

Chapter 1

Look Inside

Package Manifest

Overview

1.1 Introduction

The GSA Package Manifest File Format is designed to allow any manufacturer's package files to be imported into an SMP (System Management Point) in a common and consistent way. The manifest contains the information necessary for an SMP to download a package to an end-point, such as an EGM, and install it.

The goals of GSA Package Manifest File Format are:

1. Allow all SMP implementations to support all package files in a single, consistent way.
2. Reduce manufacturer development costs by only having to implement the SMP package import functionality once and by only having to create manifest files one way.
3. Allow an SMP to determine which packages are applicable to each end-point and, thus, reduce field service costs due to packages being installed on the wrong end-point, improving the operator experience.
4. Reduce testing costs by limiting the possible permutations in the package manifest file to the smallest set that still supports the required functionality.
5. Maintain compatibility with the G2S and S2S protocol specifications in the areas of the Download, Storage, Hardware, and other classes.

1.2 Product

The manifest file format is oriented around products. A product is a typical “product” that an operator purchases from a manufacturer, which can be downloaded using G2S or S2S. Examples of products include game content, such as *Win Like a Wild Man*, peripheral firmware updates, and even operating system updates. This approach allows operators to focus on the end products they want to download and install, rather than the intermediary packages used to construct those products. Operators do not have to understand the mapping of packages to the products that they licensed; operators can focus on the products themselves.

The manifest file format contains fields that describe features of the product that are meaningful to the operator, such as:

- The type of the product (game, OS, peripheral firmware, etc.),
- Localized icons and product descriptions that can be displayed in a system user interface,
- Information about the manufacturer, and
- The steps required to install and uninstall the product.

If the product is a game, then information about that game can also be included, such as:

- The `themeId` and `paytableId` of the game, and
- The payback percentage of the paytable.

1.3 Install Information

The manifest file format includes information that describes the steps required for an SMP to install the product on an end-point. For each step, the manufacturer can specify the parameters that should be included in the commands used by the SMP to install the product.

The manifest file format also allows the manufacturer to identify any dependencies their product might have, such as existing modules, installed hardware, available storage, specific option settings, etc.

1.4 Uninstall Information

Optionally, the manifest file format may include information that describes the steps required for an SMP to uninstall the product. Like the install information, the manufacturer can specify the parameters that should be included in the commands used by the SMP to uninstall the product. Information to uninstall a product may not be present if the product can only be installed (for example, an OS upgrade).

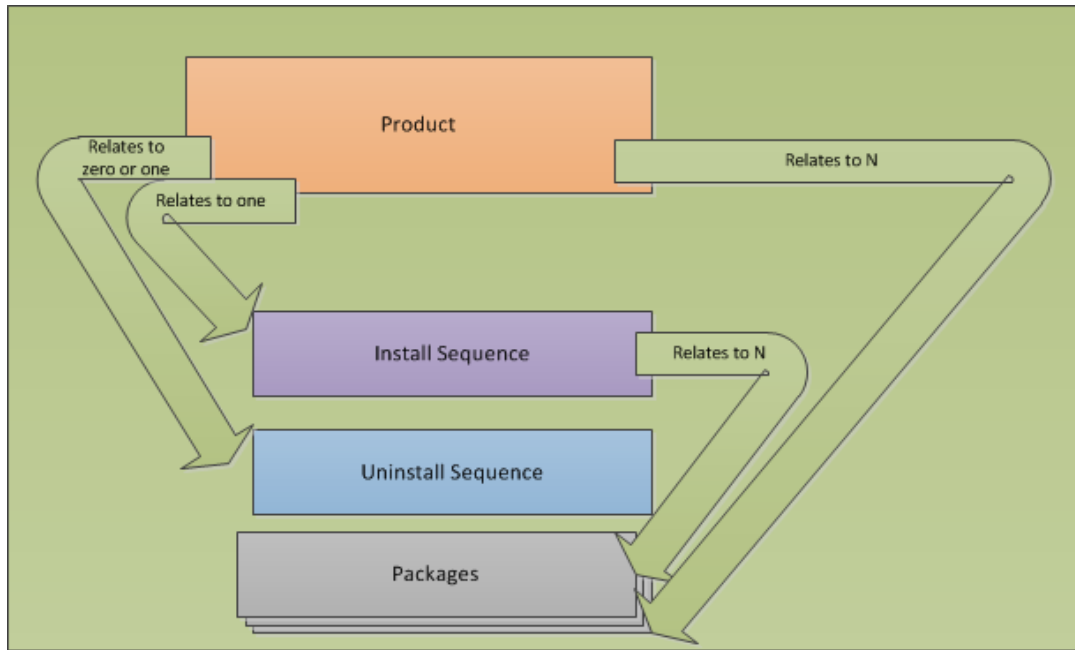
1.5 Package Information

Typically, the installation of a product will require that the packages, which contain the actual software, be downloaded to an end-point. Therefore, a package element is defined in the manifest file format that identifies various important properties of a package, such as the name of the file containing the package, the unique package identifier associated with the package, etc.

1.6 Relationship

The relationships between products, packages, install, and uninstall information is represented in the following diagram.

Figure 1.1 Relationships



1.7 Manifest File Extension

The manifest file itself MUST be written to storage media with a file extension set to “.gsaManifest”.

1.8 Product/Package Import

SMPs MUST be able to import manifests and their related files from storage media accessible to the SMPs. The storage media can be removable, fixed, or network accessible depending on the platform on which the SMP resides and its configuration.

Each manifest and its related files MUST be delivered in a single zip file. The manifest file itself MUST be located in the root directory of the zip file.

Starting at a specified directory location, an SMP should scan the directory and all subdirectories for zip files that contain a manifest file (that is, a file with the .gsaManifest file extension). How products defined in the manifest file are selected for import to the SMP is an SMP implementation decision.

One example of an import scenario is:

1. The operator inserts a CD/DVD disc from the manufacturer into a CD/DVD drive on the SMP.
2. The operator instructs the SMP to scan the CD/DVD drive for zip files containing a manifest file starting with the root directory of the CD/DVD.
3. The SMP discovers a manifest file and presents the products identified in the manifest to the operator as candidates for import.
4. The operator selects specific products for import into the SMP's repository.
5. The SMP imports the selected products along with their associated packages, install scripts, and uninstall scripts.

1.9 Scope

The scope of the GSA Package Manifest File Format includes:

1. Providing a standard method for importing any manufacturer's products and packages into an SMP.
2. Providing the information necessary for an SMP to perform a G2S or S2S download and installation of the products.
3. Providing package dependency information to an SMP that can be used to determine which packages and, thus, products can be installed on which end-points.
4. Providing the SMP with sufficient information to allow a user to understand what is in a product and how to filter products based on specific interests (for example, payback percentages) without having to first install the product on an end-point.