GetIt Local Files Documentation

Note: This is a temporary document and a work in progress. Please do not share this information without permission from Embarcadero.

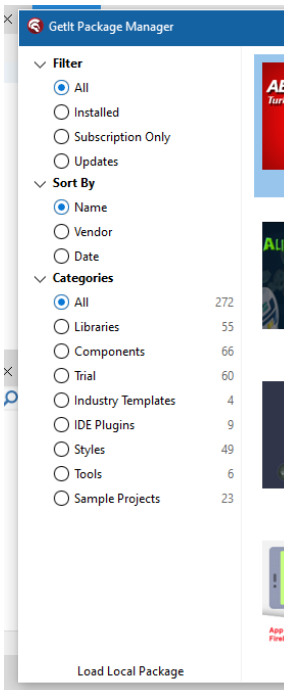
Since 11.2, RAD Studio has a feature for installing GetIt packages based on a local configuration file, rather than the content of the Getit server. This document offers details on the JSON configuration file format used for local GetIt packages..

1. Local GetIt Packages Introduction

Load Local Package (requires an active Update Subscription). This new button is visible at the bottom of the GetIt dialog sidebar, and its scope is to enable the installation of a Delphi package delivered as a local file. This can be useful in a different scenario:

- A customer can download a package and install it in RAD Studio on a computer not connected with the internet for security or other reasons. We plan to make a few key product add-ons available in my.embarcadero.com for this use case scenario.
- A package developer can use it to experiment with the creation of a GetIt package
- A company can use it to simplify the steps for installing libraries and addons in RAD Studio for multiple developers

This is the UI of the Load Local Package button. Once one or more local packages are installed, a special category will be added to the sidebar, allowing a user to see and uninstall them.



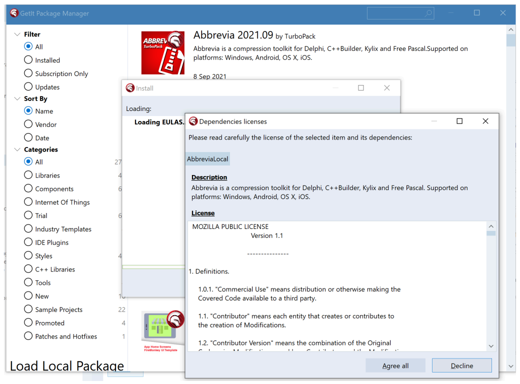
The GetIt Local Packages are defined by a JSON configuration file with a specific (and rather complex) format. The detailed documentation of this JSON GetIt local package configuration format is below.

Note: We are considering building some UI tools for editing these files.

1.1 Installing a Local GetIt Package

When you press it, the button executes a File Open dialog box to select a .json configuration file. The local GetIt packages are installed in a similar way to the online ones and are listed separately in the GetIt Package manager dialog:

- First you get prompted to accept the EULA (which can also be a local file)



Abbrevia 2021.09 by TurboPack

Abbrevia is a compression toolkit for Delphi, C++Builder, Kylix and Free Pascal.Supported on platforms: Windows, Android, OS X, iOS.

8 Sep 2021

Please read carefully the license of the selected item and its dependencies:

Description

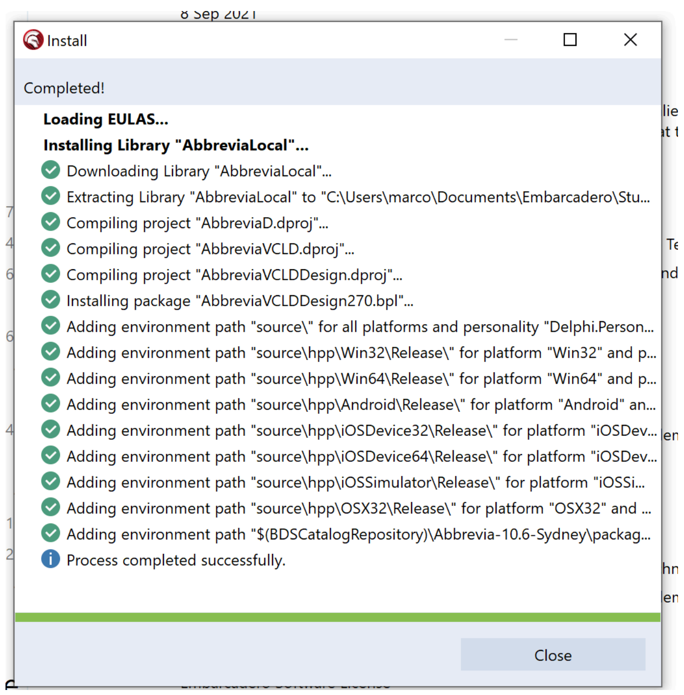
Abbrevia is a compression toolkit for Delphi, C++Builder, Kylix and Free Pascal. Supported on platforms: Windows, Android, OS X, iOS.

License

```
MOZILLA PUBLIC LICENSE
                    Version 1.1

                  ---------------

1. Definitions.

     1.0.1. "Commercial Use" means distribution or otherwise making the
     Covered Code available to a third party.

     1.1. "Contributor" means each entity that creates or contributes to
     the creation of Modifications.

     1.2. "Contributor Version" means the combination of the Original
```

- Second there is a “download or file copy” operation, depending if the actual package file is local or on the web (the file gets pushed to the CatalogRepository folder as usual in both cases)

- Third the installation actions get executed:



Install

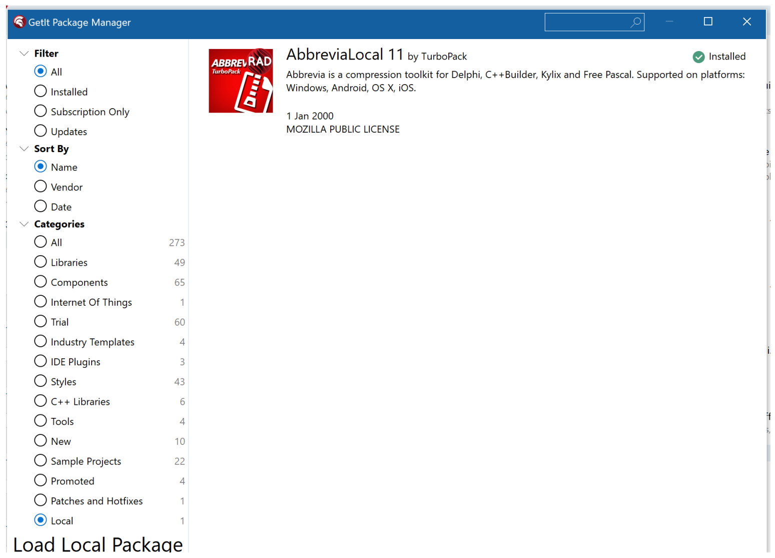
Completed!

Loading EULAS...

Installing Library "AbbreviaLocal"...

```
✓ Downloading Library "AbbreviaLocal"...
✓ Extracting Library "AbbreviaLocal" to "C:\Users\marco\Documents\Embarcadero\Stu...
✓ Compiling project "AbbreviaD.dproj"...
✓ Compiling project "AbbreviaVCLD.dproj"...
✓ Compiling project "AbbreviaVCLDDesign.dproj"...
✓ Installing package "AbbreviaVCLDDesign270.bpl"...
✓ Adding environment path "source\" for all platforms and personality "Delphi.Person...
✓ Adding environment path "source\hpp\Win32\Release\" for platform "Win32" and p...
✓ Adding environment path "source\hpp\Win64\Release\" for platform "Win64" and p...
✓ Adding environment path "source\hpp\Android\Release\" for platform "Android" an...
✓ Adding environment path "source\hpp\iOSDevice32\Release\" for platform "iOSDev...
✓ Adding environment path "source\hpp\iOSDevice64\Release\" for platform "iOSDev...
✓ Adding environment path "source\hpp\iOSSimulator\Release\" for platform "iOSSi...
✓ Adding environment path "source\hpp\OSX32\Release\" for platform "OSX32" and ...
✓ Adding environment path "$(BDSCatalogRepository)\Abbrevia-10.6-Sydney\packag...
i Process completed successfully.
```

At this point the package is installed. You need to close and reopen the package manager, and you see a new special category "Local" with local packages already installed. You can also use the item there to uninstall the local package.



GetIt Package Manager

Filter	
[✓ All]	
Installed	
Subscription Only	
Updates	

Sort By	
[✓ Name]	
Vendor	
Date	

Categories	
All	273
Libraries	49
Components	65
Internet Of Things	1
Trial	60
Industry Templates	4
IDE Plugins	3
Styles	43
C++ Libraries	6
Tools	4
New	10
Sample Projects	22
Promoted	4
Patches and Hotfixes	1
[✓ Local]	1

AbbreviaLocal 11 by TurboPack ✓ Installed

Abbrevia is a compression toolkit for Delphi, C++Builder, Kylix and Free Pascal. Supported on platforms: Windows, Android, OS X, iOS.

1 Jan 2000
MOZILLA PUBLIC LICENSE

Load Local Package

1.2 Installation folder and registry key

Notice that the IDE will save all the installed elements on repository cache, by default under the current user path at:

C:\Users\[USERNAME]\Documents\Embarcadero\Studio\[BDS version]\CatalogRepository

At the same time, information about the GetIt repository configuration and the installed packages is saved under the following registry key:

HKEY_CURRENT_USER\Software\Embarcadero\[BDS version]\CatalogRepository\Elements

2. The GetIt Local Files Format

The local file is a JSON file. It is a simplified format compared to the JSON used to configure a GetIt package on a server, simply because some of the fields make limited sense locally. The key element is that the JSON file allows you to specify, for the actual package content (which must be a ZIP file or a 7z file), for the license file and for the package image:

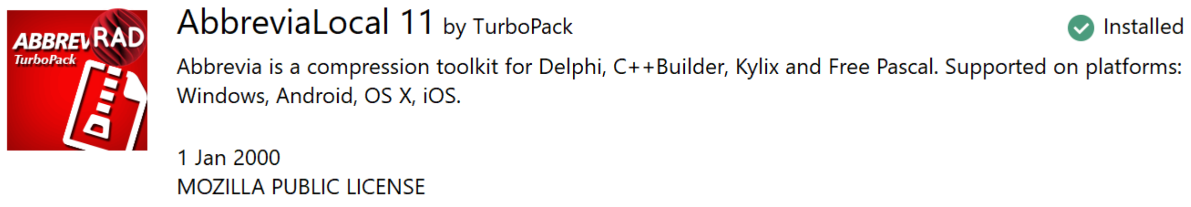
- A local file with a relative path to the JSON file
- A remote file with an http(s) URL

3. General Information

The first part lists general information about the package, the second part includes the actions to execute to install. This is a sample of the first part, each field is explained below:

```
{
  "Id": "Abbrevia-11-Local",
  "Name": "AbbreviaLocal",
  "Version": "11",
  "Description": "Abbrevia is a compression toolkit for Delphi...",
  "Vendor": "TurboPack",
  "VendorUrl": "http:\/\/www.turbopack.net",
  "Image": "Abbrevia3.png",
  "License": "MPL1.0.txt",
  "Url": "Abbrevia-20200708.zip",
  "ProjectUrl": "https:\/\/github.com\/TurboPack\/Abbrevia",
  "Modified": "2021-12-01 00:15:15",
  "LicenseName": "MOZILLA PUBLIC LICENSE",
  "RequireElevation": "0",
  "AllUsers": "0",
```

The matching item is displayed like this:



AbbreviaLocal 11 by TurboPack

Installed

Abbrevia is a compression toolkit for Delphi, C++Builder, Kylix and Free Pascal. Supported on platforms: Windows, Android, OS X, iOS.

1 Jan 2000
MOZILLA PUBLIC LICENSE

Id

This is a unique identifier for the library. This field accepts only characters from "a" to "z", from "0" to "9" and from "A" to "Z". This string value must be different from any other package Id. It is used internally, for the registry key of the package. If two packages have the same Id, you'll get an undefined behavior. It is not displayed in the GetIt dialog. This means that if you want 2 packages to coexist, they need to have a different ID. You can, instead, update a package indicating a new timestamp/version and it will be considered a new replacement version.

Name

This is the descriptive name of the library, displayed in the UI with a title font. Avoid making it too long so it fits the UI properly. Unlike the Id, this could potentially be duplicated, but clearly not recommended as it will confuse users. The name is also the primary search field. Above the name is “AbbreviaLocal”. Spaces can be used, of course.

Version

The version is another descriptive field added after the name in the description (the 11 in the image above). This version number is for the UI only, and is displayed on the item listing. It is not used for internal versioning or to manage updates (updates are driven by the value of the Modified field). You can set it to whatever value you want. For example, you may have multiple minor updates to a package but display them all as version 3. Also notice it is a string, so you can display version 1.2 or any format you like.

Description

This is a longer text description displayed below the title. Again, avoid making it too long for the UI to remain correct. If it is long, it gets truncated.

Vendor

The name of the vendor displayed as “by” near the title. Most likely, the vendor is you, so put your name or company name.

VendorUrl

This is the hyperlink of the vendor name label. The browser page will be opened if you click on the vendor name. Note you need to escape / symbols.

ProjectUrl

This is a URL of a web site that will be opened when clicking on the package name, in the GetIt UI. Note you need to escape / symbols.

Image

The image for the package. Can be a local file (relative to the JSON file, generally in the same folder) or a URL for an image to download. Notice you need to escape / symbols.

LicenseName

A descriptive name of the license, displayed in the UI

License

A URL or local file with the text of the license to display to users. The license needs to be accepted for the downloading to happen. This is true also for a local GetIt package. Note you need to escape / symbols.

Url

The URL or local file of the actual library to download (or copy). This needs to be a local ZIP file or the URL of a ZIP file. It can be a URL pointing to GitHub or another online repository, as long as you can refer to a ZIP file. While this can point at any web site, for packages you want to send to use to publish in GetIt, we recommend using only highly trusted locations (like GitHub or other high profile hosting services). In alternative, we can host internally. Notice you need to escape / symbols, as in the demo: “http:**\/\/**www.turbopack.net”

Modified

This is the modification time or timestamp. It looks like it is ignored for local files, but we need to investigate further. It should be the value displayed under the description (and above the license). It is relevant to manage the updates process. Need also to clarify if this is UTC.

RequireElevation

This is a boolean field, so it only accepts one of those values: 0 or 1. This value needs to be set to 1 if you need UAC to perform the following actions, for example to copy files under the product folder or in Program Files in general. Avoiding UAC is recommended when possible.

AllUsers

This is a boolean field, so it only accepts one of those values: 0 or 1. Install for the current users or all users. This affects registry settings. 0/False is highly recommended and the default. Specifies if the cache will be accessible by all users or only by current user.
- If value is False, cache path will be: $(BDSCatalogRepository)
- If value is True, cache path will be: $(BDSCatalogRepositoryAllUsers)

Omitted Values

Notice that the GetIt local file format does NOT include other fields of online GetIt packages like, required to publish a package on the GetIt server, including:
- Personalities (delphi, C++Builder or both)
- Minimum and maximum build
- Categories
- Platforms (almost ignored)
- SKU

4. Actions Data

Each GetIt items defines a list of actions to be performed by the RAD Studio IDE:

"Actions": [

```
          {
    “Id": "1",
    ...
  }
]
```

An individual action definition has fields like the following:

```
{
  "Id": "1",
  “ActionId": "6",
  "Type": "2",
  "RequireElevation": "0",
  "Parameter": [
    { "Parameter": "packages\\Sydney\\Delphi\\AbbreviaD.dproj" },
    { "Parameter": "Win32" },
    { "Parameter": "Release" }
  ],
  "ActionName": "CompileProject",
  "Description": "Compile AbbreviaD.dproj"
}
```

This is a summary of their role:

- **Id** is the unique identifier of the action. It must be different from any other action of the same GetIt package. The numeric value also determines the sequence of execution of actions associated with the same event (see *Type* below). That is, all After Download action will be executed in their numeric Id sequence.
- **ActionId** is the number of the actual action to be executed. The relevant actions and their IDs are listed below in this documentation. You need to refer to them by number in the JSON configuration file.
- **Type**: indicates the event (again using a numeric value, one of those listed in the *Events* section just below) the action is associated with, that is when in the process the action is triggered. In other words, this needs to be a number from 1 to 6, tied to events like After Download or Before Uninstall. The section below has the description of each type of action (or event).

- **RequireElevation** indicates if the specific actions require UAC. This works in conjunction with the same setting at the package configuration (both need to be set).
- **Parameter** is a list of one or more positional parameters. The role of a parameter strictly depends on the action being executed (each action has different number of parameters and a different value and role)
- **ActionName** is the optional description of the action and should match the ActionId numeric value (we understand this is not very logical, as the information is duplicated, and we are not certain this is required -- but it is good to have it as it keeps the JSON configuration file more readable).
- **Description** is a description of the action (displayed by the GetIt UI in the progress dialog and logged to file, along with the ActionID and ActionName)

4.1 Events

By default, all the system does with a GetIt package is expand the ZIP file in the catalog repository folder. It is possible to define actions to be executed at specific times of the installation (or uninstallation) process. Each action of a package is associated with an event:

We can use following events (the number is indicated as action "type", see above Type=2 for After download):

1: Before Download: This action happens before downloading a catalog item and cannot use any elements of the download files. It is ignored if the item is already in the cache

2: After Download: This action will be executed after download and extract a catalog item. It is ignored if the item is already in the cache

3: Before Install: This action will be executed before the installation of a catalog item.

4: After Install: This action will be executed after the installation of a catalog item.

5: Before Uninstall: This action will be executed before the uninstallation (that is removal from the cache) of a catalog item.

6: After Uninstall: This action will be executed after the uninstallation of a catalog item and cannot use any elements of the download files, as they have already been deleted at this point.

4.2 Available Actions

For each action the document includes the numeric value, the name, a summary description, a list of required and optional parameters, and some examples. The examples use a "code notation", like

```
AddOptionPath("source\", "cPasLibraryPath", "Delphi.Personality", "Win32")
```

and not the JSON notation used by the file, like:

```
 "Parameter": [
   { "Parameter": "source\" },
   { "Parameter": " "cPasLibraryPath" },
   { "Parameter": "Delphi.Personality" },
   { "Parameter": "Win32" }
 ],
```

In the future, we'll update the demos to reflect the actual JSON format. The actions are listed by number (safe for one exception), but we are considering a more logical sequence in future versions of this documentation. The list is in the initial table of contents.

1 - AddOptionPath

Adds an option path. Allowed options paths per platform/personality:

- Delphi Options:
  - Library Path: cPasLibraryPath
  - Package output directory: cPasPackageDPLOutput
  - DCP output directory: cPasPackageDCPOutput
  - Browsing path: cPasBrowsingPath
  - Debug DCU path: cPasDebugDCUPath
  - HPP output directory: cPasPackageHppOutput
  - Translated Library path: cPasTranslatedLibraryPath
  - Translated Debug DCU path: cPasTranslatedDebugLibraryPath
  - TranslatedResource path: cPasTranslatedResourcePath
- C++ Options:
  - System Include Path: cCppIncludePath
  - Library Path: cCppLibraryPath
  - Package output directory: cCppBPLOutput
  - BPI/LIB output directory: cCppBPIOutput
  - Browsing path: cCppBrowsingPath
  - Restrict refactoring path: cCppNoRefactorPath

Parameters

- Parameter 1: Specifies the path to add. Path is relative to the current repository path (for the specific package).
- Parameter 2: Specifies the option to add the path, see values above
- Parameter 3: Specifies the personality to add the path. (see personality constants values on "ToolsAPI.pas").
- Parameter 4: Optional. Specifies the platform to add the path (see below). If the parameter is empty, the path will be added to all the available platforms. Platform constants:
  - Win32, Win64, Android, iOSDevice64, Linux64, OSX64, Android64, OSXARM64
- Parameter 5: Optional. Boolean. Specifies in case of having 2 compilers for same platform (For instance: C++ Classic compiler or C++ Clang32 Compile for Win32) if the path will be added on both compilers. If we want to add path only to Clang32 compiler - use cCppIncludePath_Clang32 option and False. Value by default is true.

Examples

```
// Example 1 - Adds "source\" library path to Delphi personality on Windows 32-bit platform:

AddOptionPath("source\", "cPasLibraryPath", "Delphi.Personality", "Win32");

// Example 2 - Adds "source\" library path to Delphi personality on all platforms:

AddOptionPath("source\", "cPasLibraryPath", "Delphi.Personality");

// Example 3 - Adds "source\" library path to CBuilder personality on all platforms:

AddOptionPath("source\", "cCppIncludePath", "CPlusPlusBuilder.Personality");

 Example 4 - Adds "source\" library path from CBuilder personality on platform Win32 only on
C++ classic compiler:

AddOptionPath("source\", "cCppIncludePath", "CPlusPlusBuilder.Personality", "Win32", "False");

 Example 5 - Adds "source\" library path from CBuilder personality on platform Win32 only on
C++ Clang32 compiler:

AddOptionPath("source\", "cCppIncludePath_Clang32", "CPlusPlusBuilder.Personality", "Win32",
"False");
```

2- RemoveOptionPath

Removes an option path. Allowed options paths per platform/personality, as in the previous action.

Parameters

- Parameter 1: Specifies the path to remove. Path is relative to the current repository path.
- Parameter 2: Specifies the option to remove the path (See constants on "CodeMgr.pas").
- Parameter 3: Specifies the personality to remove the path. (see personality constants values on "ToolsAPI.pas").
- Parameter 4: Optional. Specifies the platform to remove the path (see platform constants on "PlatformConst.pas"). If the parameter is empty, the path will be removed from all the available platforms.
- Parameter 5: Optional. Boolean. Specifies in case of having 2 compilers for the same platform (For instance: The C++ Classic compiler or the C++ Clang32 compiler for Win32) if the path will be removed from both compilers. If we want to remove path only to the Clang32 compiler - use cCppIncludePath_Clang32 option and False. Value by default is true.

Examples

```
// Example 1 - Removes "source\" library path from Delphi personality on Windows 32-bit platform:

RemoveOptionPath("source\", "cPasLibraryPath", "Delphi.Personality", "Win32");

// Example 2 - Removes "source\" library path from Delphi personality on all platforms:

RemoveOptionPath("source\", "cPasLibraryPath", "Delphi.Personality");

// Example 3 - Removes "source\" library path from CBuilder personality on all platforms:

RemoveOptionPath("source\", "cCppIncludePath", "CPlusPlusBuilder.Personality");

// Example 4 - Removes "source\" library path from CBuilder personality on platform Win32 only on C++ classic compiler:

RemoveOptionPath("source\", "cCppIncludePath", "CPlusPlusBuilder.Personality", "Win32", "False");

// Example 5 - Removes "source\" library path from CBuilder personality on platform Win32 only on C++ Clang32 compiler:

 RemoveOptionPath("source\", "cCppIncludePath_Clang32", "CPlusPlusBuilder.Personality", "Win32", "False");
```

3 - ExecuteProgram

Executes a program inside the current product repository path.

Parameters

- Parameter 1: Program path to execute. Path is relative to the current repository path. Macros are allowed.

Examples

```
// Example 1 - Execute "boost_setup.exe" program:

ExecuteProgram("boost_setup.exe");

// Example 2 - Execute "sdk\platform-tools\adb.exe kill-server" program:

ExecuteProgram("sdk\platform-tools\adb.exe kill-server");
```

4 - InstallCHM // 5 - UninstallCHM

Installs /uninstalls a CHM file.

Parameters

- Parameter 1: CHM path to install/uninstall. Path should be relative to the current repository path.

Examples

```
InstallCHM("docs\Help\onguard.chm");
UninstallCHM("docs\Help\onguard.chm");
```

6 - CompileProject

Compile a project.

Parameters

- Parameter 1: project path to compile. Path is relative to the current repository path. Macros are allowed and include also
  - "$(DclUsrDpk)" -> dcluser common package path (Delphi version).
  - "$(DclUsrBpk)" -> dcluser common package path (CBuilder version).
- Parameter 2: **Optional.** List of platforms separated by ";" to compile the project. If the parameter is empty, GetIt will compile the project on all platforms (if the package is **Runtime or Runtime and Designtime**).
- Parameter 3: **Optional.** List of configurations separated by ";" to compile the project. If the parameter is empty, GetIt will compile the project using the following configurations: Release and Debug.

Examples

```
CompileProject("packages\Delphi\AbbreviaD.dproj");
```

CompileProject("packages\Delphi\AbbreviaD.dproj", "Android;Win64;Win32");

CompileProject("packages\Delphi\AbbreviaD.dproj", "Win32");

CompileProject("packages\Delphi\AbbreviaD.dproj", "", "Debug");

CompileProject("packages\Delphi\AbbreviaD.dproj", "Android;Win64;Win32", "Release;Debug");

29 - CleanProject

Clean a project

Parameters

- Same parameters as the "CompileProject" action.

Examples

CleanProject("packages\Delphi\AbbreviaD.dproj");

CleanProject("packages\Delphi\AbbreviaD.dproj", "Android;Win64;Win32");

CleanProject("packages\Delphi\AbbreviaD.dproj", "Win32");

CleanProject("packages\Delphi\AbbreviaD.dproj", "", "Debug");

CleanProject("packages\Delphi\AbbreviaD.dproj", "Android;Win64;Win32", "Release;Debug");

7- InstallPackage

Installs a package.

Parameters

- Parameter 1: bpl path to install.
- Parameter 2: Optional. Specifies if we want to install the package using BorlandIDEServices or not. In other words:
    - If the value is true, the action will add registry keys and load the package on RadStudio.
    - If the value is false, the action will add only registry keys to set the package.

  Default value is true.
- Parameter 3: Specifies if we want to show warnings. In other words:
    - If the value is true, the action will show warnings in case the package is already installed.
    - If the value is false, the action won't show warnings.

  Default value is true.

Examples
InstallPackage("OnGuardDD.bpl");

InstallPackage("OnGuardDD.bpl", "true");

InstallPackage("OnGuardDD.bpl", "false");

8 - UninstallPackage

Uninstalls a package.

Parameters
- The parameters are the same of install package

Examples
UninstallPackage("OnGuardDD.bpl");

UninstallPackage("OnGuardDD.bpl", "true");

UninstallPackage("OnGuardDD.bpl", "false");

11 - UninstallProgram

Uninstalls a program by executing following registry value "UninstallString"
on registry key:

- 32-bit system: "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\"
- 64-bit system: "HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\"

See for more details: https://msdn.microsoft.com/en-us/library/aa372105%28v=vs.85%29.aspx

Parameters
- Parameter 1: Program registry key to uninstall.

Examples
UninstallProgram("Boost Libraries for Embarcadero Developer Tools 16.0");

12 - ExecuteCommand

Executes a command line command (not a general executable) inside the current product repository path.

Parameters

- Parameter 1: Command to execute. Macros are allowed.
- Parameter 2: Optional. Specifies if we want to hide or not the command prompt window. In other words:
    - If the value is true, the Command prompt will be hidden.
    - If the value is false, the Command prompt will be shown.

  Default value is false.

Examples

```
// Example 1 - Execute command:

ExecuteCommand("echo y | sdk\tools\android.bat update sdk --no-ui --filter
platform-tools,build-tools-21.1.2,android-21");

// Example 2 - Execute command:

ExecuteCommand("sdk\platform-tools\adb.exe kill-server");

// Example 3 - Execute command on a hidden window (New since XE8 Update 1):

ExecuteCommand("sdk\platform-tools\adb.exe kill-server", "True");
```

16 - WarmNeededIDERestart

Displays a message to warm the user that RadStudio needs to be restarted to apply new changes.

Parameters

- <No parameters>

17 - InstallIDEPackage

Installs an IDE package.

Parameters

- Parameter 1: bpl path to install. Macros are allowed.
- Parameter 2: Optional. Specifies the personality to install the package:
    - Empty value: package available for all personalities.
    - "Delphi.Personality" value: package available for "Delphi" personality.
    - "CPlusPlusBuilder.Personality" value: package available for "C++Builder" personality.

  Default value is "" (package available for all personalities).

Examples

// Example 1 - Installs a package for all personalities:

```
InstallIDEPackage("$(BDSBIN)\projpageide220.bpl");
```

// Example 2 - Installs a package for "Delphi" personality:

```
InstallIDEPackage("$(BDSBIN)\delphide220.bpl", "Delphi.Personality");
```

18 - UninstallIDEPackage

Uninstalls an IDE package.

Parameters

- Same parameters of InstallIDEPackage

Examples

// Example 1 - Uninstalls a package for all personalities:

```
UninstallIDEPackage("$(BDSBIN)\projpageide220.bpl");
```

// Example 2 - Uninstalls a package for "Delphi" personality:

```
UninstallIDEPackage("$(BDSBIN)\delphide220.bpl", "Delphi.Personality");
```

19 - CopyFile

Copy a file to another path. See also action CopyFolder below.

Parameters

- Parameter 1: Original file path. Macros are allowed.
- Parameter 2: New file path. Macros are allowed.
- Parameter 3: Boolean. Optional. Flag to force directories. If the directory does not exist, specify if GetIt should create or not the directory before copying the file. Default value is "False".
- Parameter 4: Boolean. Optional. Flag to skip the process if failure when copying a file:
    - True: skip the process if failure when copying a file.
    - False: If failure, it shows a dialog with the option to retry the operation.

  Default value is "False".

Examples

// Example 1

CopyFile("Example.exe", "$(BDSBIN)\Example.exe");

// Example 2

CopyFile("Example.exe", "$(BDSBIN)\Example.exe", True);

// Example 3

CopyFile("Example.exe", "$(BDSBIN)\Example.exe", True, True);

20 - UnzipFile

Extract a compressed Zip file to a specified directory.

Parameters

- Parameter 1: Original zip file path. Macros are allowed.
- Parameter 2: Directory for extracting the file. Macros are allowed.

Examples

UnzipFile("Binaries.zip", "$(BDSBIN)\");

21 - RestartIDE

Restart the IDE.

Sets a flag to true and at the end of the installation/uninstallation process, it restarts the IDE. Before restarting, It checks if a project is already opened by the user.

Parameters

- <No parameters>

24 - AddValueToRegistry

Add a value to the windows registry.

This action takes into account the system architecture. For instance, this registry entry "HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\17.0" will be:

- On 64-bit system: "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Embarcadero\BDS\17.0"
- On 32-bit system: "HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\17.0"

Parameters

Parameter 1: Registry path. Macros are allowed.

Registry path format is following: RootKeyPath + Path

Where "RootKeyPath" could be:

- HKEY_CLASSES_ROOT
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_PERFORMANCE_DATA
- HKEY_CURRENT_CONFIG
- HKEY_DYN_DATA
- HKEY_CURRENT_USER

Parameter 2: Value name to add.

Parameter 3: Value to add.

Parameter 4: **Optional.** Type of value. The available types are:

- string
- integer
- boolean
- date
- datetime
- float
- time
- currency

Default value is "string".

Parameter 5: **Optional**. Type of value. Specifies if GetIt should expand macros from parameter 2 (Value Name) or not. **Default value is "False".**

Parameter 6: **Optional**. Type of value. Specifies if GetIt should expand macros from parameter 3 (Value to add) or not. **Default value is "False".**

Examples

```
// Example 1:

AddValueToRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\17.0", "Test",
"Hello world");

// Example 2:
```

AddValueToRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\17.0", "Test", "Hello world", "string");

// Example 3:

AddValueToRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\17.0", "Test", "True", "boolean");

// Example 4:

AddValueToRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\$(PRODUCTVERSIO N) ", "Test", "1", "integer");

// Example 5

AddValueToRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\ $(PRODUCTVERSION) ", "Test", "$(BDS)", "string", "False", "True");

25 - DeleteValueFromRegistry

Removes a value from the Windows registry.

This action takes into account the system architecture, as AddValueToRegistry does.

Parameters
Parameter 1: Registry path. Macros are allowed.

Registry path format is following: RootKeyPath + Path

Where "RootKeyPath" could be:

- HKEY_CLASSES_ROOT
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_PERFORMANCE_DATA
- HKEY_CURRENT_CONFIG
- HKEY_DYN_DATA
- HKEY_CURRENT_USER

Parameter 2: Value name to remove.

Parameter 3: **Optional**. Type of value. Specifies if GetIt should expand macros from parameter 2 (Value Name) or not. **Default value is "False".**

Examples

// Example 1:

DeleteValueFromRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\17.0", "Test");

// Example 2:

DeleteValueFromRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\
$(PRODUCTVERSION) ", "Test");

// Example 3

DeleteValueFromRegistry("HKEY_LOCAL_MACHINE\SOFTWARE\Embarcadero\BDS\
$(PRODUCTVERSION) ", "$(BDS)", "True");

32 - DeleteFile

Delete a file.

Parameters
Parameter 1: File path to delete. Macros are allowed.

Parameter 2: Boolean. **Optional**. Flag to skip the process if failure when deleting a file:

- True: skip the process if failure when deleting a file.
- False: If failure, it shows a dialog with the option to retry the operation.

Default value is "False".

Examples
// Example 1

DeleteFile("$(bds)\test.txt");

// Example 2

DeleteFile("$(bds)\test.txt", True);

33 - AddEnvironmentVariable

Add an IDE environment variable. Those variables can be used using macros and they are persistent (IDE writes a registry key "Environment variables" and a MSBUILD file "environment.proj").

For instance: $(Test)

Parameters

Parameter 1: String. Environment variable name to add.

Parameter 2: String. Environment variable value to add. Macros are allowed.

Parameter 3: Boolean. **Optional**. Specifies if GetIt must overwrite an existing variable or not.

Possible values are:

- "False": will not update an existing environment variable.
- "True": will update an existing environment variable.

Default value is "False".

Parameter 4: Boolean. **Optional**. Specifies if GetIt must expand or not macros of the value (parameter 2).

- "False": will not expand macros.
- "True": will expand macros.

Default value is "True".

Examples

```
// Example 1:

AddEnvironmentVariable("DemosDir", "$(BDSCOMMONDIR)\Samples\");

// Example 2:

AddEnvironmentVariable("DemosDir", "$(BDSCOMMONDIR)\Samples\", "True");

// Example 4:

AddEnvironmentVariable("DemosDir", "$(BDSCOMMONDIR)\Samples\", "True", "False");
```

34 - RemoveEnvironmentVariable

Remove an IDE environment variable.

Parameters

- Parameter 1: String. Environment variable name to remove.

Examples

```
RemoveEnvironmentVariable("DemosDir");
```

35 - AddTemporalVariable

Add a temporary variable. They are alive only during the process of installation/uninstallation of a catalog entry (they are not persistent).

Those variables can only be used in parameters of an action (like a macro).

For instance: $(Test)

Parameters
Parameter 1: String. Temporal variable name to add.

Parameter 2: String. Temporal variable value to add. Macros are allowed.

Parameter 3: Boolean. **Optional**. Specifies if GetIt must overwrite an existing variable or not.

Possible values are:

- "False": will not update an existing environment variable.
- "True": will update an existing environment variable.

Default value is "False".

Parameter 4: Boolean. **Optional**. Specifies if GetIt must expand only temporal variables or not on the variable value (parameter 2).

Default value is "False".

Examples

```
// Example 1:

AddEnvironmentVariable("DemosDir", "$(BDSCOMMONDIR)\Samples\");

// Example 2:

AddEnvironmentVariable("DemosDir", "$(BDSCOMMONDIR)\Samples\", "True");

// Example 3:

AddEnvironmentVariable("DemosDir", "$(BDSCOMMONDIR)\$(OtherTmpVar)\", "True", "True");
```

36 - RemoveTemporalVariable

Remove a temporary variable.

Parameters
Parameter 1: String. Temporal variable name to remove.

Examples
RemoveTemporalVariable("DemosDir");

37 - MoveFile

Move a file to another path.

If the file is in use, GetIt shows a message advising the user that the file is being used by another program and that it must close it to proceed the operation.

Parameters
Parameter 1: Original file path. Macros are allowed.

Parameter 2: New file path. Macros are allowed.

Parameter 3: Boolean. **Optional**. Flag to force directories. if the directory does not exist, specifies if GetIt should create or not the directory before copying the file. **Default value is "False".**

Parameter 4: Boolean. **Optional**. Flag to skip the process if failure when moving a file:

- True: skip the process if failure when moving a file.
- False: If failure, it shows a dialog with the option to retry the operation.

Default value is "False".

Examples
// Example 1

MoveFile("Example.exe", "$(BDSBIN)\Example.exe");

// Example 2

MoveFile("Example.exe", "$(BDSBIN)\Example.exe", True);

// Example 3

MoveFile("Example.exe", "$(BDSBIN)\Example.exe", True, True);

38 - ReplaceStrFromFile

Replace some text in the content of the file. In other words, this command allows you to modify a file that is part of the package with a string replace operation (doing a replace all, and ignoring case), so that you can inject information in the file (for example a course code file or include file) programmatically.

Parameters
Parameter 1: String. File path. Macros are allowed.

Parameter 2: String. Old string to replace if parameter 4 is set to "False".

If parameter 4 is set to "True", then we must enter a regular expression pattern.

Parameter 3: String. New string to replace. Macros are allowed if parameter 5 is set to True.

Parameter 4: Boolean. **Optional**. Specifies if we want to use a regular expression or not. **Default value is "False".**

Parameter 5: Boolean. **Optional.** Specifies if the new string to replace (Parameter 3) must be expanded before in case of using macros (for instance "$(BDS)"). **Default value is "False".**

Parameter 6: Boolean. **Optional.** If the value is "True", it skips the action without failing if the file does not exist. **Default value is "False".**

Examples
// Example 1

ReplaceStrFromFile("c:\test.txt", "Old string", "New string", False);

// Example 2

ReplaceStrFromFile("c:\test.dproj", "\<ProjectVersion\>(.+?)\<\/ProjectVersion\>", "<ProjectVersion>16.0</ProjectVersion>", True);

// Example 3 (Since BigBen or CatalogVersion = 5)

ReplaceStrFromFile("c:\test.txt", "_RADSTUDIOPATH_", "$(BDS)", False, True);

// Example 4 (Since BigBen or CatalogVersion = 5)

ReplaceStrFromFile("c:\test.txt", "_RADSTUDIOPATH_", "$(BDS)", False, True, True);

39 - CreateShortcut

Create a shortcut file

Parameter 1: String. Target file path. Macros are allowed.

Double quotes are needed when adding paths with parameters.

For instance: "$(BDS)\bin\bds.exe" "-pdelphi"

Paths with parameters are allowed

Parameter 2: String. Shortcut destination. Macros are allowed.

Parameter 3: Boolean. **Optional.** Flag to force directories (of parameter 2). if the directory does not exist, specifies if GetIt should create or not the directory before copying the file. **Default value is "False".**

Parameter 4: String. **Optional.** Shortcut description.

Parameter 5: String. **Optional.** Shortcut icon. Macros are allowed.

Examples

```
// Example 1

CreateShortcut("C:\windows\notepad.exe", "C:\Users\admin\Desktop\Notepad.lnk", False);

// Example 2

CreateShortcut("C:\windows\notepad.exe", "C:\Users\admin\Desktop\Test\Notepad.lnk", True);

// Example 3

CreateShortcut("C:\windows\notepad.exe", "$(_DESKTOP)\Notepad.lnk", True);

// Example 4 - shortcut with parameters

CreateShortcut('"$(BDS)\bin\bds.exe" "-pdelphi"', '$(_DESKTOP)\Test.lnk');

// Example 5 - shortcut with description

CreateShortcut('"$(BDS)\bin\bds.exe" "-pdelphi"', '$(_DESKTOP)\Test.lnk', False, 'Hello word');

// Example 6 - shortcut with custom icon

CreateShortcut('"$(BDS)\bin\bds.exe" "-pdelphi"', '$(_DESKTOP)\Test.lnk', False, 'Hello word',
'C:\dev\tp\images\AspNewContentPage_CS.ico');
```

40 - CopyFolder

Copy a folder with its content (files and subfolders) to another directory.

Parameter 1: String. Original directory path to copy. Macros are allowed.

Parameter 2: String. Target directory path. Macros are allowed.

Parameter 3: Boolean. **Optional.** Flag to save all copied file paths (of parameter 2) on the installation .dat GetIt file. If the value is "True", those file paths will be deleted automatically during the uninstallation process. **Default value is "False".**

Parameter 4: Boolean. **Optional**. Flag to skip the process if failure when copying a file:

- True: skip the process if failure when copying a file.
- False: If failure, it shows a dialog with the option to retry the operation.

Default value is "False".

Examples

```
// Example 1

CopyFolder("C:\Test", "C:\Users\admin\Desktop\Test", False);

// Example 2

CopyFolder("C:\Test", "C:\Users\admin\Desktop\Test", True);

// Example 3

CopyFolder("C:\Test", "C:\Users\admin\Desktop\Test", True, True);
```

41 - OpenCloseProject

Open or close a project, depending on the value of one of the parameters -- this is a fairly odd design, but hard to change it now.
Notice that since 10.4, this action also opens ".groupproj" and ".bdsgroup" group projects.

Parameter 1: string. project path to modify. Macros are allowed.

- "$(DclUsrDpk)" -> dcluser common package path (Delphi version).
- "$(DclUsrBpk)" -> dcluser common package path (CBuilder version).

Parameter 2: Boolean. **Optional.** If the value is "True", GetIt will open a project. If the value is "False", GetIt will close the project.

Default value is "True".

Parameter 3: Boolean. **Optional.** If the value is "True", GetIt will stop the installation if there is a failure when opening or closing a project. If the value is "False", GetIt will avoid the failure.

Default value is "False".

Examples

```
// Example 1

OpenCloseProject("Package\Test.dproj");

// Example 2

OpenCloseProject("Package\Test.dproj", False);

// Example 3

OpenCloseProject("Package\Test.dproj", True, True);
```

42 - MoveFolder

Move a folder with its content (files and subfolders) to another directory.

Parameter 1: String. Original directory path to move. Macros are allowed.

Parameter 2: String. Target directory path. Macros are allowed.

Parameter 3: Boolean. **Optional.** Flag to save all moved file paths (of parameter 2) on the installation .dat GetIt file. If the value is "True", those file paths will be deleted automatically during the uninstallation process. **Default value is "False".**

Parameter 4: Boolean. **Optional**. Flag to skip the process if failure when moving a file:

- True: skip the process if failure when moving a file.
- False: If failure, it shows a dialog with the option to retry the operation.

Default value is "False".

Examples

```
// Example 1

MoveFolder("C:\Test", "C:\Users\admin\Desktop\Test", False);

// Example 2

MoveFolder("C:\Test", "C:\Users\admin\Desktop\Test", True);

// Example 3

MoveFolder("C:\Test", "C:\Users\admin\Desktop\Test", True, True);
```

Predefined Temporary Variables

In general, actions can use the same macros referring to folders available in the tools configuration and other places within RAD Studio (like library folders, etc). References use the notation: *$(macro)*

These additional values can be used only in action parameters (like macros). The is a very long list, the most relevant ones are listed here:

- _CatalogRepository: catalog repository path
- _CatalogId: ID of the element that is being installed/uninstalled
- _PackageVersion: Current IDE package version suffix. For instance, for RadStudio 10 Seattle: 240
- _MYDOCUMENTS: See constant "CSIDL_MYDOCUMENTS" in Winapi.ShlObj.pas for more details
- _APPDATA: <user name>\Application Data. See constant "CSIDL_APPDATA" in Winapi.ShlObj.pas for more details.
- _WINDOWS: GetWindowsDirectory()
- _SYSTEM: GetSystemDirectory()

5. Sample Code

Link to JSON sample for Abbrevia Local
Link to complete ZIP file (including JSON, license, package ZIP, png image)
Below are the definitions of the actions of a number of different types of public packages, including different types of actions.

Alien Invasion

```
{
 "IdReadable": "AlienInvasion-1.11",
 "Name": "Alien Invasion",
 ...
 "Actions": [
  {
   "ActionName": "AddEnvironmentVariable",
   "Type": "3",
   "Description": "Add %DEMOSDIR% if not exists",
   "RequireElevation": "0",
   "Active": "1",
   "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter": "DEMOSDIR"},
        {"Parameter": "$(BDSCOMMONDIR)\\Samples"},
        {"Parameter": "False"}
      ]
    },
    {
      "ActionName": "ExecuteCommand",
      "Type": "3",
      "Description": "Copy Sample Folder if possible",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter":
         "InstallSamples.bat &quot;Src&quot; &quot;Games\\Object Pascal\\AlienInvasion-1.11&quot;"},
        {"Parameter": "True"}
      ]
    },
    {
      "ActionName": "OpenCloseProject",
      "Type": "3",
      "Description": "Open the project Alien Invasion",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter":
"$(DEMOSDIR)\\Games\\Object Pascal\\AlienInvasion-1.11\\AlienInvasion.dproj"}
      ]
    },
    {
      "ActionName": "ExecuteCommand",
      "Type": "5",
      "Description": "Remove project",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
       {
         "Parameter":
         "rmdir \/S \/Q &quot;$(DEMOSDIR)\\Games\\Object Pascal\\AlienInvasion-1.11\\&quot;"
       }
      ]
    }
  ]
}
```

Diamond VCL Premium Style

```
{
  "IdReadable": "DiamondVCLPremiumStyle-1.1",
  "Name": "Diamond VCL Premium Style",
  ...
  "Actions": [
  {
                  "ActionName": "CopyFile",
                  "Type": "3",
                  "Description": "Copy Diamond.vsf file to the Styles folder",
                  "RequireElevation": "0",
                  "Active": "1",
                  "MinPV": "",
                  "MaxPV": "",
                  "Parameter": [
                    {"Parameter":
              "$(BDSCatalogRepository)\\DiamondVCLPremiumStyle-1.1\\Diamond.vsf},
                    {"Parameter": "$(BDSCOMMONDIR)\\Styles\\Diamond.vsf"},
                    {"Parameter": "True"}
                  ]
                },
                {
                  "ActionName": "DeleteFile",
                  "Type": "5",
                  "Description": "Delete Diamond.vsf file from the Styles folder",
                  "RequireElevation": "0",
                  "Active": "1",
                  "MinPV": "",
                  "MaxPV": "",
                  "Parameter": [
                    {"Parameter": "$(BDSCOMMONDIR)\\Styles\\Diamond.vsf"}
                  ]
                }
              ],
```

FMXLinux

```
{
  "IdReadable": "FmxLinux-1.66",
  "Name": "FMXLinux",
  ...
  "Actions": [
    {
      "ActionName": "AddValueToRegistry",
      "Type": "3",
      "Description": "Registers expert library",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter": "HKEY_CURRENT_USER\\Software\\Embarcadero\\BDS\\22.0\\Experts"},
        {"Parameter": "FmxLinux"},
        {"Parameter": "$(BDSCatalogRepositoryAllUsers)\\FmxLinux-1.66\\bin\\FmuxLib.dll"}
      ]
    },
    {
      "ActionName": "AddOptionPath",
      "Type": "3",
      "Description": "Add browsing path for Linux platform",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter": "$(BDSCatalogRepositoryAllUsers)\\FmxLinux-1.66\\Lib\\Release\\"},
        {"Parameter": "cPasLibraryPath"},
        {"Parameter": "Delphi.Personality"},
        {"Parameter": "Linux64"}
      ]
  },
  {
    "ActionName": "ExecuteCommand",
    "Type": "3",
    "Description": "Open Samples folder",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": [
      {"Parameter":
        "start &quot;&quot;
&quot;$(BDSCatalogRepositoryAllUsers)\\FmxLinux-1.66\\samples\\&quot;"}
    ]
  },
  {
    "ActionName": "RestartIDE",
    "Type": "4",
    "Description": "User must restart the IDE to apply these changes",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": []
  },
  {
    "ActionName": "DeleteValueFromRegistry",
    "Type": "5",
    "Description": "Uninstalls expert library",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": [
      {"Parameter": "HKEY_CURRENT_USER\\Software\\Embarcadero\\BDS\\22.0\\Experts"},
      {"Parameter": "FmxLinux"}
    ]
  },
  {
    "ActionName": "RemoveOptionPath",
    "Type": "5",
    "Description": "Remove browsing path for Linux platform",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": [
      {"Parameter": "$(BDSCatalogRepositoryAllUsers)\\FmxLinux-1.66\\Lib\\Release\\"},
      {"Parameter": "cPasLibraryPath"},
      {"Parameter": "Delphi.Personality"},
      {"Parameter": "Linux64"}
    ]
  },
  {
    "ActionName": "RestartIDE",
    "Type": "6",
    "Description": "User must restart the IDE to apply these changes",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": []
  }
],
```

JEDI Code Library

```
{
  "IdReadable": "JEDICodeLibraryJCL-2021.09",
  "Name": "JEDI Code Library",
  ...
  "Actions": [
    {
      "ActionName": "ExecuteProgram",
      "Type": "2",
      "Description": "Run JediInstaller for Install",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter":
```

```
        "install.bat install AutoAcceptMPL AcceptWarnings AcceptErrors IgnoreRunningIDE"}
      ]
    },
    {
      "ActionName": "WarmNeededIDERestart",
      "Type": "4",
      "Description": "Restart Needed Advice",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": []
    },
    {
      "ActionName": "ExecuteProgram",
      "Type": "5",
      "Description": "Run JediInstaller for Uninstall",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter": "bin\\JediInstaller.exe uninstall IgnoreRunningIDE"}
      ]
    },
    {
      "ActionName": "WarmNeededIDERestart",
      "Type": "6",
      "Description": "Restart Needed Advice",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": []
    }
  ],
```

Xerces - C++ validating XML parser

```
{
  "IdReadable": "Xerces-2021.09",
"Name": "Xerces - C++ validating XML parser",
...
"Actions": [
 {
   "ActionName": "AddOptionPath",
   "Type": "3",
   "Description": "Add library path for Delphi.Personality",
   "RequireElevation": "0",
   "Active": "1",
   "MinPV": "",
   "MaxPV": "",
   "Parameter": [
     {"Parameter": "$(BDSCatalogRepository)\\Xerces-2021.09\\"},
     {"Parameter": "cPasLibraryPath"},
     {"Parameter": "Delphi.Personality"}
   ]
 },
 {
   "ActionName": "AddOptionPath",
   "Type": "3",
   "Description": "Add library path for Delphi.Personality",
   "RequireElevation": "0",
   "Active": "1",
   "MinPV": "",
   "MaxPV": "",
   "Parameter": [
     {"Parameter": "$(BDSCatalogRepository)\\Xerces-2021.09\\"},
     {"Parameter": "cCppIncludePath"},
     {"Parameter": "CPlusPlusBuilder.Personality"}
   ]
 },
 {
   "ActionName": "CompileProject",
   "Type": "3",
   "Description": "Compile xerces-c.cbproj project",
   "RequireElevation": "0",
   "Active": "1",
   "MinPV": "",
   "MaxPV": "",
   "Parameter": [
     {"Parameter":          "$(BDSCatalogRepository)\\Xerces-2021.09\\cbuilder\\xerces-c.cbproj"},
      {"Parameter": "Win64"},
      {"Parameter": "Release"}
    ]
  },
  {
    "ActionName": "OpenCloseProject",
    "Type": "3",
    "Description": "Open xercesGroup.groupproj project group",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": [
      {"Parameter":
"$(BDSCatalogRepository)\\Xerces-2021.09\\cbuilder\\xercesGroup.groupproj"},
      {"Parameter": "Win32"},
      {"Parameter": "Release"}
    ]
  },
  {
    "ActionName": "ExecuteCommand",
    "Type": "3",
    "Description": "Open doc folder",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": [
      {"Parameter":
       "start &quot;&quot; &quot;$(BDSCatalogRepository)\\Xerces-2021.09\\&quot;"}
    ]
  },
  {
    "ActionName": "RemoveOptionPath",
    "Type": "5",
    "Description": "Remove library path for Delphi.Personality",
    "RequireElevation": "0",
    "Active": "1",
    "MinPV": "",
    "MaxPV": "",
    "Parameter": [
        {"Parameter": "$(BDSCatalogRepository)\\Xerces-2021.09\\"},
        {"Parameter": "cPasLibraryPath"},
        {"Parameter": "Delphi.Personality"}
      ]
    },
    {
      "ActionName": "RemoveOptionPath",
      "Type": "5",
      "Description": "Remove library path for Delphi.Personality",
      "RequireElevation": "0",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter": "$(BDSCatalogRepository)\\Xerces-2021.09\\"},
        {"Parameter": "cCppIncludePath"},
        {"Parameter": "CPlusPlusBuilder.Personality"}
      ]
    }
  ],
```

XML Mapper

```
{
  "IdReadable": "XMLMapper-27",
  "Name": "XML Mapper",
  ...
  "Actions": [
    {
      "ActionName": "DeleteFile",
      "Type": "3",
      "Description": "Delete old XmlMapper.exe if exists",
      "RequireElevation": "1",
      "Active": "1",
      "MinPV": "",
      "MaxPV": "",
      "Parameter": [
        {"Parameter": "$(BDSBIN)\\XmlMapper.exe"},
        {"Parameter": "True"}
      ]
    },
{
  "ActionName": "CopyFile",
  "Type": "3",
  "Description": "Backup XmlMapper.exe file",
  "RequireElevation": "1",
  "Active": "0",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter": "$(BDSBIN)\\XmlMapper.exe"},
    {"Parameter":
     "$(BDSCatalogRepository)\\XMLMapper-27\\XmlMapper_old.exe"},
    {"Parameter": "True"},
    {"Parameter": "True"}
  ]
},
{
  "ActionName": "MoveFile",
  "Type": "3",
  "Description": "Backup XmlMapper.exe file",
  "RequireElevation": "1",
  "Active": "0",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter": "$(BDSBIN)\\XmlMapper.exe"},
    {"Parameter":
     "$(BDSCatalogRepository)\\XMLMapper-27\\XmlMapper_old.exe"},
    {"Parameter": "True"},
    {"Parameter": "True"}
  ]
},
{
  "ActionName": "CopyFile",
  "Type": "3",
  "Description": "Update IDEStyleNotifier270.bpl file",
  "RequireElevation": "1",
  "Active": "1",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter":
     "$(BDSCatalogRepository)\\XMLMapper-27\\IDEStyleNotifier270.bpl"},
    {"Parameter": "$(BDSBIN)\\IDEStyleNotifier270.bpl"},
    {"Parameter": "False"},
    {"Parameter": "False"}
  ]
},
{
  "ActionName": "CopyFile",
  "Type": "3",
  "Description": "Update XmlMapper.exe file",
  "RequireElevation": "1",
  "Active": "1",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter":
     "$(BDSCatalogRepository)\\XMLMapper-27\\XmlMapper.exe"},
    {"Parameter": "$(BDSBIN)\\XmlMapper.exe"},
    {"Parameter": "False"},
    {"Parameter": "False"}
  ]
},
{
  "ActionName": "InstallPackage",
  "Type": "3",
  "Description": "Install IDEStyleNotifier270.bpl package",
  "RequireElevation": "0",
  "Active": "1",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter": "IDEStyleNotifier270.bpl"}
  ]
},
{
  "ActionName": "UninstallPackage",
  "Type": "5",
  "Description": "Uninstall IDEStyleNotifier270.bpl package",
  "RequireElevation": "0",
  "Active": "1",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter": "IDEStyleNotifier270.bpl"}
  ]
},
{
  "ActionName": "DeleteFile",
  "Type": "5",
  "Description": "Delete IDEStyleNotifier270.bpl file",
  "RequireElevation": "1",
  "Active": "1",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter": "$(BDSBIN)\\IDEStyleNotifier270.bpl"},
    {"Parameter": "False"}
  ]
},
{
  "ActionName": "DeleteFile",
  "Type": "5",
  "Description": "Delete XmlMapper.exe file",
  "RequireElevation": "1",
  "Active": "1",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
    {"Parameter": "$(BDSBIN)\\XmlMapper.exe"},
    {"Parameter": "False"}
  ]
},
{
  "ActionName": "MoveFile",
  "Type": "5",
  "Description": "Restore XmlMapper.exe file",
  "RequireElevation": "1",
  "Active": "0",
  "MinPV": "",
  "MaxPV": "",
  "Parameter": [
      {"Parameter":
       "$(BDSCatalogRepository)\\XMLMapper-27\\XmlMapper_old.exe"},
      {"Parameter": "$(BDSBIN)\\XmlMapper.exe"},
      {"Parameter": "True"},
      {"Parameter": "True"}
    ]
  }
],
```