



# **User Guide**

## **Persistence DB Configuration Tool**

Version 1.1

<16/01/2016>

# Content

---

<b>1. Introduction .....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Support information .....	1
1.3 Revision History .....	1
1.4 Abbreviations & Terminology.....	2
<b>2. Unpacking/Installing .....</b>	<b>3</b>
2.1 Downloading Software & tools.....	3
2.2 Setting up the environment.....	3
2.3 Get the sources .....	3
2.4 Build the tool .....	4
2.5 Running the tool.....	4
<b>3. Persistence DB Configuration Tool .....</b>	<b>5</b>
3.1 Persistence Administration Service .....	5
3.1.1 Installation Scope .....	5
3.2 Data organization.....	6
3.3 JSON Configuration Files .....	8
3.4 Application Installation Configuration .....	8
3.4.1 JSON File Format.....	8
3.4.2 Example .....	8
3.5 Resource Configuration Table Configuration .....	9
3.5.1 JSON File Format.....	10
3.5.2 Example .....	10
3.6 Default Data.....	11
3.6.1 JSON File Format.....	11
3.6.2 Example .....	11
3.7 Install Exception.....	12
<b>4. Graphical User Interface .....</b>	<b>13</b>
4.1 User Interface Details .....	14
4.1.1 Area 1 – Application Area .....	14
4.1.2 Area 2 – Resource Area.....	14
4.1.3 Area 3 – Resource Configuration Area .....	14
4.1.4 Area 4 - Default Data Area .....	14
<b>5. Tool Handling .....</b>	<b>15</b>
5.1 Open an existing configuration .....	15

5.2	Add an application .....	15
5.2.1	Installation Rules .....	16
5.3	Add a resource .....	17
5.3.1	Install Exception .....	17
5.4	Resource Configuration .....	18
5.5	Add default data.....	19
5.5.1	Key-value default data.....	19
5.5.2	File default data.....	19
5.6	Generating JSON files .....	20
<b>6.</b>	<b>Appendix.....</b>	<b>21</b>

# List of Tables & Figures

---

## Tables

Table 1 - Revision History .....	1
Table 2 - Abbreviations & Terminology .....	2
Table 3 - Affected data and allowed exception .....	12

## Figures

Figure 1 - Content Resource Installation Details .....	7
Figure 2 - Resource Installation File Details .....	7
Figure 3 - User interface overview .....	13
Figure 4 - User interface details .....	14
Figure 5 - Application details .....	15
Figure 6 - Resource details .....	17
Figure 7 - Resource configuration details .....	18
Figure 8 - Key-value default data .....	19
Figure 9 - File default data .....	19

# 1. Introduction

---

## 1.1 Purpose

The purpose of this document is to provide detailed information how to use the Persistence Database Configuration tool. Building the tool is also part of this document

## 1.2 Support information

For inquiries there is a mailing list available and a bug tracker to file bugs.

- Mailing list:
  - <http://lists.genivi.org/mailman/listinfo/genivi-persistence>
- Bug tracker
  - [http://bugs.genivi.org/enter\\_bug.cgi](http://bugs.genivi.org/enter_bug.cgi)

## 1.3 Revision History

Revision	Change	Date
1.0	Initial creation	30.10.2015
1.1	Changed document license to CC BY-SA 4.0	16.01.2016

**Table 1 - Revision History**

## 1.4 Abbreviations & Terminology

Abbreviation	Description
RCT	Resource Configuration Table
PCL	Persistence Client Library
PAS	Persistence Administration Service

**Table 2 - Abbreviations & Terminology**

## 2. Unpacking/Installing

---

This chapter summarizes the information about where to find the git repository for the source code and how to build the tool.

### 2.1 Downloading Software & tools

To build the tool Eclipse for RCP & RAP Developers is needed. It is recommended to use version 4.5 or greater, which can be downloaded from the following location: <https://www.eclipse.org/downloads/>.

Java 8 RTE (run time environment) must be also installed, which is available on the following location: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Please make sure that JAVA RTE and Eclipse are matching e.g. 32-bit or 64-bit.

The source code is available in the following git repository:

<http://git.projects.genivi.org/?p=persistence/persistence-client-library.git;a=summary>

It is recommended to check out the tool from the git repository within Eclipse, which is described in the next section.

### 2.2 Setting up the environment

Normally git support is already available in Eclipse, but if not installed use Eclipse Marketplace to install this.

Checkout & Import 6 Eclipse projects (4 plugins & 2 features) by adding the git repository to Eclipse:

File → Import → Git → Projects from Git.

The wizard will guide you through the import.

### 2.3 Get the sources

The sources are hosted on GENIVI projects git repository.

Use the following command to get the GUI tool:

- `git clone ssh://git-genivi@git.projects.genivi.org/eclipse-json-gui`

## 2.4 Build the tool

After you have successfully imported the Eclipse project, open the “**platform.target**” in the **PersistenceConfigurationToolUI** project:

- Click "**Set as Target platform**" at the top right of the form
  - This may take some time to download necessary code

Now Open "**com.xse.optstack.persconftool.product**" in the **PersistenceConfigurationToolUI** project

- Open "**Eclipse Product export wizard**" in "**Exporting**" on the "**Overview**" page
- Insert a target directory OR archive file
  - Directory: Tool is exported into a directory and can directly be used
  - Archive: Tool is exported as an archive
- Select "**Export for multiple platforms**"
  - Deselect "**Generate p2 repository**"
- Click "**Next**"
- Select the needed platforms
- Click "**Finish**"

## 2.5 Running the tool

Now start the executable in the folder you have set above.



# 3. Persistence DB Configuration Tool

---

The Persistence Database Configuration tool is being used to provide a graphical tool to the system integrator in order to generate the JSON configuration files.

The JSON configuration files are used to:

- Create the default file / folder structure
- Create resource configuration table database
- Create databases for default data and configurable default data

The configuration files are used by the persistence administration service during the software loading (software update) process of the head unit.

## 3.1 Persistence Administration Service

One part of the Persistence Administration Service (PAS) responsibilities is to setup the Persistence subsystem.

This includes the following tasks:

- Create default application folders including links to shared data
- Deploy the default content
- Create local database for each application
- Create shared databases
- Provide application specific links to shared databases (group/ public)
- Setup of application file system access policies
- Delete, copy, backup and restore files (files and databases)

### 3.1.1 Installation Scope

The installation scope of the persistence administration service is similar to the internal Persistence file system structure.

The format for the installation should be flexible, which means the following should be considered:

- Installation of new application data
- Update/uninstall of application data (as whole)
- Install/update/uninstall individual resources e.g. install new key for an app, while existing keys are untouched
- Configuration of single or many apps
- Partial updates of resources using masks (key types)

## 3.2 Data organization

The following image shows the content of the resource installation file which is a dedicated compressed folder structure. It provides the JSON format configuration files for the PAS to setup the persistence data.

In the root folder of the resource installation - 3 folders represent the deployed persistence environment.

### 1. Apps

- installation data folders – one per application  
***installRules.json***  
list the applications to be affected by the installation for each application,  
the type of install is specified

### 2. Groups

- installation data folders - one per group  
follow the naming convention  
***installRules.json***  
list the groups to be affected by the installation for each group, the type of install is specified

### 3. Public

- public – installation data  
folder to maintain the same structure as for the installation data folders for applications and groups  
***installRules.json***  
specifies the type of installation for the public data

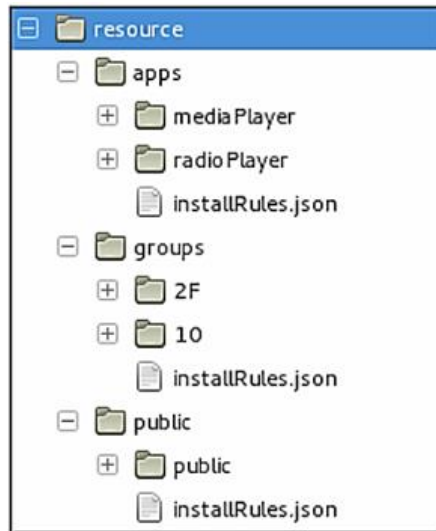


Figure 1 - Content Resource Installation Details

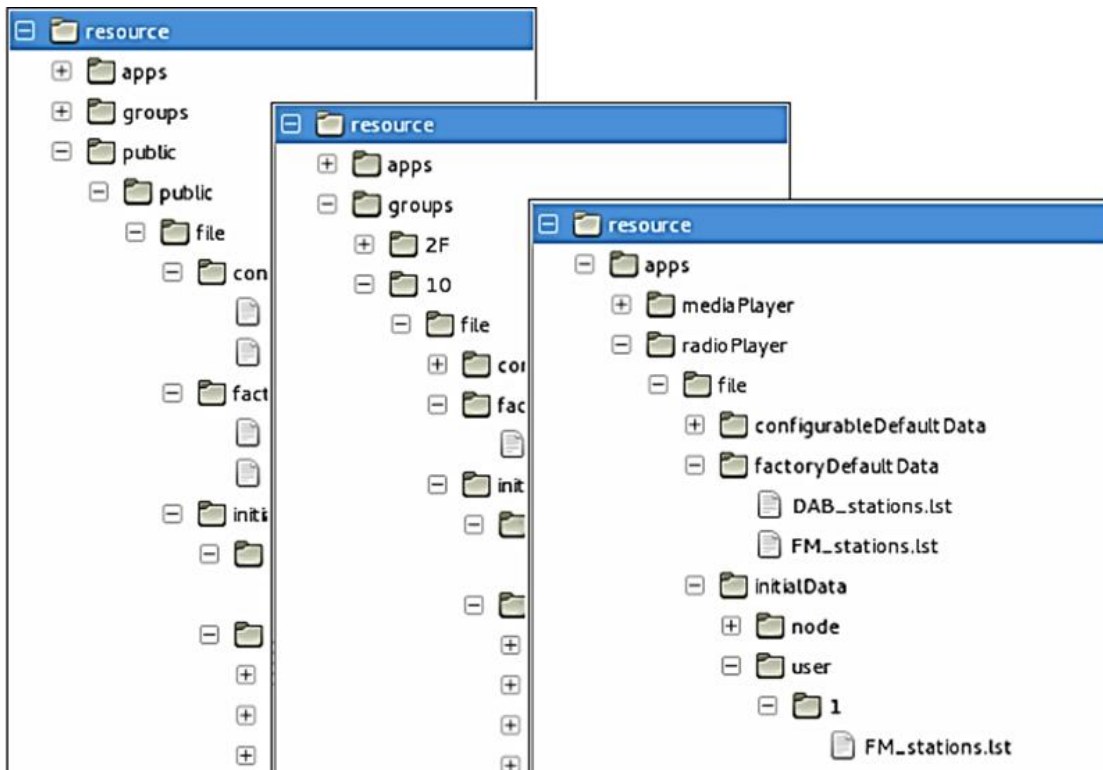


Figure 2 - Resource Installation File Details

## 3.3 JSON Configuration Files

Basically there are three different types of JSON configuration files available:

- Application installation configuration
- Resource configuration
- Default data configuration

Most of the JSON configuration files have a version information tag inside, but this is currently not used by persistence and will not be written via this tool.

There are also other JSON files used to provide default data for files and database resources as well as installation exception files. The format and content of the files will be described in the following sections.

For more details about the JSON file, please refer to the Persistence Administration User Manual (see 6. Appendix)

## 3.4 Application Installation Configuration

There are different types of installations available to install an application:

- New install
- Uninstall
- Update defaults
- Skip factory defaults
- Skip config defaults
- ...

### 3.4.1 JSON File Format

```
{  
    "[APPLICATIONNAME]":"[RULE]",  
    ...  
}
```

The JSON xml files will be generated by this tool.

### 3.4.2 Example

```
{  
    "Navigation":"PersAdminCfgInstallRules_NewInstall",  
    "AppX":"PersAdminCfgInstallRules_Uninstall",  
}
```

## 3.5 Resource Configuration Table Configuration

As the name says the database is responsible to configure the resource.

For the resource there are different configuration options available:

- Policy:
  - Cached:
    - resource will be cached when the value gets modified
  - Write through:
    - changed data will directly end up on the memory device
- Permission:
  - Read only:
    - resource data can only be read and not be modified
  - Read/write:
    - resource data can be modified
- Storage:
  - Local:
    - application local data (no other application is able to read this data)
  - Shared:
    - one “master” application is able to modify (write) the data and other application only can read the data
  - Custom type:
    - custom data type like e.g. secure, early, emergency
- Max Size:
  - The maximum size of the content
- Responsible:
  - The responsible application
  - Insert here the application ID of the “master” who is able to modify the data
- Custom Name:
  - If custom storage type
- Type:
  - Key
    - Resource will be stored as key-value item and accessed via the key-value API
  - file
    - Resource will be stored as a file and accessed via the file API
  - CustomID
    - Just an ID, could be used for special purposes when the resource item is configured as a custom resource.

## 3.5.1 JSON File Format

```
{ "config_appl" : "[APPLICATIONNAME]",
  "version" : "[VERSION]",
  "resources" : {
    "[ENTRYNAME]" : {
      "policy" : "[POLICY]",
      "permission" : "[PERMISSION]",
      "storage" : "[STORAGE]",
      "type": "[TYPE]",
      "max_size" : "[MAXSIZE]",
      "responsible" : "[APPLICATIONNAME]",
      "customPlugin" : [PLUGINNAME],
      "customID" : "[HASHVALUE]"
    }
  }
}
```

The JSON xml files will be generated by this tool.

## 3.5.2 Example

```
{ "config_appl": "Nav",
  "version": "0.1.0",
  "resources": {
    "pos/last_position": {
      "policy": "cached",
      "permission": "RW",
      "storage": "local",
      "max_size": "2048",
      "responsible": "Nav",
      "custom_name": "na",
      "type": "key",
      "customID": "edf1bc",
    }
  },
}
```

## 3.6 Default Data

There are two types of default data available, factory default data and configurable default data.

### 3.6.1 JSON File Format

```
{
"config_appl" : "[APPLICATIONNAME]",
"version" : "[VERSION]",
"resources" :
  {
    "[ENTRYNAME]" :
      {
        "size" : "[SIZE]",
        "data" : "[DATA]",
      }
  },
}
```

The JSON xml files will be generated by this tool.

### 3.6.2 Example

```
{
"config_appl" : "lt-persistence_client_library_test",
"version" : "0.1.0", "resources" : {
  "69" : {
    "size" : "5",
    "data" : "4546474849" },
  "status/open_document" : {
    "size" : "10",
    "data" : "57545f202f7661722f6f70"}
  },
}
```

## 3.7 Install Exception

Exceptions are defined in *installExceptions.json* in each app/group's install folder.

Affected data and allowed exception for each install rule		Default action on resource's data	Affected data				Valid exceptions		
			Factory default	Config default	Non default	Cached (not managed over RCT)	update	Don't touch	delete
Install rule per application or group	New-install	Update	X	X	X	X	-	-	-
	Un-install	Delete	X	X	X	X	-	-	-
	Don't touch	-	-	-	-	-	-	-	-
	Update-all	Update	X	X	X	-		X	
	Update-all-skip-default-factory	Update	-	X	X	-	-	X	-
	Update-all-skip-default-config	Update	X	-	X	-	-	X	-
	Update-all-skip-default-all	Update	-	-	X	-	-	X	-
	Update-default-factory	Update	X	-	-	-	-	X	-
	Update-default-config	Update	-	X	-	-	-	X	-
	Update-default-all	Update	X	X	-	-	-	X	-
	Update-set-of-resources	Exception	X	X	X	-	X	-	X
	Uninstall-non-default	delete	-	-	X	-	-	X	-

Table 3 - Affected data and allowed exception



# 4. Graphical User Interface

The persistence database configuration tool helps to easily create the JSON configuration files.

It provides the functionality to add and modify applications and resources including its default data.

The graphical tool is able to export the JSON configuration files as described in the previous sections, and it is able to read JSON configurations and display its content in the graphical tool.

The following example shows how the tool looks like.

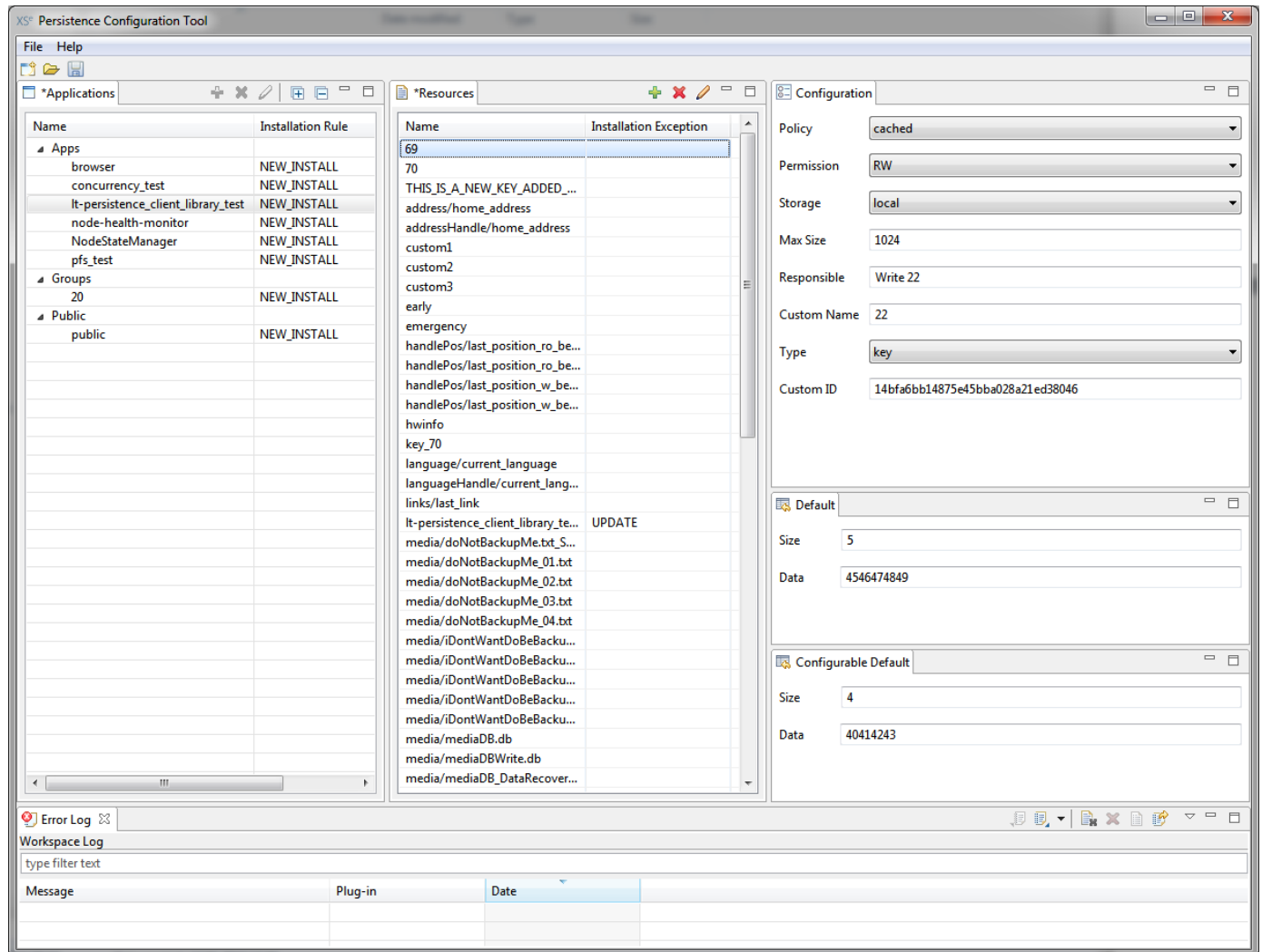


Figure 3 - User interface overview

# 4.1 User Interface Details

The user interface is divided into four different main areas.

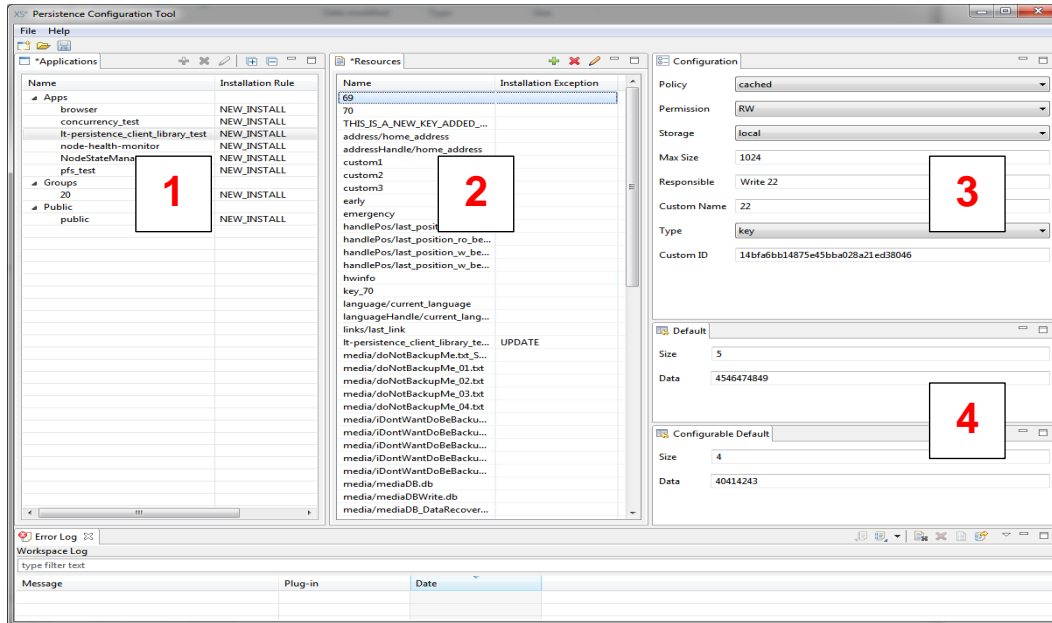


Figure 4 - User interface details

## 4.1.1 Area 1 – Application Area

The area 1 gives an overview about the different applications and groups available in the system as well as the public data. It is possible to add applications and groups by pressing the "+" Button.

When adding an application and groups a drop box is available to select predefined installation rules. Renaming and deleting of applications or groups is possible.

## 4.1.2 Area 2 – Resource Area

The area 2 gives an overview about the available resources for an application or group. The content of this area changes when selecting an application or group from the area 1. It is possible to add new resources or delete resources, as well as rename resources. When selecting a resource in area 2, the resource configuration items of the resource will be displayed in area 3.

## 4.1.3 Area 3 – Resource Configuration Area

The area 3 provides the details about the resource configuration of each resource.

The resource configuration provides a drop box where predefined values can be selected or an input field if variable data can be entered.

## 4.1.4 Area 4 - Default Data Area

The area 4 displays the factory default data and the configurable default data of the resource selected in area 2.

The data takes a string value as input and during generation it converts this string into a hex ASCII value. The size tag is fixed and will display the size according to the string length.

# 5. Tool Handling

## 5.1 Open an existing configuration

To open an existing configuration select in the “File” tab the entry “open” and choose a folder with an existing configuration. Make sure to select the folder which contains the resource folder, otherwise an error will be returned.

The persistence client library (PCL) test data can be used as an example. The configuration files can be found in the GENIVI PCL git repository in the following location:

- Repository: <http://git.projects.genivi.org/persistence/persistence-client-library.git>
- Location: `../test/data/PAS_data.tar.gz.`

## 5.2 Add an application

To add a resource you first need to add an application.

If the resource is an application local resource add the application under the Apps section.

To add shared data create the resource under a group in the Groups section of if the shared data is public add it under the Public section.

Now just click the “+” icon at the top of the application and enter a name for the application or the resource.

The name for the application must be unique in the system and this name will be used as application when using the Persistence Client Library API.

In the column “Installation Rule” a predefined installation rule can be selected.

For details about the installation rule please refer to section 3.3

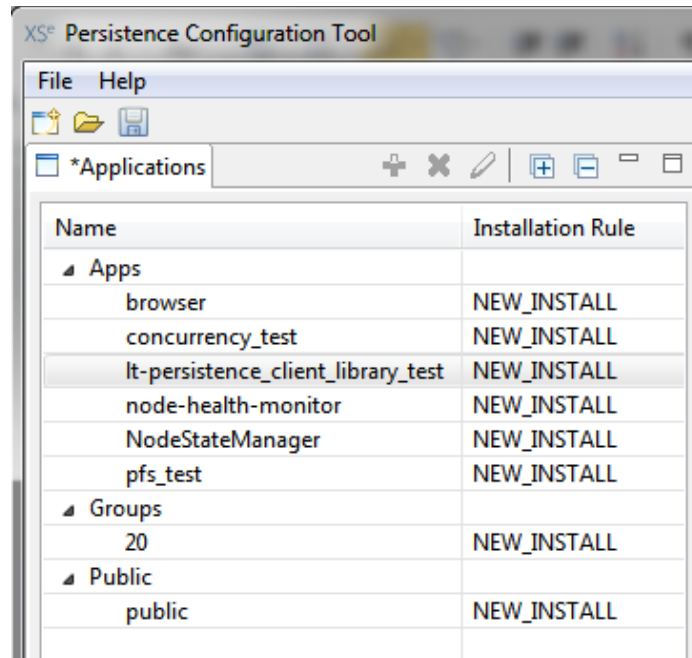


Figure 5 - Application details

## 5.2.1 Installation Rules

The installation rule allows the clear specification of what actions to be performed for each application or group.

- NewInstall
  - Application/group data is overwritten (eventual existent data is deleted)
- Uninstall
  - deletes application/group data  
Consistency must be assured (if an application has to be uninstalled, it has also to be removed from all the groups it was part of)
- DontTouch
  - Don't touch the application's or group's data
- UpdateAll (merge)
  - New resourceIDs are installed resourceIDs no longer available in the new RCT are uninstalled.  
Data for resourceIDs available in both old and new RCT is merged (new overwrite old data).  
Covers also the change of policy!!!
- UpdateAllSkipDefaultFactory
- UpdateAllSkipDefaultConfig
- UpdateAllSkipDefaultAll
- UpdateDefaultFactory
- UpdateDefaultConfig
- UpdateDefaultAll
- UninstallNonDefault
- UpdateSetOfResources
  - When only one or a few resources need to be updated

## 5.3 Add a resource

To add a resource navigate to the application or group in the Application area and select the application or the group.

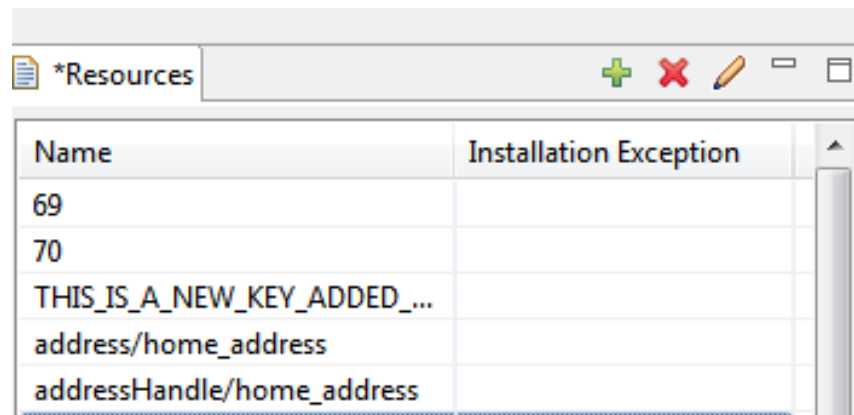
In the Resource section the already available resources of the application or group appear.

To add new resource, just click the green “+” icon in the top of the Resource area and enter a name of the resource.

To rename or delete a resource just click the corresponding icon in the top of the Resource area. Use the pencil for renaming a resource or the red “X” to delete the resource.

In the column “Install Exception” a predefined exception can be selected if necessary.

For details about the installation exception please refer to the next section.



Name	Installation Exception
69	
70	
THIS_IS_A_NEW_KEY_ADDED_...	
address/home_address	
addressHandle/home_address	

Figure 6 - Resource details

### 5.3.1 Install Exception

For each resource there is also an installation exception available. If a single resource shall be treated in a special way during the installation three options are available:

- Update
  - Update this single resource during installation process
- Don't touch
  - Don't do anything during installation process
- Delete
  - Delete this single resource during installation process

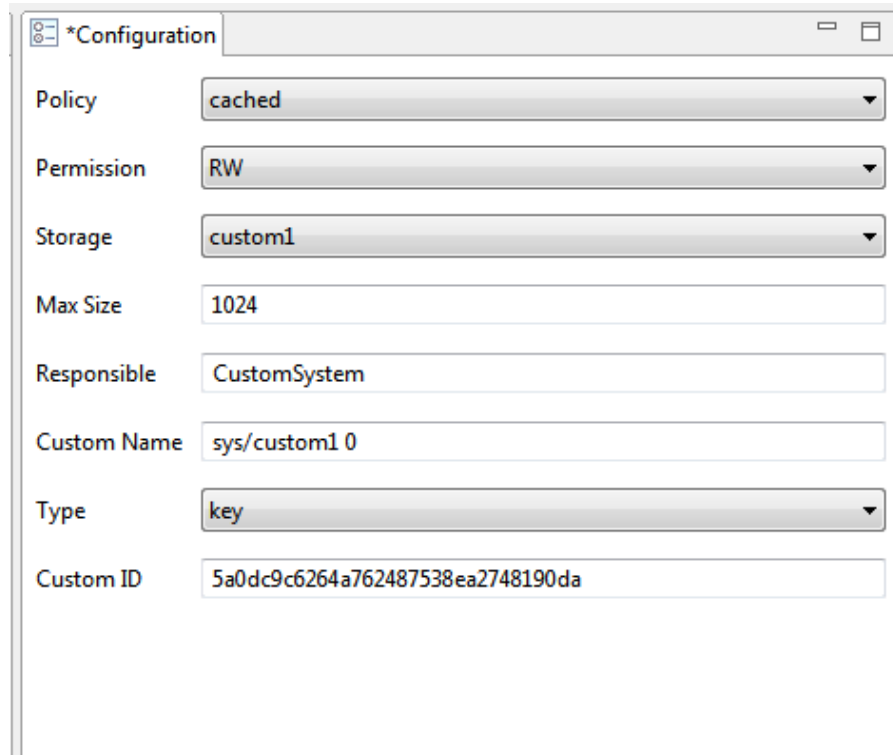
## 5.4 Resource Configuration

The following chapter contains information how the resource configuration is done.

For several of the configuration items predefined values are available via a dropdown list, just select the desired entry

For some of the configuration items a free text can be entered.

A description of the configuration items can be found in the chapter 3.5



The screenshot shows a configuration window titled '\*Configuration'. It contains the following fields and values:

Field	Value
Policy	cached
Permission	RW
Storage	custom1
Max Size	1024
Responsible	CustomSystem
Custom Name	sys/custom1 0
Type	key
Custom ID	5a0dc9c6264a762487538ea2748190da

Figure 7 - Resource configuration details

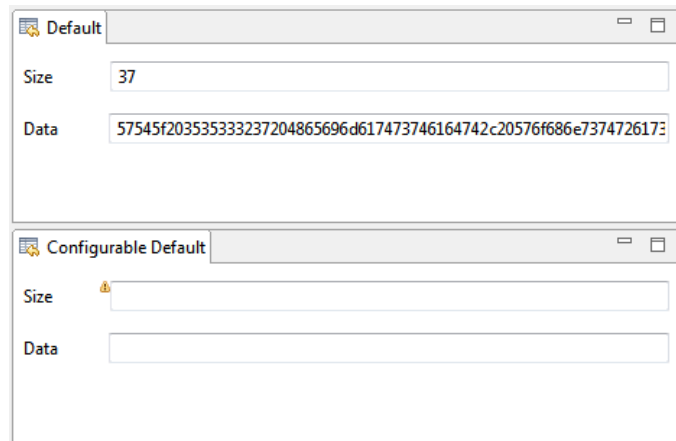
## 5.5 Add default data

In this area the default data and configurable default data can be added. Depending on the resource type (key-value or file), enter the binary data for key-value items directly or select a file containing the default data for files.

### 5.5.1 Key-value default data

The default data needs to be entered in hex numbers.

Use a text to hex converter if text will be stored, like <http://www.unit-conversion.info/texttools/hexadecimal/#data>

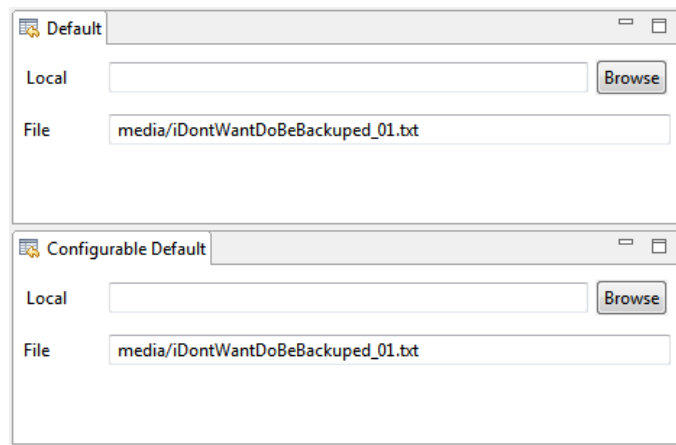


The image shows two panels from a software interface. The top panel, titled 'Default', has a 'Size' field with the value '37' and a 'Data' field containing a long hexadecimal string: '57545f203535333237204865696d617473746164742c20576f686e7374726173'. The bottom panel, titled 'Configurable Default', has empty 'Size' and 'Data' fields. A small warning icon is visible next to the 'Size' field in the 'Configurable Default' panel.

Figure 8 - Key-value default data

### 5.5.2 File default data

For default file data just select the file containing the default data.



The image shows two panels from a software interface. The top panel, titled 'Default', has a 'Local' field with an empty text box and a 'Browse' button to its right. Below it, the 'File' field contains the path 'media/iDontWantDoBeBackupid\_01.txt'. The bottom panel, titled 'Configurable Default', has a 'Local' field with an empty text box and a 'Browse' button to its right. Below it, the 'File' field contains the path 'media/iDontWantDoBeBackupid\_01.txt'.

Figure 9 - File default data

## 5.6 Generating JSON files

The JSON configuration files will be automatically generated every time the data will be saved. In the given storing location there will be two folders generated, a header folder and a resource folder.

The resource folder contains the JSON files used by the persistence administration service (PAS) to setup the file and folder structure as well as the different databases (resource configuration table and default databases).

Just create a tar archive form this folder and it can be used by the PAS.

Setting up persistence data:

```
persadmin_tool install /path_to_test_data/name_of_the_archive.tar.gz
```

The header folder contains different header files named according to the application names used in the GUI (application section).

The header files contain defines for the resource names which can be easily used in an application using PCL.

You just include the corresponding header and use defines for the resource ID instead of a string.



# 6. Appendix

---

Appendix 1 - Related Documents..... 21

## Appendix 1: Related Documents

Document	Description	Version	Link
Persistence Client Library User Manual	User Manual	2.4	<a href="http://docs.projects.genivi.org/persistence-client-library/1.0/GENIVI_Persistence_Client_Library_UserManual.pdf">http://docs.projects.genivi.org/persistence-client-library/1.0/GENIVI_Persistence_Client_Library_UserManual.pdf</a>
Persistence Administration User Manual	User Manual	1.1.0	<a href="http://git.projects.genivi.org/?p=persistence/persistence-administrator.git;a=blob_plain;f=Administrator/doc/GENIVI_Persistence_Admin_ComponentSpecification.pdf;hb=1b168f21d12c8ba36655546c400cfd961241092b">http://git.projects.genivi.org/?p=persistence/persistence-administrator.git;a=blob_plain;f=Administrator/doc/GENIVI_Persistence_Admin_ComponentSpecification.pdf;hb=1b168f21d12c8ba36655546c400cfd961241092b</a>
Persistence Architecture Documentation	Architecture Documentation	1.5	<a href="http://docs.projects.genivi.org/persistence-client-library/1.0/GENIVI_Persistence_ArchitectureDocumentation.pdf">http://docs.projects.genivi.org/persistence-client-library/1.0/GENIVI_Persistence_ArchitectureDocumentation.pdf</a>

### Appendix 1 - Related Documents