What is a package?

A package is a single compressed file (.tar.gz) with file extension **.nipkg** that contain 2 subsets of files

* Control files : package metadata
  + The control files are used to describe package relationships like dependencies, preconditions, requirements, etc. and also additional properties pertaining to the package like version, description, architecture, etc.
* Data files : the payload

**Structure of an .nipkg**

|  |
| --- |
| <pkg-name>.nipkg        <- <.ar>          control.tar.gz   <- <control files>              control      <- mandatory - Package metadata          data.tar.gz      <- <package payload>              instructions <- optional  - Custom actions XML file              ...          <- optional  - The payload of the package          debian-binary    <- mandatory - File containing the string "2.0\n" |

For file packages:

The payload needs to live inside different directories that map to an implied system location, a full list of the root names and mappings can be found here: **See Agent File Path Resolver Document**

Package file naming convention

Because files can always get renamed, package management will never use the file name to make decisions. Information about a package will only be collected from the spec file inside the package. The file name helps users identify a package in a folder, but serves no other purpose.

The package file name uses the following format:

<package-name>\_<package-version>\_<architecture>.nipkg

Control file attributes

| **Attribute Name**  **---**  **nipkgbuild Control File Variable** | **Required** | **Type** | **Default** | **Description** | **Example** |
| --- | --- | --- | --- | --- | --- |
| Package  ---  nipkgbuild\_PKG\_NAME | Yes | String | - | This field identifies the package and must be unique.  The Package serves as a *unique* identification for a package and is created by concatenating the *vendor-domain* and *label* strings.  The name must be a user friendly string that a customer would type on the command line:    Package names must consist only of lower case letters (a-z), digits (0-9), plus (+) and minus (-) signs, and periods (.).    Package names must be at least two characters long    Package names must start with an alphanumeric character.  **Expected Values:**  The name must match the regular expression  ^[a-z0-9][a-z0-9.+-]{2,}$  *warning)*Please note that uppercase is NOT permitted  **NI Guideline:**   * National Instruments packages should start with "ni-" * Packages that plug-into other products should follow the following convention:   + ni-<your product name>-<the product name you plug into>-support   + E.G. ni-daqmx-labview-support | Good Names:  ni-measure-2015, ni-curl, ni-labview-2015, ni-daqmx, ni-daqmx-labview-2015-support    Bad Names:  Curl, LabVIEW\_2015, NationalInstrumentsGPIB, labview |
| Source  ---  - | No | String | *NULL* | This field originally describes the source package that the binary package came from, it doesn't make sense in this context.  **Expected Values:**  *warning)*We are only parsing this to maintain compatibility with the debian format, this field is not needed and will not be used. | - |
| Version  ---  nipkgbuild\_PKG\_VERSION  (set automatically by build tools based on the nibuild 'package' file version) | Yes | String | - | This is set automatically by build tools based on the nibuild 'package' file version.  The version number will use the debian format:  <https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Version>  **Expected Values:**  NI will use the version policy:  <major>.<minor>.<update>.<lowOrderWindowsVersion>-0+<nibuild phaseLetter><nibuild buildNumber>  ex. nibuild packge version 15.1.0f7 -> installer version 15.1.0.49159-0+f7  *warning)*For dual citizen wininst packages, this should be the same (but translated to debian) version as your core MSI's ProductVersion property.   * For example, a MSI ProductVersion of 15.10.49153 will become an NIPM Version of 15.1.0.49153-0+f1. * But, if you are using an iBuild style version number, such as 16.0.0.449, then 16.0.0.449-0 is OK, as is 16.0.0.449-0+f1 if it's your f1.   More information [here](https://nitalk.jiveon.com/thread/80912) and [here](http://niweb.natinst.com/confluence/x/dgklHQ) (search for "*Does the version in my dual citizen MSI need to exactly match the version of the package file that contains it*") | 15.0.0.49152-0+f0 |
| Section  ---  nipkg\_PKG\_SECTION  (defaults to Infrastructure) | No | String | *NULL* | This field specifies an application area into which the package has been classified.  **Choose between one of these values:**   * **Programming Environments**   + Examples: LabVIEW, CVI/LabWindows, Measurement Studio * **Application Software**   + Examples: DIAdem, TestStand * **Add-Ons**   + Plugins and Toolkits belong to this category. * **Drivers** * **Runtime** * **Utilities**   + Examples: Update Service * **Documentation** Guided Help, In-product help, etc. * **Infrastructure** Infrastructure packages (packages with XB-UserVisible:no)   These values are available, but should not be used unless instructed:   * **"Other Software"**and**"LabVIEW"**   + Any legacy WinMIF products that are not dual citizens. This value is automatically added by WIA, so don't ever use these attribute values! * **"Library**" and "**Firmware**"   + These categories may be used in the future, but we don't have clear guidance on when or how. Don't use them in Card-A. * "**Feed**"   + Packages that install \*.repo files. The store item publishing process creates these packages. | Application Software |
| Priority  ---  nipkg\_PKG\_PRIORITY  (defaults to standard) | No | String | *NULL* | This field represents how important it is that the user have the package installed. According to the debian format:  <https://www.debian.org/doc/debian-policy/ch-archive.html#s-priorities>  **Expected Values:**  **required**  Packages which are necessary for the proper functioning of the system (usually, this means that nipkg functionality depends on these packages).  Removing a required package may cause your system to become totally broken and you may not even be able to use nipkg to put things back, so only do so if you know what you are doing. Systems with only the required packages are probably unusable, but they do have enough functionality to allow the other tools to install more software.  **important**  The important packages are just a bare minimum of commonly-expected and necessary tools.  **standard**  This is what will be installed by default if the user doesn't select anything else. It doesn't include many large applications.  **optional**  (In a sense everything that isn't required is optional, but that's not what is meant here.) This is all the software that you might reasonably want to install if you didn't know what it was and don't have specialized requirements. This is a much larger system. Note that optional packages should not conflict with each other.  **extra**  This contains all packages that conflict with others with required, important, standard or optional priorities, or are only likely to be useful if you already know what they are or have specialized requirements (such as packages containing only detached debugging symbols).  *warning)* We are only parsing this to maintain compatibility with the debian format, this field is not needed and will not be used. | standard |
| Architecture  ---  nipkgbuild\_PKG\_ARCHITECTURE  (defaults to $(PROC) as set by nibuild, and converted to a valid value) | Yes | String | - | Indicates the OS architectures supported by this package, can be a comma separated list if you support more than one.  **Expected Values:**  ~~windows\_x86,~~ windows\_x64  *warning)* For Cardassia, only a single value of **windows\_x64** is supported - even if your product is bit netural. See Rally [US98428](https://rally1.rallydev.com/#/26019714992d/detail/userstory/51266686211) for the future plan. | windows\_x64 |
| Essential  ---  - | No | Boolean | no | If set to yes then the package management system will refuse to remove the package (upgrading and replacing it is still possible).  *warning)***Do not use this** . If you have any reason to use this, please come and talk to the Package Management Team.  **Expected Values:**  yes, no | no |
| Depends  ---  nipkgbuild\_PKG\_DEPENDS | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | This declares an absolute dependency  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Pre-Depends  ---  - | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | Same as depends.  *warning)***Do not use this** .  We are only parsing this to maintain compatibility with the debian format, this field will be treated as another Depends relationship.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Recommends  ---  nipkgbuild\_PKG\_RECOMMENDS | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | This declares a strong, but not absolute, dependency.  The Recommends field should list packages that would be found together with this one in all but unusual installations.  ***warning)***Recommends relationships must only be declared to packages with XB-UserVisible set to true.  See [CAR #622572](http://nippm.natinst.com/itg/web/knta/crt/RequestDetail.jsp?REQUEST_ID=622572) for details.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Suggests  ---  nipkgbuild\_PKG\_SUGGESTS | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | This is used to declare that one package may be more useful with one or more others. Using this field tells the packaging system and the user that the listed packages are related to this one and can perhaps enhance its usefulness, but that installing this one without them is perfectly reasonable.  ***warning)*** Suggests relationships must only be declared to packages with XB-UserVisible set to true.  See [CAR #622572](http://nippm.natinst.com/itg/web/knta/crt/RequestDetail.jsp?REQUEST_ID=622572) for details.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Conflicts  ---  nipkgbuild\_PKG\_CONFLICTS | No | String relationship array | *NULL* | When one binary package declares a conflict with another nipkg  will refuse to allow them to be installed on the system at the same time.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Breaks  ---  - | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | Same as Conflicts  *warning)***Do not use this** .  We are only parsing this to maintain compatibility with the debian format, this field will be treated as another Conflicts relationship.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Provides  ---  nipkgbuild\_PKG\_PROVIDES | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | Other package relationship fields can declare a relation with "virtual packages". A virtual package is one which appears in the Provides control field of another package.  The effect is as if the package(s) which provide a particular virtual package name had been listed by name everywhere the virtual package name appears.  If there are both concrete and virtual packages of the same name, then the dependency may be satisfied (or the conflict caused) by either the concrete package with the name in question or any other concrete package which provides the virtual package with the name in question. This is so that, for example, supposing we have:  Package: foo Depends: bar  and someone else releases an enhanced version of the bar package they can say:  Package: bar-plus Provides: bar  and the bar-plus package will now also satisfy the dependency for the foo package.  **Notes**   * This attribute can optionally specify an explicit version that it provides, such as *Provides: bar (= 16.0).*If the version is not specified, then it satisfies any dependency. You MUST use = or use no explicit version at all. In other words, you MUST NOT use <<, >>, <=, or >=... * [See this page for more details on how Provides behaves.](http://niweb.natinst.com/confluence/display/ss/Additional+details+on+relationship+types)   **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Replaces  ---  nipkgbuild\_PKG\_REPLACES | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | Packages can declare in their control file that they should completely replace other packages.  Normally, Conflicts should be used in conjunction with Replaces.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Supplements  --  nipkgbuild\_PKG\_SUPPLEMENTS | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | This field is similar to Recommends but works in the opposite direction. It is used to declare that a package can supplement the functionality of another package.  ***warning)***Supplements relationships must only be declared to packages with XB-UserVisible set to true.  See [CAR #622572](http://nippm.natinst.com/itg/web/knta/crt/RequestDetail.jsp?REQUEST_ID=622572) for details.  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Enhances  ---  nipkgbuild\_PKG\_ENHANCES | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | This field is similar to Suggests but works in the opposite direction. It is used to declare that a package can enhance the functionality of another package.  ***warning)***Enhances relationships must only be declared to packages with XB-UserVisible set to true.  See [CAR #622572](http://nippm.natinst.com/itg/web/knta/crt/RequestDetail.jsp?REQUEST_ID=622572) for details.  **Based on:**  <https://www.debian.org/doc/debian-policy/ch-relationships.html>  **Expected Values:**  See the String relationship array section for more information about the syntax. | - |
| Installed-Size  ---  nipkgbuild\_PKG\_INSTALLEDSIZE | No | Long | 0 | It gives an estimate of the total amount of disk space required to install the named package  **Expected Values**  The disk space is given as the integer value of the estimated installed size in bytes, divided by 1024 and rounded up.  **Guidelines**  All NI Packages should define the Installed-Size, as this is used to calculate installation footprint. | 25485141 |
| Maintainer  ---  nipkgbuild\_PKG\_MAINTAINER  (set automatically) | Yes | String | - | The package maintainer's name and email address.  **Expected Values:**  The name must come first, then the email address inside angle brackets <> (in RFC822 format). | National Instruments <support@ni.com> |
| Description  ---  nipkgbuild\_PKG\_DESCRIPTION  (defaults to "This is an infrastructure package for NI Software.") | Yes | Multi-line | - | The Description field contains a description of the binary package, consisting of two parts, the synopsis or the short description, and the long description.  It is a multi-line field with the following format:  **Description Control Field**   |  | | --- | | Description: <single line synopsis>   <extended description over several lines> |   warning) We recommend to use the 2 starting white space characters for your extended descriptions. Read **Multi-line attributes** below for details | Package: acpi-events  Description: Scripts for handling many ACPI events  It only includes scripts for events that can be supported with some level of safety cross platform. |
| Homepage  ---  nipkgbuild\_PKG\_HOMEPAGE  (set automatically) | No | String | *NULL* | URL that the package vendor can use to provide more detailed information about the package  **Expected Values:**  A URL | http://www.ni.com/ |
| Size  ---  - | No | Long  Repository Attribute | 0 | It gives an estimate of the total amount of disk space that the .nipkg use.  **Expected Values:**  The disk space is given as the integer value of the estimated installed size in bytes, divided by 1024 and rounded up.  ***warning)*** This field is only used inside a  ***Packages*** file, and should **NOT** live on the Control file | 684484 |
| Filename  ---  - | No | Multiline  Repository Attribute | *NULL* | File name of the package being described, relative to the root of the Packages file.  **Expected Values:**  string with the relative path to the package file name.  ***warning)*** This field is only used inside a ***Packages***file, and should **NOT** live on the Control file | ..\..\pool\ni-pkg\_1.0.0.2-0+d2\_windows\_x64.nipkg |
| MD5Sum  ---  - | No | String  Repository Attribute | *NULL* | MD5 hash used to verify the integrity of the package once downloaded.  ***warning)*** This field is only used inside a  ***Packages*** file, and should **NOT** live on the Control file. | ac2aacc3c208b93cd1914b98a5210004 |
| Built-Using  ---  TBD | No | String | *NULL* | warning)  To be defined | - |
| XB-Plugin  ---  nipkgbuild\_PKG\_PLUGIN  (set automatically) | Yes | String | *NULL* | The Agent that will be used to install the Package.  **Expected Values:**  file | wininst | eula  warning)   "opkg | simple" are also supported values, but are only used for testing purposes. | file |
| XB-ReleaseNotes  ---  nipkgbuild\_PKG\_RELEASENOTES | No | Multiline | *NULL* | Human readable release notes for the package. Includes information about what's new in this version.  warning)  XB-ReleaseNotes is a multiline field. The first line of the field value (the part on the same line as XB-ReleaseNotes:) is always empty. The content of the field is expressed as continuation lines. Read **Multi-line attributes** below for details  Refer to [Policies and Guidelines for NI Packages](https://nitalk.jiveon.com/docs/DOC-311915), *The Release Notes (XB-ReleaseNotes;)* section for more information about the default string to use. | XB-ReleaseNotes:    Bug fixes and security updates.  DAQExpress - Refer to the NI website at http://www.ni.com/pdf/manuals/376809a.html for installation instructions, known issues, bug fixes, compatibility information, and other important notices about this product.  LabVIWE NXG - Refer to the NI website at http://www.ni.com/pdf/manuals/376808a.html for installation instructions, known issues, bug fixes, compatibility information, and other important notices about this product. |
| XB-HardwareSupport  ---  nipkgbuild\_PKG\_HARDWARESUPPORT | No | Multiline | *NULL* | When a package provides functionality for specific hardware product(s), it can lists such products in this section. This allows tools to recommend software to a customer based on the hardware on his system.  This attribute is meant to address the case in which the hardware is connected to a system, but a driver has not been installed.  Only top level user visible packages with the Section as Driver should define this attribute.  **Expected Values**  One supported hardware per line using one of the next 2 formats:  **Format 1:**  This is the recommended format.  **Hardware Support Format**   |  | | --- | | BUS\_##,VID\_####,PID\_#### |   BUS\_##   * The ## part should be a 2 digit Hex number (padded with zeros if less than two digits) represents the bus type, available types are:  | **Bus Type** | **Decimal** | **HEX** | | --- | --- | --- | | BuiltInSystem | 0 | 00 | | PciPxi | 1 | 01 | | Usb | 2 | 02 | | Gpib | 3 | 03 | | Vxi | 4 | 04 | | Serial | 5 | 05 | | TcpIp | 6 | 06 | | cRio | 7 | 07 | | Scxi | 8 | 08 | | cDaq | 9 | 09 | | SwitchBlock | 10 | 0A | | Scc | 11 | 0B | | FireWire | 12 | 0C | | Accessory | 13 | 0D | | Can | 14 | 0E | | SwitchBlockDevice | 15 | 0F |  * SystemAPI JSON Reference:   <http://jabba.natinst.com:8472/fsm/perforce/sa/ss/sysapi/trunk/16.0/source/mxSystemApi.json?view#line_16>  VID\_####   * The #### part should be a 4 digit Hex number (padded with zeros if less than four digits) that represents the VID (VendorNumber) * SystemAPI JSON Reference:   <http://jabba.natinst.com:8472/fsm/perforce/sa/ss/sysapi/trunk/16.0/source/mxSystemApi.json?view#line_1018>  PID\_####   * The #### part should be a 4 digit Hex number (padded with zeros if less than four digits) that represents the PID (ModelNumber) * SystemAPI JSON Reference:   <http://jabba.natinst.com:8472/fsm/perforce/sa/ss/sysapi/trunk/16.0/source/mxSystemApi.json?view#line_1050>  **OR Format 2:**   * 3rd party instruments will use this format * Network devices will use this format.   **Hardware Support Format**   |  | | --- | | VendorName,ModelName |   Comma separated VendorName and ModelName  **warning)   Important**   * Only packages marked as XB-UserVisible can declare HardwareSupport. * Because this is a multiline attribute:   + The first line of the field value (the part on the same line as XB-HardwareSupport:) is always empty.   + You should start each new line with two or more spaces. | XB-HardwareSupport:    BUS\_02,VID\_3923,PID\_717B    BUS\_02,VID\_3923,PID\_74B7    National Instruments,PXI-8110 |
| XB-SoftwareSupport  ---  nipkgbuild\_PKG\_SOFTWARESUPPORT | No | String Array | *NULL* | warning)  To be defined |  |
| XB-OsSupport  ---  nipkgbuild\_PKG\_OSSUPPORT | No | String Array | *NULL* | warning)  To be defined |  |
| XB-LanguageSupport  ---   nipkgbuild\_PKG\_LANGUAGESUPPORT | No | String Array | *NULL* | The list of languages that the payload supports.  **Expected Values**:  Comma separated 2 letter language codes following the ISO 639-1 Code standard | en, es, fr, de, zh |
| XB-StoreProduct  ---  nipkgbuild\_PKG\_STOREPRODUCT | No | Boolean | no | Whether the package represent a product on the Store.  Store Products are:   * Virtual packages * User visible packages * Equivalent to current "Distributions", "Suites" or any piece of software that is currently downloadable from ni.com * Recommend and Suggest top-level packages from other Products.   Read <https://nitalk.jiveon.com/docs/DOC-311915> for more info.  **Expected Value:**  <yes|no> | no |
| XB-Tags  ---  nipkgbuild\_PKG\_TAGS | No | String Array | *NULL* | List of strings that can be used to optionally tag a package. These are free form strings that can be used during searches for packages.  A package can have multiple tags.  **Expected Values**:  Array of strings | configuration, deployment, image, software installation |
| XB-UserVisible  ---  nipkgbuild\_PKG\_USERVISIBLE | No | Boolean | no | If yes:   * The package will be displayed on the UI. * The package is considered to be installed by a user, thus not installed to fulfill some dependency. (warning)   Only applicable on windows \*.nipkg packages) * **IMPORTANT** :   + All existing pieces of software that are currently downloadable from ni.com should be marked as UserVisible.   + If you want a package to be searchable and installable from the store, you should mark it as UserVisible too.   If no:   * The package will be hidden by default on the UI.   **Expected Value:**  <yes|no>   * yes : for Top Level Packages * no : for Infrastructure Packages | no |
| XB-RepositoryName  ---  - | No | String | *NULL* | The repository name where the package came from.  ***warning)*** This field is only returned by the solver when listing packages and should **NOT** live on the Control file. | nirvana-repository |
| XB-DisplayName  ---  nipkgbuild\_PKG\_DISPLAYNAME | No | String | *NULL* | User friendly package name.  **NI Guideline:**   * Should not contain the version in the name * Should not contain the "development stage" (development, alpha, beta, final)   + That information can be part of the Description Synopsis (It will appear aside the package name on the UI)   + Differentiate between packages by using the version field. * Should not contain the "trunk" (trunk, bajor, cardassia, etc)   + That information can be part of the Description Synopsis (It will appear aside the package name on the UI)   + Differentiate between packages by using the version field. | NI Measure |
| XB-DisplayVersion  ---  nipkgbuild\_PKG\_DISPLAYVERSION  (set automatically based on package file version) | No | String | *NULL* | User friendly display version.  **NI Guideline:**   * Should be in the form: Major.Minor.Update * Do not include the "development stage" words (dev, alpha, beta, final) | 15.0.0 |
| XB-Eula  ---  nipkgbuild\_PKG\_EULA | No | [String relationship array](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=482346760#NIPackageManagement:Howtobuildapackage&supportedattributes-StringRelationshipArray) | *NULL* | List of EULAs and their minimum versions required to be displayed before installing this package. A mapping of EULA types to Package Manager EULA names can be found in Perforce at //NIInstallers/export/Legal/license/nipkgeulamapping.csv  **warning)   Important**   * You must add eulaLocatorMinimumVersion to the declaration to pull it into the Package Locator and feed   + To know the version of the EULA package, check the package repo here: \\us-aus-argo\ni\nipkg\packages\eula   + For example: *nipkgbuild\_PKG\_EULA := eula-ni-standard eulaLocatorMinimumVersion=17.0.0* * If you have multiple EULAs, separate them with a comma. * Only use Package Manager EULA names from the second column of "nipkgeulamapping.csv". Do not use Legal Package Names or EULA Index names. * Package Manager EULA names always start with the "eula-" prefix. * Virtual packages do not need to declare EULAs because they have no payload. * All top-level and infrastucture packages that have a payload should declare the EULAs that are required for the payload they install. * Packages that only install 3rd-party software should not specify the NI standard EULA. They should specify the EULA for the 3rd-party software they install. * Note that only visible EULAs need to be listed in your package control file. So, if your EULA is not listed in the above .csv, then you do not need to list it in your package control file. * If porting to a dual citizen WinInst package, your EULAs should probably be the same in your package as in your WinMIF PDF. To convert between the PDF EULA name and the control file EULA name:   + For each <LegalPackage> tag in the PDF, search the following file to convert the PDF <LegalPackage> tag to the WinMIF EULA name (they are often the same, but not always):  //NIInstallers/export/Legal/license/LegalPackageIndex.xml   + Search the .csv file above for the WinMIF EULA name.     - If it’s not found, then it’s not a visible EULA, so don’t there is nothing to enter in the package control file.     - If it is found, then add it to the package control file as described above. | **Correct EULA declarations**  eula-ni-standard eulaLocatorMinimumVersion=16.0.0  eula-ni-standard eulaLocatorMinimumVersion=16.0.0, eula-ms-msxml-6.0 eulaLocatorMinimumVersion=16.0.0  eula-ivi eulaLocatorMinimumVersion=16.0.0    **Incorrect EULA declarations**  NIStandardReleased  NIStandardReleased, MSXML60  IVI  eula-ni-standard eula-ms-msxml-6.0  eula-ni-standard; eula-ms-msxml-6.0  eula-ni-standard |
| XB-RepositoryNames | No | String Array | *NULL* | The list of repository names that were installed as part of the payload.  This attribute is used to update recently installed repositories.  **Expected Values**:  Comma separated repository names installed by the payload.  **warning)   Important**   * Only packages marked as Section: Repository can declare this attribute. | XB-RepositoryNames: ni-product-repo0, ni-product-repo1 |
| XB-WinInstType | No | String | NULL | Special types of WIA package.  **Expected Values:**  legacy | dual  If NULL, it means it was installed by Package Manager (normal use case).  Note: Do not manually author this attribute in the control file; it gets automatically added by the WinMIF Recorder. | legacy |
| XB-Message      or  XB-Message-1, XB-Message-2, etc.  ---  For nipkgbuild, use a custom control file attribute (See XB-<YourAttribute>). | No | Multiline |  | Contains an important message to display before installing the package. Consists of two parts, a header and a body. (keyword: NIExtraFeatureInfo)  It is a multi-line field with the following format:  **Description Control Field**   |  | | --- | | XB-Message(-n): <header>   <body> |   warning) We recommend to use the 2 trailing white space characters for your extended descriptions. Read **Multi-line attributes** below for details.  The message will always be shown in the Package Manager UI before installing the package. See the [Preinstall Notification Feature Specification](http://niweb.natinst.com/confluence/display/AST/WIA+Package+PreInstall+Notification+and+Blocking+Feature+-+Solution+for+Cardassia+A) for more details. | XB-Message-1: Microsoft Visual C++ 2015 Run-Time issue  You must update Windows to install Microsoft KB 2919355, which is prerequisite for the Microsoft Visual C++ 2015 Run-Time software.  .  For more information, visit [ni.com/info](http://ni.com/info) and enter the Info Code: KB2919355. XB-Message-2: Loss of support for LabVIEW 2015 and earlier  If you have driver support for LabVIEW 2015 or earlier installed then the support will be removed when installing this package. |
| XB-CompatibilityVersion | No | Long |  | nipkg.exe automatically sets this attribute to a predetermined value when it packs a package. Package Manager checks this value before installing packages to ensure that an outdated version of Package Manager does not try to install a package that is newer. |  |
| XB-ActivateAfterInstallation | No | Boolean | no | **Expected Value:**  <yes|no>    If yes:   * When the package is being installed by the UI, then NIPM will attempt to call License Manager's [ActivateAll](http://niweb.natinst.com/confluence/pages/viewpage.action?pageId=412125626#Startup&ActivationExperiencesSpecification-4.1.4EnterSerialNumbers(ActivateAll)) ([US87870](https://rally1.rallydev.com/#/26019714992d/detail/userstory/45415968393))   If no:   * Package Manager will not attempt to call License Manager for activation, as long as there is not other package being installed at the same time that set's it's ActivateAfterInstallation flag to "yes". | no |
| XB-PackageFeeds | No | String Array | NULL | **warning) Package authors MUST NOT use this attribute.**  Comma delimited array of feed package names. This is used by the store to track feeds installed with products. Package authors **must not** use this attribute.  More info can be found here: <https://nitalk.jiveon.com/docs/DOC-378515/> | XB-PackageFeeds: ni-488.2-17.0.0-repository, ni-488.2-17.0.0-critical-updates-repository |
| XB-NIManagedFeed | No | Boolean | no | **warning) Package authors MUST NOT use this attribute.**  This is used by the store to indicate that an installed feed came from the store. Package authors **must not**use this attribute.  More info can be found here: <https://nitalk.jiveon.com/docs/DOC-378515/> | no |
| XB-<YourAttribute>  ---  Custom control file attributes may be specified using the "makovar\_NIPKG\_CUSTOM\_" prefix to the name of the attribute to be set.  Custom attributes specified this way will appear in the control file as "XB-(name):(value)".  Example:  makovar\_NIPKG\_CUSTOM\_MyCustomAttribute = yes  becomes  XB-MyCustomAttribute: yes | No | Multiline | *NULL* | Any other NI Custom Meta-data Attribute should be prefixed with "XB-" and will be ignored by the solver.  Read **Multi-line attributes** below. | XB-MyCustomAttribute |

String Relationship array

These fields all have a uniform syntax. They are a list of package names separated by commas.

The package names listed may also include lists of alternative package names, separated by vertical bar (pipe) symbols |. In such a case, that part of the dependency can be satisfied by any one of the alternative packages.

All of the fields except for Provides may restrict their applicability to particular versions of each named package. This is done in parentheses after each individual package name; the parentheses should contain a relation from the list below followed by a version number, in the format described above.

The relations allowed are <<, <=, =, >= and >> for strictly earlier, earlier or equal, exactly equal, later or equal and strictly later, respectively.

**Note: Do not use < or >. These relations actually mean <= and >=, respectively, in package manager. They are supported for debian compatibility, but debian has deprecated them because they are misleading.**

Whitespace may appear at any point in the version specification.

Example:

Depends: dependency01 (>= 2.2.1), dependency02 | dependency03

Multi-line Attributes

The value of a multiline field may comprise multiple continuation lines. The first line of the value, the part on the same line as the field name, often has special significance (Like in the description field) or may have to be empty (like on XB-ReleaseNotes). Other lines are added following the same syntax as the continuation lines of the folded fields. Whitespace, including newlines, is significant in the values of multiline fields.

The lines in multi-line attributes can have these formats:

* Those starting with a single space are part of a paragraph. Successive lines of this form will be word-wrapped when displayed. The leading space will usually be stripped off. The line must contain at least one non-whitespace character.
* Those starting with two or more spaces. These will be displayed verbatim. If the display cannot be panned horizontally, the displaying program will line wrap them "hard" (i.e., without taking account of word breaks). If it can they will be allowed to trail off to the right. None, one or two initial spaces may be deleted, but the number of spaces deleted from each line will be the same (so that you can have indenting work correctly, for example). The line must contain at least one non-whitespace character.
* Those containing a single space followed by a single full stop character. These are rendered as blank lines. This is the *only* way to get a blank line.

For example:

**Multiline Attribute**

|  |
| --- |
| Package: acpi-events  Description: Scripts for handling many ACPI events   It only includes scripts for events that can be supported with   some level of safety cross platform.   .    It is able to:   \* Detect loss and gain of AC power, lid closure, and the press of a   number of specific buttons (on Asus, IBM, Lenovo, Panasonic, Sony   and Toshiba laptops).   .   \* Suspend, hibernate and resume the computer, with workarounds for   hardware that needs it.   .   \* On some laptops, set screen brightness.   .   .    This package contains scripts to    react to various ACPI events. |

Will be rendered on the UI like this:

**Multi-line Attribute: Single line synopsis**

|  |
| --- |
| Scripts for handling many ACPI events |

**Multi-line Attribute: Extended description over several lines**

|  |
| --- |
| It only includes scripts for events that can be supported with some level of safety cross platform.  It is able to:  \* Detect loss and gain of AC power, lid closure, and the press of a number of specific buttons (on Asus, IBM, Lenovo, Panasonic, Sony and Toshiba laptops).  \* Suspend, hibernate and resume the computer, with workarounds for hardware that needs it.  \* On some laptops, set screen brightness.    This package contains scripts to  react to various ACPI events. |

warning) We recommend to use the 2 trailing white space characters for your extended descriptions.

Control file example

**Control File Example**

|  |
| --- |
| Package:ni-product-name Version:15.0.0.35-0+d36 Section:Application Architecture:windows\_x64 Essential:no Depends:ni-product-dependency-01 (>= 15.0.0), ni-product-dependency-02 (>= 13.0.0) Recommends:ni-recommended-package Suggests:ni-suggested-package Conflicts:ni-product-conflict (<< 15.0.0) Provides:ni-product-name (= 15.0.0.35-0+d36) Replaces:ni-pre-product-name Supplements:ni-supplemented-packageEnhances:ni-enhanced-package  Installed-Size:25485141  Maintainer:National Instruments <support@ni.com>  Description:Provides basic functionality for the X product    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.    Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.    Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.  Homepage:http://www.ni.com/  XB-Plugin:file  XB-ReleaseNotes:    Fix a potential startup crash.    Authentication did not work with alias.    Improved Interfaces.  XB-HardwareSupport:    BUS\_02,VID\_3923,PID\_717B    BUS\_02,VID\_3923,PID\_74B7    National Instruments,PXI-8110  XB-SoftwareSupport: <to be defined>  XB-OsSupport: <to be defined>  XB-LanguageSupport:en, es, fr, de, zh  XB-StoreProduct:yes  XB-Tags:configuration, product, measurement  XB-UserVisible:yes  XB-DisplayName:NI Product Name  XB-DisplayVersion:15.0.0  XB-Eula:eula-ni-standard eulaLocatorMinimumVersion=17.0.0, eula-ms-msxml-6.0 eulaLocatorMinimumVersion=17.0.0 |

How to build a package?

Having a package directory structure (like the one described above) on C:\my-package\

**How to build a package**

|  |
| --- |
| C:\nipkg.exe pack C:\my-package\ |

Will create the .nipkg file.