, mode, where the data files are compiled into a shared library. ICU used to be able to dynamically load these shared libraries, but as of ICU 2.0, such support has been removed. This mode is still useful for two main purposes: to build ICU itself, as the ICU data is packaged as a shared library by default; and to build resource bundles that are linked to the application that uses them. Such resource bundles can then be placed anywhere where the system's dynamic linker will be looking for shared libraries, instead of being forced to live inside the ICU data directory.

The **static** packaging mode is similar to the shared library one except that it produces a static library.

Finally, **pkgdata** supports a **files** mode which simply copies the data files instead of packaging them as a single file or library. This mode is mainly intended to provide support for building ICU before it is packaged as separate small packages for distribution with operating systems such as Debian GNU/Linux for example. Please refer to the packaging documentation in the ICU source distribution for further information on the use of this mode.

pkgdata builds, packages, installs, or cleans the appropriate data based on the options given without the need to call GNU **make** anymore.

OPTIONS

–h, **–?**, **––help**
: Print help about usage and exit.

–v, **––verbose**
: Display extra informative messages during execution.

–c, **––copyright**
: Include a copyright notice in the binary data.

–C, **––comment** *comment*
: Includes the specified *comment* in the resulting data instead of the ICU copyright notice.

–m, **––mode** *mode*
: Set the packaging *mode* to be used by **pkgdata**. The different modes and their meaning are explained in the **DESCRIPTION** section above. The valid mode names are **common** (or **archive**), **dll** (or **library**), and **files**.

–O, **––bldopt** *options*
: Specify options for the builder. The builder is used internally by **pkgdata** to generate the correct packaged file. Such options include, but are not limited to, setting variables used by **make**(1)

during the build of the packaged file. Note: If **icu-config** is available, then this option is not needed.

−p, **−−name** *name*
Set the packaged file name to *name*. This name is also used as the default entry point name after having been turned into a valid C identifier.

−e, **−−entrypoint** *name*
Set the data entry point (used for linking against the data in a shared library form) to *name*. The default entry point name is the name set by the **−n**, **−−name** option.

−r, **−−revision** *version*
Enable versioning of the shared library produced in **dll**, or **library**, mode. The version number has the format *major.minor.patchlevel* and all parts except for *major* are optional. If only *major* is supplied then the version is assumed to be *major*.0 for versioning purposes.

−F, **−−rebuild**
Force the rebuilding of all data and their repackaging.

−I, **−−install**
Install the packaged file (or all the files in the **files** mode). If the variable **DESTDIR** is set it will be used for installation.

−s, **−−sourcedir** *source*
Set the source directory to *source*. The default source directory is the current directory.

−d, **−−destdir** *destination*
Set the destination directory to *destination*. The default destination directory is the current directory.

−T, **−−tempdir** *directory*
Set the directory used to generate temporary files to *directory*. The default temporary directory is the same as the destination directory as set by the **−d**, **−−destdir** option.

AUTHORS

Steven Loomis
Yves Arrouye

VERSION

63.2

COPYRIGHT