CA Client Automation

Packager and Installer for Windows Administration Guide

12.9

CA Technologies Product References

This documentation set references to the following CA products:

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA Asset Portfolio Management (CA APM)
- CA Business Intelligence
- CA Common Services™
- CA Desktop Migration Manager (CA DMM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Mobile Device Management (CA MDM)
- CA Network and Systems Management (CA NSM)
- CA Patch Manager
- CA Process Automation
- CA Service Desk Manager
- CA WorldView™

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Introduction to the Packager and Installer for Windows

This section contains the following topics:

Software Management Packager and Installer

CA Technologies Software Management solution consists of two main components, the Packager and the Installer. The Packager packages software and data into products. These products must be registered in the Software Package Library, from where you distribute and install them on target computers in your network. The Installer must be installed on the target computers, where it provides facilities for an unattended installation and removal of software products.

Product Packaging on Windows

During the packaging process on Windows, you install the product to be packaged on the reference system of the Packaging Computer, as if you were going to use the product on that computer. This reference installation is recorded automatically by the Packager in a number of archive files, which are used for a later installation of the product on target computers.

The reference system on the Packaging Computer includes the file system, installed services, and the registry of the Packaging Computer. Additionally, you can specify files, installation parameters, and other necessary changes to customize the product you are going to package (reference system configuration).

The Packager records all the information required for a later product installation on a target computer in a product in the SXP packaging format, named an SXP product. The Packager also provides all relevant files, directories, or information that is generated automatically during product packaging in a compressed format in the files*n*.cmp and ufiles*n*.cmp archive files (CMP archive files; the ufiles*n*.cmp file contains user-specific information).

To avoid large CMP archive file sizes, the Packager creates multiple CMP archive files of limited size and suffixes the names of these files with a sequence number, such as files1.cmp, files2.cmp, and so on.

You can create SXP products for Windows only on a Packaging Computer that runs a Microsoft Windows operating system later than Windows NT. (For more information, see "Supported Operating Environments" chapter in the *CA Client Automation Readme*, which is available as part of the CA Client Automation documentation set on the installation DVD.)

With the Packager you can create 64-bit application packages, provided the Packager runs on a 64-bit operating system. The 64-bit SXP product that is created can be installed only in a 64-bit operating environment.

Once you have created an SXP product, you can optionally convert it to an MSI product, that is, a product in the Microsoft Installer (MSI) format.

Note: MSI products that have been converted in a 64-bit operating environment have the platform type set to “x64” and cannot be installed on target computers running a 32-bit operating system, even if a 32-bit operating system is specified as a “Target OS” in the info.sxp archive file.

Packaging Computer Restrictions

Only the operating system software with service packs and the Packager for Windows must be installed on the Packaging Computer. If you require an application other than the operating system on the Packaging Computer, install that application on both the Packaging Computer and the target computers.

Violations of this rule can corrupt the installation of the SXP product on a target computer.

Before You Use Packager and Installer for Windows

Before you start creating and installing the SXP products, you must have knowledge about the following tasks:

- Configure the reference system of the Packager for Windows
- Use the Packager for Windows to generate SXP products

For information about these tasks, study the Reference System Configuration and Using the Packager for Windows chapters, respectively.

If you are an advanced user, read the following topics:

- SXP Product Conversion to MSI Product (see page 48)
- How the Installer Installs Products (see page 64)
- Installation of Converted MSI Products (see page 49)
- View the Results of Installation Jobs (see page 85)
- Trace Feature (see page 86)

Chapter 2: Reference System Configuration

This section contains the following topics:

About the Reference System on the Packaging Computer

The environment on the Packaging Computer is known as the reference system. The reference system is used during the packaging process on Windows to install the product to be packaged as if you were going to use it on that computer.

The reference system includes the file system, installed services, and the registry of the Packaging Computer. You can configure the reference system to customize the product you are going to package, that is, you can specify files, installation parameters, and can make other changes.

Aspects of Reference Installation Recording

In most cases of product packaging, the automatic recording using the default settings of the reference installation on the Packaging Computer meets all the needs. This so-called automatic method of packaging monitors all changes within the registry and the file system, and the factory settings after installation of the Packager accommodate most of the recording tasks.

In the following cases, adapt the settings to the environment of a particular recording.

Reference system performance

The more comprehensive the data on the reference system, the longer it takes to check for changes. For this reason, it can be useful to exclude certain drives from the check, particularly local drives not used for installation and network drives.

Changes not directly related to the installed product

On every computer, files and registry entries are changed continuously during runtime without relation to any software installation, for example, updates of the pagefile and temporarily stored Internet files. By default, these changes are not recorded. A list of excluded files is kept in the Packager and can be extended if needed.

The list of excluded files is automatically extended by files that change while a reference installation is not in progress. Most of these files are log files. To recognize most of these files, we recommend that you terminate the Packager and reboot the system after the Packager has been started for the first time and the Backup Files have been created.

In general, any directories in which temporary files are created (such as c:\temp) should not be recorded.

Protection of current system files

To ensure that the reference system remains consistent, system files are backed up, such as when a product introduces a different version of a Dynamic Link Library (DLL) file. When the product being created is installed, the original DLL file is overwritten. It is restored after the packaging process has completed. Otherwise, permanent changes are made to the reference system. By default, the Packager for Windows backs up the entire Windows directory.

This behavior can be switched off, if the Packager is used with a virtual machine. In this case, the Packager system can be restored by switching back to a virtual machine snapshot.

Partial changes made to INI and ASCII files

In general, if the product being created makes partial changes to existing ASCII files on the Packaging Computer, only those partial changes must be reproduced when the product is distributed to and installed on a target computer.

These ASCII files must be identified so that the reference system can handle them as special cases. Otherwise, under normal circumstances, the entire ASCII file would be added to the packaged version, and the file of the same name on the target computer would be replaced.

All INI files are monitored for partial changes.

Reference System Settings

You can modify the Packager's reference system settings by using the Configure Reference System dialog available from the File menu. Set the options on this dialog to maintain the reference system.

The tabs on the Configure Reference System dialog let you modify the following reference system settings:

- Paths to scan (see page 16)
- Paths to exclude (see page 18)
- Keys to scan (see page 17)
- Keys to exclude (see page 17)
- Expandable registry values (see page 19)
- ASCII files (see page 19)
- Defaults (see page 20)
- Backup files (see page 20)
- Product files filter (see page 21)

When you click OK on the Configure Reference System dialog, all changes on all tabs are applied. When you click Apply, only the values for the current tab are saved if they have been modified. After you click Apply, you cannot cancel the changes by clicking Cancel.

For some tabs, you can set a recursive option, represented by a folder symbol with the letter r. Gray folder symbols indicate defaults that cannot be changed. Yellow folder symbols contain either defaults or values you have already set that can be changed again if necessary.

Paths to Scan

Specify the paths that you want to be scanned on the Paths to scan tab of the Configure Reference System dialog.

If you do not specify a path on the Paths to scan tab, changes made to the path during the reference installation are not recorded, and the path is not reset to its original state after the reference installation has completed.

During a reference installation, the Packager records the following items:

- Files and directories that are added when the product is installed (additional files)
- Files and directories that are deleted when the product is installed (deleted files)
- Files that are modified when the product is installed (modified files), based on size and date

You can specify either an entire drive or a specific path. If you specify a path, only that path is checked. For example, if you specify c:, the entire c: drive is checked. If you specify c:\win, the c:\win directory and all of its subdirectories are checked.

To add another directory to the path list, use the Browse button, or enter the directory name directly in the Path field. Removing a folder from the path list does not affect the Paths to exclude tab.

To add or remove special folders from the paths list, use the Special Folder dialog. Removing a special folder from the path list automatically adds it to the list of available folders. Adding a special folder to the path list removes the folder from the list of available folders.

In general, additional and modified files and directories are copied completely to the product. The exception is ASCII files. For each ASCII file, you can specify that only the changes to the file are recorded. By default, changes to any file with the .ini extension are always recorded in this way.

For deleted files and directories, the Packager records that the marked files or directories are to be deleted on the target computer when the product is installed.

If existing files or directories are deleted or modified on the Packaging Computer's reference system while the product is being installed, they can be restored only if the following requirements are met:

- The files and directories are removed during the reference installation at the time indicated by the Packager dialogs.
- The files or directories were specified in the Backup Files tab and, therefore, were backed up.

Keys to Scan

Specify the keys that you want to be scanned on the Keys to scan tab of the Configure Reference System dialog.

During a reference installation, the Packager records the following items:

- Registry keys and values that are added when the product is installed (additional keys)
- Registry keys and values that are deleted when the product is installed (deleted keys)
- Registry values that are modified when the product is installed (modified values)

If you do not specify a key on the Keys to scan tab, changes made to the key during the reference installation are not recorded and, therefore, the key is not reset to its original state after the reference installation has completed.

During a reference installation, only these areas of the registry are checked. A key entry is interpreted recursively.

You can use only the two preset main keys. There is no need to specify other keys, because all other keys are subkeys of these two keys.

If keys and values are deleted from the registry while the product is being installed, these changes are reversed by the Packager at the end of the packaging process so that the reference system is not changed.

Keys to Exclude

During a reference installation, the Packager does not check for or record changes in the registry made to the areas (roots and keys) specified on the Keys to exclude tab of the Configure Reference System dialog.

If you specify a key on the Keys to exclude tab, changes made to the key during the reference installation are not recorded and, therefore, the key is not reset to its original state after the reference installation has completed.

The registry areas to be excluded are dynamic areas that are modified by Windows at run time. This dynamic modification at run time could corrupt the reference installation, unless the keys for these registry areas are excluded from the packaging process. By default, these registry keys are excluded from the packaging process, through the Keys to exclude tab.

Normally, you do not need to enter any keys on this dialog, because of the default settings. However, you may want to exclude additional keys, such as keys belonging to products that were not installed by the Packager for Windows or that may be installed in the future. To exclude the proper keys for these purposes, you must know how the registry works.

To exclude a registry area, in addition to the areas excluded by default, browse the registry to select a registry key to be excluded.

If you select the Recursively option, all keys and values under this key are also excluded from the check.

Paths to Exclude

Use the Paths to exclude tab to specify drives and paths that are not checked for changes during a reference installation.

If you specify a path on the Paths to exclude tab, changes made to the path during the reference installation are not recorded. However, the path is reset to its original state after the reference installation has completed.

Gray folder symbols indicate defaults that cannot be changed. Yellow folder symbols contain either defaults or values you have set that can be changed, if needed.

In the Path field, you enter the name of the path to be excluded.

If you select the Recursively option, all subdirectories of all paths you specify are excluded.

In the File mask field, you enter the name of a file or a file group to be excluded.

The following table includes examples of file masks:

Path	File Mask
C:\	cmos.ram
C:\	pagefile.sys
C:\	boot.ini
C:\tmp	trace.*
C:\test	*.*

Note: Some of the above examples are default entries set by the Packager; they cannot be modified.

Expandable Registry Values

The Expandable registry values tab lets you expand or overwrite values in the registry. You can use this tab to predefine which registry values can be expanded. Some of the values to be entered here are preset by the reference system, as shown in the following example:

[HKEY_LOCAL_MACHINE]\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\Path

Any data recorded in values specified here is prefixed or appended to the value on the target system.

Values specified on this tab must be in the range defined by the Keys to scan tab and Keys to exclude tab.

The values you specify must be of the REG_SZ (string) type, REG_EXPAND_SZ (expand string) type, or REG_MULTI_SZ.

If the registry values you specify on the Expandable Registry Values tab are expanded during a reference installation, the Packager records whether the original registry value was appended or prefixed.

You can identify the type of expansion (append or prefix) in the sreg.sxp or ureg.sxp file.

Partial Changes for ASCII Files

Use the ASCII files tab to specify ASCII files to monitor for partial changes. If a product being created makes partial changes to the existing ASCII files on the Packaging Computer, only those partial changes should be reproduced when the product is distributed to and installed on a target computer.

You must enter the path names of these ASCII files on the ASCII files tab to prevent the entire ASCII file from being added to the packaged version. Otherwise, the ASCII file of the same name on the target computer is replaced.

You can add nonexisting ASCII files to the list.

The Packager records changes for ASCII files specified on this tab in script files. These script files are referred to in the asc*nnnn*.ini file, which also contains the complete path of the ASCII file. When the product is installed on a target computer, these changes are reproduced on the target computer in the ASCII file of the same name. If the product is uninstalled from the target computer, the changes to these ASCII files are canceled.

Restoring Default Settings

You can use the Defaults tab to reset all the reference system settings to their factory settings.

When the Packager is installed, it sets default values for all tabs on the Configure Reference System dialog. However, when you create a product or configure the reference system, you can change one or more of the values on those tabs.

Using the Defaults tab of the Configure Reference System dialog, you can restore these values to their factory settings. The default settings apply to all products you later create.

Backup Files

The Backup files tab lets you select paths and files to protect by creating backups. The Backup tab is not displayed when the reference system reset phase is disabled (see Reference System Reset (see page 21)).

The first time the Packager is used, the drives, paths, and files you specify on the Backup files tab are backed up automatically, so that they can later be restored.

If you select the Recursively option, all subdirectories are also backed up in the archive.

If you make changes on this tab and click Apply or OK, the changes are automatically incorporated in the backup archive, and the Update Reference System dialog automatically appears.

Disk Space Considerations

Ensure that sufficient disk space is available on the Packaging Computer to store the backup file. The directories are backed up in compressed form.

ASCII Files Considerations

The files specified on the ASCII files tab do not need to be entered here because the reference system automatically backs up and restores these files.

Managing Product Files Using a Filter

Use the Product files filter tab to perform actions for a certain set of product files by specifying a product file mask.

For example, you can assign or delete file options for all files in the same group, that is, for all files with the same file extension; or you can remove a set of product files that are selected through a product file mask.

Reference System Reset

During the packaging process, the product installation on the reference system of the Packaging Computer is recorded. After package creation has completed, the reference system is by default reset to its original state.

In the Tools, Options menu of the Packager main menu, you can enable or disable resetting the reference system. By default, this option, named Enable Reset Reference System, is selected.

If the Enable Reset Reference System option is not selected (reset is disabled), all products that are installed on the reference system during package creation will remain there and are cumulated.

Disabling the reference system reset on the Packaging Computer can be of use when the Packager is installed on a virtual machine. In this case the reference system can be reset quickly by using a virtual snapshot.

Note: When reset is disabled, the Packager needs no backup files of the reference system. All backup-related tasks and GUI items, for example, the Backup files tab in the Configure Reference System dialog, are removed from the Packager's user interface.

Reference System Update

To protect the reference system from permanent changes caused by reference installations, the files of the reference system are backed up.

However, if you want to replace backed up files or paths with updated versions, for example, after installing a Windows service pack, you can use this function to update the reference system.

If you are working with a reference installation and have installed a service pack on the reference system, that service pack must also be installed on the target computers to ensure consistency.

Select either of the following methods for updating the reference system. Both methods require the Packager for Windows to be started first, and both generate a complete backup of all files.

Include new, deleted, and modified system files

To ensure new, deleted, and modified system files are included in the update, select File, Backup System Files from the Packager main menu. The resulting process completely updates the reference system.

Include additional files

To include additional files in the update, select File, Configure Reference System from the Packager main menu. On the Configure Reference System dialog select the Backup files tab and make your changes. When you click Apply or OK, the update is activated. The Packager automatically stores the list of backed up files (from the Backup files tab).

Note: As long as the backup is not created, the Packager reminds you to save your reference system.

File Options

During a reference installation, two installation or removal options apply to the recorded files:

- Version-dependent installation
- Reference counter-dependent uninstallation

You can assign one or both of these options to each individual file. Use the Product Files Filter tab on the Configure Reference System dialog to assign these options to groups of files that have the same path name and file extension.

On the Configure Reference System dialog the options are represented by the following icons:

Indicates version-dependent installation of the files

Indicates reference counter-dependent uninstallation of the files

Indicates that both options apply to the files

Version-Dependent Installation

The version-dependent file installation option ensures that a previously installed version of a file on the target computer is overwritten only, if it is older than the file to be installed. For this mechanism to work the file must have a version ID assigned. The Installer checks the internal version ID of the file during installation of the product on the target computer.

All product files with an internal version ID are automatically assigned the version-dependent installation option. All .dll and .exe files generally have a version ID.

To specify the version-dependent installation option, select the appropriate file path name on the Product Files Filter tab of the Configure Reference System dialog and click the version-dependency button.

Reference Counter-Dependent Uninstallation

The reference counter-dependent uninstallation option controls the multiple use of a file. When you apply this file option, a counter (reference counter) on the target computer is updated upon file installation and removal. If it does not already exist, the reference counter is created. The counter increases 1 each time the file is installed and decreases 1 each time the file is removed. The Installer deletes this file (and the reference counter) only, if the counter resets to zero during uninstallation.

For example, if a DLL file such as vbxxx.dll is part of an application program, this DLL file is installed for the first time when you install the application program. If you install any other application program that contains the same DLL file (vbxxx.dll), the reference counter for the file is incremented. When you remove any of the application programs that use the same DLL file, the reference counter is decremented. Normally, the file is deleted, however, when using the reference counter option, the file is retained to ensure that the other application programs that need vbxxx.dll are able to run.

The reference counter-dependent uninstallation file option is effective only, if it is specified for all products on the target computer that use this file. We recommend that you apply this option .

To select the reference counter-dependent uninstallation option, select the appropriate file path name on the Product Files Filter tab of the Configure Reference System dialog and click the reference counter-dependency button.

Important! Apply the reference counter-dependent uninstallation option only for files used by several application programs and only when necessary! Unnecessary counters increase the registry size and slow down installation. Moreover, this file option is effective only, if it is applied to all products installed on the target computer that contain this file.

Product Archive Configuration

The Packager for Windows manages products in a product archive. The product archive is organized hierarchically and is mapped to a directory tree. You can configure the root of this tree by selecting File, Open Product Archive.

On the appearing Open Product Archive dialog, you can enter the directory name of the new product archive you want to create, or you can enter the name of an existing product archive that you want to modify.

The default directory for the product archive of the reference system is known as SxpArchive and is on the drive where the Packager directory is installed:

Drive_of_Packager_directory\SxpArchive

Important! Do not create the product archive under the Packager directory, because it is deleted, if the Packager is uninstalled.

Product Archive on a Network Share

When the product archive resides on a network share and the Packager is rebooted, the Packager cannot access the product archive on some Windows platforms, even if the network connection has been reestablished by the DSM Explorer.

In this case, the Packager provides a user and password dialog so that you get access to the product archive again.

Chapter 3: Using the Packager for Windows

Use the Packager for Windows to create, modify, and register regular products, such as packaged software or data and parameter products, such as packaged parameters. Both types of products are created in the SXP packaging format and are referred to as SXP products. After an SXP product has been registered in the Software Package Library, it can be distributed to and installed on target computers.

This section contains the following topics:

Product Archive Window

When you start the Packager for Windows, the first window displayed is the Product Archive window.

The Product Archive window has an extended menu bar and toolbar and contains three panes, which display the following information:

- Archive tree
- General information
- Archive files

The archive tree pane, which is located on the left side of the window, displays all SXP products contained in the product archive, in a tree view. The first level of the tree represents the entire archive, the second level represents all products (product level), and the third level represents all versions (version level) of a product.

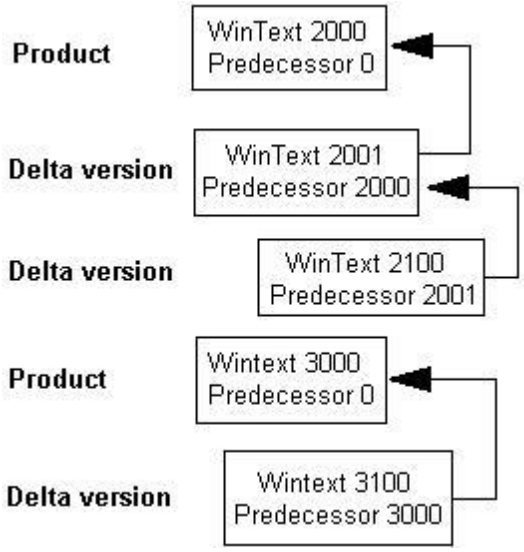
The general information pane, on the top right, displays the archive name and information about the currently selected SXP product, such as subject, version, predecessor, and size.

The archive files pane, on the lower right, lists all SXP files, script files, and compressed archive files (CMP files) for the selected product version. If the version has subdirectories, only the archive files contained in the currently active subdirectory are displayed.

Product Versions

Product versions are either complete or delta versions. A complete version does not have a predecessor. A delta version is an update, not a complete product. A delta version depends on a predecessor of the same product. This predecessor must have a lower version number than the delta version updating it.

The following illustration shows examples of complete and delta versions of an SXP product:



Packaging Methods

In Windows environments, software and data can be packaged using the automatic method, with the reference installation on the Packaging Computer or the manual method with manual SXP file creation.

Automatic Method

The automatic method is based on a reference installation on the dedicated Packaging Computer, which lets you install software on the Packaging Computer using the software's original installation program, such as setup.exe. It is the easiest method to use, because it automatically generates an SXP product and provides quick results.

When you perform the reference installation, you install the product on the Packaging Computer as if you were going to use it on that computer. All changes made by the product installation are recorded by the Packager in the SXP product. Any other manual changes, such as copying files, are also recorded.

For more information, see How to Package Products Using the Automatic Method (see page 30).

Manual Method

The manual method is designed primarily for products to install on the target computer in unattended mode with their original setup programs. These products must supply a response file with their setup program to enable unattended installation. For more information, see How to Package Products Using the Manual Method (see page 45).

Package Programs that Write Hardware Configuration Data

Some programs write hardware configuration data into the application's binary files. Generally, this information cannot be modified on the target computer.

Follow these steps:

- Package the program on a reference system that has the same hardware configuration as the target computer.
- Use the manual method of packaging, in which the original setup is executed on the target computer.

How to Package Products Using the Automatic Method

Use the automatic method to generate an SXP product. This method automatically creates most of the SXP archive files for a product and most of the required entries in those archive files. However, some features of the Packager and Installer for Windows can be used only if you create entries manually in the appropriate SXP archive files after the initial product has been generated.

Follow these steps:

1. Prepare to create a product version (see page 30).
2. Determine which type of product to create (see page 32).
3. Create a product version (see page 34), a combined or delta product, or a delta version (see page 39).
4. Specify enhancements to the automatic method (see page 41), if necessary.

After you have created your products, test them before registering them in the Software Package Library.

Prepare to Create a Product Version

Assign a product name and version in the product archive before you generate a product and register it. The product name and version is used to manage the product through software delivery functions.

Products are created using the Create Product Version dialog, where you enter the name and the version for the product. Optionally, you can specify the product name and version from the vendor by editing the info.sxp.

The Create Product Version dialog is the first in a series of dialogs for the packaging process. You can cancel the packaging process from any of these dialogs by clicking Cancel. If you cancel the packaging process, the Packaging Computer is automatically reset to the state it was in before you began creating the product version.

Select the Packaging Method

Follow these steps:

- If you want to create a product, or trying to create a product for the first time, then use the standard mode (automatic method). This method uses the reference installation and presents an absolute minimum of dialogs for required input.
- If you are an experienced user and want to change or enhance the configuration of the product or, for example, create delta and combined products, then use the custom mode (manual method). This method also uses the reference installation but provides additional dialogs to perform your tasks.
- If you want to create a product manually based on templates of SXP archive files, select the custom mode (manual method).

You select the packaging method on the Create Product Version dialog, which is the first in a series of dialogs for the packaging process.

Configure and Update the Reference System

When you use the automatic method, the Packager records all system changes resulting from the installation of the product. Therefore, the configuration of the reference system is important.

If you are creating the first product on the Packaging Computer and click OK on the Create Product Version dialog, the Packager automatically updates the reference system and displays the Updating Reference System message. This process can take several minutes, depending on the processing speed and the number of files on the computer.

To view the current reference system configuration and make any necessary changes, click the Configure Reference System button to open the Configure Reference System dialog.

System State Backup

Before the Packager begins the reference installation, it backs up the configured reference system (all files and configuration data), including ASCII files that are to have only partial changes made. The system state backup function is not performed when the reference system reset phase is disabled (see Reference System Reset (see page 21)).

When the backup is running, the Packager displays the processing status message:

Saving System State

Determine Which Type of Product to Create

Depending on the option you select from the Create Product Version dialog, create any of the following product types:

- Product (see page 32)
- Delta product (see page 33)
- Delta version (see page 33)
- Combined product (see page 34)

Product

A product is an application that runs independently and can be installed individually.

You can create a product if any of the following is true:

- The product you are packaging can be installed independently and run independently, without requiring any software other than the operating system. This is a product with no dependencies.
- The product you are packaging requires a previously packaged product for proper operation but not for installation. This is a product with dependencies.

  For example, suppose you have defined client parameters and created a parameter product. You must always install the parameter product on the target computers before the actual program can be properly run. Therefore, you must define a dependency for the packaged product on the parameter product.

  Another example is a desktop publishing program that requires specific fonts and printer drivers to produce the correct output. These fonts and printer drivers are not essential at installation time, but must later be installed on the users' computers and defined as dependencies.

  Dependencies are defined in the info.sxp archive file, after the initial packaging process has completed.
- The product you are packaging requires that certain software not be installed on the target computer. This is a product with an incompatible dependency.

  Specifying incompatible dependencies can be useful to account for incompatibilities between specific products, a company's hard disk restrictions, or other company policies.

  To use the incompatible dependency feature, enter a product name and version preceded by the exclamation point (!) operator in the #InternalDependence# section of the info.sxp archive file. The specified product must not be installed on the target computer if you want to install the product you just packaged.

Delta Product

A delta product is a complete product that requires one or several previously packaged products to be already installed on the target computer. A delta product always has a different name than the products it depends on.

For example, suppose you want to package an emulation program that can be installed on the Packaging Computer only if an SNA layer program has already been installed. However, you do not want to package and distribute the SNA layer together with the emulation program, because you want to be able to update each product independently.

In this case, you are creating a delta product for the emulation program. First you install the SNA layer program. Then you choose the delta mode. The Packager for Windows records only the changes from the emulation program into the emulation product, and not the changes from the SNA layer program.

If the prerequisite product has not been installed on the target computer when the Installer attempts to install the delta product, the Installer cancels the installation of the delta product and displays an error message.

Delta Version

A delta version is only an update. It is not a complete product. For a delta version to be installed correctly, a previous version of the same product, called a predecessor, must already be installed on the target computer.

If the predecessor product has not been installed on the target computer when the Installer attempts to install the delta version, the Installer automatically installs the predecessor first and then all delta versions.

A delta version can be created in two ways. The situation determines which procedure to use:

- You know exactly which files need to be updated or added to enable the newer version of the product to run. This may be the case, if you are a software developer, or if the software supplier has given you precise instructions. In this case, follow the instructions in Create Product Version.
- You do not know which files have been modified since the last version, there are too many files to track, or you want to change only the registry and not the file system. In this case, follow the instructions for creating a delta version in Create a Combined or Delta Product or a Delta Version (see page 39).

Combined Product

A combined product is a product that usually consists of two or more previously recorded products or delta products, such as all previous delta versions. A new product can also be installed, for example, when an update for an already packaged product is published

These products are packaged together so that they can be installed together on the target computers. The products can be one or more previously archived products and they can be either executable programs or data.

For example, two or more operating system service packs that were packaged separately and must be installed individually can be considered combined products.

Create Product Version

When you create a product version (with or without dependencies) or a delta version, follow these steps:

1. Install predecessor (if creating a delta version)
2. Install the software
3. Use the software
4. Make a list of file system changes (Custom Mode)
5. Specify install and delete options for files and directories (Custom Mode)
6. Specify enhancements to the automatic method (optional)

Important! Use this process to create a delta version only if you know exactly which files need to be updated or added in order for the newer version of the product to run.

Install the Software

After you click Continue on the Create Product Version dialog, the Installing Software dialog appears and the reference installation begins. All changes made to the reference system, beginning with those made from the Installing Software dialog to those made from the Using the Software dialog, are incorporated in the archive product.

During the reference installation, you may be asked to log off and on again, or to restart the computer. In such situations, the Packager for Windows ensures that, when the system restarts, it resumes the packaging process from the exact point at which it was interrupted.

Important! If you are going to package an MSI product you must ensure that all desired features of the MSI product are completely installed on the Packaging Computer before packaging the MSI product. No feature must be installed with the Installed on First Use option. This complete installation is necessary, as during the default installation of an MSI product, for example, Microsoft Office, some of its features are only published and not installed, and therefore cannot be packaged in the SXP product. To install the desired features completely, choose the MSI Custom Installation and, in the features tree, select the option to install all features on the local drive.

Use the Software

After you have completed the reference installation, close all windows except the Packager window, and click Continue on the Installing Software dialog. The Using the Software dialog appears.

You are prompted to start and close the software programs you have just installed. This action is necessary because many applications create their entries in the registry only when they are started for the first time. If these registry entries are not recorded now, with the reference installation, you will not be able to delete them later when the product is uninstalled on the target computer.

If you do not want the changes made by the programs at initial startup to be recorded, you do not need to start and close these programs.

Before you click Continue on the Using the Software dialog, you can still make changes to the product you are installing. For example, you can add or remove files; or insert parameters in product-specific files, as described in the topic Specify Enhancements to the Automatic Method.

When you click Continue, the Analyzing System dialog appears, and the Packager begins comparing the last saved system state before the reference installation with the current system state after the installation. The differences between these two states comprise the data from which the SXP archive files are created.

List of File System Changes

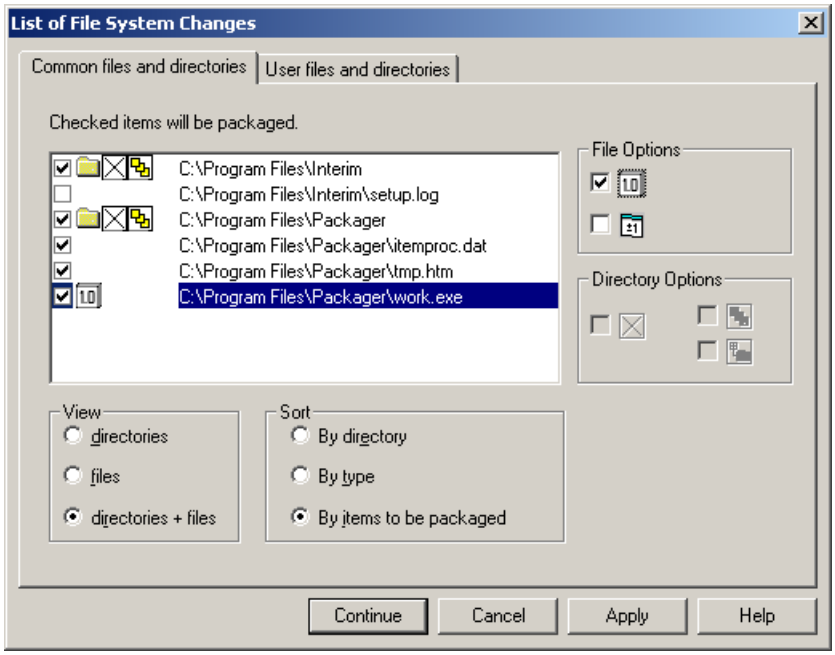
After all system information has been compared and the changes have been recorded, the Analyzing System dialog closes, and the List of File System Changes dialog appears.

All files that have been created or modified during the reference installation are displayed on the Common files and directories tab or the User files and directories tab.

The List of File System Changes dialog may display files that have been changed in the Paths to Scan area and also in the Paths to Exclude area of the Configure Reference System dialog. You may include files to the product or remove files.

Common Files and Directories

The following screenshot shows a sample Common files and directories tab:



The settings on the Common files and directories tab of the List of File System Changes dialog are applied to the product in the files.sxp and dirs.sxp archive files.

User Files and Directories

You can configure the user-specific files in the same way as the common product files, using the version-dependency button and the reference counter-dependency button. In addition, two options are available for user-specific product files, as follows:

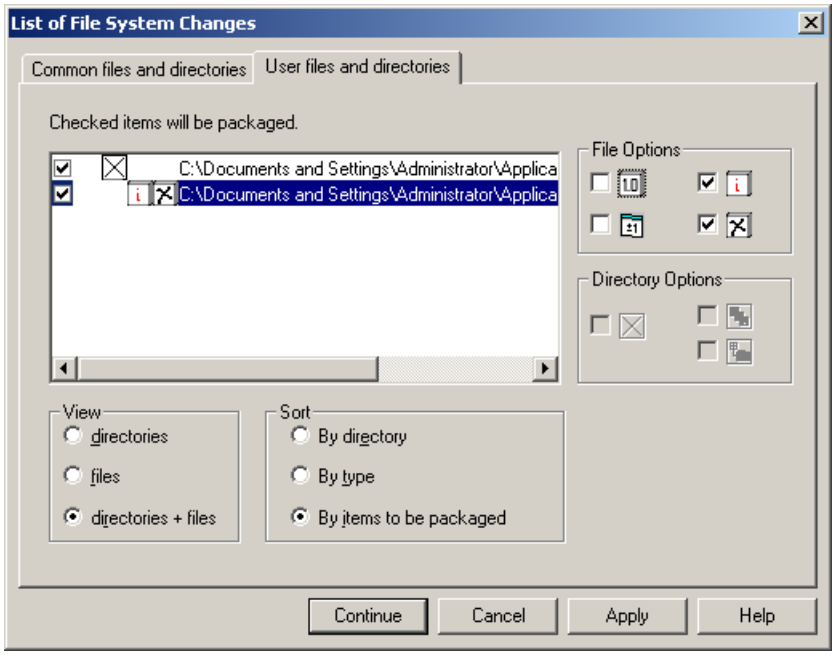
Install file only when file does not exist

This option protects user-specific files against overwriting by newly installed software. For example, if a user has already created a dictionary for office applications and a more recent office application is to be installed, this option ensures that the user's dictionary file is not overwritten by the application's default dictionary.

Do not remove file during uninstall

This option helps the user to keep a user-specific file (for example, a dictionary) on the target computer even if the application to which it belongs is uninstalled. The dictionary file may also be useful for other applications that remain on the target computer.

The following screenshot shows a sample User Files and Directories tab:



The settings on the User files and directories tab of the List of File System Changes dialog are applied to the product in the ufiles.sxp and udirs.sxp archive files.

Specify Install and Delete Options for Files

On the List of File System Changes you can specify install and delete options for each file on the target computer, as follows:

- Select the check box next to the version-dependency button to assign the version-dependent installation option (see page 23).
- Select the check box next to the reference counter-dependency button to assign the reference counter-dependent uninstallation option (see page 24).

Specify Install and Delete Options for Directories

New directories created during the reference installation, except user-specific directories, are displayed on the List of File System Changes dialog.

Root Directories

Root directories are new folders that are not sub-folders of another new folder. The Installer on the target computer installs the product in the specified root directories.

A root directory is specified in the SXP archive files by the internal parameter SxpRootDir*n*, where *n* is the number of the root directory.

Example 1

After the packaging process a parameter is specified, instead of the fixed path, for the root directory on the target computer:

```
SxpRootDir1=E:\GraphicWorkshop
```

Example 2

The ProductXRoot parameter is a client parameter that must be configured individually for the target computer.

```
SxpRootDir1=$(ProductXRoot)
```

Options for Deleting Directories

Directory delete options are assigned to each directory. Each delete option is represented by a different symbol. You can individually change the delete option for each directory.

You must uncheck the delete folder symbol (directory is not deleted) for each multi-use directory and all of its higher-level directories. A multi-use directory is a directory used by two or more applications.

Create a Combined or Delta Product or a Delta Version

To create a combined or delta product or a delta version perform the following steps:

1. Select the Custom mode using reference installation method on the Create Product Version dialog.
2. Specify a parameter set (optional)
3. Install previously created products
4. Select the product type to create a combined or delta product or a delta version

This is the standard method to create a delta version. Using this method you need not know which changes have been made since the last version.

Specify a Parameter Set

If a product to be installed in the step "Install previously created products" uses client specified parameters, these external parameters are substituted by one of the following methods:

- The default parameter set is used. The default parameter set is defined for every product in the product archive file sxpparam.ini. Use the function Check Parameters in the Product Archive to create this file.
- An individual parameter set is defined for the packaging system. You can edit this set using the Client Parameters editor for the computer object Packager. These entries will take precedence over the default set defined in sxpparam.ini.

The temporary file param.ini is recreated every time by the Packager and cannot be used for substituting parameters manually. Use the Client Parameter editor for the Packager PC instead.

Install Previously Created Products

After you click Custom mode using reference installation on the Create Product Version dialog, the Install Previously Created Products dialog appears. This dialog lets you install the previously archived products that are required to create a combined product, a delta product, or a delta version on your reference system.

If you select an archived version of a delta product, the Packager automatically installs the version you selected including all predecessor versions, and the product on which it is based.

Important! Manual entries that you make in the SXP archive files of the archived products (details on user rights, pre- or post-programs, and so on) are not included in the product to be packaged.

Select a version from the list and click Install. You can install any of the versions listed. However, before continuing, you must account for any dependencies the version may have, on any other products that are included in the list. For example, if the version you want to install requires a software product named XYZ 5000 as an installation prerequisite, you must make sure that XYZ 5000 is installed on the reference system before continuing.

During the installation of the archive version, you may be asked to log off and on again or to restart the computer, if changes have been made to the system. In this situation, the Packager for Windows ensures that when the system restarts, it resumes the packaging process from the exact point at which it was stopped.

Important! When previously archived products are installed, the original.sxp archive file is not evaluated, the pre- and post-programs are not executed, and an installation with the original setup file is not performed.

Select Type of Product to Create

After the archived products have been installed, the Product Type dialog appears, prompting you to choose whether you want to create a combined product or a delta product.

combined product

If you select Combined product, the Installing Software dialog appears. The products that you are going to install now are packaged in one SXP product, together with the already installed archived product versions.

delta product

In case of delta products the Analyzing System dialog field is shown, while the Packager is comparing the system state before the reference installation with the current system state after the installation of the archived versions. This data is used later on to reset to the initial state.

The Packager now backs up the configured reference system a second time, so that only the differences from this second backup is later used for creating the SXP archive files. During this time, the system displays the processing status message: Saving System State

After this, the Installing Software dialog appears to install the complete product. The system is analyzed again, comparing the previous and the current system state.

Finally the Specify Install and Delete Options for Directories dialog appears. Click Continue to complete the reference installation. When the reference system reset phase is enabled (see Reference System Reset (see page 21)), all changes to the reference system are reverted.

You can register the product to the Software Package Library or convert the product into an MSI product.

Specify Enhancements to the Automatic Method

When you are creating a product, a combined product, a delta product, or a delta version, you can modify the archive files after you have finished the packaging process.

If you discover that the reference system settings do not meet your needs, you can reconfigure the reference system and create the product again.

Edit Archive Files

To open and edit an archive file, double-click the icon of the archive file in the Product Archive window, or right-click the icon and select Edit Archive Files from the context menu.

When you double-click the icon of an archive file in the Product Archive window, an archive file editor window appears displaying the contents of the file.

You can select the section you want to edit from the Section list box, if available. The current parameters are listed under Contents. For some archive files you must add or edit the content directly in the input area. When the editor window is open, the Packager main menu is adapted and provides editing functions for the archive file on the Edit menu and functions for saving and closing the archive file on the File menu.

A short syntax description of the section entries can be found in the footer of the editor window. If you want more detailed information about the structure and syntax of the archive file, open the Packager's online help and click the file name in the Archive Files Overview help topic.

If you double-click a CMP archive file (files*n*.cmp or ufiles*n*.cmp), the built-in CMP Editor launches and provides all functions you need to modify the compressed files. To edit the CMP archive files, use the CMP Editor only, because this ensures the correct handling of special characters independent of the used codepage.

The majority of SXP files are created automatically and assigned data by the reference installation. Therefore, it is not necessary to edit all SXP archive files and all sections of the SXP archive files.

The following topics provide basic descriptions for manual entries you can generate.

- Dependency Specification (see page 42)
- Use of Pre- and Post-Programs (see page 42)
- ASCII File Modification (see page 43)
- Use of Registry and Desktop (see page 43)
- Parameter Creation (see page 43)

Dependency Specification

Any dependency that exists between archive products must be entered in the info.sxp file.

Internal dependencies are product dependencies on the same target computer.

External dependencies refer to products that are installed on another computer, such as client-server applications.

Use of Pre- and Post-Programs

To use the functionality of pre- and post-programs, you must create the actions.sxp file manually, using the Create archive file method. You can add the pre- and post-programs to the product by means of the Import archive file method.

If you use parameters in the pre- and post-programs, you must list the files containing parameters in the #ReplaceParams# or #ReplaceParamsUNICODE# sections.

ASCII File Modification

The reference installation creates an SXP archive file, asc*nnnn*.sxp, for every ASCII file that is specified during the configuration of the reference system, and that is changed during the reference installation.

Two scripts, named asc*nnnn*.ins and asc*nnnn*.dei, are created. These scripts modify the ASCII file during installation and removal of the product, respectively.

The only exceptions are the autoexec.bat and config.sys files. A different procedure applies for these files.

How to Handle Registry and Desktop Changes

If a product makes user-specific or group-specific entries in the registry or modifies the desktop, the ureg.sxp and ulinks.sxp archive files are created automatically with the reference installation.

By default, entries are created in these files, making the product accessible for all users of the target computer. Therefore, if you want to restrict one or more users from accessing the product on the target computer, you must edit the ureg.sxp and ulinks.sxp archive files.

If you start operating system programs, such as notepad.exe, during the reference installation, these programs may change the registry. These registry changes are recorded in the ureg.sxp file. You should delete these entries in the ureg.sxp archive file before completing the SXP product version.

If you need to restrict access rights for files and keys in the registry, you must manually create the permis.sxp archive file.

Parameter Creation

Parameters can be used in archive files. You can specify either internal or external parameters. Internal parameters are replaced by values that are specified internally by the packaged program or the operating system. External parameters are replaced by values that are specified externally in files that are created during the installation of an SXP parameter product on the target computer.

Each parameter is replaced by the values in the following sequence:

1. The Installer checks for the parameter in the internal parameters.
2. If the parameter is not found there, the Installer checks for it in the external parameters, which are located in special parameter files.
3. If the parameter is not found there, the Installer checks for it in the default parameter set included in the package.

The sequence in which the parameter files are searched corresponds to the sequence in which these files are specified in the Installer's configuration file.

The Installer checks the following archive files to see if they contain parameters and replaces the parameter values automatically:

- All SXP archive files
- All scripts specified in original.sxp and asc*nnnn*.sxp files
- All archive files specified in the #ReplaceParams# section of the files.sxp file. These archive files must be stored in the files.cmp file.
- All files specified in the #ReplaceParams# and #ReplaceParamsUNICODE# sections of the actions.sxp file. These files must not be stored in the files.cmp file.

Replacing SXP Parameters in UNICODE Files

To replace SXP parameters in UNICODE files external to the SXP package, you must specify the UNICODE file names in the #ReplaceParamsUNICODE# section of the files.sxp archive file.

To edit the files.sxp archive file, double-click the files.sxp icon in the Packager GUI. The files.sxp editor opens where you can select the #ReplaceParamsUNICODE# section from the Section drop-down list.

Create and Use Scripts to Modify Text Files

You can write script files that contain instructions for modifying text files. The Packager uses the scripts to adapt ASCII files for specific target computers.

In your scripts, you can use parameters as variables for values specific to a target computer.

To write a script for use with a product, you must use the Packager scripting language. You will find the description of the Packager scripting language in the *Software Management Packager Help*.

How to Package Products Using the Manual Method

The manual method for packaging products is designed primarily for products that are to be installed in unattended mode on the target computer, using their original setup programs. Such a product must supply a response file with its setup program to enable unattended installation.

After the product has been distributed to a target computer and the installation has begun, the Installer executes the product's original setup or installation program. The manual method requires detailed knowledge of how to respond to the installation program's prompts and also how the response file is structured.

The process of creating a product with the manual method includes the following tasks:

1. Create a product version and select Custom mode, Create templates to be edited manually (see page 45).
2. Specify the archive files to be created (see page 45).
3. Perform additional tasks to create archive files (see page 46).

After you have packaged the products, you can register them in the Software Package Library.

Create a Product Version and Select Custom Mode Using Templates

Use the Create Product Version dialog to specify the name and version of the product and select a packaging method.

To select the packaging method

1. Click Custom mode
2. Click Create Templates to be Edited Manually

   The Create Templates for Archive Files dialog appears.
3. Select the SXP archive files you want to create and include in the product.

Specify Archive Files to Create

The manual method creates empty files that must be populated after the initial packaging process has completed. You can complete these files using the Packager's editor. To start the editor, right-click the archive file and select Edit Archive Files.

All SXP archive files can also be created manually after the packaging process has completed, using the main Product Archive window.

Perform Additional Tasks to Create Archive Files

Up to this point, you have only created empty archive files. The following tasks complete the archive file creation:

- Copy the original installation media
- Write a script to modify the response file locally
- Modify the original.sxp file

Copy the Original Installation Media

Create a subdirectory under the version directory and copy the original installation media (DVD) into it.

Use a Script to Modify the Response File Locally

Optionally, you can write a script to modify the response file locally during installation on the target computer.

The script must contain parameters that allow the response file to be modified for each target computer. Copy the script into the version directory, or a subdirectory, on the Packaging Computer.

To write a script for use with a product, you must use the Packager scripting language. You will find the description of the Packager scripting language in the *Software Management Packager Help*.

Modify the Original.sxp File

In the original.sxp file you must modify the #ResponseFiles#, #ScriptFiles#, and #Actions# sections, as follows:

#ResponseFiles# and #ScriptFiles# sections

In the #ResponseFiles# section, enter the path name of the response files you are using. In the #ScriptFiles# section, enter the path name of the script file that modifies the response file.

If you are using more than one response file, list the path names of the script files in the #ScriptFiles# section in the same sequence as the path names of the associated response files listed in the #ResponseFiles# section.

All path names must begin with the internal $(SxpSrvRelDir) parameter. On the target computer, this parameter contains the path name for the archive files.

#Actions# section

Enter the installation and removal instructions for the product. Specify the path name of the original setup program for unattended installation and removal, as shown in the following example:

Install = $(SxpSrvRelDir)\setup.exe –s –SMS

Uninstall = $(SxpSrvRelDir)\setup.exe -uninstall

Prerequisites for SXP Packager on Windows 8 and 8.1

If you tried creating packages using SXP Packager on the computer, verify that.NET framework 3.5 is installed and enabled on the Windows 8 or Windows Server 2012 computer using the following steps:

Windows 8 and 8.1:

Click Control Panel, Programs, and Features, Turn Windows features on or off. For more information, see http://msdn.microsoft.com/en-us/library/hh506443.aspx.

Windows Server 2012 and Windows Server 2012 R2:

Install .NET Framework 3.5 using the Add Roles and Features Wizard. For more information, see http://technet.microsoft.com/en-us/library/hh83180.aspx.

Product Deletion

You can delete a product, including all of its versions, or you can delete a specific version of a product.

When you delete a product or version, you remove it from the product archive on the Packaging Computer. This deletion does not remove the product or version from the target computer or from the Software Package Library.

SXP Product Conversion to MSI Product

After you have completed packaging an SXP product using the automatic method, you can optionally convert it to a software product in the Microsoft Installer (MSI) packaging format. The converted MSI product is stored in the product archive with the same version number expanded by .msi.

Converted MSI products are registered in the Software Package Library and installed on target computers, as SXP products are.

During the SXP to MSI conversion, all files are added to a single .cab file in the MSI package. The maximum size of a single .cab file is 2 GB. This means that SXP packages larger than 2 GB cannot be converted by the SXP to MSI Converter.

During installation of an MSI product the Microsoft Installer sets properties. Default values for the properties are predefined in the MSI product in the Property table. Each package contains a minimum set of properties. Some of these properties are reserved and must not be changed, for instance, ProductCode, whereas others may be customized, for instance, REBOOTPROMPT.

When converting an SXP product to an MSI product, SXP parameters are mapped to MSI properties. For each external (client) parameter found in the SXP product, a public property with a default value is created in the property table. The values are taken from the sxpparam.ini file, if an appropriate entry exists. Otherwise, the property value is set to Undefined, and the user must specify a default value. Some internal parameters are not automatically mapped to MSI properties and therefore must be edited by the user.

Because the MSI format does not support all SXP features, there are restrictions for the SXP to MSI conversion. For example, in SXP formatted files any user-specific item can be configured for installation for one or multiple users or groups, and flexible assignment of users and groups is provided. Because this behavior is not supported by MSI, any user-specific object will be installed for all users.

Any problems during conversion, such as restrictions detected, are listed in the file Comment.txt, which is created in the MSI product directory. In the terminating message the user is informed about the entries in the Comment.txt.

SXP Product Version Registration

After you have packaged software or data, you must register the SXP product in the Software Package Library of a manager system. You can browse for an available manager system on the Register a Version of a Product dialog.

You can select the security provider from the drop-down list on the Register a Version of a Product dialog

When you register a version, all files and subdirectories in the version directory you select on the Packaging Computer are transferred to the Software Package Library and made available for distribution.

Important! Ensure that you have the permission to register software in the Software Package Library.

Installation of Converted MSI Products

Converted MSI products are registered in the Software Package Library and installed on target computers similarly to other MSI products. However, it is required that the Software Delivery agent must be installed on the target computer, before a converted MSI product can be installed.

The following considerations apply to the installation of converted MSI products:

- Converted MSI products must be installed on the machine agent only. They are then available for all users of that target computer.
- Converted MSI products can be advertised, provided a nonuser-specific link exists in the product. When a user triggers the installation of an advertised MSI product, the product will be installed for all users.
- During installation of an MSI package, the Microsoft Installer sets properties. Default values for the properties are predefined in the MSI package in the Property table. If a parameter product is installed on the machine, these default values are overwritten by the client parameter values set on this target machine.
- During installation or removal of converted MSI packages, the Windows Installer logging is active. The output is provided in the job output file. In addition, the SXP Installer's trace mechanism is used.
- The installed MSI product has no entry in the Add/Remove Programs option of the Windows Control Panel. The product can be managed only through software delivery functions, and not by single users from their desktops.

Parameter Archive

All client parameters (see page 50) (computer-specific parameter settings) are stored in four .ini files (see page 58) that are collectively known as the parameter archive.

Client Parameters

Client parameters are SXP products. They are created and saved on the Packaging Computer and then registered in the Software Package Library and distributed to the target computers.

Client parameters customize installations. They are wildcards or placeholders for values that identify a specific target computer. Examples include the IP address and the installation path for a specific target computer. During the installation on a target computer, client parameters are replaced by the values defined in both internal (see page 50) and external (see page 51) parameters. These values can also be retrieved from environment variables and registry keys.

Client parameters, whether they are defined for an individual computer or a group of computers, are always defined for computers and not for users.

You can edit client parameters using the client parameter editor (see page 52).

Internal Parameters

Internal parameters are reserved parameters that are generated automatically by the Packager during the packaging process. All internal parameters have the reserved prefix Sxp in their names.

SxpRootDir is a special internal parameter. It generates the installation paths of a product. This parameter is specified in the info.sxp archive file. If necessary, you can change the installation paths defined in this file.

All other internal parameters are assigned values during installation on the target computers. These other parameters cannot be changed or edited. They represent the Windows programs folder, the windows directory, or any other well known Windows path.

External Parameters

External parameters are defined after the initial packaging process has completed.

External parameters are optional but can be useful in customized installations. If you want to use external parameters, package and register them before distributing them to the target computers.

If you are specifying an external parameter, never start its name with the letters Sxp, as this prefix is reserved for the names of internal parameters.

Many administrators work with one parameter product that stores the parameters of all products. However, you can define several different versions of parameter products, if needed.

Note: If an SXP product has an internal dependency on a parameter product, the parameter product must be installed on the target computers before the SXP product is installed.

Enter and Edit Client Parameters

You can use the following methods to enter parameters for all target computers:

- Enter all required parameters under the default icon. To do this, select the default object in the right pane, and then create the individual entries, using either of the following ways:
  - From the Packager main menu, select Edit, Parameter, Add
  - Right-click the default object to display the shortcut menu and select Add parameter.
- Move PC objects into the same group when you want to set the same parameter settings for all of them at one time. After you move the PCs to the same group, you can set the necessary group parameters (see Group Objects (see page 55)).
- Move PCs that have or will have PC-specific parameters to the bottom of the archive tree. Select and set the values for each one individually (see PC Objects (see page 57)).

You must specify parameters for the default object before you specify them for the Group objects or PC objects. Therefore, you must first specify the type, range of values, default value, and description for each parameter in the default object. You can then specify parameter values for either Group objects or PC objects.

Information about naming standards can be found under Parameter Names (see page 52).

If a large number of parameters need to be entered, you can use tools for automatic parameter generation (see page 52).

Parameter Names

Parameter names should be unique on each target computer. For example, when assigning a parameter for the installation directory of a product being installed, specify a name such as *productname*.InstallDir, not simply InstallDir.

However, if you want a new parameter value distributed to the target computer to overwrite the previous value, you should use the same parameter name.

To reference these parameter values in archive files and scripts, you must enter them in the following format:
$(*parametername*)

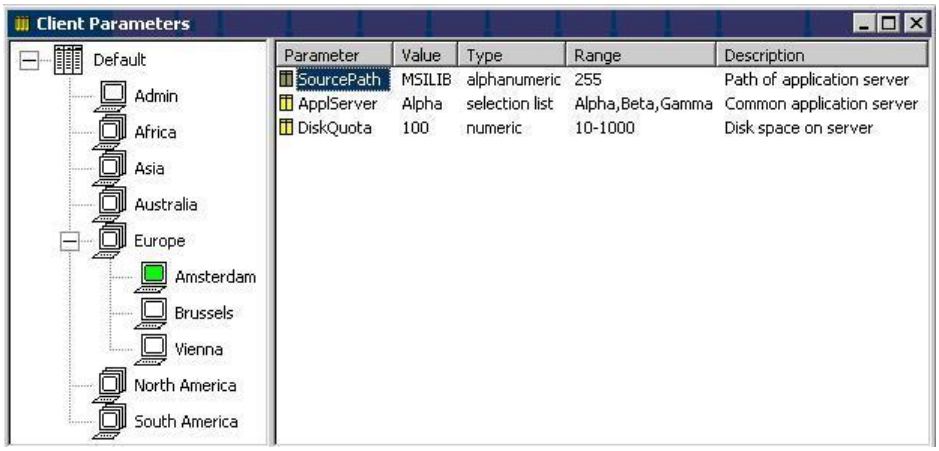
Automatic Parameter Generation

Entering parameters manually is not practical for a large number of target computers. Instead, you can generate the parameter files automatically with popular third-party software tools, such as Perl and AWK. Doing so lets you group the PCs as you choose and enter client parameters for all PCs in the group at once.

Client Parameters Window

To edit client parameters, select Edit, Client parameters from the Packager main menu. The Client Parameters window appears, where you can define all parameters, for any SXP product, with values specific to each individual PC in your system (PC-specific values). All of the parameters contain default values, from the default object, that can optionally be overwritten with either group-specific values or computer-specific values.

A sample Client Parameters window is shown following:



The Client Parameters window consists of two panes.

- The left pane shows all client parameters and the default object the client parameters belong to in a tree structure.
- The right pane displays the parameters of the default object, when you click it in the left pane. The group objects and PC objects appear below the default object. These objects contain all parameters of the default object, although some values differ from the default values.

The following icons are used in the left pane of the Client Parameters window:

Indicates the default object (see page 54) (yellow symbol). The default object contains all parameters. Initially, there are no parameters; you must create them.

Indicates a group object (see page 55) (red symbol). A group object contains parameters for a specific group of target computers that deviate from the default object.

Initially, there are no group objects; you must create them. When you create a group object, it inherits all default parameters. You can change the parameter values, but not their type or length.

Indicates a PC object (see page 57) (green symbol). A PC object contains parameters for a specific target computer that deviate from either the default object, the PC group to which the computer belongs, or both. Each target computer that requires individual parameters requires its own entry.

Initially, the PC object Admin represents the Packager. When you create a PC object, it inherits all parameters from the group object or default object from which it was created. You can change the values of the parameters, but not their type or length.

The following icons are used in the right pane of the Client Parameters window:

Indicates a default parameter value.

Indicates a group-specific parameter value.

Indicates a PC-specific parameter value.

You must first assign default values, and then, if needed, assign either group-specific or computer-specific deviations from the defaults. Group-specific values take precedence over default values, and computer-specific values take precedence over both default values and group-specific values.

Default Object

The default object is located at the top level of the tree and contains all parameters. Right-click the default object to display the three shortcut menu options:

Add Group

Creates a group object that inherits the default values.

Add PC

Creates a PC object that inherits the parameter values of the closest object hierarchically in the tree (default or group).

Create parameter product

Passes on the client parameters as an SXP product in the product archive.

To administer default parameters

1. Select the default object in the left pane.
2. Select a specific parameter in the right pane
3. Right-click the parameter and select one of the following options from the context menu:

   Add parameter

   Creates a parameter in the default object.

   Edit parameter

   Modifies the values, range, and the description of a parameter.

   Delete parameter

   Deletes a parameter from the default object.

   Create parameter product

   Passes the client parameters as an SXP product in the product archive.

You must first assign default values, and then, if needed, assign either group-specific or computer-specific deviations from the defaults. Group-specific values take precedence over default values, and computer-specific values take precedence over both default values and group-specific values.

Important! If you delete a parameter from the default section, that parameter is no longer available for any new group or PC objects. In addition, the deleted parameter is removed from any existing group and PC objects.

Group Objects

The group object contains all parameters for a group of PCs and is located at the second level of the tree. Create group objects when you want to assign parameter values for the group that are different than the default values. Generally, create the group first and add the PC objects to it. Then, modify the necessary parameter values for the group.

To create a group object

1. Select the default object.
2. Right-click the default object and select Add Group from the shortcut menu.

To administer a computer group

1. Select the group object in the left pane
2. Right-click the group object and select one of the following options from the shortcut menu:

   Delete Group

   Deletes a group object from the second level of the tree.

   Add PC

   Creates a PC object in the third level of the tree.

   Create parameter product

   Passes the client parameters as an SXP product in the product archive.

   Important! Each computer that you create can belong to only one group.

To administer parameters for a group

1. Select the group object in the left pane.
2. Select a specific parameter in the right pane.
3. Right-click this parameter and select one of the following options from the shortcut menu.

   Edit parameter

   Modifies the parameter value within the defined range of values. The values also apply for lower level PCs, if you have not set a different value for this parameter.

   Create parameter product

   Passes the client parameters as an SXP product in the product archive.

Group-specific values take precedence over default values, and computer-specific values take precedence over both default values and group-specific values.

PC Objects

PC objects contain all parameters, some of which may deviate from the parameters of the closest object hierarchically (default or group). A PC object can be located either directly below the default object or at the third level of the tree.

To create a PC object

1. Select the default object
2. Right-click the default object and select Add PC from the shortcut menu.
3. Name the PC object according to your site standards.
4. Right-click the PC object to display two shortcut options:

   Delete PC

   Removes a PC object from the second or third level of the tree.

   Create parameter product

   Passes the client parameters as an SXP product in the product archive.

To administer parameters for a PC object

1. Select the PC object in the left pane.
2. Right-click the PC object and select one of the following options from the shortcut menu:

   Edit parameter

   Modifies the parameter value for the selected PC within the value range.

   Create parameter product

   Passes the client parameters as an SXP product in the product archive.

To move a PC object from one group to another

1. Select the PC object in the left pane.
2. Drag and drop the PC object onto a group object.

All parameter values for the group are applied to any PC you move to the group, except for PC-specific parameter values.

Client Parameter Files and Parameter Format

The client parameters are stored in the following ini files:

- default.ini (see page 58)
- group.ini (see page 59)
- pc.ini (see page 59)
- tree.ini (see page 60)

To reference the parameters in scripts and archive files, use the following format for the parameters (see page 52):

$(*parametername*)

default.ini File

The default.ini file contains one section named [default]. This section specifies the name and type of each client parameter, as well as a default, range of values, and description. Three types of parameters are supported:

Numeric Entries (n)

Use the following structure for numeric entries:

parametername=n,default,minimumvalue,maximumvalue,
description

Alphanumeric Entries (a)

Use the following structure for alphanumeric entries:

parametername=a,default,stringlength,description

List Entries (l)

Use the following structure for list entries:

parametername=l,default,numberoflistelements,
listelement1,listelement2,...,
listelementn,description

A sample default.ini file follows:

```
[default]
parameter1=n,5,0,10,Parameter1 numeric
parameter4=l,stockcontrol,4,clearing,stockcontrol,purchase,orders,Parameter4 selection list
parameter3=a,controlling 66,6,Parameter3 alphanumeric
parameter2=n,33,0,100,Parameter2 numeric
```

group.ini File

The group.ini file contains several sections. Each section must be named the same as the PC group to which its parameters apply. Each section contains only the parameters that differ from the default parameter values in the default.ini file.

The entries in the group.ini file consist of the name and value of the parameter only, since the type and range of a parameter cannot be changed.

A sample group.ini file follows:

```
[group1]
parameter4=clearing

[group2]
parameter3=payroll99

[group3]
parameter1=9
```

pc.ini File

The pc.ini file contains the deviations for each PC from its PC group, or from the default object if the PC is not assigned to a PC group.

If there is no section for a PC in this file, the defaults for its PC group file (in the group.ini file) apply automatically. If the PC is not assigned to a PC group, the defaults from the default.ini file apply automatically.

Each PC that has at least one parameter different from the default receives its own section in this file. Each section has the same name as the PC to which its parameters apply.

```
[pc0_1]
parameter2=48
parameter3=08/15

[pc1_1]
parameter3=1xyz

[pc2_1]
parameter1=8

[pc2_2]
parameter4=purchase
```

tree.ini File

The tree.ini file specifies the names of each PC group and its members. Each section is named for a PC group. The individual PCs in a group are listed in the section named for their group.

In addition, the tree.ini file contains a section named [default], which lists PCs that do not belong to any group.

A sample tree.ini file follows:

```
[default]
pc0_1

[group1]
pc1_1

[group2]
pc2_1
pc2_2
```

Parameter Products

Parameter products contain packaged parameters, while regular products contain packaged executable software or data. The following characteristics apply to parameter products:

- All parameters are always stored and distributed together, rather than individual parameters from specific objects.
- The client parameters are stored in the product archive. They are not automatically registered.
- The parameter product must be registered in the Software Package Library before being installed on the target computers.

Important! You should always distribute and install the parameter product before you distribute the product itself.

Client Parameter Registration

After you have created the parameter product, you must register it in the Software Package Library. When you register the parameter product, it is copied from the Packaging Computer to the Software Package Library and made available for distribution to the target computers.

Parameter products are stored in the product archive. The procedure for registering parameter products is the same as that one used to register regular products.

Chapter 4: Using the Installer for Windows

This section contains the following topics:

About the Installer for Windows

The Installer for Windows (Installer) is a software management component that resides on the Windows target computers in the network. You install it on the target computers as part of the software delivery agent functionality.

The Installer receives, installs, and removes SXP products. The Installer lets you easily upgrade and downgrade installed software products, without the need to uninstall or remove older products manually before upgrading or downgrading. The installed product is automatically completely uninstalled by the Installer, before the new product version is installed on the target computer.

The Installer for Windows consists of two main components, the Installer function and the Logon function.

Installer function

The Installer function checks for pending install or uninstall jobs from a manager, and executes them. However, the Installer cannot directly execute the user-defined settings, for example, from the ureg.sxp and ulinks.sxp files. Instead, it prepares these files for execution by the Logon function.

Logon function

The Logon function implements the treatment of user-defined settings for the installed products. Because many settings can be processed only if the user is actually logged on, the Logon function is started automatically when the user logs on.

The Logon function can also be configured to interact directly with the Installer, so that settings can be processed immediately for the current user.

How the Installer Installs Products

When the Installer detects and executes an installation job, it creates a local archive for every product that will be installed. In the local archive, the Installer stores selected product data, such as all SXP archive files. A local product archive is deleted only if the associated product is removed.

The process used by the Installer to install a product is divided into several steps. Not all steps are executed for all products. Which steps are executed depends on the type of SXP archive files included with the product.

For each step, an SXP file is transferred to the local archive, and then processed by the Installer. The Installer replaces all of the parameters used in the SXP archive files with values that are valid for the target computer.

Therefore, each step is the result of the Installer's processing a specific SXP file. If no SXP file was created during the packaging process for an individual product, the corresponding step is omitted when the product is installed on the target computer.

SXP files containing user-specific data are only prepared for the Logon function, which performs the actual user profile update when the Installer has finished or at the next log on.

The installation process includes the following steps:

1. Check info.sxp (Product) (see page 65)
2. Check actions.sxp (Pre-programs) (see page 65)
3. Check gina.sxp (see page 65)
4. Check uactions,sxp (see page 66)
5. Check original.sxp (Original setup) (see page 66)
6. Check dirs.sxp and udirs.sxp (Directories) (see page 66)
7. Check files.sxp and ufiles.sxp (Product files) (see page 67)
8. Check gac.sxp (see page 67)

9. Check sreg.sxp, ureg.sxp, sregdel.sxp, and uregdel.sxp (Registry) (see page 67)
10. Check ascnnnn.sxp (ASCII files) (see page 68)
11. Check ininnnn.sxp and uininnnn.sxp (.ini files) (see page 68)
12. Check desktop.sxp (Desktop) (see page 68)
13. Check services.sxp (Services) (Windows NT only) (see page 69)
14. Check links.sxp and ulinks.sxp (Links) (see page 69)
15. Check permis.sxp (Permissions) (Windows NT only) (see page 69)
16. Check actions.sxp (Post-programs) (see page 69)

The proceeding after installation is considered under Proceeding on Successful and Unsuccessful Installations (see page 70).

Check info.sxp (Product)

In this step, the Installer checks the info.sxp file and performs the following actions:

- Check whether the product can be used on the system.
- Check for dependencies on other products (internal and external).
- Remove the SxpRootDirn parameter (root directory n) from the file for later use and sets it to the correct value
- Establish the reset level. If a reset level was specified in the info.sxp file, for a specific product being installed, the Installer uses it. Otherwise, the Installer uses the reset level set in the Program section of the sxpeng*xx*.ini file.

Note: If the installation fails, the changes that can be canceled depend on the value of the reset level.

Check actions.sxp (Pre-Programs)

In this step, the Installer checks the actions.sxp file and executes the pre-programs in sequence (#InsPreActions# section).

Check gina.sxp

In this step, the Installer checks the gina.sxp information and installs and configures the GINA-related files.

Check uactions.sxp

In this step, the Installer prepares the uactions.sxp information for the Logon function.

Check original.sxp (Original Setup)

In this step, the Installer checks the original.sxp file and performs the following actions:

- Copy the response file (#ResponseFiles# section) to the local archive.
- Modify the response file, using the appropriate script (RFile*n* = SFile*n*).
- Execute the original setup program.

Check dirs.sxp and udirs.sxp (Directories)

In this step, the Installer checks the dirs.sxp file and performs the following actions:

- Edit the #InsDelDirs# section, and deletes the corresponding directories and files from the target computer.
- Edit #InsDelDirsWithSubs# section, and deletes the corresponding directories with subdirectories and files from the target computer.
- Edit the #InsAddDirs# section, and creates the corresponding directories on the target computer.
- Edit the #InsSetDirAttr# section, and sets the corresponding directory attributes on the target computer.

The Installer also prepares the udirs.sxp information for the Logon function.

Check files.sxp and ufiles.sxp (Product Files)

In this step, the Installer checks the files.sxp file and performs the following actions:

- Edit the #InsDelFiles# section, and deletes the corresponding files from the target computer.
- Mark files that cannot be deleted now, and deletes them at the next boot.
- Edit the #CmpArchiveFiles# section, and unpacks the corresponding files from the files.cmp archive file.
- Install or ignore each version-dependent file in the product being installed, according to the version-dependent installation setting specified for the file. It also updates the reference counter dependent uninstall setting for the file, if applicable.
- Mark files that cannot be overwritten now, and overwrites them at the next boot.
- Edit the #ReplaceParams# section, and replaces the corresponding parameters in unpacked files.

The Installer also prepares the ufiles.sxp information for the Logon function.

Check gac.sxp (Global Assembly Cache)

In this step, the Installer checks the gac.sxp file (GAC = Global Assembly Cache), which contains sections that specify assemblies to install or remove.

Assemblies specified in the #CreateAssembly*n*# section of the gac.sxp are installed in the GAC, if they were not yet installed, and the Installer creates a reference to the current product. If a reference already exists (that is, the assembly was already installed), only a new reference is created. If the reference entries are missing in a #CreateAssembly*n*# section, the assembly is installed without a reference.

The Installer removes an assembly specified in the #DeleteAssembly*n*# section of the gac.sxp from the GAC, provided the last reference will be deleted. Otherwise only the reference to the current product is deleted.

Check sregdel.sxp, sreg.sxp, uregdel.sxp, and ureg.sxp (Registry)

In this step, the Installer checks the sregdel.sxp and sreg.sxp files and modifies the registry accordingly. The Installer also creates new keys and sets values for both new and existing keys.

The Installer also prepares the uregdel.sxp and ureg.sxp information for the Logon function.

Check ascnnnn.sxp (ASCII Files)

In this step, the Installer checks the #ScriptFiles# section in the asc*nnnn*.sxp file for installation scripts (asc*nnnn*.ins) that modify ASCII files. If any are found, the Installer executes each script found for each SXP archive file in the installation job being executed. By default, all changes are appended at the end of the ASCII file.

Check ininnnn.sxp and uininnnn.sxp (.ini Files)

In this step, the Installer checks the ini*nnnn*.sxp file and performs the following actions:

- Edit the #InsDelEntries# section of the ini*nnnn*.sxp file, and delete or modify the corresponding entries in the ini files.
- Edit the #InsAddEntries# section of the ini*nnnn*.sxp file, and creates or modifies the corresponding entries in the ini files.
- Edit the #InsDelSections# section of the ini*nnnn*.sxp file, and deletes the corresponding sections from the ini files.

In the list, the term ini file refers to all .ini files that were included in the reference installation for the product.

The Installer also prepares the uini*nnnn*.sxp information for the Logon function.

Check desktop.sxp (Desktop)

In this step, the Installer checks the (legacy) desktop.sxp file and performs the following actions:

- Create the folder (new shells only).
- Edit the #InsFolders# section.
- Create the links (new shells only).
- Edit the #InsLink*n*# section.
- Create the icons.
- Edit the #InsItem*n*# section.

Check services.sxp (Services)

(Windows NT only)

In this step, the Installer checks the services.sxp file and performs the following actions:

- Stop services
- Delete services
- Create services
- Start services

Check links.sxp and ulinks.sxp (Links)

In this step, the Installer checks the links.sxp file and modifies the desktop accordingly.

The Installer also prepares the ulinks.sxp information for the Logon function.

Check permis.sxp (Permissions)

(Windows NT only)

In this step, the Installer checks the permis.sxp file and performs the following actions:

- Edit the #Permis*n*# section.
- Install the Access ACEs and the Denied ACEs.

Check actions.sxp (Post-Programs)

In this step, the Installer checks the actions.sxp file and executes the post-programs specified in the #InsPostActions# section, in the order in which they are specified.

Proceeding on Successful and Unsuccessful Installations

If the installation succeeds, the user of the target computer may be able to use the newly installed application immediately, or he may need to log out and on again before using the new application. This depends on the software installed and the Installer configuration.

If the installation fails the Installer cancels the changes made, according to the current reset level. System data is recorded and stored in the backup directory during installation, if a reset is needed.

For more information, see Configure Updates to Be Activated Immediately (see page 76).

How the Installer Uninstalls Products

The process used by the Installer to uninstall a product, is divided into several steps. Not all steps are executed for all products. Which steps are executed depends on the type of SXP archive files included with the product.

For each step, the Installer refers to the SXP file that was transferred to the local archive at installation time. Therefore, each step is the result of the Installer processing a specific SXP file. If no SXP file was created during the packaging process for an individual product or parameter product, the corresponding step is omitted when the product is uninstalled on the target computer.

SXP files containing user-specific data are only prepared for the Logon function, which performs the actual user profile update when the Installer has finished or at the next log on.

The uninstallation process includes the following steps:

1. Check actions.sxp (Pre-programs) (see page 71)
2. Check gina.sxp (see page 71)
3. Check services.sxp (Services) (Windows NT only) (see page 71)
4. Check original.sxp (Original setup) (see page 72)

5. Check ascnnnn.sxp (ASCII files) (see page 72)
6. Check uininnnn.sxp (.ini files) (see page 72)
7. Check ulinks.sxp (Links) (see page 72)
8. Check gac.sxp (Global Assembly Cache) (see page 73)
9. Check links.sxp (Links) (see page 73)
10. Check ininnnn.sxp (.ini files) (see page 73)
11. Check ufiles.sxp (Product files) (see page 73)
12. Check files.sxp (Product files) (see page 74)
13. Check udirs.sxp (Directories) (see page 74)
14. Check dirs.sxp (Directories) (see page 74)
15. Check sreg.sxp (Registry) (see page 74)
16. Check desktop.sxp (Desktops) (see page 75)
17. Check uactions.sxp (Post-programs) (see page 75)
18. Check actions.sxp (Post-programs) (see page 75)

The proceeding in case of an unsuccessful product removal is considered under Unsuccessful Product Removals (see page 75).

Check actions.sxp (Pre-Programs)

In this step, the Installer checks the actions.sxp file and executes the pre-programs in the order in which they are listed.

Check gina.sxp

In this step, the Installer checks the gina.sxp information and removes the GINA.

Check services.sxp (Services)

(Windows NT only)

In this step, the Installer checks the services.sxp file and performs the following actions:

- Stop services
- Delete services
- Create services
- Start services

Check original.sxp (Original Setup)

In this step, the Installer checks the original.sxp file and executes the original setup program of the product that was packaged.

Check ascnnnn.sxp (ASCII Files)

In this step, the Installer checks the #ScriptFiles# section in the asc*nnnn*.sxp file for uninstall scripts (asc*nnnn*.dei) that modify ASCII files. If any are found, the Installer executes each script found for each SXP archive file in the installation job being executed. By default, all changes are appended at the end of the ASCII file.

However, if an uninstall script specifies lines to be removed, the Installer does not remove those lines if they are also used by another product installed on the target computer.

Check uininnnn.sxp

In this step, the Installer checks the uini*nnnn*.sxp file and performs the following actions for every SXP archive file in the uninstall job:

- Edit the #DeiAddEntries# section, taking into account whether the lines to be removed are also used by other products.
- Edit the #DeiDelSections# section, and delete the corresponding entries from the ini files.
- Edit the #DeiDelEntries# section, and delete or modify the corresponding entries in the ini files.
- Edit the #DeiAddEntries# section, and create or modify the corresponding entries in the ini files.

In the list, the term ini file refers to all .ini files that were included in the reference installation for the product.

Entries used by other installed products are not removed.

Check ulinks.sxp (Links)

In this step, the Installer prepares the ulinks.sxp information for the Logon function.

Check gac.sxp (Global Assembly Cache)

In this step, the Installer checks the gac.sxp file (GAC = Global Assembly Cache), which contains sections that specify assemblies to remove.

The Installer removes an assembly specified in the #DeleteAssembly*n*# section of the gac.sxp from the GAC, provided the last reference will be deleted. Otherwise only the reference to the current product is deleted.

Check links.sxp (Links)

In this step, the Installer checks the links.sxp file and modifies the desktop accordingly.

Check ininnnn.sxp (.ini Files)

In this step, the Installer checks the ini*nnnn*.sxp file and performs the following actions for every SXP archive file in the uninstall job:

- Edit the #DeiAddEntries# section, taking into account whether the lines to be removed are also used by other products.
- Edit the #DeiDelSections# section, and delete the corresponding entries from the ini files.
- Edit the #DeiDelEntries# section, and delete or modify the corresponding entries in the ini files.
- Edit the #DeiAddEntries# section, and create or modify the corresponding entries in the ini files.

In the list, the term ini file refers to all .ini files that were included in the reference installation for the product.

Entries used by other installed products are not removed.

Check ufiles.sxp (Product Files)

In this step, the Installer checks the ufiles.sxp file and performs the following actions:

- Edit #FilesInArchives# section, and delete the corresponding files from the target computer.
- Mark files that cannot be deleted now, and delete them at the next reboot.

Check files.sxp (Product Files)

In this step, the Installer checks the files.sxp file and performs the following actions:

- Edit #FilesInArchives# section, and delete the corresponding files from the target computer.
- Mark files that cannot be deleted now, and cannot delete them at the next reboot.

Check udirs.sxp (Directories)

In this step, the Installer checks the udirs.sxp file and performs the following actions:

- Edit the #DeiDelDirs# section, and delete the corresponding directories from the target computer.
- Edit the #DeiDelDirsWithSubs# section, and delete the corresponding directories and subdirectories from the target computer.
- Edit the #DeiAddDirs# section, and create the corresponding directories on the target computer.

Check dirs.sxp (Directories)

In this step, the Installer checks the dirs.sxp file and performs the following actions:

- Edit the #DeiDelDirs# section, and delete the corresponding directories from the target computer.
- Edit the #DeiDelDirsWithSubs# section, and delete the corresponding directories and subdirectories from the target computer.
- Edit the #DeiAddDirs# section, and create the corresponding directories on the target computer.

Check sreg.sxp (Registry)

In this step, the Installer checks the sreg.sxp file and modifies the registry, deleting keys and key values, or changing key values. The Installer does not delete keys or values that are used by other products on the target computer.

Check desktop.sxp (Desktops)

In this step, the Installer checks the desktop.sxp file and performs the following actions:

- For new shells only, edit the #DeiLinks# section and delete the links named in that section.
- For new shells only, edit the #DeiFolders# section and delete the folders named in that section.
- Edit the #InsItem*n*# section and perform the following actions:
  - Remove icons from the Windows Explorer
  - Buffer statements in a file in the backup directory; after uninstallation is completed, this file is copied to the individual user files in the user's directory

Check uactions.sxp (Post-Programs)

In this step, the Installer checks the uactions.sxp file and executes the user-specific post-programs, in the order in which they are listed in that file.

Check actions.sxp (Post-Programs)

In this step, the Installer checks the actions.sxp file and executes the post-programs, in the order in which they are listed in that file.

Unsuccessful Product Removals

If any errors occur during product removal, the Installer notes them but continues the uninstallation process and removes as much of the product as possible. If you receive any error messages, select the error log to help determine whether you need to delete or change files manually, to complete uninstalling the product.

For more information, see Configure Updates to Be Activated Immediately (see page 76).

Configure Updates to Be Activated Immediately

For each user defined on a Windows NT computer, Windows NT stores user-defined data for the desktop and registry in the user's profile. Whenever a user logs in to the computer, the Logon function checks this data and resets the profile for the user, making any necessary updates.

Use the Show=32 parameter to specify that sxplogxx.exe is to be activated immediately after updates are made to the profile.

So, the user is not required to log out and on again to activate these updates.

Boot Procedure and Boot Levels

After the Installer has processed its jobs (install, uninstall and update), a system start (boot or logoff/logon) is initiated when a respective boot level has been set or when a system restart is necessary (for example, because of locked files). The system start can take different forms depending on the boot level and the condition of the system.

The installation boot level is set automatically by the reference installation, however, it can be modified manually. You can set boot levels for installation and uninstallation separately. If an uninstallation boot level is not specified, the value of the installation boot level is used.

The Installer can change the boot level to a higher value, if a logoff/logon or reboot is necessary to install the product.

Optionally, you can specify or change a boot level for a product in the info.sxp archive file. You have the possibilities to force a reboot or logon/logoff after the product is installed.

The following table provides information about the boot level, the initiating entity, and the corresponding action:

Boot Level	Action	Comment
0	Target driven	If circumstances during installation or removal of the product on the target computer require a reboot (for example, when a file to be replaced is locked on the target computer), the Installer initiates this reboot. Default, when the Packaging Computer has not been rebooted during the reference installation.

Boot Level	Action	Comment
1	Logoff required	Requires a user to log off and on to activate changes to the Windows desktop and registry.
3	Restart after batch	Automatically initiates a system reboot at the end of the transaction. Default, when the Packaging Computer has been rebooted during the reference installation.
4	Restart after job	Initiates a system reboot immediately after the end of the installation or removal of the product.

For boot levels 1 and 3 the following applies: In a transaction with several installation and uninstallation jobs, the uninstallation jobs are executed first. The highest uninstallation boot level that occurs is recorded and the necessary system boot is performed at the end of all the uninstallation jobs. Then, the same procedure is applied to all installation jobs (using the highest installation boot level).

The boot level 2 no longer applies with this Packager version. If an older package that contains the boot level 2 entry is being edited, the boot level is internally set to "1—Logoff required".

Modification Rules for autoexec.bat

The Installer follows specific modification rules for the autoexec.bat file when performing installation (see page 78) and uninstallation (see page 79) jobs on target computers.

How autoexec.bat Is Modified at Installation

When the autoexec.bat file is prepared for installation on the target computer, all entries in the *ascnnnn*.sxp file that are not already in the autoexec.bat file are copied to the autoexec.bat file.

This copy process does not apply to lines that contain one of the following commands: LH, LOADHIGH, or SET.

The Installer processes lines that include these commands as follows:

LH and LOADHIGH

In a program call, the LH and LOADHIGH commands are not evaluated. For example, if the autoexec.bat file contains the program call LH C:\DRIVER.EXE 1024 and the asc*nnnn*.sxp file contains the entry C:\DRIVER.EXE, the C:\DRIVER.EXE entry from asc*nnnn*.sxp is added to autoexec.bat.

SET

You can optionally use the SET command in the autoexec.bat file to assign a value to a variable. However, if both the autoexec.bat file and the asc*nnnn*.sxp file specify two different settings for the same variable, the setting in the asc*nnnn*.sxp file is used.

For example, if the autoexec.bat file specifies SET TEST=5 and the asc*nnnn*.sxp file specifies TEST=10, when the program is installed, the TEST=10 setting is used.

However, there is one exception:

The following variable assignment is added to the autoexec.bat file as an extension to the variable value:

(*variablename*=%*variablevalue*%*value*)

This statement never overwrites any previous variable settings.

The variable value extension is entered at the end of the file. You should note this value in case the new path must be used before the startup process reaches the end of the autoexec.bat file (for example, for a command). At system startup, the autoexec.bat file is evaluated sequentially.

Each extension of the PATH variable performed by the installation job is entered as an extension in the installation file.

For example, an application extends the following line:

```
PATH=C:\DOS;C:\WIN;
```

By adding C:\TEST, the entry in the asc*nnnn*.sxp file becomes

```
PATH= %PATH%C:TEST
```

The previous entry would extend the PATH variable by C:\TEST; during installation and would not overwrite any part of the existing entry.

How autoexec.bat Is Modified at Removal

When an uninstallation job is executed, all entries in the asc*nnnn*.sxp file of the application to be uninstalled are removed from the autoexec.bat file, provided they are not needed by another application installed on the target computer.

The Installer processes as follows lines that contain one of the commands LH, LOADHIGH, or SET:

LH and LOADHIGH

The Installer compares program calls with the lines containing the LH and LOADHIGH commands. The LH and LOADHIGH commands are ignored in the comparison. For example, if the autoexec.bat file contains the LH C:\DRIVER.EXE program call, and the asc*nnnn*.sxp file also contains the C:\DRIVER.EXE entry, the line is deleted if no other application on the target computer needs the DRIVER.EXE program.

SET

You can optionally use the SET command in the autoexec.bat file to assign a value to a variable. If you do so, and if the entry you specify is to be deleted by the uninstall job, the entry is deleted only if the variable named in the SET command is not used by another program on the target computer.

Modification Rules for config.sys

The Installer follows specific modification rules for the config.sys file when performing installation (see page 79) and uninstallation (see page 81) jobs on target computers.

How config.sys Is Modified at Installation

All entries in the asc*nnnn*.sxp file of the application to be installed are copied to the target computer's config.sys file, if those entries are not already present in that file.

This procedure does not apply to lines that contain one of the following commands: BUFFERS, STACKS, FILES, FCBS, LASTDRIVE, DEVICE, DEVICEHIGH, INSTALL, or SET.

The Installer processes lines that include these commands as follows:

BUFFERS, STACKS, FILES, or FCBS

If the asc*nnnn*.sxp file contains an entry with a BUFFERS, STACKS, FILES, or FCBS command, the Installer compares the entry from the asc*nnnn*.sxp file and the entry from the config.sys file, and uses the higher value as a new entry.

For example, a BUFFERS=50 installation entry overwrites a BUFFERS=30 entry, but not a BUFFERS=60 entry. If there are more than two values (as with STACKS), the highest combined value is used. For example, from STACKS 9,30 and STACKS 5,35, the Installer uses STACKS 9,35.

LASTDRIVE

The LASTDRIVE command is handled similarly. A LASTDRIVE=F entry in the installation file overwrites a LASTDRIVE=D entry in the config.sys file, but not a LASTDRIVE=G entry.

DEVICE and DEVICEHIGH

The DEVICE and DEVICEHIGH commands are used to load drivers. The Installer defines the file name of the driver, such as EMM386.EXE, and compares it with both the installation entry and the current entry in the config.sys file. If the two driver names are the same, the config.sys entry is overwritten by the installation entry, even if the full path names are different. For example, the DEVICE=C:\DOS\EMM386.EXE installation entry overwrites the config.sys DEVICEHIGH=D:\WIN\EMM386.EXE entry.

Important! When the DEVICE and DEVICEHIGH commands are compared, only the driver name is considered; the exact command specified (DEVICE or DEVICEHIGH) is not important.

INSTALL

The modification rule for the INSTALL command corresponds to that for DEVICE and DEVICEHIGH.

SET

You can optionally use the SET command in the config.sys file to assign a value to a variable, using the same rules as those used for the autoexec.bat file.

How config.sys Is Modified at Removal

When an uninstallation job is executed, all entries in the asc*nnnn*.sxp file of the application to be uninstalled are removed from the target computer's config.sys file, provided they are not needed by any other application installed on the target computer.

This Installer processes as follows lines that contain one of these commands, BUFFERS, STACKS, FILES, FCBS, LASTDRIVE, DEVICE, DEVICEHIGH, INSTALL, and SET:

BUFFERS, STACKS, FILES, and FCBS

If the asc*nnnn*.sxp file contains an entry with a BUFFERS, STACKS, FILES, or FCBS command, the Installer checks whether another application installed on the target computer needs the entry:

- If no, the entry is deleted.
- If yes, the next highest entry is used for the other applications.

For example, if the config.sys file contains BUFFERS=50 and the installation file also contains BUFFERS=50; and if two other applications have the entries BUFFERS=40 and BUFFERS=30, BUFFERS=50 is replaced by BUFFERS=40.

LASTDRIVE

The LASTDRIVE command is handled the same way as the BUFFERS, STACKS, FILES, and FCBS commands.

DEVICE and DEVICEHIGH

The DEVICE and DEVICEHIGH commands are used to load drivers. The Installer defines the file name of the driver, such as EMM386.EXE, and compares this with the driver names used by the other installed applications. If another application is using this driver, the line in the config.sys file is not deleted.

Important! When the DEVICE and DEVICEHIGH commands are compared, only the driver name is considered; the exact command specified (DEVICE or DEVICEHIGH) is not important.

INSTALL

The modification rule for the INSTALL command corresponds to the modification rule for the DEVICE and DEVICEHIGH commands.

SET

You can optionally use the SET command, in the config.sys file, to assign a value to a variable. If you do so, and if the entry you specify is to be deleted by the uninstall job, the entry is deleted only if the variable named in the SET command is not used by another program on the target computer.

Modification Rules for win.ini and system.ini

The Installer follows specific modification rules for the win.ini and system.ini files when performing installation (see page 82) and uninstallation (see page 82) jobs on target computers.

How win.ini and system.ini Are Modified at Installation

When the win.ini and system.ini files are modified for installation on the target computer, all entries in the asc*nnnn*.sxp file are copied to the specified sections in the .ini files. Entries in the .ini files are overwritten by any matching entries in the asc*nnnn*.sxp file.

The following exceptions apply:

Load and run entries

The load and run entries in the [windows] section of the win.ini file are only extended by installation entries.

For example, suppose the win.ini file contains the following entry:

```
load=c:\tools\fun.exe
```

The installation entry is:

```
load=c:\dark\dark.exe
```

Then the new extended entry in the win.ini file becomes:

```
load=c:\tools\fun.exe c:\dark\dark.exe
```

[386Enh] section

All drivers with device= are entered in the system.ini file in the [386Enh] section. No existing device= entries are overwritten in the [386Enh] section; instead, existing entries are added.

How win.ini and system.ini Are Modified at Removal

All entries that are contained in the asc*nnnn*.sxp file of the application to be uninstalled are removed from the config.sys file if they are not needed by other applications that are installed on the target computer.

The following exceptions apply:

Load and run entries

The load and run entries in the [windows] section of the win.ini file do not have a corresponding entry in the installation file; therefore, they are not removed.

[386Enh] section

The drivers (entry: device=) are deleted that were written to the [386Enh] section in the system.ini file by the application that is being removed.

Chapter 5: Diagnostics and Troubleshooting

If you encounter errors while using the Packager and Installer for Windows, the Trace feature, available for Packager and Installer, and the output of installations on the target computer help you obtain detailed diagnostic information.

This section contains the following topics:

View the Results of Installation Jobs

To obtain detailed diagnostic information about software installation jobs, you can view the output of the installation jobs on the target computers.

Follow these steps:

1. Expand Jobs, Software Jobs, All Software Jobs on the Explorer tree of your domain manager.

   The All Software Jobs node displays a list of all scheduled and completed job containers on this domain.

2. Double-click a job container on that list.

   The list of target computers to which the job container was sent is displayed in the right pane.

3. Double-click a target computer on that list.

   The jobs for that target computer are displayed in the right pane.

4. Right-click a job and select Properties from the context menu.

   The Job Container Target Job Properties dialog opens, displaying tabs with details about the selected job:

   Computer Job

   Provides status and error information in the Status Message field, if the installation job has failed. If errors occurred when the job was executed, the error messages are displayed here (for example, SXP000290 - Product cannot be used for this system).

   Click the Status Help button to search for more details about reason and solution for an error message, for example, SXP000290.

Delivery Trace

Displays the trace information that is provided by the Data Transport Service (DTS) component.

Job output

Displays the output that is provided by the Software Management Installer on the target computer. If the trace feature was turned on when the job was executed, the Job output tab provides trace information. If the trace feature was turned off when the job was executed, the output tab has fewer details, and, for jobs with no errors, it can be empty.

Trace Features

You can activate tracing for the Packager, the Installer, or both. Tracing for the Installer can be activated on any number of target computers.

Packager Trace

For diagnostic purposes, you can optionally record the activities of the Packager in a trace file, called trace.txt, on the Packaging Computer. The trace file is available in the trace subdirectory of the Packager installation directory. If necessary, the Packager creates backup files called trace.ba0, trace.ba1, and so on. The trace.txt file always contains the recent information.

To enable the trace for the Packager, select Tools, Options from the Packager main menu and select Enable Trace on the appearing Options dialog. To disable the Packager trace, clear the Enable Trace check box on the Options dialog. Protocol and Error messages are always written into the trace files, even if the trace feature is not enabled.

Installer Trace

For diagnostic purposes, you can specify that the activities of the Installer must be recorded in a trace file on the target computer.

To specify this for the target computer you want to trace, select the agent configuration procedure 'SM Installer: Enable Trace' from the Properties menu of the software delivery agent plug-in in the DSM Explorer.

To transfer the trace file contents to the output file of the job, use the agent activate procedures 'SM Installer: Get Latest Trace' or 'SM Installer: Get All Traces'.

Trace information is also created for the installation of user-specific items. To query this information, use the agent activate procedure 'SM Installer: Get User Trace'.

Instead of using a trace, you can query user-specific history by issuing the agent activate procedure 'SM Installer: Get User History'.

The same mechanism is used to record activities of the Microsoft Installer (MSI) during installation of converted MSI products. This MSI trace consists of up to three parts:

- Contents of SxpOutputFile
- Current property/parameter settings
- Trace messages

Log File Collection Tool dsminfo

CA Technologies provides the dsmInfo tool, which collects diagnostic information from systems that have Client Automation installed. The data collected is compressed into a single file that contains log files, system information, directory structures, and registry and environment information. This diagnostic tool is available in the Client Automation product installation media under the DiagnosticTools folder.

If a problem with Client Automation is reproducible, then run the following command to change the trace level to DETAIL:

```
cftrace -c set -l DETAIL
```

Reproduce the problem and collect the diagnostic information with the dsmInfo tool.

Notes:

For more information about this tool, see the DSMInfoReadMe.txt file available under the DiagnosticTools folder in the product installation media.

The dsmInfo tool produces ".7z" files by default. These files provide better compression than zip files, so uploading to CA Technologies is easier.

Index