



forward faster

▶▶ **SSA® ERP LN 6.1**

Product Maintenance and Control (PMC) Distributor User Guide

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any way or by any means, including, without limitation, photocopying or recording, without the prior written consent of Baan™ International B.V.

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Baan International B.V.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Baan International. and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company pursuant to a separate agreement ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above.

Although Baan International, has taken due care to ensure that the material included in this publication is accurate and complete, Baan International, cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Baan International, does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Baan International B.V., is a wholly subsidiary of SSA® Global Technologies™, Inc.

Trademark Acknowledgements

SSA Global, SSA and the SSA Global logo are trademarks of SSA Global Technologies, Inc. in the United States and/or other countries. Baan is a trademark of Baan International B.V., in the United States and/or other countries.

All other company, product, trade or service names referenced may be trademarks of their respective owners.

Publication Information

Document code U8617A US

Document group User Documentation

Document title Product Maintenance and Control
(PMC) Distributor User Guide

Subject SSA® ERP LN 6.1

Created on 6/29/2005 15:14:00

Table of Contents

Publication Information

About this document

Chapter 1 Introduction.....	1-1
PMC Introduction.....	1-1
PMC Benefits.....	1-2
Chapter 2 PMC overview.....	2-1
PMC Architecture.....	2-2
Where to find the PMC module.....	2-2
Dependencies.....	2-4
Individual solutions.....	2-5
Collections.....	2-5
Service Packs and patches.....	2-6
Justification of Service Packs and patches.....	2-6
Service Packs.....	2-7
Patches.....	2-9
Distributor's Policy.....	2-9
PMC Distributor functionality.....	2-10
PMC Recipient functionality.....	2-11
Chapter 3 Distributor procedure.....	3-1
Setup.....	3-1
Setup procedure.....	3-2

Parameters.....	3-2
base VRCs.....	3-3
base VRC combinations.....	3-3
To create updates.....	3-3
To create solutions.....	3-4
To create collections.....	3-6
To create patches.....	3-7
To create Service Packs.....	3-8
Service Pack development.....	3-10
System setup for standard products.....	3-11
System setup for derived products.....	3-12
Rules to upgrade derived products.....	3-14
Building Service Packs and the SSA Installation Wizard.....	3-15
Customizations.....	3-15
Patches and customizations.....	3-16
Service Packs and Customizations.....	3-18
Dependencies for customizations.....	3-21
Other.....	3-22

Chapter 4 Detailed session descriptions.....4-1

Parameters (ttpmc0100s000).....	4-1
Base VRC's (ttpmc0110m000).....	4-4
Base VRC Combinations (ttpmc0111m000).....	4-5
Base VRC Combinations (ttpmc0111m100).....	4-6
Base VRCs by Base VRC Combination.....	4-7
Solutions (ttpmc1100m000).....	4-8
Print Solution/Patch (ttpmc1400m000).....	4-12
Defects by Solution (ttpmc1110m000).....	4-13
Components by Solution (ttpmc1520m000).....	4-14
Component by Solution (ttpmc1120s000).....	4-16

Connect Components to PMC Solution (ttpmc1221m000).....	4-17
Print Components by Solution (ttpmc1420m000).....	4-25
Dependencies (ttpmc1140m000).....	4-26
Generate Dependencies (ttpmc1240s000).....	4-28
Print Dependencies (Multilevel) (ttpmc1440m000).....	4-29
Export Solution/Patch (ttpmc1200s000).....	4-30
Export Solution Multi-level (ttpmc1202s000).....	4-31
Collections (ttpmc1503m000).....	4-31
Collections (ttpmc1101s000).....	4-33
Solutions by Collection (ttpmc1151m000).....	4-34
Generate Collection (ttpmc1251s000).....	4-34
Validate Collection (ttpmc1402s000).....	4-35
Export Collection (ttpmc1210s000).....	4-36
Print Collection (ttpmc1403m000).....	4-37
Patches (ttpmc1501m000).....	4-38
Generate Patch (ttpmc1250s000).....	4-41
Password by Patch (ttpmc1105s000).....	4-42
Validate Patch (ttpmc1401s000).....	4-43
Patch Header (ttpmc1106m000).....	4-44
Packages by Patch Header (ttpmc1107m000).....	4-50
Generate Obsolete Solutions (ttpmc1252s000).....	4-52
Generate Patch for New Base VRC (ttpmc1253s000).....	4-53
Maintenance History (ttpmc1560m000).....	4-56
General Maintenance History (ttpmc1561m000).....	4-58
Print Maintenance History (ttpmc1460m000).....	4-59

Appendix A Distributor session summary.....A-1

PMC Distributor Sessions.....	A-1
-------------------------------	-----

Appendix B Glossary.....B-1

Index

About this document

Document summary

This document describes the Product Maintenance and Control (PMC) module for the Distributor of Service Packs and individual Solutions.

This document is also a composite of components that are available in the SSA Web Help.

For the content of this guide refer to the Table of Contents.

For the explanation of PMC terms refer to the Glossary.

To find a topic quickly please use the Index.

Comments?

We continually review and improve our documentation. Any remarks/requests for information concerning this document or topic are appreciated. Please e-mail your comments to documentation@ssaglobal.com.

In your e-mail, refer to the document number and title. More specific information will enable us to process feedback efficiently.

Chapter 1

Introduction

1

This chapter provides an introduction to the Product Maintenance and Control (PMC) module.

PMC Introduction

To help you manage software updates to your SSA ERP LN system, SSA Global offers the Product Maintenance and Control (PMC) Tool – an efficient, highly effective tool for managing functional software updates (Service Packs) and other software updates (Individual Solutions).

The PMC module manages the installation of Service Packs and Individual Solutions. The PMC module is delivered to all customers as part of the master CD-ROM of SSA ERP LN. The PMC module is used to check software updates for completeness and customization interference.

Without a tool such as PMC, software updates could be installed irrespective of other software updates that were already installed. Technically, an update that contains an earlier version of a software component than the version already present on your system, could be installed. In this case, you lose the updates contained in the original software components.

The PMC module contains two sub-modules.

- PMC distributor
- PMC recipient

PMC Benefits

The PMC module is an easy-to-use tool that enables you to install, configure, and run new software with confidence. The PMC module simultaneously addresses a wide range of software update challenges, including:

- Automatic checks for updates for any previous dependencies, to ensure that these exist on the system on which you install the update.
- Analysis of potential conflicts with system customizations at installation time, which enables the user to identify any conflicting customizations.
- Option to maintain copies of all previous software components. This feature enables the user to roll back updates easily to any point in history.
- Overview of all Service Packs and Individual Solutions, that are installed.

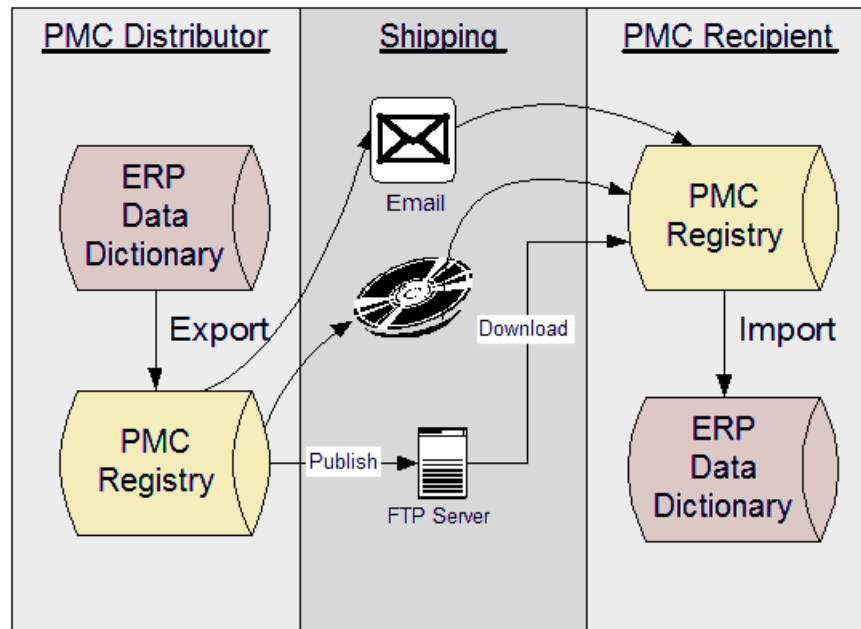
This chapter provides an overview of the Product Maintenance and Control (PMC) module.

This chapter contains the following sections:

- PMC Architecture
- Where to find the PMC module
- Dependencies
- Individual Solutions
- Collections
- Service Packs and Patches
- PMC Distributor functionality
- PMC Recipient functionality

PMC Architecture

The following figure illustrates the architecture of the PMC module:



The software developer creates or updates software components in the *Data Dictionary* of the developers ERP system. Software is packaged by the PMC distributor and stored in the PMC Registry. The software is published on an FTP server. Software can also be shipped on other media, such as CD-ROM, or can be sent by e-mail.

The customer can download the software from an FTP Server, or can order a software CD-ROM. PMC recipient builds or updates the PMC Registry on the customer's ERP system. Software in the customer's PMC Registry can be imported in the customer's data dictionary.

Where to find the PMC module

The PMC module is part of SSA Enterprise Server (Tools). The software supplier is the distributor of the solutions, and the user or customer of the applications is the recipient. The PMC module contains functionality and sessions for both the distributor and recipient role.

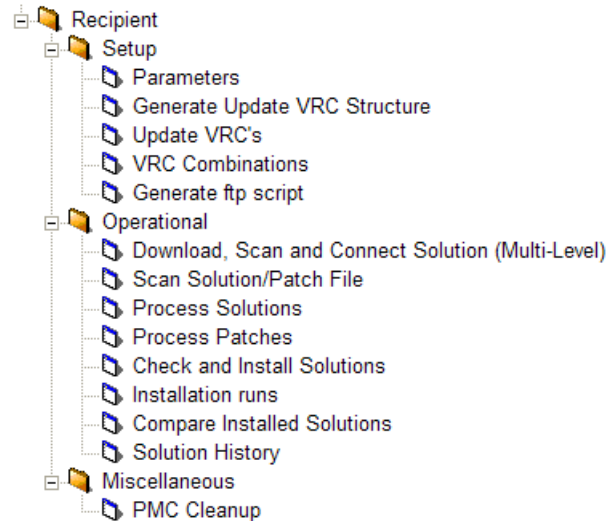
To navigate to the recipient part of the PMC module take the following steps:

1. Log on to SSA ERP.
2. On the ERP Menu Browser, click **Tools --> Software Installation --> Miscellaneous --> Recipient**.
3. Click **Setup, Operational, and Miscellaneous**.

Alternatively, you can take the following steps to navigate to the recipient part of the PMC module:

1. Log on to SSA ERP.
2. On the ERP Menu Browser, click **Tools --> Application Development --> Product Maintenance and Control --> Recipient**.
3. Click **Setup, Operational and Miscellaneous**.

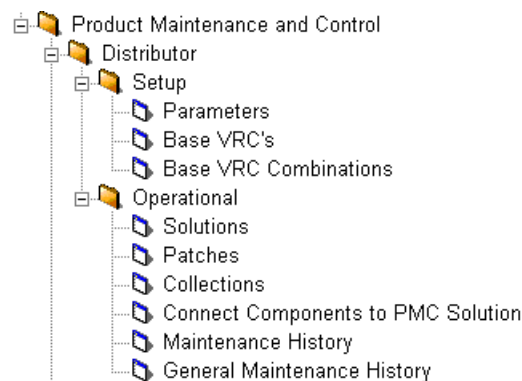
The following figure shows the Recipient sessions:



To navigate to the distributor part of the PMC module take the following steps:

1. Log on to SSA ERP.
2. On the ERP Menu Browser, click **Tools --> Application Development --> Product Maintenance and Control --> Distributor**.
3. Click **Setup and Operational**.

The following figure shows the Distributor sessions:



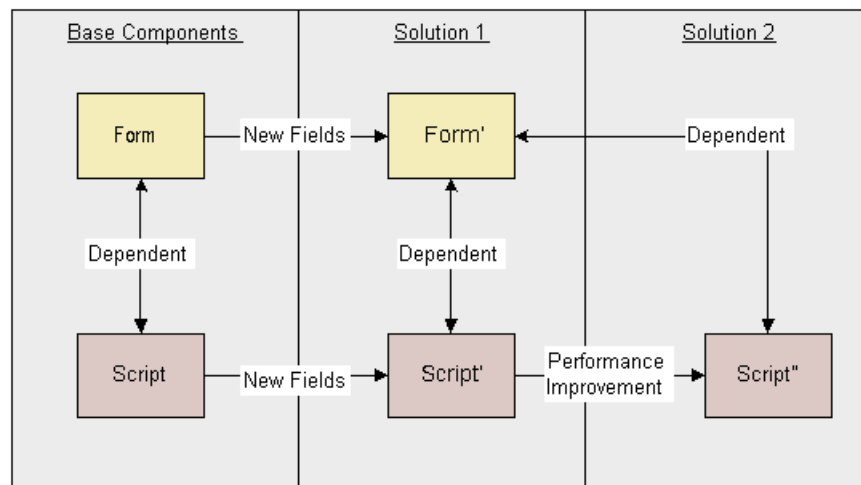
Dependencies

When the PMC recipient installs software updates, the recipient checks the presence of the required depending solutions.

Three dependency types exist:

- Pre-requisites
- Co-requisites
- Post-requisites

The following figure shows a pre-requisite dependency: solution 1 must be installed before solution 2.



The following section describes a customer scenario.

Example

A dependency requires you to install the previous version of a software component first. Suppose that you installed SSA ERP LN from the master CD-ROM without customizations or solutions. You receive an updated version of the Copy Sales Order (tdsls4201s000) session, which contains the session object, script, and forms that are interdependent of each other.

You decide not to install the dump and, a few months later, you discover that a change is required in the script to correct a problem that is keeping your sales force from selling an item. The change is sent as a dump that contains only the session object and script. You can install this dump only if you installed the previous version of the software component, because the dump requires the proper form of the first solution. PMC automatically installs the previous solution.

Suppose you deliberately decided not to install a solution, for example, because you do not know if the existing version of the Copy Sales Order (tdsls4201s000) session was customized. The PMC module provides

protection against this as well. In this case PMC informs you that the current version is customized. You cannot overwrite software, because PMC uses update VRC levels and maintains copies of all previous software components. This setup enables you to roll back updates easily to any point in history.

Individual solutions

A PMC Distributor creates and publishes individual solutions.

Individual solutions created for standard products of SSA Global are available for download 24 hours a day, seven days a week, from the SSA OnePoint Support Web site www.support.baan.com for SSA-owned products.

You can only download these solutions one by one.

After you download and scan a solution, the PMC recipient checks the dependencies with other solutions. In some cases, other dependent solutions might be missing at the installation system and must be downloaded and scanned, as well. The process to install individual solutions can take a long time if many dependent solutions must be downloaded and scanned.

You can use the Multilevel Download functionality to download all the dependent solutions, without user interaction, to simplify and speed up the download process.

Collections

Collection dumps enable customers to update their system with the latest software.

Collections are groups of all individual solutions that are released in a particular period.

These individual solutions are gathered at the PMC distributor side in a single dump file.

Collections for standard products of SSA Global are published on an FTP server.

You can scan collections in a single action. While you scan the collection with the Scan Solution/Patch File (tppmc2200s000) session, the dump file is split into all individual solutions that were included in the collection.

Afterwards, you can install the individual solutions.

You can use collection dumps in two ways: for pro-active installation or corrective installation.

- **Pro-active installation**

The user installs all individual solutions of the collection. The user runs the most up-to-date version of the software, including the latest released fixes.

- **Corrective installation**

The individual solutions of the collection are not installed. The solutions have status *Available* in the PMC registry. If the user runs into a problem, and a solution is advised for this problem, the user can install that solution immediately. All dependent solutions will be installed automatically.

Service Packs and patches

Periodically, individual solutions can be bundled into Service Packs or patches. In general, these types of updates contain solutions created in a larger time period than a collection. In the PMC Tool, the term 'patch' is used both for Service Packs and patches. The patch entity is known at both the PMC distributor and PMC recipient side. Service Packs and patches are an indivisible set of solutions. You cannot install or uninstall individual solutions that belong to a Service Pack or a patch by the PMC recipient. You can only install or uninstall entire Service Packs and patches. Dependencies between Service Packs and patches can exist.

The subsequent sections describe the justification, characteristics, and differences of Service Packs and patches.

Justification of Service Packs and patches

During the life cycle of a product, in general, two types of changes are implemented in the product:

- Corrective fixes for defects
- Functional enhancements to further enrich the product

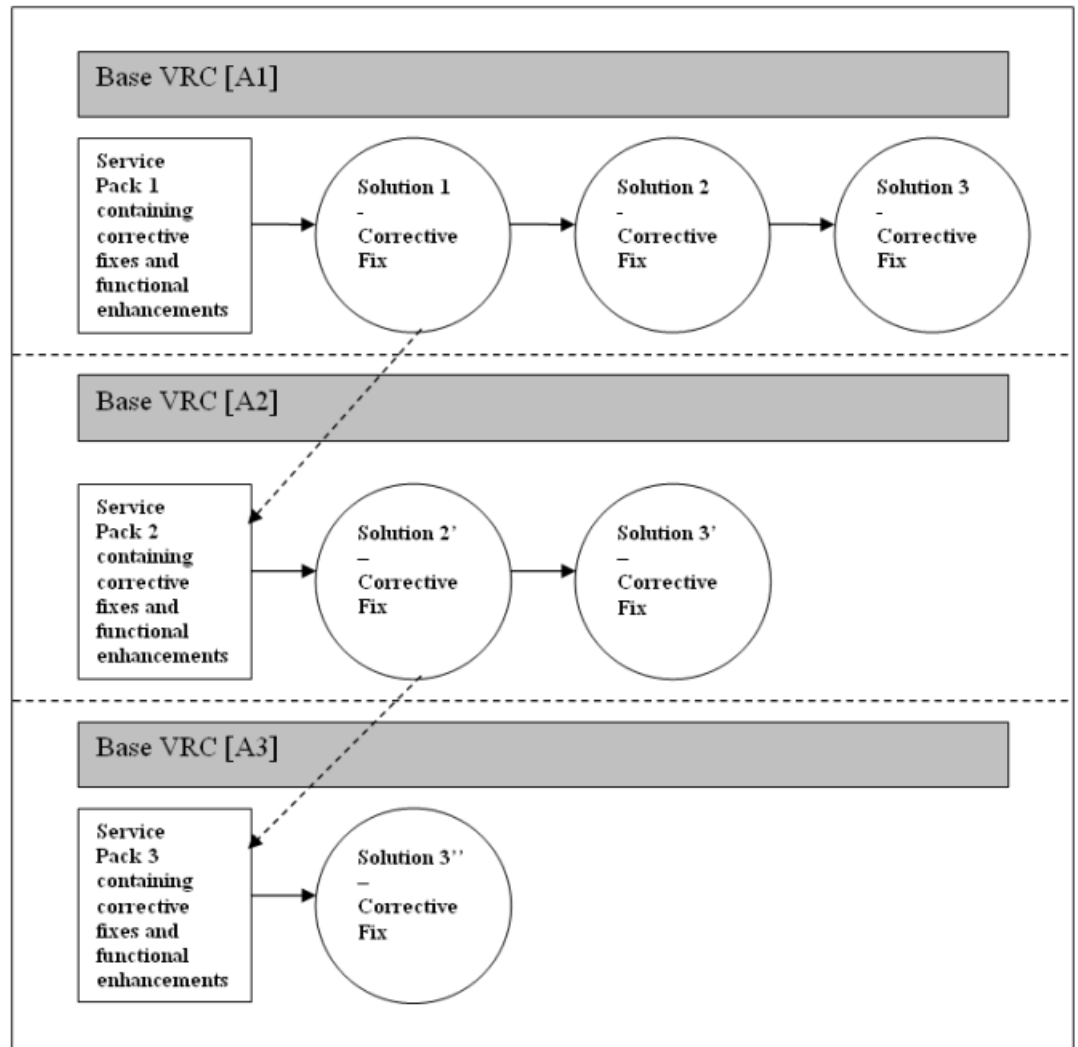
Functional enhancements are often accompanied by changes in the data model, changes in the user interface, and changes in the behavior of the application. In general, functional enhancements require a careful preparation before installation in an operational environment. PMC generates dependencies between solutions, that contain the same components to guarantee that the solutions are installed in the correct order and no necessary solutions are skipped. In some cases, users must first install a number of functional enhancements before the users can install a corrective fix for a relative small defect. Installation of the functional enhancements can be required at unexpected and undesirable moments.

Service Packs enable you to separate functional enhancements from the flow of corrective fixes. This separation enables users to adopt and implement functional enhancements in a more controlled way. Patches do not offer this possibility.

Service Packs

Service Packs enable you to separate corrective fixes from functional enhancements. Multiple parallel maintenance baselines exist for the product. Every new baseline is linked to a unique base VRC. Functional enhancements are, in principle, only implemented at the start of a new baseline.

The Service Packs mechanism is as follows:



Service Pack 1 is linked to base VRC A1. The Service Pack contains a number of corrective fixes and some functional enhancements. Corrective maintenance is performed for this Service Pack, which results in solutions 1, 2, and 3 in the previous figure.

In parallel, the software distributor develops Service Pack 2, which is linked to base VRC A2. Whenever possible, the distributor immediately ports and includes corrective solutions created on top of Service Pack 1 in Service

Pack 2. Refer to solution 1 and the dashed arrow in the previous figure. These ported solutions are registered in Service Pack 2 as so-called obsolete solutions.

The same process repeats when the distributor develops Service Pack 3, which is linked to base VRC A3.

After the release of Service Pack 2 and 3, the distributor also creates corrective solutions on top of these Service Packs.

When Service Pack 2 is being installed at a recipient, the PMC Tool checks whether all solutions that were installed on top of Service Pack 1 are also available for Service Pack 2.

These solutions can be available in two different shapes for Service Pack 2:

- **As an obsolete solution**
Included in Service Pack 2. Refer to solution 1 and the dashed arrow in the previous figure. These obsolete solutions are integrated in Service Pack 2 and do not have a separate physical solution dump.
- **As an individual solution**
Available on top of Service Pack. Refer to solutions 2' and 3' in the previous figure. These so-called missing solutions are not included in Service Pack 2 and do have a physical solution dump.

Note: If Service Pack 1 is installed in multiple update VRCs, the PMC Tool performs the missing solution check in all update VRCs, that contain Service Pack 1.

The PMC Tools will refuse to install Service Pack 2 if not all solutions installed on top of Service Pack 1 are also available for Service Pack 2. Otherwise, old defects would come back after you install Service Pack 2. The distributor is responsible for ensuring that all solutions of the preceding Service Pack are also available for the subsequent Service Pack before publishing the subsequent Service Pack.

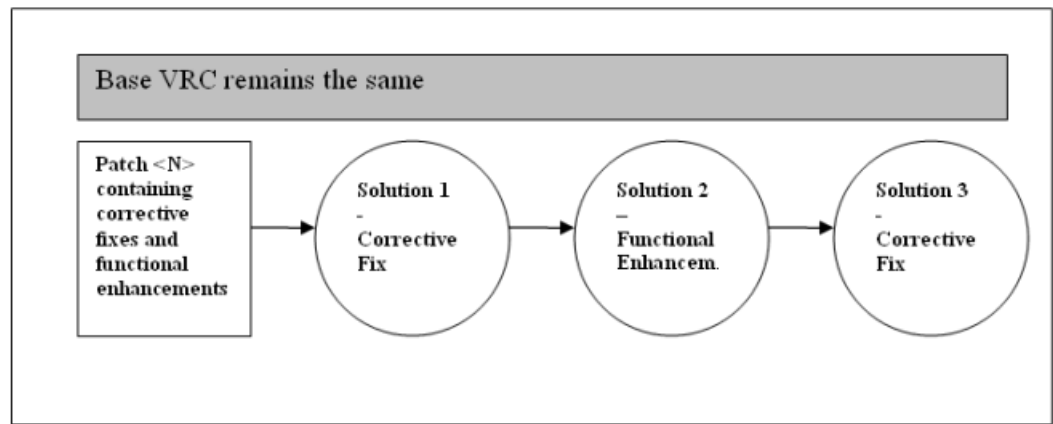
Service Packs are cumulative. Every new Service Pack also includes all preceding Service Packs. This enables you to skip the installation of Service Packs at the PMC recipient. For example, you can immediately upgrade from Service Pack 1 to Service Pack 3 without installing Service Pack 2 in between. The installation process is optimized in such a way that only components that are changed as compared to the current installed Service Pack will be installed.

You can install Service Packs in an existing update VRC. Therefore, creating a new update VRC is unnecessary. As a result, the VRC structure at a recipient system can remain unchanged. However, the base VRC linked to the update VRC will change. If desired, you can, of course, also install a new Service Pack in a new update VRC.

Patches

Patches do not enable you to separate corrective fixes from functional enhancements. Only one maintenance baseline of the product exists. Changes are always implemented in the latest version of a software component. The base VRC remains the same. Functional enhancements will be included in the chain of depending corrective solutions.

The Patch mechanism is as follows:



To install solution 3 in the this picture, you must first install solution 2, which contains a functional enhancement.

Patches are not cumulative, which means you cannot skip the installation of Patches at a recipient's location. For example, before you install Patch 3, you always must install Patch 2 in advance. You cannot immediately upgrade from Patch 1 to Patch 3.

Distributor's Policy

The software distributor decides if Service Packs or Patches are being applied. A mix of both types is also possible:

- When applying Service Packs, the distributor can decide to release a functional enhancement as an individual PMC solution on top of one of the released Service Packs.
- When applying Service Packs, the distributor can decide to periodically also release Patches on top of the Service Pack.

When applying Service Packs, the distributor's policy determines how many base VRCs are maintained in parallel and how long the base VRCs are maintained.

PMC Distributor functionality

The software developer generates additional software and makes software changes on the existing SSA ERP LN software packages.

The supplier of the SSA ERP LN software uses the PMC Distributor functionality to manage the software updates and prepares those updates for delivery to the customers.

The following is a process overview of the PMC Distributor module.

- **Create individual solutions**
 - a. Define a unique identifier for the solution and a brief description.
 - b. Link one or more software components to the solution.
 - c. Define or generate the dependencies on other solutions.
 - d. Export the solution, the software dumps are now created.
 - e. Release the solution, the solution is now ready for delivery.
- **Create collections**
 - a. Define a unique identifier for the collection and a brief description.
 - b. Link one or more individual solutions to the collection.
 - c. Export the collection.
 - d. Release the collection.
- **Create patches**
 - a. Define a unique identifier for the patch and a brief description.
 - b. Link one or more individual solutions to the patch.
 - c. Define or generate the dependencies on other patches.
 - d. Validate the patch.
 - e. Export the patch.
 - f. Release the patch.
- **Create service packs**
 - a. Define a new base VRC for the service pack.
 - b. Define a unique identifier for the service pack and a brief description.
 - c. Define the dependency on previous service packs.
 - d. Generate the service pack, in other words, link all components in a specified VRC to the service pack.
 - e. Validate the service pack.
 - f. Export the service pack.
 - g. Release the service pack.
- **Additional distributor functionality**
 - Multilevel export.

- SCM integration.
- Maintenance History.

PMC Recipient functionality

Customers use the PMC Recipient module to install the SSA ERP LN software updates on their ERP system.

The following is an overview of the PMC Recipient module.

- **Download solutions**
You can download solutions from the FTP server or copy the solutions from a medium such as a CD-ROM.
- **Scan and connect solutions**
The downloaded solution files can be scanned. In other words, the user must extract and store the solutions into the PMC registry. The extracted solutions are also connected to an update VRC.
- **Process solutions**
 - a. Check to install. Report the following: Pre- and post-installation instructions, customized components, and missing dependent solutions.
 - b. Install the solutions: Store the software components in the data dictionary, report additional post-installation instructions.
- **Additional recipient functionality**
 - Uninstall
 - Multilevel download
 - Compare installed solutions
 - Solution History
 - PMC cleanup
 - Copy PMC registry
 - View installation runs.

Chapter 3

Distributor procedure

3

This chapter describes the PMC Distributor procedures. This chapter contains the following sections:

- Setup
- To create updates
- Service Pack development
- Customizations

Setup

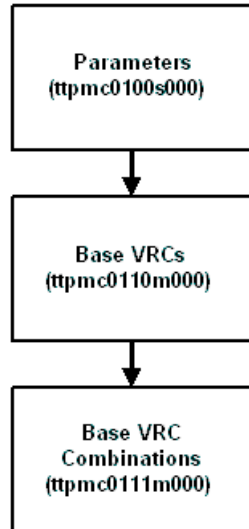
This section describes the procedures for the update distributor to set up the PMC administration.

The setup consists of the following sections:

- Setup procedure
- Parameters
- Base VRCs
- Base VRC combinations

Setup procedure

This section provides a flowchart that describes the steps of the distributor set-up procedure. The subsequent sections provide a detailed explanation of the procedure.



Distributor Set-up Procedure

Parameters

The very first step in the distributor set-up procedure is to define the PMC parameters.

The following two group boxess are available:

- **Recipient**
Even if you do not use the PMC recipient part of the PMC module, for technical reasons, you must fill in the parameters in this group box.
- **Distributor**
The PMC distributor part of the PMC module uses these parameters, and therefore you must fill in these parameters.
Most of the parameters are directories on the operating system on which the solution dumps will be stored.

For details, refer to the *Parameters (ttpmc0100s000)* (p. 4-1) session.

Note

Because the definition of the parameters is a once only step, you must be aware of the impact if you change the parameters afterwards.

If you change the directories on the operating system, you must move the directories and files according to the new directory specifications.

base VRCs

You must define a base VRC before you can create and link solutions, collections, patches, and Service Packs to the base VRC.

A base VRC is an administrative identifier for a software product and is not a physical VRC, but must always be connected to a physical VRC, which contains the software components.

The PMC distributor should connect a base VRC to an export VRC, the PMC recipient should connect a base VRC to an update VRC.

Note

A Service Pack always starts a new base VRC.

For details, refer to the *Base VRC's (ttpmc0110m000)* (p. 4-4) session.

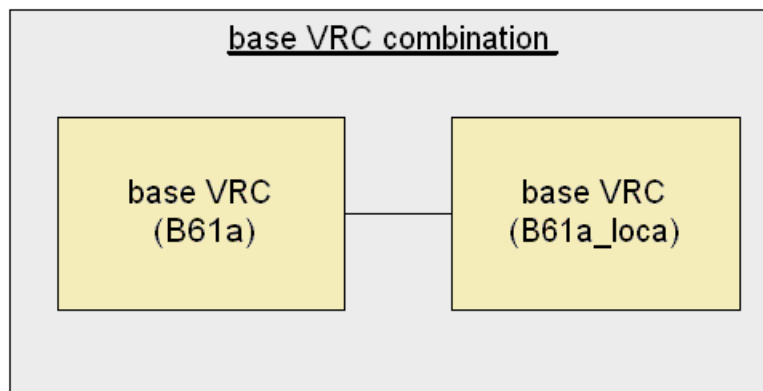
base VRC combinations

Base VRC combinations are required to define relations between base VRCs.

Any base VRC must always be a member of a base VRC combination, even if no relations to other base VRCs exist.

If you have not linked a base VRC to a base VRC combination, you cannot create updates.

The following example shows a base VRC combination with two base VRCs with a relation:



For details refer to the *Base VRC Combinations (ttpmc0111m000)* (p. 4-5) session.

To create updates

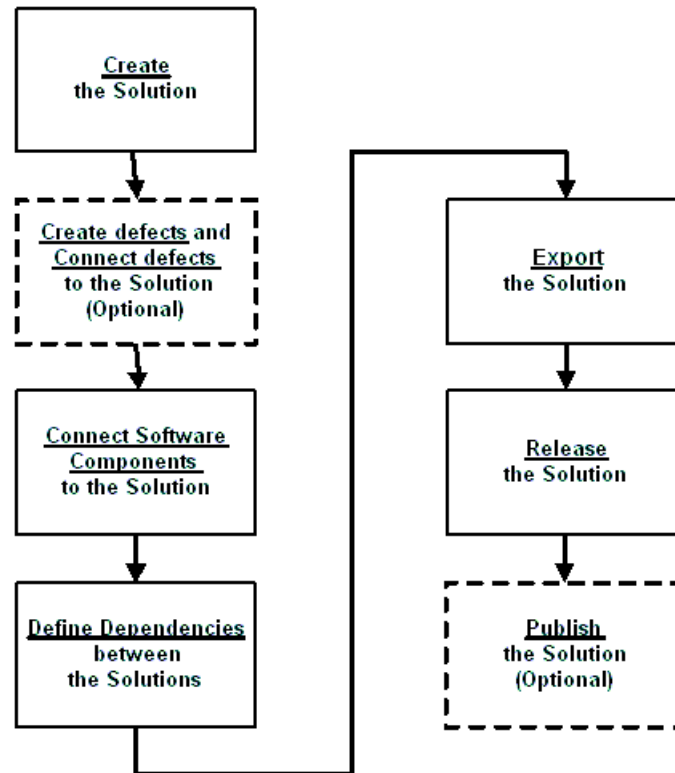
This section describes the following PMC distributor procedures to create software updates:

- To create solutions
- To create collections

- To create patches
- To create Service Packs

To create solutions

This section describes the procedure steps to create solutions:



You can create or maintain solutions in the *Solutions (ttpmc1100m000)* (p. 4-8) session.

This session is the central session from which you can maintain all aspects of solutions.

The initial status of a solution after creation is In Progress.

If you create a solution, you must take the following steps:

- **Create defects and connect the defects to the solution**
You can create and connect the defects in the *Defects by Solution (ttpmc1110m000)* (p. 4-13) session.
This step is optional.
- **Connect software components to the solution**
To connect software components to the solution, use the *Components by Solution (ttpmc1520m000)* (p. 4-14) session and the *Component by Solution (ttpmc1120s000)* (p. 4-16) session.

You can also connect software components to a solution in the *Connect Components to PMC Solution* (ttpmc1221m000) (p. 4-17) session.

- **Define dependencies between solutions**

To define dependencies between solutions, run the *Dependencies* (ttpmc1140m000) (p. 4-26) session.

- **Export the solution**

To export a solution, you must use the *Export Solution/Patch* (ttpmc1200s000) (p. 4-30) session.

- **Release the solution**

To release a solution, you must change the *Status* from Exported to Released in the *Solutions* (ttpmc1100m000) (p. 4-8) session.

You can only release a solution if the *Status* of the solution is Exported.

- **Publish the solution**

To publish a solution, select the *Published* check box in the *Solutions* (ttpmc1100m000) (p. 4-8) session.

You can only publish a solution if the *Status* of the solution is Released.

This step is optional.

- **View or print solutions**

To view solutions, you must use the *Components by solution* (ttpmc1521m000) session.

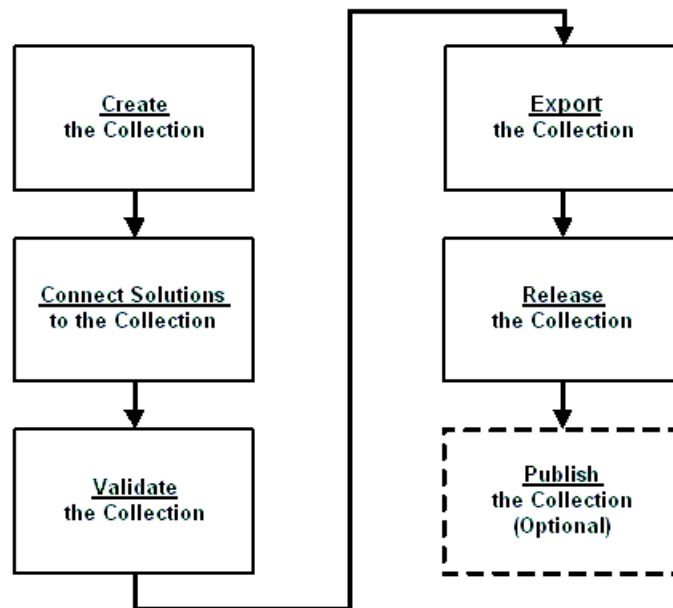
To print solutions, you must use the *Print Components by Solution* (ttpmc1420m000) (p. 4-25) session.

- **View the solution history.**

To view the solution history, you must use the *Maintenance History* (ttpmc1560m000) (p. 4-56) session.

To create collections

This section shows the procedure steps to create collections:



You can create or maintain collections in the *Collections (ttpmc1503m000)* (p. 4-31) session.

This session is the central session from which you can maintain all aspects of collections.

The initial status of a collection after creation is In Progress.

After you create a collection, take the following steps:

- **Connect solutions to the collection**
You can connect solutions to a collection in the *Solutions by Collection (ttpmc1151m000)* (p. 4-34) session.
- **Validate the collection**
You can validate collections in the *Validate Collection (ttpmc1402s000)* (p. 4-35) session.
- **Export the collection**
You can export a collection in the *Export Collection (ttpmc1210s000)* (p. 4-36) session.
- **Release the collection**
To release a collection, you must change the *Collection Status* from Exported to Released in the *Collections (ttpmc1101s000)* (p. 4-33) session.
You can only release a collection if the *Collection Status* is Exported.

- **Publish the collection**

To publish a collection, select the *Published* check box in the *Collections* (*ttpmc1101s000*) (p. 4-33) session.

You can only publish a collection if the *Collection Status* is Released.

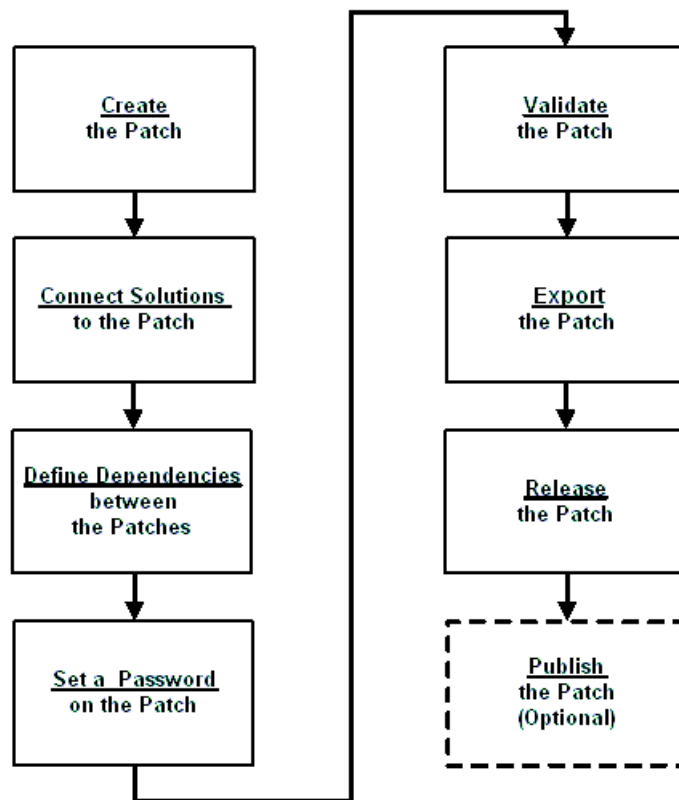
This step is optional.

- **View the collection history**

To view the collection history, use the *Maintenance History* (*ttpmc1560m000*) (p. 4-56) session.

To create patches

This section describes the procedure to create patches:



You can create or maintain patches in the *Patches* (*ttpmc1501m000*) (p. 4-38) session.

This session is the central session from which you can maintain all aspects of patches.

You must clear the New Base VRC check box. Otherwise, the patch is a Service Pack.

The initial status of a patch after creation is In Progress.

After you create a patch, you can take the following steps:

- **Connect solutions to the patch**
To connect solutions to a patch, you must use either the *Solutions* (*ttpmc1100m000*) (p. 4-8) session or the *Generate Patch* (*ttpmc1250s000*) (p. 4-41) session.
- **Define dependencies between patches**
To define dependencies between patches, you must use the *Dependencies* (*ttpmc1140m000*) (p. 4-26) session.
- **Set or change the password of the patch**
To set or change the password of a patch, you must use the *Password by Patch* (*ttpmc1105s000*) (p. 4-42) session.
- **Validate the patch**
To validate a patch, you must use the *Validate Patch* (*ttpmc1401s000*) (p. 4-43) session.
- **Export the patch**
To export a patch, you must use the *Export Solution/Patch* (*ttpmc1200s000*) (p. 4-30) session.
- **Release the patch**
To release a patch, you must change the *Status* from Exported to Released in the *Patches* (*ttpmc1501m000*) (p. 4-38) session.
You can only release a patch if the *Status* is Exported.
- **Publish the patch**
To publish a patch, select the *Published* check box in the *Patches* (*ttpmc1501m000*) (p. 4-38) session.
You can only publish a patch if the *Status* of the patch is Released.
This step is optional.
- **View the patch history**
You can view the patch history in the *Maintenance History* (*ttpmc1560m000*) (p. 4-56) session.

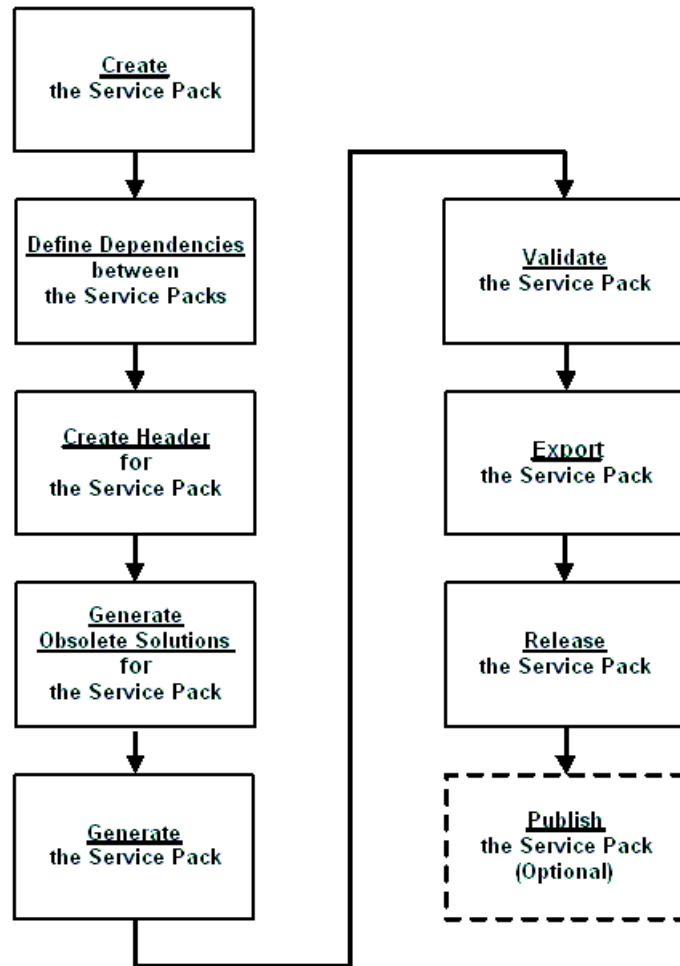
To create Service Packs

Note

The process to create Service Packs requires a specific setup of your development environment.

The *Service Pack development* (p. 3-10) section describes in detail how the setup must look.

This section describes the procedure steps to create Service Packs:



You can create or maintain Service Packs in the *Patches (ttpmc1501m000)* (p. 4-38) session.

This session is the central session from which you maintain all aspects of Service Packs.

You must select the New Base VRC check box, otherwise, the patch is not a Service Pack.

The initial status of a Service Pack after creation is **In progress**.

If you have created a Service Pack, you must take the following steps:

- **Define dependencies between Service Packs**
Define dependencies between Service Packs in the *Dependencies (ttpmc1140m000)* (p. 4-26) session.

- **Create the Service Pack header**
Create the Service Pack header in the *Patch Header (ttpmc1106m000)* (p. 4-44) session.
- **Generate obsolete solutions for the Service Pack**
Generate obsolete solutions for the Service Pack in the *Generate Obsolete Solutions (ttpmc1252s000)* (p. 4-52) session.
- **Generate the Service Pack**
Generate a Service Pack in the *Generate Patch for New Base VRC (ttpmc1253s000)* (p. 4-53) session.
- **Validate the Service Pack**
Validate a Service Pack in the *Validate Patch (ttpmc1401s000)* (p. 4-43) session.
- **Export the Service Pack**
Export a Service Pack in the *Export Solution/Patch (ttpmc1200s000)* (p. 4-30) session.
- **Release the Service Pack**
To release a Service Pack, change the *Status* from Exported to Released in the *Patches (ttpmc1501m000)* (p. 4-38) session.
You can only release a Service Pack if the *Status* of the Service Pack is Exported.
- **Publish the Service Pack**
To publish a Service Pack, you must select the *Published* check box in the *Patches (ttpmc1501m000)* (p. 4-38) session.
You can only publish a Service Pack if the *Status* of the Service Pack is Release.
This step is optional.
- **View the Service Pack history**
You can view the Service Pack history in the *Maintenance History (ttpmc1560m000)* (p. 4-56) session.

Service Pack development

This section describes various aspects applicable to the Service Pack development and build process.

This section describes the following topics:

- System setup for standard products
- System setup for derived products
- Rules for upgrading derived products
- To build Service Packs and the SSA Installation Wizard

System setup for standard products

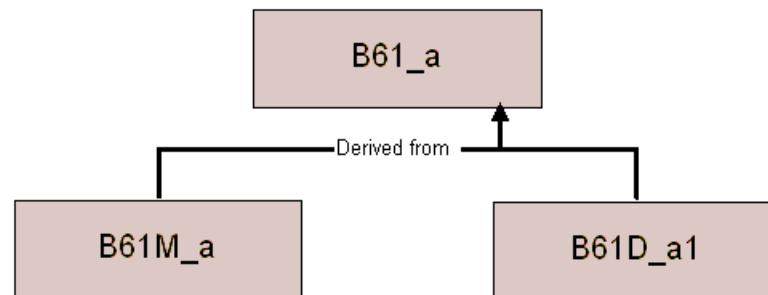
The subsequent setup assumes that the distributor side includes a combined development/maintenance environment. This combined environment supports the maintenance of already released Service Packs, development of a new Service Pack, and the physical creation of solutions and Service Packs.

Note

The following examples use the following conventions for VRC naming:

- **Version: B61<X>**
X = **M** aintenance, **D** evelopment
- **Release: a<Y>**
Y = Service Pack number
- **Customer: <zzzz>**
zzzz = extension/localization code

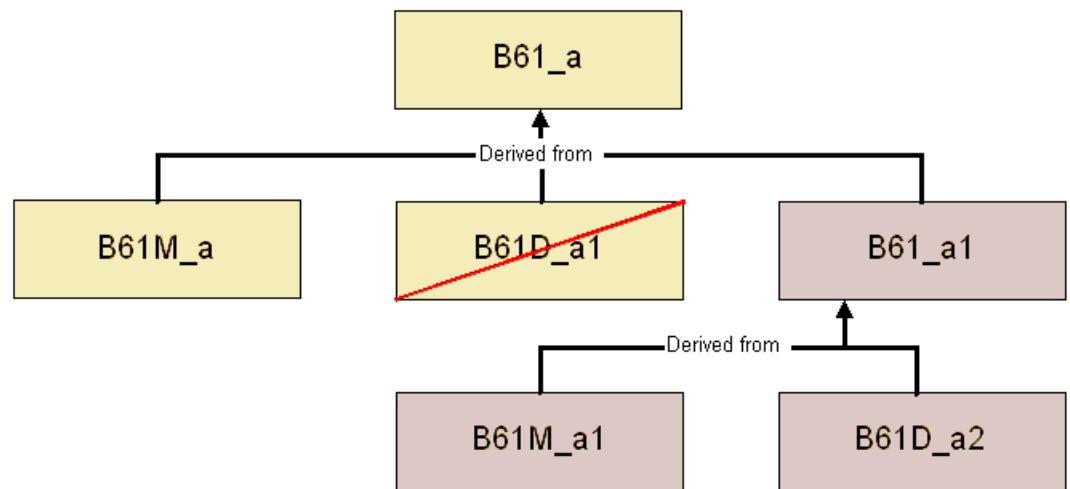
The following figures illustrate the results over time:



As soon as the initial master product, B61_a, which for the sake of simplicity is referred to simply as Service Pack 0 in the remainder of this document, is released, the maintenance starts for B61_a in the maintenance VRC B61M_a. The development of the new Service Pack is carried out in B61D_a1. All fixes in B61M_a are ported to B61D_a1. In the PMC registry of B61D_a1, these fixes are registered as obsolete solutions. After B61D_a1 is frozen, a final Service Pack is created for the Service Pack and installed in the same environment to enable you to perform maintenance on the Service Pack.

Note

The SCM revision data is not available in the Service Pack VRC B61_a1, because the export/import software does not support the transfer of revisions. Because the original development VRC B61D_a1 is still present, you can still view revisions.



Currently, you can maintain B61M_a and B61M_a1, Service Pack 0 and Service Pack 1, in parallel. You must port bug fixes that are created for Service Pack 0 between the freeze moment and the time of installation of Service Pack 1 first. Service Pack 2, B61D_a2, is currently in development. You can repeat this cycle unlimited times for each new Service Pack.

System setup for derived products

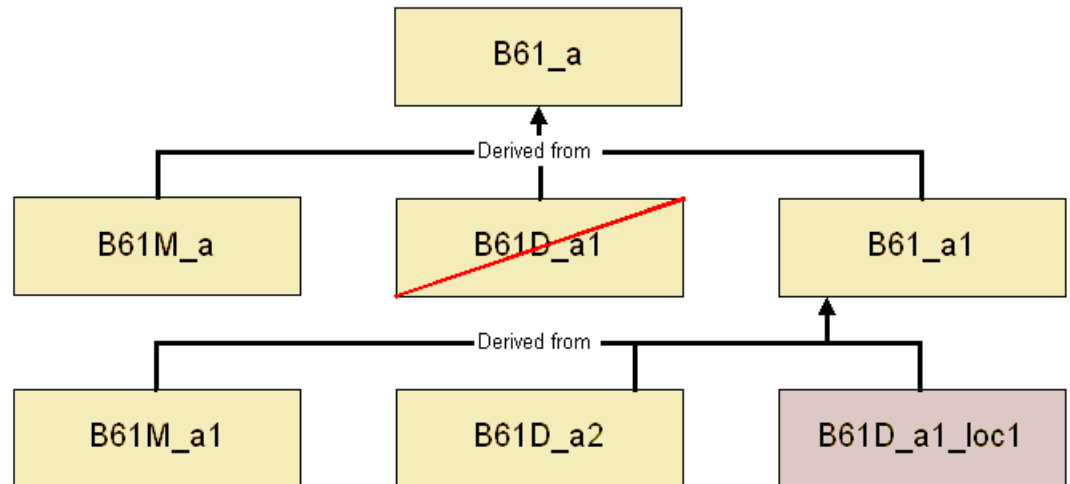
The setup becomes a bit more complex if derived products such as localizations or extensions enter the scope.

Note

Customizations are also a type of derived products. For customizations, the situation differs somewhat. The section *Customizations* (p. 3-15) describes how to deal with customizations.

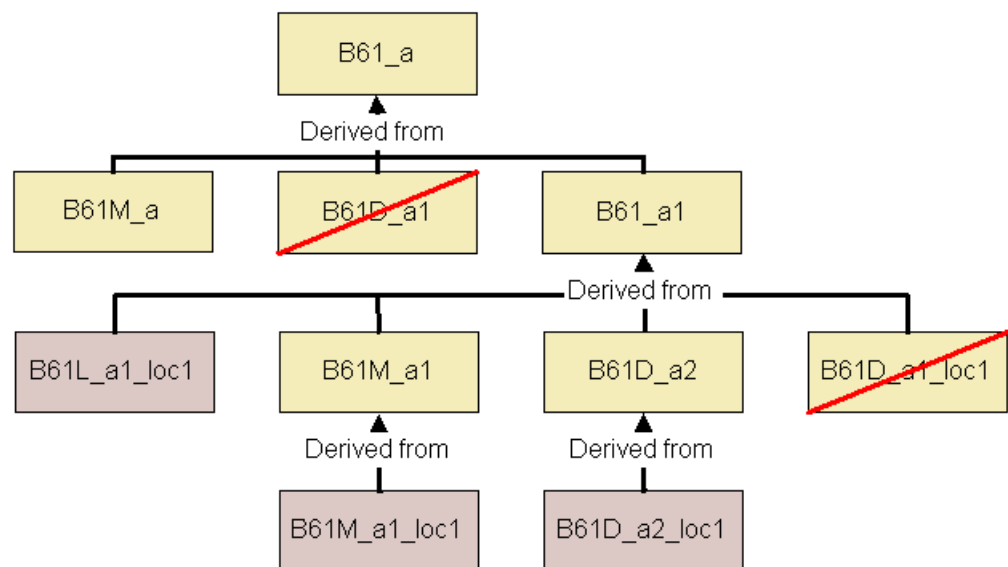
Suppose you must create a localization on top of Service Pack 1.

A new VRC is created on top of Service Pack 1, B61D_a1_loc1:



The localization is built on top of the standard Service Pack and is delivered as a Service Pack in PMC. Therefore, you must not create master media for derived products. As soon as Service Pack 0 for the localization is available, the Service Pack is installed in the maintenance VRC, B61M_a1_loc1.

The localization is also copied to a development VRC B61D_a2_loc1:



The first action that you must perform is to merge all bug-fixes of the standard VRC B61M_a1 into the localization maintenance VRC B61M_a1_loc1 and to create PMC solutions for the localization.

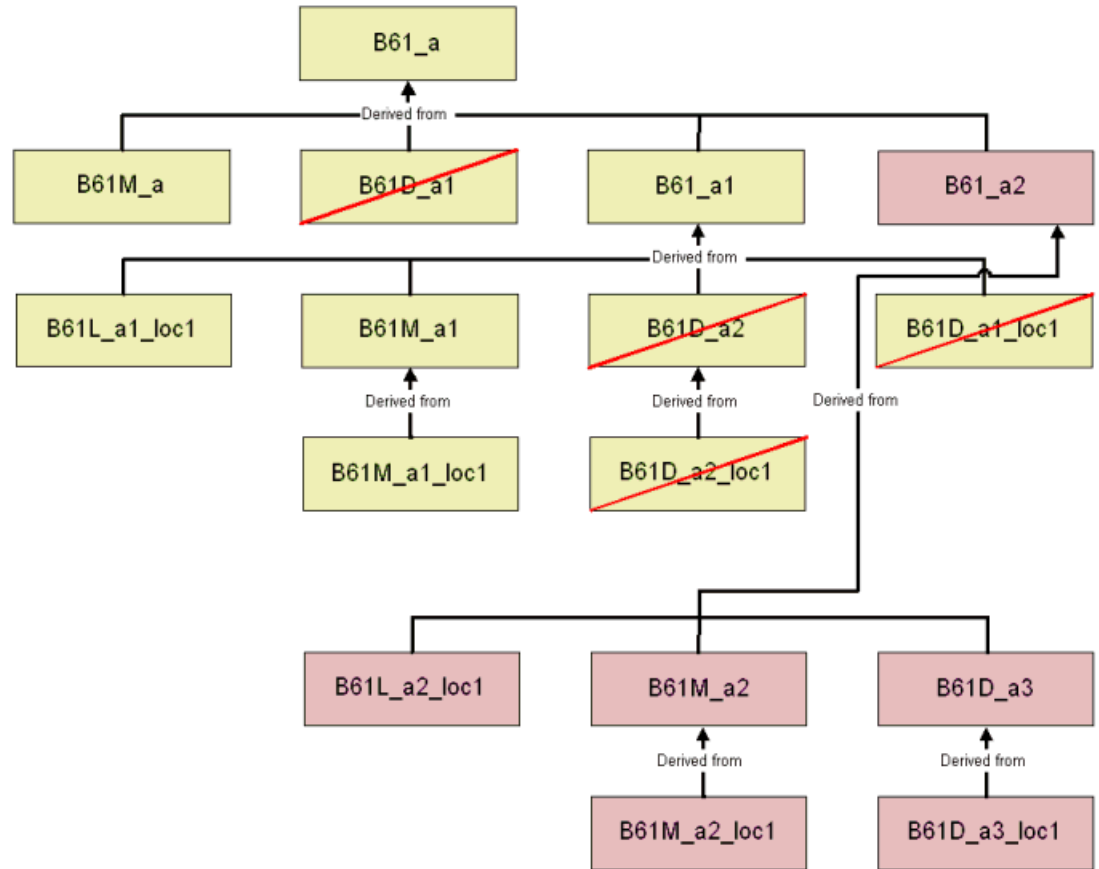
In addition, you must merge the B61D_a2 functional enhancements to B61D_a2_loc1.

From this moment on, the localization follows the same Service Pack cycle as the standard software.

Repeat the same approach for Service Pack 2.

The physical Service Packs for the standard software and the localization are created and installed in a separate VRC tree B61_a2 and B61L_a2_loc1. New maintenance and development VRCs are created.

The maintenance and development VRC for the localization are populated with the components from the installed Service Pack B61L_a2_loc1.



Rules to upgrade derived products

The following rules apply with respect to the creation of Service Packs for derived products:

- You must only build Service Packs for derived products on top of Service Packs of the parent product. Do not build Service Packs on top of a Service Pack plus a number of individual solutions of the parent product. The reason for this is because you can only define PMC dependencies between the Service Pack of the standard and the derived products.

You cannot define PMC dependencies from the Service Pack of the derived product to individual solutions of the parent product.

- The Service Pack frequency of derived products can be greater than the Service Pack frequency of the parent product. Therefore, you can build multiple derived Service Packs on top of a single Service Pack of the parent product.
- The Service Pack frequency of derived products cannot be lower than the Service Pack frequency of the parent product. If you do not update a derived product in time, you will most likely face various compatibility issues between the parent and the derived product.

Building Service Packs and the SSA Installation Wizard

PMC Service Packs are installable units that you can install by means of the SSA Installation Wizard. Metadata that is included in a Service Pack controls the behavior of the SSA Installation Wizard. The following aspects are relevant if you want to make your Service Pack suitable for installation with the SSA Installation Wizard:

- Before you build a Service Pack, be sure to fill in the installable unit key, name, and version are filled at your PMC base VRC. If you do not fill in these values, the SSA Installation Wizard will not recognize the installable unit. For more details, refer to the *Base VRC's (ttpmc0110m000) (p. 4-4)* session.
- Before you export a Service Pack, be sure that the Service Pack header contains the correct properties. Based on the properties defined in the Service Pack header, the SSA Installation Wizard can automatically perform particular actions at a recipient system, such as create new package combinations, create new update VRCs, and link new update VRCs to an existing or new package combination. For more details, refer to the *Patch Header (ttpmc1106m000) (p. 4-44)* session.

Customizations

This section describes various aspects relevant to the development and maintenance process of customizations.

Customizations are, in most cases, built by organizations other than the producer of the related parent standard product.

The vendor who builds customizations installs the standard parent product in an update VRC.

The customization is built on top of the installed parent standard product.

This situation is specific in the sense that PMC serves both as recipient and distributor in such a development environment.

The following sections describe best practices how to develop and maintain customizations.

- Patches and customizations
- Service Packs and customizations
- Dependencies for customizations
- Other

Patches and customizations

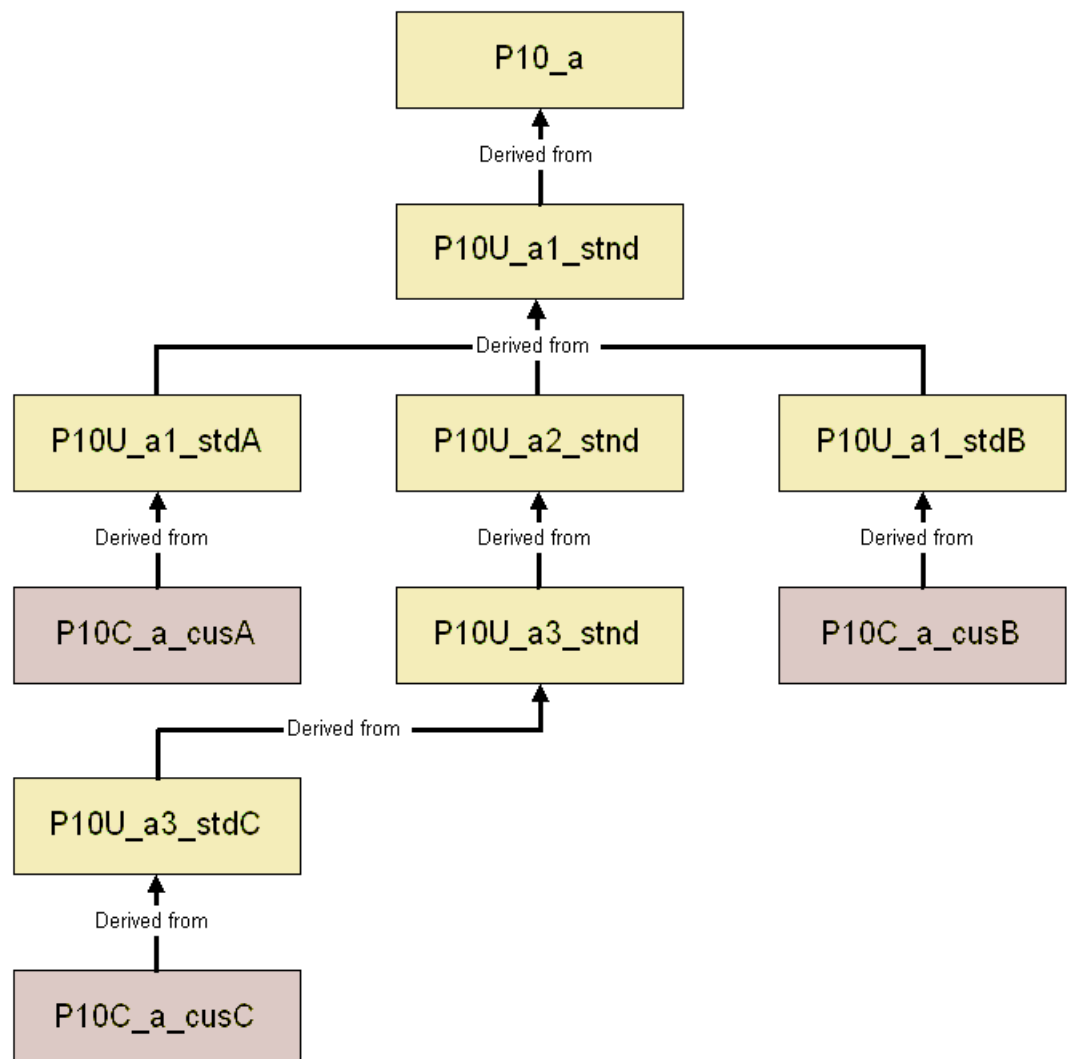
In some cases, the vendor of the standard parent product applies patches rather than Service Packs as a method to provide incremental updates to the market. As opposed to Service Packs, patches are not cumulative. The installation of a patch in an update VRC requires that you also install all preceding update VRCs in the same update VRC. This issue has consequences for the VRC setup at the distributor.

In most cases, you use a single environment to develop and to maintain multiple customizations. You build every customization on a specific patch. In addition, you must periodically upgrade existing customizations to a newer patch. As a result, you must have multiple Service Packs of the standard parent product installed in separate VRCs. The following method is the best way to achieve this:

Install patch 1 of the standard parent product in update VRC P10U_a1_std. When patch 2 is available, install the patch in a new, separate update VRC P10U_a2_std, which is derived from P10U_a1_std. Patch 2 will have patch 1 as a prerequisite. You cannot install patch 2 in the new update VRC P10U_a2_std because patch 1 is not installed in this new update VRC. Therefore, run the Copy Solution Registry to Derived Update VRC (ttpmc2290m000) session and copy the registry of update VRC P10_a1_std into P10_a2_std. After you perform this step, you can install patch 2 directly in the P10U_a2_std update VRC. You can then repeat this same process for subsequent patches. The P10U_a<n>_std VRCs must only contain net patches. No additional solutions must be installed in these VRCs that do not belong to the relevant patch.

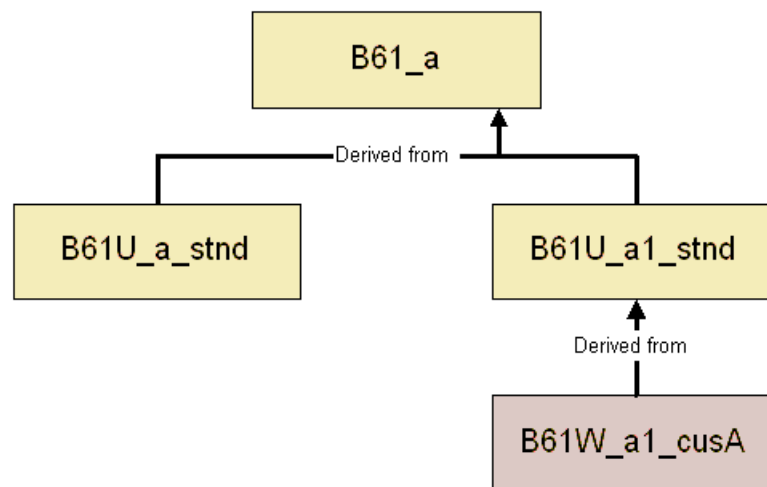
At this point, you have a VRC tree in which the net patches are installed in separate VRCs. In general, customizations are developed based on a net patch. However, in some cases customizations are not derived from a net patch, or a customer might want to install parent standard product solutions that have components which overlap with the customization. As a result, the customer must have the option to derive the customization VRC from any possible maintenance stage of the parent standard product. The P10U_a<n>_std update VRCs are introduced to achieve this. For every customization an intermediate VRC layer is introduced in which standard solutions can be installed that do not belong to the underlying patch. You can again use the Copy Solution Registry to Derived Update VRC (ttpmc2290m000) session to populate the PMC registry for the P10U_a<n>_std update VRCs.

The customization VRCs P10C_a_cusA, P10C_a_cusB, and P10C_a_cusC represent various various customizations that are derived from various patches. Suppose customization P10C_a_cusA is developed based on the net patch 1. During development of the customization, the P10U_a1_stdA VRC will still be empty. After the customization is released and the customer installs the customization, the customer might want to install solutions of the standard parent product that also require an update of the customization. In this case, the customization vendor can install the solutions of the parent standard product in the P10U_a1_stdA VRC, port the solution to the P10C_a_cusA and provide a corresponding solution for the customization VRC. Installation of solutions in the P10U_a1_stdA VRC will only have impact on the P10C_a_cusA customization and will not impact the other customizations in the environment.



Service Packs and Customizations

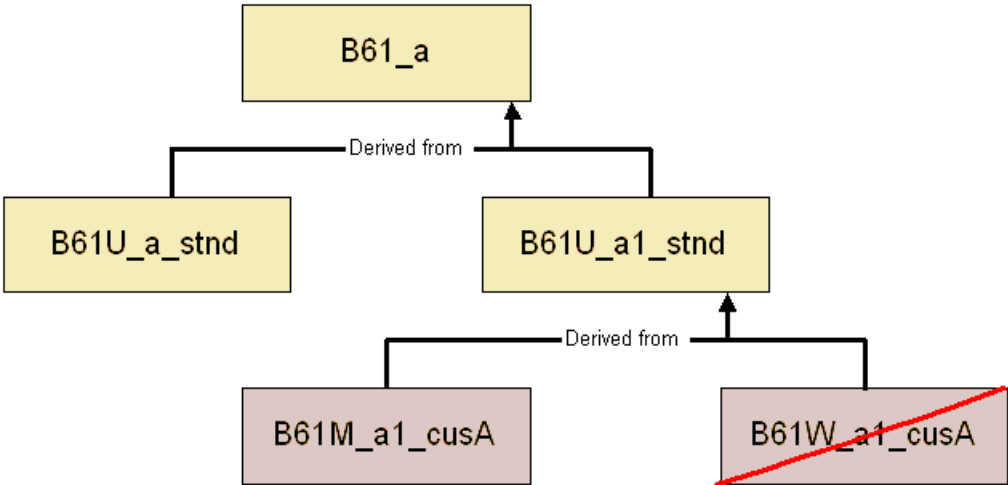
To a certain extent, the system setup is comparable to the setup described for localizations and extensions described in the section *Service Pack development* (p. 3-10) . The main difference lies in the fact that Service Packs for the parent standard product are now installed in an update VRC by means of PMC. Another differentiator is the fact that Service Packs are cumulative, which means that Service Packs also contain all preceding Service Packs. To upgrade a customization to a higher Service Pack level, having a VRC available that only contains the net changes of a Service Pack of the standard parent product can be useful. Suppose you must create a customization on top of Service Pack 1 for a parent standard product. Service Pack 1 is installed in the B61U_a1_stnd update VRC and a new VRC B61W_a1_cusA is developed on top of the Service Pack.



The customization is built on top of the standard Service Pack and will be delivered as a Service Pack in PMC, so no master media must be created for the customization. As soon as Service Pack 0 for the customization is available, the Service Pack is installed in the maintenance VRC B61M_a1_cusA.

Note

In theory, you can also skip the installation of the SP in the B61M_a1_cus1 VRC and start maintenance immediately in the B61D_a1_cus1 VRC. However, best practice is to install and maintain the official build that customers also receive.



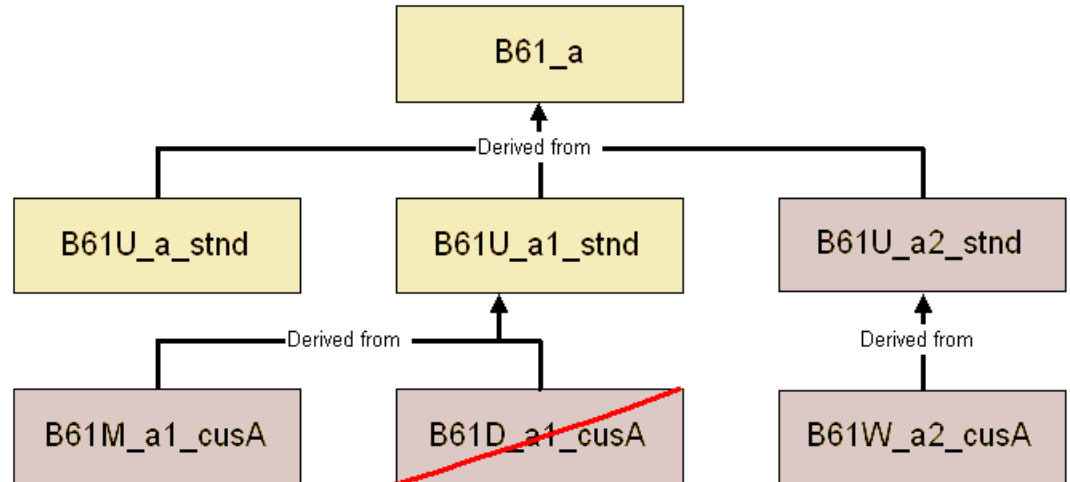
Solutions for Service Pack 1 of the parent standard product will be installed in the B61U_a1_stnd VRC. When you scan solutions for the parent standard product, you can also print a customization report. The customization report provides an overview of which components must be updated in the B61M_a1_cusA VRC. As soon as you install the standard solution, you can merge the changes to the B61M_a1_cusA VRC and create a corresponding PMC solution for the customization.

The approach for subsequent Service Packs is as follows: Install Service Pack 2 for the parent standard product in the B61U_a2_stnd VRC, which is derived from B61_a VRC. Copy the contents of the B61M_a1_cusA VRC to the B61W_a2_cusA VRC. As a result, all maintenance changes you create on top of Service Pack 0 of the customization until the moment of the copy action immediately become available in the B61W_a2_cusA VRC. After the copy, you must also port all maintenance changes in the B61M_a1_cusA VRC to the B61W_a2_cusA VRC. Service Pack 2 for the standard product is cumulative, which means that all changes implemented in Service Pack 1 are also present in Service Pack 2. Therefore, you must create an overview of all components that were changed in Service Pack 2 as compared to Service Pack 1, and these components also must be present in the customization. To achieve this, you can use the Compare Package VRC's (ttadv6450m000) session. In this session, specify values in the following fields as follows:

VRC 1	B61U_a1_stnd
VRC 2	B61U_a2_stnd
Where Software Components present in VRC	B61W_a2_cusA

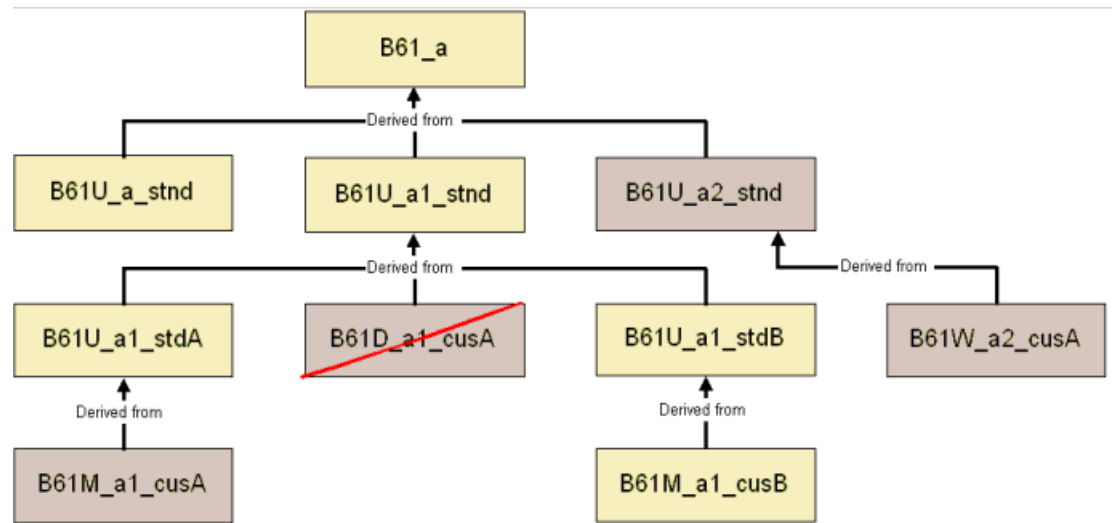
Using these settings enable you to create a net overview of customized components that must be updated to be compatible with the Service Pack 2

of the parent standard product. You can then merge all changes implemented in Service Pack 2 of the standard product from the B61U_2_std VRC into the B61W_a2_cusA VRC. You can then build Service Pack 1 for the customization, based on the contents of the B61D_a2_cusA VRC.



You can repeat this procedure for an unlimited number of Service Packs. In addition, you can develop and maintain multiple customizations in parallel on various levels of the standard parent product.

If multiple customization VRCs are derived from, for example, the B61U_a1_std VRC, then for every customization, an additional update VRC layer B61U_a1_std is required between the customization VRC and the B61U_a2_std VRC. You can use the additional update VRC layer to install additional solutions of the parent product, which you must port to the customization. You must use the Copy Solution Registry to Derived Update VRC (ttpmc2290m000) session to populate the PMC registry for the B61U_a1_std update VRCs. Installation of additional solutions in the B61U_a1_stdA update VRC only impacts the B61M_a_cusA VRC and does not influence the B61M_a1_cusB VRC.



Dependencies for customizations

Individual solutions, patches and/or Service Packs for customizations must have dependencies to corresponding updates of the parent standard product. Not having these types of dependencies can result in runtime compatibility problems if only a solution for the standard parent product is installed, while installation of an update for the customizations was also required.

The best way to establish these dependencies is to define co-requisite dependencies between the updates of the parent product and the corresponding updates for the customized product. The co-requisite dependencies are generated automatically if the customization uses the same solution numbers that were also used in the parent standard product. Of course this method only works if the base VRC of the customization, together with the base VRC of the standard parent product, are part of the same base VRC combination at the distributor side. At the recipient side, the update VRCs for the parent standard product and the customization must also be included in a VRC combination.

The co-requisite relationship will only be in one direction: from the customization to the standard parent product. This one-directional co-requisite makes it impossible to install a solution for the customization without simultaneously installing the corresponding standard solution for the parent product. However, as long as you have not scanned the customized solution, you can still install a solution for the standard parent product without simultaneously installing the corresponding solution of the customization. This can result in run time compatibility problems. The customization report solves this problem. When you check standard solutions in a customized environment, you can optionally print a customization report, which signals customized components and informs the recipient that, together with the standard solution, an update for the customization must also be installed.

Other

The facts described in the sections *Rules to upgrade derived products* (p. 3-14) and *Building Service Packs and the SSA Installation Wizard* (p. 3-15) are also applicable to the development and maintenance of customizations.

Chapter 4

Detailed session descriptions

4

Parameters (ttpmc0100s000)

Use this session to maintain the parameters for the PMC module.

Several sessions in the PMC module require predefined data that is dependent on the environment. This data is stored in the parameter table. These parameters consist chiefly of the paths for import dumps and export dumps of solutions. If a directory does not exist, the directory is created after the user confirms.

Note

If you do not use PMC Distributor, leave the fields in the **Distributor** group box set to the default values.

The following figure shows the form of the session with the default field values:

The screenshot shows a 'Parameters' dialog box with the following fields and values:

- Recipient:**
 - Path for Individual Solution Dumps: `${BSE}/pmc/import/`
 - Path for Uninstall Dumps: `${BSE}/pmc/uninstall/`
- Distributor:**
 - ☐ Attach components during check-out
 - ☐ Insert PMC ident string in scripts
 - Path for Export Dumps: `${BSE}/pmc/export/`
 - Path for Multi-level Export Dumps: `${BSE}/pmc/multilevel/`
 - ☒ Copy Published Solutions
 - Path for Published Solutions: `${BSE}/pmc/published/`

Buttons on the right: OK, Close, Save, Revert, New..., Help.

NOTE: All paths must be unique!!

Modify `ttpmc0100s000` `000`

Fields	Description
Recipient	
Path for Individual Solution Dumps	A directory in which individual solution dumps are placed during the scanning process at the recipient system. If a solution must be installed, the dump is read from this directory.
Path for Uninstall Dumps	A directory in which dumps that are required to uninstall a solution or patch are placed. The Install Solution/Patch (ttpmc2205s000) session or the Check and Install Solutions (ttpmc2204s000) session creates these dumps. The Uninstall Solution/Patch (ttpmc2208s000) session reads the dumps from this directory.
Distributor	
Attach components during check-out	If this check box is selected, you are asked to which PMC solution a component must be linked, if a component that is not yet linked to a solution is checked out using Software Configuration Management (SCM).
Insert PMC ident string in scripts	If this check box is selected, and a <u>program script</u> is checked out using SCM, an ident string containing the code of the solution is added to the source code of the script.

Note

This functionality is only available if the *Attach components during check-out* (p. 4-2) check box is also selected.

Path for Export Dumps A directory in which dumps are placed at the distributor side. The *Export Solution/Patch* (ttpmc1200s000) (p. 4-30) session places the dump in this directory. When you create a patch or collection, the individual solution dumps are read from this directory and concatenated in the patch dump, which in turn are placed in this directory.

Note

The solution file name is composed of the solution code and a constant suffix *.tar.gz*, for example, solution code 1005 becomes *1005.tar.gz*.

Note

Collection dumps are placed in the Multilevel Export directory.

Path for Multi-level Export Dumps A directory in which dumps are placed at the distributor side. The *Export Solution Multi-level* (ttpmc1202s000) (p. 4-31) session places the dump in this directory. Exported collections are also placed in this directory.

Note

The solution file name is composed of the solution code and a constant suffix *.tar.gz*. For example, solution code 1005 becomes *1005.tar.gz*.

Copy Published Solutions If this check box is selected, the dumps of solutions, collections, or patches that were marked as *Published* are copied to a separate directory immediately upon release of these dumps.

Path for Published Solutions A directory in which dumps are placed at the distributor side when the solutions are published. This directory is a location from which you can copy the solutions, for example, to an external FTP server. Customers can connect to that FTP server to download the software updates they want to install. The *Copy Published Solutions* field indicates whether or not this copy action must be carried out.

Solutions are stored in separate subdirectories by base VRC.

The customer part of the base VRC will be added to the solution file name. If the customer part of the base VRC is empty, the string **stnd** will be added to the solution file name. Refer to the following examples:

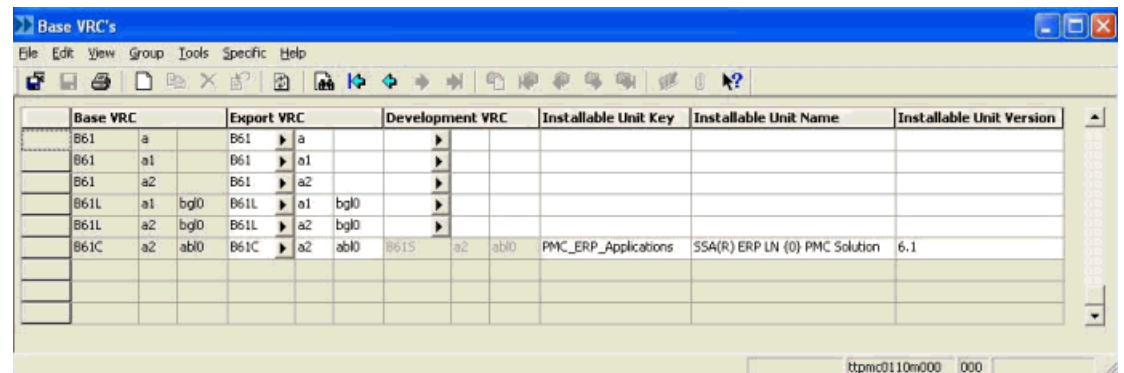
base VRC	solution code	solution file name
B61_a	1005	1005stnd.tar.gz
B61L_a_Spl0	1005	1005Spl0.tar.gz

Base VRC's (ttpmc0110m000)

Use this session to maintain the relations between base VRCs and export VRCs.

The reason why you must specify the relation between export VRCs and base VRCs is as follows: a solution must be linked to a base VRC, because the base VRC specifies for which version of the software the solution is intended. To assemble the solution, you must link components to the solution. However, a component can occur multiple times, in various VRCs. So, how does the system know which of these components you intend? To make this clear, you must link the VRC of the components, that is, the export VRC, to the base VRC of the solution.

In addition to the base VRC and the export VRC, the session also shows the development VRC that is linked to each export VRC. The development VRC is derived from the export VRC. If SCM is activated for the export VRC in one or more packages with the same SCM development VRC, PMC uses the SCM development VRC as the development VRC.



Fields	Description
Base VRC	<u>base VRC</u>
Export VRC	<u>export VRC</u> The export VRCs must exist in the Package VRCs (ttadv111) table for at least one package.

Development VRC	<u>development VRC</u> If you export the solution, and if one or more components exist in the development VRC, a warning message appears.
Installable Unit Key	PMC Solutions are exported as installable units. At the customer side, to store this installable unit in the staging area, you can use this key. The SSA installation wizard supports the installation of products derived from SSA ERP LN. If you create Service Packs that the SSA installation wizard can handle, enter PMC_ERP_Applications in this field. If you create PMC solutions for products that the SSA installation wizard does not support, leave this field blank.
Installable Unit Name	The name of the installable unit that appears in the SSA ERP Installer. If you create Service Packs that the SSA installation wizard can handle, enter SSA(R) ERP LN {0} PMC Solutions in this field. If you create PMC solutions for products that the SSA installation wizard does not support, leave this field blank.
Installable Unit Version	The version of the installable unit, which appears in the SSA ERP Installer. If you create Service Packs that the SSA installation wizard can handle, enter 6.1 in this field. If you create PMC solutions for products that the SSA installation wizard does not support, leave this field blank.

Commands	Description
Solutions	Starts the <i>Solutions (ttpmc1100m000)</i> (p. 4-8) session. Before you can start this session, the <u>base VRC</u> must be a member of a <u>base VRC combination</u> .
Patches	Starts the <i>Patches (ttpmc1501m000)</i> (p. 4-38) session. Before you can start this session, the <u>base VRC</u> must be a member of a <u>base VRC combination</u> .
Print Solution/Patch	Starts the <i>Print Solution/Patch (ttpmc1400m000)</i> (p. 4-12) session.

Base VRC Combinations (ttpmc0111m000)

Use this session to enter the base VRC combinations.

Base VRC Combinations

File Edit View Tools Specific Help

Base VRC Combination: b61asp1

Description: Service Pack 1 SSA ERP LN 6.1

PMC Version: 1

Base VRCs by Base VRC Combination

Base VRC	
B61	a1

Modify ttpmc0111m100 000

Fields
Description

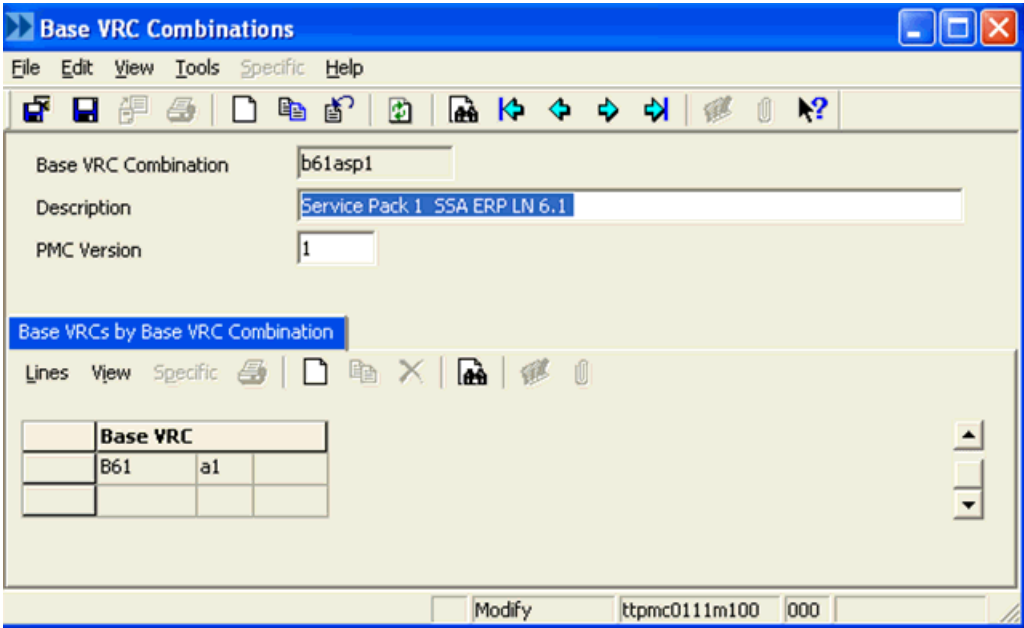
Base VRC Combination The code and the description of the base VRC combination.

PMC Version PMC version

Base VRC Combinations (ttpmc0111m100)

Use this session to maintain base VRC combinations and to link base VRCs to base VRC combinations.

You can start the session from the *Base VRC Combinations (ttpmc0111m000)* (p. 4-5) session.



Fields	Description
Base VRC Combination	The code and the description of the base VRC combination.
PMC Version	<u>PMC version</u> .

Base VRCs by Base VRC Combination

Use this session to link Base VRCs to a Base VRC Combination. You can start the session, as shown in the following figure, from the *Base VRC Combinations (ttpmc0111m000)* (p. 4-5) session.

Base VRC Combination: b61asp1

Description: Service Pack 1 SSA ERP LN 6.1

PMC Version: 1

Base VRCs by Base VRC Combination

Base VRC	a1
B61	

Modify ttpmc0111m100 000

Fields	Description
--------	-------------

Base VRC	<u>Base VRC</u>
----------	-----------------

Solutions (ttpmc1100m000)

You can use this Use this session to maintain solutions.

The scope of this session covers the life cycle of a solution from the solution's creation to the solution's release. In this session, you can maintain the basic data of a solution.

You can use this session to change the status of a solution manually under certain conditions.

The following figure shows the **Overview** tab of the session:

Solution Code	Description	Obsolete	Status	Patch Code
149538	display problems	☐	Released	
149564	In Electronic Bank Statements, distribution of bank files is	☒	Released	
149584	Planned receipt date cannot be changed	☐	Released	
149586	In Maintenance Sales Order Wrong rates are fetched	☐	Released	
149603	Batch stck up at In Use status	☐	Released	
149614	order policy message incorrect	☐	Released	

The following figure shows the **Details** tab of the session:

Fields	Description
Base VRC	<u>base VRC</u>
Export VRC	<u>export VRC</u>

Base VRC Combination base VRC combination

Solution Code	A unique identification of the solution in a <u>base VRC</u>. The solution code has the following characteristics: ■ Datatype: alpha-numerical string. ■ Maximum length: 14 characters. ■ Forbidden characters: All characters that have a special meaning in UNIX, such as \$, *, and so on. If you enter a forbidden character, the code will not be accepted.
Obsolete Solution	This check box identifies a solution as an <u>obsolete solution</u> .
Status	<u>status</u>
Patch	If the solution is also part of a <u>patch</u> , the patch code appears in this field.
Published	If this check box is selected,, the solution is published immediately upon releas. If a solution is published, the solution is copied to the directory specified in the <i>Path for Published Solutions (p. 4-3)</i> field in the <i>Parameters (ttpmc0100s000) (p. 4-1)</i> session. Note If the <i>Copy Published Solutions (p. 4-3)</i> check box in the <i>Parameters (ttpmc0100s000) (p. 4-1)</i> session is cleared, the solution cannot be published, regardless of the selection of the current check box.
Text	If this If this check box is selected,, a text appears with additional information. Click Text to add, read, or change the text.
Patch	The code of the <u>patch</u> to which a solution can be linked. Note To select a patch for this field, you must start the current session from the Specific menu of the <i>Patches (ttpmc1501m000) (p. 4-38)</i> session.
Commands	Description
Defects by solution	Starts the <i>Defects by Solution (ttpmc1110m000) (p. 4-13)</i> session.
Print Solution/Patch	Starts the <i>Print Solution/Patch (ttpmc1400m000) (p. 4-12)</i> session.

Print Maintenance History	Starts the <i>Print Maintenance History (ttpmc1460m000)</i> (p. 4-59) session.
Print Dependencies (Multi-level)	Starts the <i>Print Dependencies (Multilevel) (ttpmc1440m000)</i> (p. 4-29) session.
Components by solution	Starts the <i>Components by Solution (ttpmc1520m000)</i> (p. 4-14) session.
Check-in	check in
Check-out	check out
Dependencies	Starts the <i>Dependencies (ttpmc1140m000)</i> (p. 4-26) session.
Generate Dependencies	Starts the <i>Generate Dependencies (ttpmc1240s000)</i> (p. 4-28) session and sets the status to Dependencies Defined .
Link Solution to Patch	Links the solution to the patch that appears in the <i>Patch</i> field. This option is only available if the <i>Patch</i> field displays a patch that is not yet released. The solutions you want to link must be released. This option is only available if you start the current session from the SpecificSpecific menu of the <i>Patches (ttpmc1501m000)</i> (p. 4-38) session. A solution can only be part of one patch.
Remove Solution from Patch	Removes the solution from the patch that appears in the <i>Patch</i> field. This option is only available if the <i>Patch</i> field displays a patch that is not yet released. To select a patch, you must start the current session from the SpecificSpecific menu of the <i>Patches (ttpmc1501m000)</i> (p. 4-38) session.
Export solution	Starts the <i>Export Solution/Patch (ttpmc1200s000)</i> (p. 4-30) session. This option is only available if the <i>Status</i> of the solution is Dependencies Defined. Note that the session that starts has no user interface, and that the solution is exported immediately.
Export Solution Multi Level	Starts the <i>Export Solution Multi-level (ttpmc1202s000)</i> (p. 4-31) session. This option is only available if the <i>Status</i> of the solution is Released. Note that the session that starts has no user interface, and that the solution is exported immediately.
Maintenance History	Starts the <i>Maintenance History (ttpmc1560m000)</i> (p. 4-56) session.

Change Current Package Starts the Change Current Package VRC of User (ttadv0140s000) session.
VRC of User

Print Solution/Patch (ttpmc1400m000)

Use this session to print solutions and patches.

Fields	Description
Solutions Range From/To	
Base VRC	<u>base VRC</u>
Solution	A unique identification code of the solution in a <u>base VRC</u> .
Status	<u>status</u>
Type	<u>software component</u>
Duration Range From/To	
Export Date	The date on which the solution or <u>patch</u> was exported.
Export Time	The time at which the solution or <u>patch</u> was exported.
Include Print Option	

Defects	If this check box is selected, the defects linked to the solution are printed.
Dependencies	If this check box is selected, the dependencies between the solutions or <u>patches</u> are printed.
Components	If this check box is selected, the components linked to the solutions are printed.
Solutions	If this check box is selected,, the solutions that are linked to the patches are printed.
Patch Header	If this check box is selected, and you print <u>patches</u> , the patch header that is linked to the patch is printed.

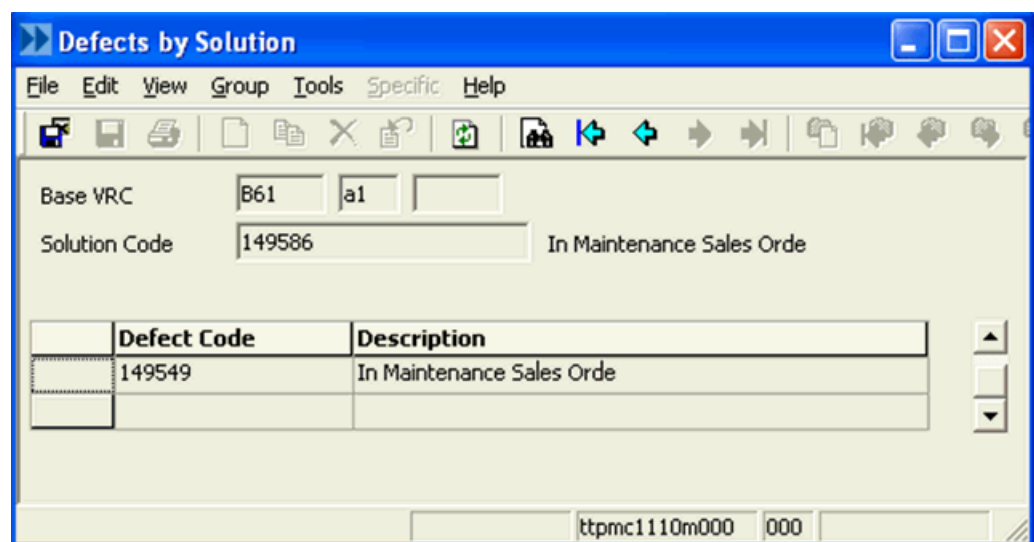
Commands	Description
Print	Prints the report.

Defects by Solution (ttpmc1110m000)

Use this session to specify the defects that a solution solves.

This information is for administrative purposes only. The defects are not included in the final PMC solution dump.

You can start this session, as shown in the following figure, from the **Specific** menu of the *Solutions (ttpmc1100m000)* (p. 4-8) session.



Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Defect Code	A unique identification of the defect.

Components by Solution (ttpmc1520m000)

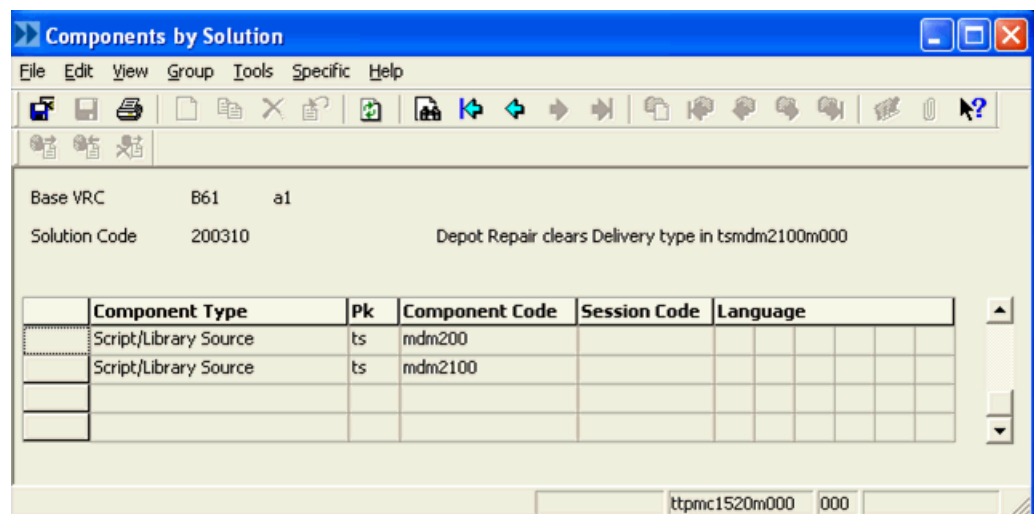
Use this session to connect the modified components to a solution.

If only one session uses a component, the session code appears in the *Session* (p. 4-15) field. If more than one session uses the component, an asterisk (*) appears in that field. Use the *Sessions by Component* form command on the **Specific** menu to display all the sessions that use the component.

For the language-dependent components, all base languages for which the component is present in the export VRC appear by default. This applies to labels, messages, and questions.

For other components that have a description in different languages, you can use language 2 (English).

You can only start this session, as shown in the following figure, from the **Specific** menu of the *Solutions* (ttpmc1100m000) (p. 4-8) session.



Fields	Description
Base VRC	<u>base VRC</u>

Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Component Type	<u>software component</u>
Pk	The <u>package</u> to which the component belongs. Note A solution can relate to more than one package. The Tools package can not be combined with other packages. The types of packages that you can use is determined by the first package entered.
Component Code	The component identification, for example, module and session code. If you zoom to a session to select a component, the component type you selected in the <i>Component Type</i> field determines the session that starts .
Session	The session that uses the component. If more than one session uses the component, an asterisk (*) appears. Use the <i>Sessions by Component</i> form command on the Specific menu to display all the sessions that use the component.
Language	The language code of the component. This field is only filled for language-dependent components, such as labels, messages, and questions or software components that have language-dependent descriptions. By default, this field contains the language codes of the languages for which the component exists in the relevant <u>VRC</u>: for Tools the Tools VRC, for others packages the <u>development VRC</u> or <u>export VRC</u>.

Commands	Description
Check-out	<u>check out</u>
Check-in	<u>check in</u>
Sessions by Component	Starts the Sessions by Component (ttpmc1125m000) session.
Print Components by Solution	Starts the <i>Print Components by Solution</i> (ttpmc1420m000) (p. 4-25) session.
Change Current Package VRC of User	Starts the Change Current Package VRC of User (ttadv0140s000) session.

Undo Check-Out undo check-out

Component by Solution (ttpmc1120s000)

Use this session to link a component to a solution.

Component by Solution

File View Tools Specific Help

Base VRC B61 a1

Solution Code 200310 Depot Repair clears Delivery t

Component Information

Component Type Script/Library Source

Session Code

Package ts

Component Code mdm200

Language ☐ ☐ ☐ ☐ ☐ ☐

ttpmc1120s000 000

Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Component Type	<u>software component</u>
Session Code	The session code of the session that uses the modified component.
Package	The <u>package</u> to which the component belongs.
	Note A solution can relate to multiple packages. You cannot combine the Tools package with other packages. The types of packages that you can use is determined by the first package type you enter.
Component Code	The component identification, for example, module and session code.

If you zoom to a session to select a component, the session that starts is determined by the component type you selected in the *Component Type* field.

Language	The language of the component. This field is only specified for language-dependent software components, such as labels, messages, and questions, or software components that have language-dependent descriptions. This fields are, by default, filled with the language codes of the languages for which the software component exists in the relevant <u>VRC</u>. For SSA ERP Tools (tt), the tools VRC, for others packages, the development VRC or export VRC.
----------	--

Commands	Description
Sessions by Component	Starts the Sessions by Component (ttpmc1125m000) session.
Check Out	<u>check out</u>
Check In	<u>check in</u>
Undo Check-Out	<u>undo check-out</u>
Change Current Package VRC of User	Starts the Change Current Package VRC of User (ttadv0140s000) session.

Connect Components to PMC Solution (ttpmc1221m000)

Use this session to connect software components present in the export VRC to a PMC solution.

The session contains the following tabs:

- **Environment:** Specify the PMC solution to which the software components must be connected, and the options you want to use.

The screenshot shows the 'Connect Components to PMC Solution' dialog box with the 'Selection' tab selected. The dialog is divided into several sections:

- Environment:** Contains 'Base VRC' (three input boxes) and 'Solution Code' (one input box).
- Options:** Contains 'Max. Number of Components' (input box with '0'), 'What to do when Max. Number is Exceeded' (dropdown menu with 'Cancel' selected), and three radio buttons: 'Language-Dependent Components Only' (unchecked), 'Not-Expired Components' (selected), 'Expired Components' (unchecked), and 'Not-Expired and Expired Components' (unchecked).
- Source:** Contains a list of source types with checkboxes: 'Export VRC', 'Other Solution', 'Patch', 'Sequential File', and 'Cust. Admin.'. To the right of these are input fields for 'Base VRC', 'Solution', 'Patch', 'Project', 'Project Part', and 'Customization Part'. The 'Project Part' and 'Customization Part' fields have 'From' and 'To' sub-fields, with the 'To' fields containing 'ZZZ'.

- **Selection** Specify the component types and the component code ranges.

Connect Components to PMC Solution

Environment: Selection

General

From To

Package -

Language -

Components

From To

☒ Sessions -

☒ Integrated Sessions -

☒ Reports -

☒ Forms -

☒ Menus -

☒ Scripts -

☐ Objects Only

☒ Program Functions -

☒ Tables 000 -

☒ Domains -

☒ Messages -

☒ Questions -

☒ Business Objects -

☒ Labels -

☒ Charts -

☒ Additional files -

☒ Subscription Templates -

Fields	Description
Environment	
Base VRC	<u>base VRC</u>
Solution Code	Enter an existing solution code or zoom to the <i>Solutions (ttpmc1100m000)</i> (p. 4-8) session to select a solution. The code identifies the solution to which the components are connected. Note If the following conditions are met, the last character of the solution code must be numeric, for example, SOL003. ■ In the <i>Max. Number of Components</i> (p. 4-19) field, a maximum number of components is specified. ■ In the <i>What to do when Max. Number is Exceeded</i> (p. 4-20) field, the option Next Solution Number is selected.
Max. Number of Components	Specify the maximum number of components that can be connected to the solution.

This option is available because a large number of components can cause problems at the recipient side. In that case, the components must be distributed over several solutions.

Note

If the specified number is zero, no limitation applies.

What to do when Max. Number is Exceeded	Specify how the process must proceed if the maximum number of components specified in the <i>Max. Number of Components</i> (p. 4-19) field is exceeded for the current solution.
---	--

Various choices of what to do if the maximum number of components is exceeded

The following values specify how the process of linking components to a solution is continued if the specified maximum number of components is exceeded.

- **Cancel**

The process is canceled, and no components are linked at all.

- **Stop**

The process is stopped. A number of components equal to the specified maximum number is linked to the solution, but the remaining components are ignored.

- **Specify Other Solution**

The process pauses, and you are prompted to specify another solution number for the remaining components. If the maximum number is exceeded again, you are prompted again, and so on, until all components are processed. If you specify no other solution number, the process stops, and the remaining components are ignored.

Note:

You must specify an existing solution number, or create a solution the moment you are prompted to specify the solution number.

- **Next Solution Number**

The process itself selects another solution. This process is repeated if the maximum number is reached again, until all components are processed.

Notes:

- You can only use this option if the last character of the initial solution code is a numeric character.
- Solutions that are already released are skipped.
- Solutions that already contain the specified maximum number of components are skipped.

- The process creates a new solution if the next number does not identify an existing solution.
- If the next number does not exist because the range is full, the process stops, and the remaining components are ignored. For example, after SOL999, no new solution can be created.
- Use this option only if you are sure that the initial solution number is not followed by solution numbers of solutions that are used for other purposes.

Language-dependent Components only	If this check box is selected, only language-dependent components are linked to the solution. As a result, the following language-independent components are excluded : ■ Scripts ■ Objects ■ Functions ■ Non-enumerated domains. This option is particularly useful for solutions that serve translation purposes.
Selection of components	The user can select one of the following choices: ■ Non-expired components ■ Expired components ■ Non-expired and expired components
Export VRC	If this check box is selected, the components that are present in the export VRC of the specified <u>base VRC</u> are all linked to the solution. Use the options on the Selection tab to select or exclude component types and ranges.
Other Solution	If this check box is selected, you can specify a solution whose component list will be used to link components to your current solution. Use the options on the Selection tab to select or exclude component types and ranges. This option is helpful if you want to deliver a solution whose components are (almost) the same as that of another solution.
Base VRC	<u>base VRC</u>
Solution	The solution number of the solution whose component list you want to use for your current solution.

Enter a solution code or zoom to the *Solutions (ttpmc1100m000)* (p. 4-8) session to select a solution.

Patch

If this check box is selected, you can specify a patch whose component list will be used to link components to your current solution.

Use the options on the **Selection** tab to select or exclude component types and ranges.

This option is helpful if you want to deliver a solution whose components are (nearly) the same as that of a patch, for example, software translations.

Base VRC base VRC

Patch

The patch number of the patch whose component list you want to use for your current solution.

Enter a patch code or zoom to the *Patches (ttpmc1501m000)* (p. 4-38) session to select a patch.

Sequential File

If this check box is selected, you can specify a sequential file name whose records specify the components that will be linked to the solution.

Use the options on the **Selection** tab to select or exclude component types and ranges.

The records in the file must have the following format:

<comp.type>|<package>|<comp.code>|<languages>

comp.type must be a numeric value, as specified in the following table:

comp.type	description
1	Message
2	Question
3	Session
4	Domain
5	Table Definition
6	Menu
7	Form
8	Script/Library Source

comp.type	description
9	Script/Library Object
10	Report (source+object)
11	Label
13	Function Source
14	Business Object
16	Chart
17	Integrated Session
18	Additional File
19	Subscription Template

Cust. Admin. If this check box is selected, you can specify a customization project whose component list will be used to link components to your current solution. Use the options on the **Selection** tab to select or exclude component types and ranges.

Project Enter a project code or zoom to the Projects (ttadv1170m000) session to select a project.

Project Part Enter a project part code or zoom to the Customization Parts by Project (ttadv1175m000) session to select a project part.

Customization Part Enter a customization part code or zoom to the Customization Parts by Project (ttadv1175m000) session to select a customization part.

Selection

Package package

Language A code that identifies the language.

Sessions Enter a session code or zoom to the Sessions (ttadv2500m000) session to select a session.

Integrated Sessions	Enter an <u>integrated session</u> code or zoom to the Sessions (ttadv2500m000) session to select a session.
Reports	Enter a <u>report</u> code or zoom to the Reports (ttadv3530m000) session to select a report.
Forms	Enter a <u>form</u> code or zoom to the Forms (ttadv3500m000) session to select a form.
Menus	Enter a <u>menu</u> code or zoom to the Menus (ttadv3560m000) session to select a menu.
Scripts	Enter a <u>program script</u> code or zoom to the Program Scripts / Libraries (ttadv2530m000) session to select a script.
Objects Only	If this check box is selected, only the objects of the scripts are linked to the solution. Scripts are excluded. Note This option is only available if you select the Scripts check box.
Program Functions	Enter a <u>function</u> code or zoom to the Functions (ttadv2560m000) session to select a function.
Tables	Enter a <u>table</u> code or zoom to the Table Definitions (ttadv4526m000) session to select a table.
Domains	Enter a <u>domain</u> code or zoom to the Domains (ttadv4500m000) session to select a domain.
Messages	Enter a <u>message</u> code or zoom to the Messages (ttadv4551m000) session to select a message.
Questions	Enter a <u>question</u> code or zoom to the Questions (ttadv4561m000) session to select a question.
Business Objects	Enter a <u>Business Object</u> code or zoom to the Business Objects (ttadv7500m000) session to select a Business Object.
Labels	Enter a <u>label</u> code or zoom to the Labels (ttadv1540m000) session to select a label.

Charts Enter an chart code or zoom to the Chart Manager Application Data (ttchm1500m000) session to select a chart.

Additional files Enter an additional file code or zoom to the Additional Files (ttadv2570m000) session to select an additional file.

Subscription Templates Enter a subscription template code or zoom to the Subscription Templates (ttadv7565m000) session to select a subscription template.

Commands	Description
----------	-------------

Connect	Click to connect the components to the solution.
---------	--

Print Components by Solution (ttpmc1420m000)

Use this session to print components connected to a solution.

Specify a range of base VRCs, solutions, component types, packages, and component codes. All the components found within the range will be printed, including the sessions in which the components are used and the export date.

Fields	Description
--------	-------------

Base VRC	<u>base VRC</u>
----------	-----------------

Package	<u>package</u>
---------	----------------

Solution	The unique identification code of the solution.
----------	---

Type software component

Code A unique identification code of the component.

Commands	Description
----------	-------------

Print	Click to print the report.
-------	----------------------------

Dependencies (ttpmc1140m000)

Use this session to define the dependencies between solutions or patches. You can generate the dependencies automatically, or define the dependencies manually.

If not all dependencies that would be generated are present, you receive a warning during the export of the solution or patch.

After you define all the dependencies, you can change the solution status to **Dependencies defined** and the solution is ready to be exported. You can change the status in the details session of the *Solutions (ttpmc1100m000)* (p. 4-8) session.

If the status of the solution was **Exported**, and you change the dependencies, then the solution status is reset to **Dependencies defined**.

You can only start the current session through the **Specific** menu in the *Solutions (ttpmc1100m000)* (p. 4-8) session.

Related VRC	Solution	Description	Type	Manual
B61 a1	P50010010007	Components Patch SPB61a1	Pre-requisite	☐

ttpmc1140m000 000

Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Related VRC	The <u>base VRC</u> of the solution or <u>patch</u> to which the current solution or patch is related. The related VRC can be a different base VRC if the dependency type is <u>co-requisite</u> or a <u>prerequisite</u> from a <u>Service Pack</u> to the previous Service Pack.
Solution	The solution or <u>patch</u> with which a dependency relation exists.
Type	<u>dependency</u> You must enter dependencies of type co-requisite and prerequisite for the solution being created. You must add post-requisites dependencies to the already released solution and the post-requisite dependency must point to the solution being created. Note Be sure to add the post-requisite dependency before you release the new solution. The post-requisite dependency is transferred to the recipient in the dump of the solution being created. This implies that this dependency is only known at the recipient side after the newly created solution has been scanned.
Manual	If this check box is selected, the dependency was not generated, but defined manually. Note If you generate or regenerate the dependencies, the <i>Generate Dependencies (ttpmc1240s000)</i> (p. 4-28) session start. In this session, you can specify whether or not you want to delete existing manually defined dependencies.

Commands	Description
Generate Dependencies	Starts the <i>Generate Dependencies (ttpmc1240s000)</i> (p. 4-28) session.

Print Dependencies (Multi-level) Starts the *Print Dependencies (Multilevel)* (ttpmc1440m000) (p. 4-29) session.

Generate Dependencies (ttpmc1240s000)

Use this session to generate the dependency relations between solutions that must be installed together at a customer system.

The following dependency relations are generated:

- Prerequisite relations.
- Co-requisite relations to the same solutions in other base VRCs, if these solutions have one or more components in common.

Conditions for co-requisites:

- Base VRCs must belong to the same base VRC combination.
- Solution numbers must be identical.

The following dependency relations cannot be generated, but must be specified manually:

- Co-requisite relations to solutions in the same base VRC
- Post-requisite relations

To start this session, as shown in the following figure, you can click *Generate Dependencies* on the **Specific** menu of the *Dependencies* (ttpmc1140m000) (p. 4-26) session, which in turn you must start from the **Specific** menu of the *Solutions* (ttpmc1100m000) (p. 4-8) session.

Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Status	<u>status</u>

Delete Also Manually Entered	If this check box is selected, the dependencies you might have entered manually are deleted when you regenerate the dependencies.
------------------------------	---

Regenerate Dependencies	If you select this If this check box is selected, all existing dependency relations are deleted, and the relations are regenerated.
-------------------------	---

Note

Manually entered relations are not deleted, unless you select the *Delete Also Manually Entered* (p. 4-29) check box.

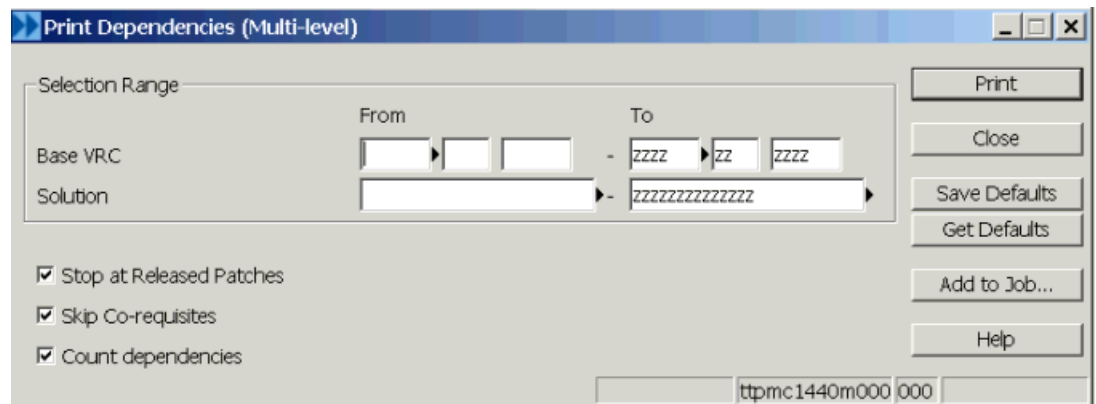
Commands	Description
Generate	Generates the dependency relations. Existing dependency relations are all retained, unless you select the <i>Regenerate Dependencies</i> (p. 4-29) check box and, optionally, the <i>Delete Also Manually Entered</i> (p. 4-29) check box.

Print Dependencies (Multilevel) (ttpmc1440m000)

Use this session to print the dependencies.

The report is built according to the following steps:

1. All solutions that have a post-requisite relation with the solutions in the selection ranges are listed. These related solutions are printed single-level. The solutions are added to the set of selected solutions to print the dependencies separately. The solutions receive level 1.
2. All solutions that have a co-requisite relation with the solutions in the selection ranges are listed. These solutions are also printed single-level and added to the print selection as described for the post-requisites. These solutions receive level 1.
3. The solutions that have a prerequisite relation with the solutions in the selection ranges are printed. Each entry is preceded by a level number, ranging from 1 to 35. Level 1 is a direct prerequisite, level 2 is a prerequisite of a level 1 dependency, and so on.



The dialog box titled "Print Dependencies (Multi-level)" contains the following elements:

- Selection Range:**
 - Base VRC:** Fields for "From" (empty) and "To" (zzzz zz zzzz).
 - Solution:** Fields for "From" (empty) and "To" (zzzzzzzzzzzzzzzzzzzz).
- Checkboxes:**
 - ☒ Stop at Released Patches
 - ☒ Skip Co-requisites
 - ☒ Count dependencies
- Buttons:** Print, Close, Save Defaults, Get Defaults, Add to Job..., Help.
- Status Bar:** ttpmc1440m000 000

Fields	Description
Base VRC	<u>base VRC</u>
Solution	The unique identification code of the solution in a <u>base VRC</u> .
Stop at Released Patches	If this check box is selected, and a solution is found with the status Released , the chain of related solutions is cut off at that point.
Skip Co-requisites	If this check box is selected, solutions with a <u>corequisite</u> relation are not printed.
Count dependencies	If this check box is selected, the number of dependency relations is counted.

Commands	Description
Print	Prints the dependencies.

Export Solution/Patch (ttpmc1200s000)

Use this session to create the export dump for a solution or a patch.

The solution dump contains an export dump of the components modified for the solution and a PMC information file. The patch dump consists of the concatenation of several individual solution files, and is also preceded by a PMC information file. The export dump that is created is the input for the Scan Solution/Patch File (ttpmc2200s000) session at the recipient side.

The status of the solution that must be exported must be at least **Dependencies Defined**. After the solution is exported, the status of the solution is set to **Exported**. Each status change is recorded in the

maintenance history, which you can view in the *Maintenance History* (*ttpmc1560m000*) (p. 4-56) session.

- To export solutions, you must start this session from the **Specific** menu of the *Solutions* (*ttpmc1100m000*) (p. 4-8) session.
- To export patches, you must start this session from the **Specific** menu of the *Patches* (*ttpmc1501m000*) (p. 4-38) session.
- The current session has no user interface. The solution or patch is exported immediately.
- To export a solution together with the solution's prerequisite solutions, you must use the *Export Solution Multi-level* (*ttpmc1202s000*) (p. 4-31) session, which you can also start from the **Specific** menu of the *Solutions* (*ttpmc1100m000*) (p. 4-8) session.

Export Solution Multi-level (*ttpmc1202s000*)

Use this session to create a multilevel export dump for a released solution. A multilevel export dump of a solution contains not only the solution, but also the prerequisite solutions for that solution.

Before you can export a solution with this session, the status must be **Released**. Exporting the solution does not change this status.

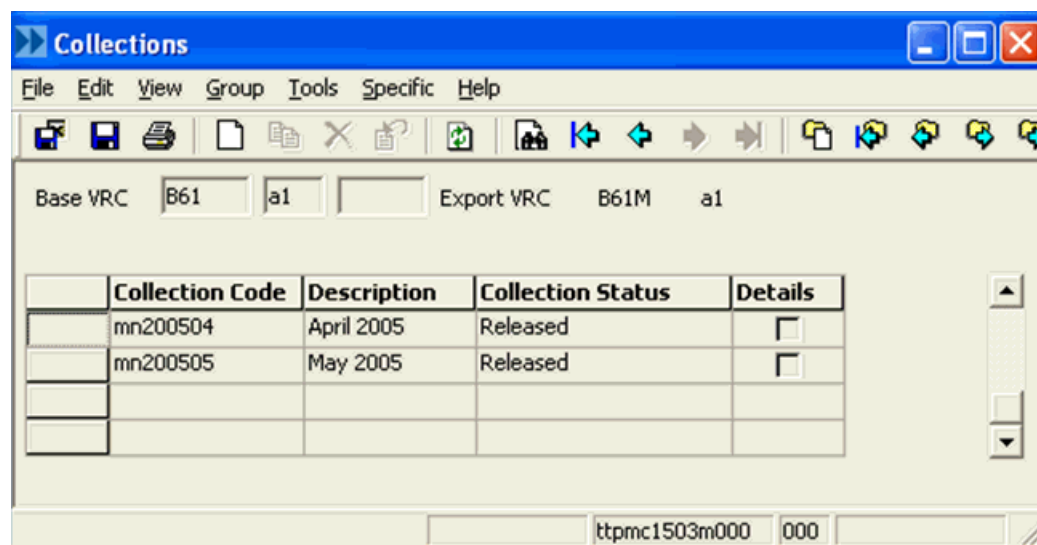
Notes:

- You must start this session from the **Specific** menu of the *Solutions* (*ttpmc1100m000*) (p. 4-8) session.
- The current session has no user interface. The solution or patch is exported immediately.
- The export process continues until a solution that is linked to a patch with status **Released** is reached.

Collections (*ttpmc1503m000*)

Use this session to maintain collections.

After you link the solutions to the collection, you must validate and export the collection, which you can do using the commands on the **Specific** menu of the session.



Fields	Description
Base VRC	<u>base VRC</u>
Export VRC	<u>export VRC</u>
Collection Code	The code and the description of the <u>collection</u> .
Collection Status	<u>status</u>
Details	If this check box is selected, text with additional information is present. Click Text to add, read, or change the text.
Commands	Description
Solutions by Collection	Starts the Solutions by Collection (ttpmc1551m000) session.
Validate Collection	Starts the <i>Validate Collection</i> (ttpmc1402s000) (p. 4-35) session. This option is only available if the <u>collection</u> is not yet exported or released. If the collection was exported, you must first set the status to Dependencies Defined .
Export Collection	Starts the <i>Export Collection</i> (ttpmc1210s000) (p. 4-36) session. This option is only available if the <u>collection</u> is not yet released.
Maintenance History	Starts the <i>Maintenance History</i> (ttpmc1560m000) (p. 4-56) session.

Print Collection

Starts the *Print Collection* (ttpmc1403m000) (p. 4-37) session.

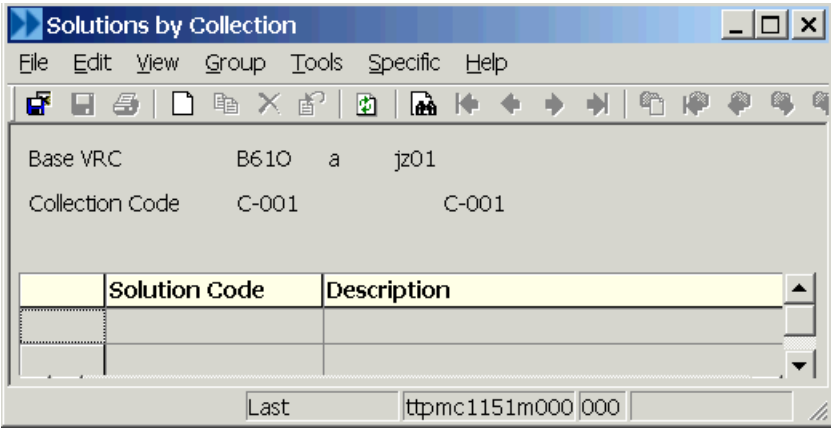
Collections (ttpmc1101s000)

Use this session to maintain collections.

Fields	Description
Base VRC	<u>base VRC</u>
Export VRC	<u>export VRC</u>
Collection Code	The code and description of the <u>collection</u> .
Collection Status	<u>solution status distributor</u>
Published	If this check box is selected, the <u>collection</u> is published the moment the collection is released. If a collection is published, the collection is copied to the directory specified in the <i>Path for Published Solutions</i> (p. 4-3) field in the <i>Parameters</i> (ttpmc0100s000) (p. 4-1) session. Note If the <i>Copy Published Solutions</i> (p. 4-3) check box in the <i>Parameters</i> (ttpmc0100s000) (p. 4-1) session is cleared, the collection cannot be published, regardless of the setting of this check box.
Details	If this check box is selected, a text is present with additional information. Click Text to add, read, or change the text.

Solutions by Collection (ttpmc1151m000)

Use this session to add solutions to a collection.
You can only link solutions to a single collection. You cannot include solutions that are already a part of another collection.
You can add a range of solutions simultaneously, using the *Generate Collection* command on the **Specific** menu. This command starts the *Generate Collection (ttpmc1251s000)* (p. 4-34) session.
You usually start the Solutions by Collection session from the **Specific** menu of the *Collections (ttpmc1503m000)* (p. 4-31) session.



Fields	Description
Base VRC	<u>base VRC</u>
Collection Code	The code and the description of the <u>collection</u> .
Solution Code	A unique identification of the solution in a <u>base VRC</u> .

Commands	Description
Generate Collection	Starts the <i>Generate Collection (ttpmc1251s000)</i> (p. 4-34) session.

Generate Collection (ttpmc1251s000)

Use this session to to add a range of solutions to a collection.
You can start the session from the **Specific** menu of the *Solutions by Collection (ttpmc1151m000)* (p. 4-34) session.

This session only selects solutions with the status **Released** and solutions that are not linked to another collection.

In the *Solutions by Collection (ttpmc1151m000)* (p. 4-34) session, you can remove solutions from the collection.

Fields	Description
Base VRC	<u>base VRC</u>
Collection Code	The identification and the description of the <u>collection</u> in a <u>base VRC</u> .
Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Export Date	The date on which the solution was exported.
Export Time	The time at which the solution was exported.

Commands	Description
Generate	Generates the <u>collection</u> .

Validate Collection (ttpmc1402s000)

Use this session to validate a collection.

A collection must be validated before the *Export Collection (ttpmc1210s000)* (p. 4-36) session can export the collection.

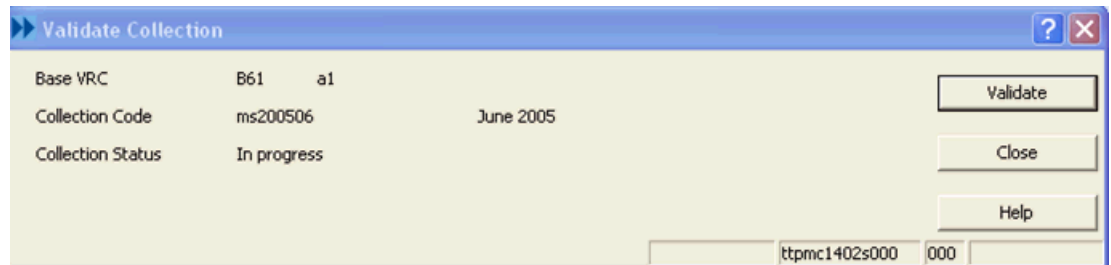
The validation does the following:

- Checks whether the dumps of the solutions are present in the *Path for Export Dumps* (p. 4-3) directory that is specified in the *Parameters (ttpmc0100s000)* (p. 4-1) session. The dumps of the solutions are placed in this directory when the solutions are exported individually,

which you can do through the *Export Solution/Patch (ttpmc1200s000)* (p. 4-30) session.

- Checks whether the status of the solutions is **Released**.
- Checks if any solution in the collection has a post-requisite relation with another solution, or a co-requisite relation with another solution in the same base VRC, and whether the solution with which this relation exists is also part of the collection.

If you export a collection, the Validate Collection session starts automatically. If any of the checks fails, the export stops. You can also use the Validate Collection session manually from the **Specific** menu of the *Collections (ttpmc1503m000)* (p. 4-31) session.



Fields	Description
Base VRC	<u>base VRC</u>
Collection Code	A unique identification of the <u>collection</u> in the <u>base VRC</u> .
Collection Status	<u>status</u>

Commands	Description
Validate	Validates the <u>collection</u> .

Export Collection (ttpmc1210s000)

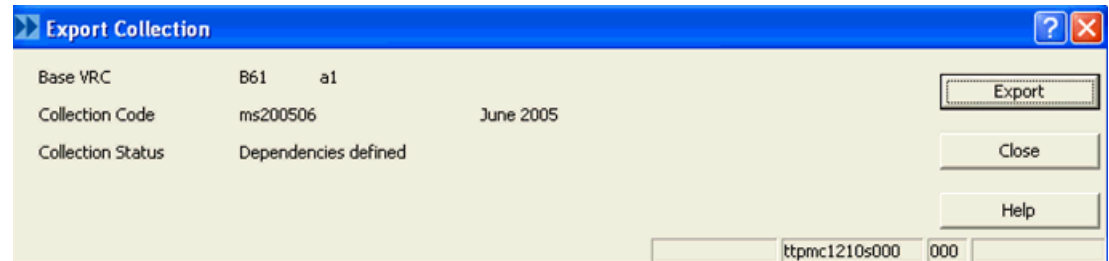
Use this session to export a collection, which means that a dump of the collection is made.

You must start this session from the **Specific** menu of the *Collections (ttpmc1503m000)* (p. 4-31) session.

Before a collection is exported, the collection is validated by means of the *Validate Collection (ttpmc1402s000)* (p. 4-35) session. If the collection is not valid, the export is stopped.

You can also start the *Validate Collection (ttpmc1402s000)* (p. 4-35) session from the **Specific** menu of the *Collections (ttpmc1503m000)* (p. 4-31) session to validate the collection manually.

If a collection is exported, a dump is made in the directory where the multilevel dumps of the solutions are stored. You specify this directory in the *Path for Multi-level Export Dumps (p. 4-3)* field in the *Parameters (ttpmc0100s000)* (p. 4-1) session.



Fields	Description
Base VRC	<u>base VRC</u>
Collection Code	A unique identification of the <u>collection</u> in the <u>base VRC</u> .
Collection Status	<u>status distributor</u>

Commands	Description
Export	Exports the <u>collection</u> .

Print Collection (ttpmc1403m000)

Use this session to report of the contents of a collection.

The following figure shows the form of the session:

Fields	Description
Base VRC	<u>base VRC</u>
Collection Code	The identification code of the <u>collection</u> .
Details	If this check box is selected, print the details of the <u>collections</u> .
Solutions by Collection	If this check box is selected, the solutions in the <u>collections</u> by collection.

Commands	Description
Print	Prints the report of the <u>collections</u> .

Patches (ttpmc1501m000)

Use this session to to maintain patches.

The following figure shows the **Overview** tab of the session:

Patch Code	Description	New Base VRC	Status
SPB61a1	Service Pack 1 for SSA ERP LN 6.1	☒	Released

The following figure shows the **Detail** tab of the session:

Fields	Description
Base VRC	<u>base VRC</u>
Export VRC	<u>export VRC</u>
Base VRC Combination	<u>base VRC combination</u>
Patch Code	A unique identification of the <u>patch</u> in a <u>base VRC</u>

Patch Description	The description of the <u>patch</u> .
New Base VRC	This field is only applicable for Service Packs. If this check box is selected, the base VRC is a <u>Service Pack</u> ; otherwise, the batch VRC is a <u>patch</u> .
Patch Sequence Number	The patch sequence number. You can only enter a value for Service Packs. For each new Service Pack, you must increase the number by one.
Status	<u>status</u>
Published	If this check box is selected, the patch is published.
Commands	Description
Print Solution/Patch	Starts the <i>Print Solution/Patch (ttpmc1400m000)</i> (p. 4-12) session.
Print Dependencies (Multi-level)	Starts the <i>Print Dependencies (Multilevel) (ttpmc1440m000)</i> (p. 4-29) session.
Print Maintenance History	Starts the <i>Print Maintenance History (ttpmc1460m000)</i> (p. 4-59) session.
Generate Obsolete Solutions	Starts the <i>Generate Obsolete Solutions (ttpmc1252s000)</i> (p. 4-52) session. This option is only available if the following conditions are met: ■ The patch is a <u>Service Pack</u>. ■ A previous Service Pack exists that has a <u>prerequisitedependency</u>. ■ The <u>status</u> of the previous Service Pack is Released.
Generate Patch	Starts the <i>Generate Patch (ttpmc1250s000)</i> (p. 4-41) session for a <u>patch</u> or the <i>Generate Patch for New Base VRC (ttpmc1253s000)</i> (p. 4-53) session for a <u>Service Pack</u>. This option is only available if the <u>status</u> of the patch is not yet Released.
Solutions	Starts the <i>Solutions (ttpmc1100m000)</i> (p. 4-8) session, which lists all the solutions for the selected patch.
Dependencies	Starts the <i>Dependencies (ttpmc1140m000)</i> (p. 4-26) session.

Validate Patch	Starts the <i>Validate Patch (ttpmc1401s000)</i> (p. 4-43) session.
Export Patch	Starts the <i>Export Solution/Patch (ttpmc1200s000)</i> (p. 4-30) session. You can only export a <u>patch</u> if the <u>status</u> of the patch is Dependencies Defined .
Maintenance History	Starts the <i>Maintenance History (ttpmc1560m000)</i> (p. 4-56) session.
Change Password	Starts the <i>Password by Patch (ttpmc1105s000)</i> (p. 4-42) session. This option is only available if the following conditions are met: ■ The patch is not a <u>Service Pack</u>. ■ The <u>status</u> of the <u>patch</u> is not Released.

Generate Patch (ttpmc1250s000)

Use this session to generate a patch.

The generation of a patch only links a group of solutions to the patch. To link solutions one by one to a patch, you can use the *Link Solution to Patch* command on the **Specific** menu of the *Solutions (ttpmc1100m000)* (p. 4-8) session.

After you link the solutions to a patch, you must define the dependencies of the patch to other patches, and you must validate and export the patch. To carry out these actions, you can use the commands on the **Specific** menu of the *Patches (ttpmc1501m000)* (p. 4-38) session.

Note

You cannot include solutions that are already part of another patch.

Using the *Remove Solution from Patch* command on the **Specific** menu of the *Solutions (ttpmc1100m000)* (p. 4-8) session, you can remove a solution from a patch.

The following figure shows the form of the session:

Fields	Description
Base VRC	<u>base VRC</u>
Patch	A unique identification of the <u>patch</u> in a <u>base VRC</u> .
Solution Code	A unique identification of the solution in a <u>base VRC</u> .
Export Date	The date on which the solution was exported.
Export Time	The time at which the solution was exported.

Commands	Description
Generate	Generates the <u>patch</u> .

Password by Patch (ttpmc1105s000)

Use this session to enter a password for a patch. A patch is intended to be installed as a whole. To break up a patch to install the solutions the patch contains separately, you must enter a password.

If you want to specify a password, you must specify the password before you run the *Export Solution/Patch (ttpmc1200s000)* (p. 4-30) session.

You can only start this session from the **SpecificSpecific** menu or the toolbar of the *Patches (ttpmc1501m000)* (p. 4-38) details session.

Fields	Description
Base VRC	<u>base VRC</u>
Patch Code	A unique identification of the patch in a <u>base VRC</u> .

New Password Enter the password in this field.

Reenter Pass- Enter the password a second time to confirm the new
word password.

Validate Patch (ttpmc1401s000)

Use this session to validate a patch or Service Pack.

You must validate a patch or Service Pack before you can export the patch or Service Pack by means of the *Export Solution/Patch (ttpmc1200s000)* (p. 4-30) session.

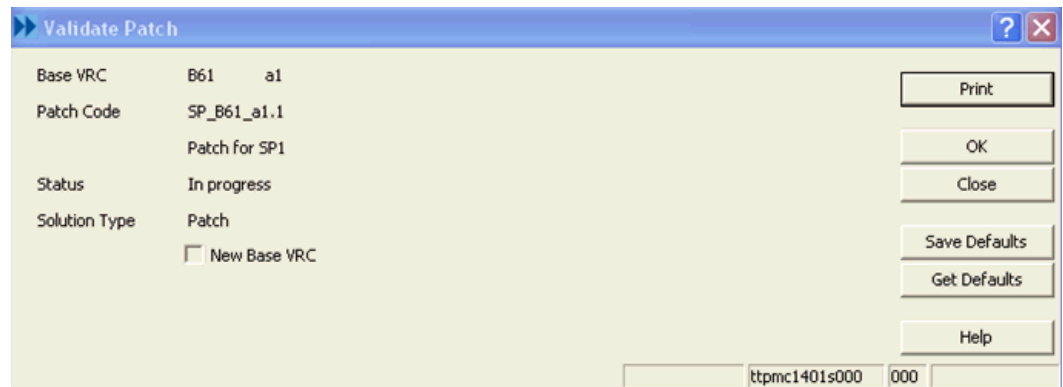
The validation performs the following:

- Checks whether the dumps of the solutions are present in the *Path for Export Dumps* (p. 4-3) directory specified in the *Parameters (ttpmc0100s000)* (p. 4-1) session. The dumps of the solutions are placed in this directory when the solutions are exported individually, which you can do through the *Export Solution/Patch (ttpmc1200s000)* (p. 4-30) session.
- Checks whether the status of the solutions is **Released**, which can only be the case if the solution was exported.
- Checks if any solution in the patch has a post-requisite dependency relation with another solution, and whether the other solution is also present in the patch.
- If any solution in the patch has a co-requisite relation with another solution, whether the other solution is present in another patch, and a co-requisite relation is defined between the current patch and the other patch.

For Service Packs, the validation performs the following additional checks:

- The patch header must be present.
- The installable unit key name and the version of the base VRC must be present.
- The prerequisite to the preceding Service Pack must be defined.
- All solutions created in the previous Service Pack must exist as obsolete solutions in this Service Pack.

If you export a patch or Service Pack, the current session starts automatically. If any of the checks fails, the export stops. You can also run this session manually from the **Specific** menu of the *Patches (ttpmc1501m000)* (p. 4-38) session.



Fields	Description
Base VRC	<u>base VRC</u>
Patch Code	A unique identification of the <u>patch</u> in a <u>base VRC</u> .
Status	<u>status</u>
Solution Type	Always: <u>Patch</u> .
New Base VRC	If this check box is selected, the patch is a Service Pack.

Commands	Description
Print	Prints the report of the validation.

Patch Header (ttpmc1106m000)

Use this session to maintain patch headers.

You can only create a patch header for a Service Pack.

A patch header contains metadata that steers the installation of Service Packs in a recipient environment if you use the SSA installation wizard.

This session creates an XML patch header that you can view if you click **Text**.

Fields	Description
Base VRC	<u>base VRC</u>
Patch Code	The unique identification of a solution in a <u>base VRC</u> .
Action for Package Combination	The SSA Installation Wizard can automatically perform particular actions at a recipient system, such as create new package combinations, create new update VRCs, and link new update VRCs to an existing or new package combination. This field controls which action is being performed with respect to Package Combinations (PCs) when you install a Service Pack in a PMC recipient environment using the SSA Installation Wizard. The following options are available: No Action No specific action is being performed. No PCs are being created or changed. New PC, All Packages This option is relevant for derived products. Suppose that a few packages are changed for a derived product. A new package combination will be created. The package VRCs of the changed packages of the derived product will be linked to the new PC. Moreover, the package VRCs of the unchanged packages of the parent product will also be linked to the new PC. Example: A recipient environment contains a PC with the following package VRCs:

Package	VRC
ci	B61U_a_stnd
tc	B61U_a_stnd
td	B61U_a_stnd
tf	B61U_a_stnd
wh	B61U_a_stnd

A Service Pack for a derived product has been created. The Service Pack header field *Action for Package Combination* contains the value **New PC, All Packages** and the Service Pack contains the following changed packages:

Package	VRC
tc	B61U_a_derv
td	B61U_a_derv

If the Service Pack for the derived product is installed in the recipient environment, a new PC will be created that contains the following package VRCs:

Package	VRC
ci	B61U_a_stnd
tc	B61U_a_derv
td	B61U_a_derv
tf	B61U_a_stnd
wh	B61U_a_stnd

New PC, Only Packages of Base VRC

This option is relevant if you want to deliver new products that are not derived from a parent product and that are stand-alone products. A new PC is being created that will only contain package VRCs for the packages that are included in the Service Pack.

Example:

A recipient environment contains a PC that has the following package VRCs:

Package	VRC
ci	B61U_a_stnd
tc	B61U_a_stnd
tf	B61U_a_stnd
wh	B61U_a_stnd

A Service Pack for a new independent product has been created. The Service Pack header field *Action for Package Combination* contains the value **New PC, Only Packages of Base VRC** and the Service Pack contains the following packages:

Package	VRC
lw	B61U_a_indp
lz	B61U_a_indp

If the Service Pack for the independent product is installed in the recipient environment, a new PC will be created. This new PC will only contain the package VRCs of the new, independent product. The existing PC will remain unchanged.

Add Packages to existing PC

This option is relevant if you want to deliver new products that are not derived from a parent product, and that are stand-alone products. However, you want to include the package VRCs of the new product in an existing PC. New packages present in the Service Pack will be linked to an existing PC. If no PCs exist in the recipient environment available, the Service Pack will be installed, but the update VRCs are not linked to a PC.

Example:

A recipient environment contains a PC that has the following package VRCs:

Package	VRC
ci	B61U_a_stnd
tc	B61U_a_stnd
tf	B61U_a_stnd

Package	VRC
wh	B61U_a_stnd

A Service Pack for a new independent product has been created. The Service Pack header field *Action for Package Combination* contains the value **Add Packages to existing PC** and the Service Pack contains the following package:

Package	VRC
tm	2.6_a

The existing PC will be extended during installation of the Service Pack. After installation of the Service Pack, the PC will contain the following package VRCs:

Package	VRC
ci	B61U_a_stnd
tc	B61U_a_stnd
tf	B61U_a_stnd
wh	B61U_a_stnd
tm	2.6_a

Note:

If no PC was available before the installation of the Service Pack, no PC will be created at all during installation of the SP. Refer also to the **Create New PC if no PC exists** option.

Create New PC if no PC exists

This option is similar to the preceding **Add Packages to existing PC** option. There is one difference: In case there are no existing PCs in the recipient environment available to which the package VRCs can be added, a new PC will be created and the Service Pack will be installed.

Package Combination You can only enter a package combination if the *Action for Package Combination* field contains one of the following values:

- New PC, All Packages
- New PC, Only Packages of Base VRC
- Create New PC if no PC exists

In the SSA Installation Wizard, the package combination you enter appears as the default, however, you can overwrite this value. The package combination will be created in the recipient environment.

Package Combination description	The description of the package combination. In the SSA Installation Wizard, this description appears as the default, but can be overwritten.
---------------------------------	--

Derived From Base VRC	This field is only available if the <i>Action for Package Combination</i> field contains the value New PC, All Packages . Using this field, you can define relationships between the base VRC of <u>derived product</u> and the related parent product. You use this relationship if you create and derive update VRCs at a recipient during installation of a Service Pack. You must always define the relationship to the base VRC of the Service Pack on which the Service Pack of the derived product is built.
-----------------------	--

Example:

The following products are available:

- Standard product PARENT
- Derived product LOC1 which is derived from PARENT
- Derived product LOC2 which is derived from LOC1

Two Service Packs are available. In this scenario, the **Derived From Base VRC** must be populated as follows:

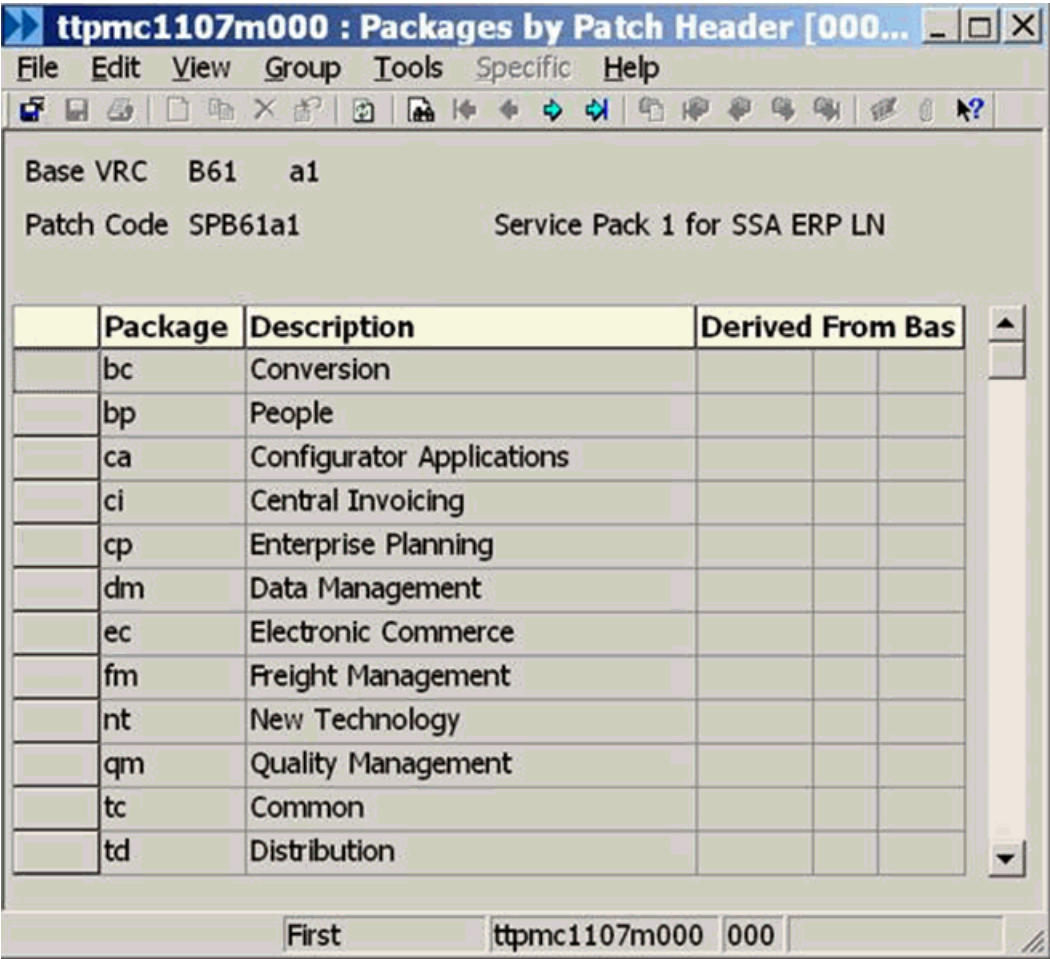
Product	Service Pack	Base VRC	Derived From Base VRC
PARENT	1	B61_a1	
PARENT	2	B61_a2	
LOC1	1	B61L_a1_loc1	B61_a1
LOC1	2	B61L_a2_loc1	B61_a2
LOC2	1	B61L_a1_loc2	B61L_a1_loc1
LOC2	2	B61L_a2_loc2	B61L_a2_loc1

Commands	Description
Packages	This command starts the <i>Packages by Patch Header (ttpmc1107m000)</i> (p. 4-50) session.

Default This command reverts to the defaults determined by PMC, based on the VRC derivation structure.

Packages by Patch Header (ttpmc1107m000)

Use this session to maintain packages by patch header.
You can start this session from the *Patch Header (ttpmc1106m000)* (p. 4-44) session.



Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification for the <u>solution</u> in a <u>base VRC</u> .

Package The package code.

Derived From Base VRC

Using this field, you can define relationships between the base VRC of a derived product and the related parent product. You use this relationship if you create and derive update VRCs at a recipient during installation of a Service Pack. You must always define the relationship to the base VRC of the Service Pack on which the Service Pack of the derived product is built.

Example:

The following products are available:

- Standard product PARENT with package p1, p2, and p3
- Derived product LOC1, which is derived from PARENT. Packages p1 and p2 are changed. Note that package p3 is not changed for LOC1. A new package, p4, was introduced for LOC1.
- Derived product LOC2, which is derived from LOC1. Packages p1, p2, p3, and p4 are changed. Note that package p3 is changed for LOC2.

Two Service Packs are available. In this scenario, you must populate the **Derived From Base VRC** as follows:

Product	Service Pack	Pack-age	Base VRC	Derived From Base VRC
PARENT	1	p1	B61_a1	
PARENT	1	p2	B61_a1	
PARENT	1	p3	B61_a1	
PARENT	2	p1	B61_a2	
PARENT	2	p2	B61_a2	
PARENT	2	p3	B61_a2	
LOC1	1	p1	B61L_a1_loc1	B61_a1
LOC1	1	p2	B61L_a1_loc1	B61_a1
LOC1	1	p4	B61L_a1_loc1	
LOC1	2	p1	B61L_a2_loc1	B61_a2
LOC1	2	p2	B61L_a2_loc1	B61_a2

LOC1	2	p4	B61L_a2_loc1	
LOC2	1	p1	B61L_a1_loc2	B61L_a1_loc1
LOC2	1	p2	B61L_a1_loc2	B61L_a1_loc1
LOC2	1	p3	B61L_a1_loc2	B61_a1
LOC2	1	p4	B61L_a1_loc2	B61L_a1_loc1
LOC2	2	p1	B61L_a2_loc2	B61L_a2_loc1
LOC2	2	p2	B61L_a2_loc2	B61L_a2_loc1
LOC2	2	p3	B61L_a2_loc2	B61_a2
LOC2	2	p4	B61L_a2_loc2	B61L_a2_loc1

Generate Obsolete Solutions (ttpmc1252s000)

Use this session to, as shown in the following figure, to generate obsolete solutions.

This session is one of the steps in the process to create a Service Pack. Obsolete solutions are created for all solutions of the previous base VRC that have the statusReleased.

You can start this session from the *Patches (ttpmc1501m000)* (p. 4-38) session.

Fields	Description
Base VRC	<u>base VRC</u>
Patch	The <u>patch</u> code.

Base VRC Previous Patch base VRC, version

Previous Patch The previous patch code.

Selection Range

Solution The Solution code.

Export Date / Time The export date and time of the previous patch.

Commands Description

Generate Starts the generate obsolete solutions process.

Generate Patch for New Base VRC (ttpmc1253s000)

Use this session to generate Service Packs.

You can start this session from the *Patches (ttpmc1501m000)* (p. 4-38) session.

You can only generate one Service Pack for a base VRC.

ttpmc1253s000 : Generate Patch for New Base VRC [000] Gem Build

Base VRC: B61 a1

Patch: SPB61a1
Service Pack 1 for SSA ERP LN 6.1

Actions

- ☒ Link Obsolete Solutions
- ☒ Create Solutions for Export VRC Export VRC: B61D a1 Default
- ☐ Create Solutions for Previous VRCs Export VRC: B61 a Default
- ☐ Export Solutions
- ☐ Validate Patch after Generation

Options for Creating Solutions

Max. Number of Components: 1500

- ☐ Language-Dependent Components Only
- ☐ Objects Only

Selection for Creating Solutions

Language: From 2 To 2

Buttons: Generate, Close, Save Defaults, Get Defaults, Add to Job..., Help

Status: ttpmc1253s000 000

Fields	Description
Base VRC	<u>base VRC</u>
Patch	The <u>patch</u> code.
Actions	
Link Obsolete Solutions	If this check box is selected, <u>Obsolete Solutions</u> , created with the <i>Generate Obsolete Solutions (ttpmc1252s000)</i> (p. 4-52) session, are linked to this Service Pack.
Create Solutions for Export VRC	If this check box is selected, <u>solutions</u> of the current base VRC are generated for this Service Pack. Software components of the given export VRC are linked to the created Solutions. The user has the option to delete the current solutions of this Service Pack. The created solution codes have a prefix PS and a sequence number as a suffix. Note This action does not create obsolete solutions. Note For this action, the <i>Connect Components to PMC Solution (ttpmc1221m000)</i> (p. 4-17) session is called.
Export VRC	By default, the <u>export VRC</u> of the base VRC is shown. You can enter a different export VRC. This export VRC must contain all components that are changed for the Service Pack. This export VRC can contain all components, including the unchanged components. In the latter case, you do not need to select the Create Solutions for Previous VRCs check box.
Create Solutions for Previous VRCs	If this check box is selected, <u>solutions</u> of the preceding base VRC will be created for this Service Pack. All software components that are not present in the export VRC of the Service Pack you want to generate are selected from the preceding Service Pack. The user has the option to delete the current solutions of this Service Pack.

Note

This action does not create obsolete solutions.

Note

Generating subsequent Service Packs with separating software components that are changed and that are not changed, increases the installation speed at the recipient side.

PMC only installs the software components that are changed after the previously installed Service Pack.

Export VRC	By default, the derived-from VRC of the other given <u>export VRC</u> is shown. You can enter a different export VRC. This export VRC must contain the preceding Service Pack.
Export Solutions	If this check box is selected, the solutions are exported. As a result, a dump with the components is stored on the file system of the operating system. Note For this action, the <i>Export Solution/Patch (ttpmc1200s000)</i> (p. 4-30) session is called.
Validate Patch after Generation	If this check box is selected, the generated Service Pack is validated. Note For this action, the <i>Validate Patch (ttpmc1401s000)</i> (p. 4-43) session is called.

Options for Creating Solutions

Max. Number of Components	The maximum number of software components of the solutions in a Service Pack. Using this field, you can control the size of the generated solutions and, with that, the size of the logical transactions when the solution is installed at a recipient. For example, if 80 software components exist and the Max. Number of Components is 30, three solutions are generated. The first and second solutions contain 30 components, and the third solution contains 20 components.
Language-Dependent Components Only	If this check box is selected, only language-dependent components are linked to the solution.

As a result, the following non-language-independent components are excluded:

- Scripts
- Objects
- Functions
- Non-enumerated domains

This option is particularly useful for solutions that serve translation purposes.

Selection for Creating Solutions

Language	The language code.
----------	--------------------

Commands	Description
----------	-------------

Generate	Starts the Service Pack generation process.
----------	---

Maintenance History (ttpmc1560m000)

Use this session to display the status changes to a solution or patch for a base VRC.

Every status change of a solution or patch status is recorded. This session shows all these recorded changes. If you delete a solution, this fact is recorded with the new status as blank.

Usually, to start this session, you can double-click a record in the *General Maintenance History (ttpmc1561m000)* (p. 4-58) session. In that case, the current session shows the maintenance history for the relevant solution or patch up until the specified time.

Maintenance History (Current View: Maintenance History)

File Edit View Group Tools Specific Help

Base VRC B61 a1

Solution Code 203134

Cluster not copied to the Follow-up Service Order / Activity

	Date	Time	Ser.	User	Status
	06/27/05	08:35:32	1	athammin	In progress
	06/27/05	08:35:32	2	athammin	Dependencies defined
	06/27/05	08:35:43	1	athammin	Exported
	06/27/05	08:35:46	1	athammin	Released

ttpmc1560m000 000

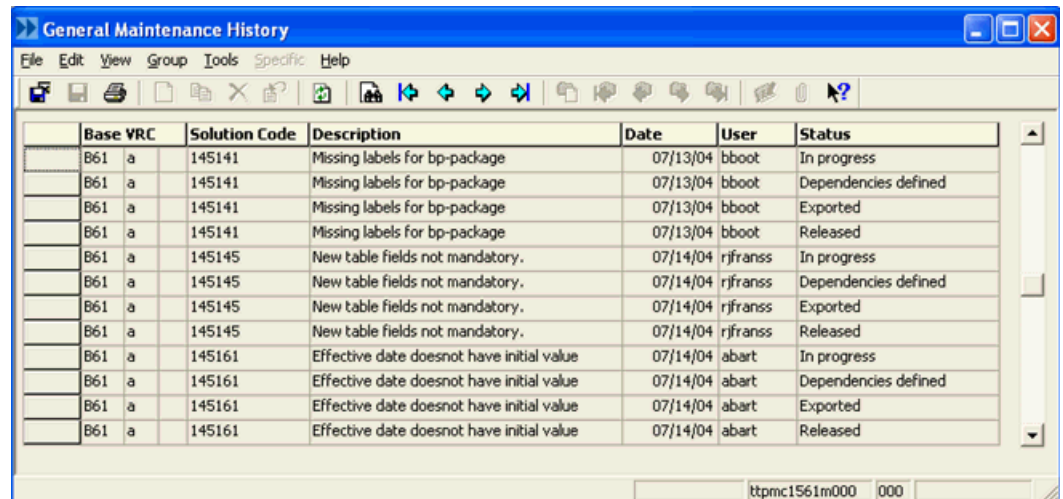
Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification of the solution or <u>patch</u> in a <u>base VRC</u> .
Date	The date on which the status of the solution or <u>patch</u> was changed.
Time	The time when the status of the solution or <u>patch</u> changed.
Ser.	Number to make the key unique.
User	The name of the user who changed the solution or <u>patch</u> .
Status	The new status of the solution or <u>patch</u> .

Commands	Description
Print Maintenance History	Starts the <i>Print Maintenance History (ttpmc1460m000)</i> (p. 4-59) session.

General Maintenance History (ttpmc1561m000)

Use this session to display the status changes of all solutions and patches. Every status change of a solution or patch is recorded. This session shows all these recorded changes. If you delete a solution or patch, this fact is recorded with the new status as blank.

If you double-click a record, the *Maintenance History (ttpmc1560m000)* (p. 4-56) session starts and the the maintenance history for the selected solution or patch appears up until the specified time.



Base VRC	Solution Code	Description	Date	User	Status
B61 a	145141	Missing labels for bp-package	07/13/04	bboot	In progress
B61 a	145141	Missing labels for bp-package	07/13/04	bboot	Dependencies defined
B61 a	145141	Missing labels for bp-package	07/13/04	bboot	Exported
B61 a	145141	Missing labels for bp-package	07/13/04	bboot	Released
B61 a	145145	New table fields not mandatory.	07/14/04	rjfranss	In progress
B61 a	145145	New table fields not mandatory.	07/14/04	rjfranss	Dependencies defined
B61 a	145145	New table fields not mandatory.	07/14/04	rjfranss	Exported
B61 a	145145	New table fields not mandatory.	07/14/04	rjfranss	Released
B61 a	145161	Effective date doesnot have initial value	07/14/04	abart	In progress
B61 a	145161	Effective date doesnot have initial value	07/14/04	abart	Dependencies defined
B61 a	145161	Effective date doesnot have initial value	07/14/04	abart	Exported
B61 a	145161	Effective date doesnot have initial value	07/14/04	abart	Released

Fields	Description
Base VRC	<u>base VRC</u>
Solution Code	A unique identification of the solution or <u>patch</u> in a <u>base VRC</u> .
Date	The date on which the status of the solution or <u>patch</u> was changed.
User	The user who changed the status of the solution or <u>patch</u> .

Status The new status of the solution or patch.

Commands	Description
Print Maintenance History	Starts the <i>Print Maintenance History (ttpmc1460m000)</i> (p. 4-59) session.

Print Maintenance History (ttpmc1460m000)

Use this session to print the history of changes for a solution or patch.

This session prints the same data that appears in the *General Maintenance History (ttpmc1561m000)* (p. 4-58) session.

The screenshot shows the 'Print Maintenance History' dialog box. It features a 'From' and 'To' section for filtering data. Under 'Solutions Range', there are input fields for Base VRC, Solution, Status (a dropdown menu), User, and Serial. Under 'Duration Range', there are input fields for Date and Time. On the right side of the dialog, there are buttons for 'Print', 'Close', 'Save Defaults', 'Get Defaults', 'Add to Job...', and 'Help'. At the bottom right, there is a text field containing the session identifier 'ttpmc1460m000 000'.

Fields	Description
Base VRC	<u>base VRC</u>
Solution	The unique identification code of the solution in the <u>base VRC</u> .
Status	<u>status</u>
User	The name of the user who changed the solution or <u>patch</u> .
Serial	A number that makes records with the same data unique.
Date	The date at which the solution or <u>patch</u> was changed.

Time The time at which the solution or patch was changed.

Commands	Description
----------	-------------

Print	Prints the report.
-------	--------------------

Appendix A

Distributor session summary

A

PMC Distributor Sessions

Title	Description
<i>Parameters (ttpmc0100s000) (p. 4-1)</i>	Maintain the parameters for the PMC module.
<i>Base VRC's (ttpmc0110m000) (p. 4-4)</i>	Maintain the relation between <u>base VRCs</u> and <u>export VRCs</u> .
<i>Base VRC Combinations (ttpmc0111m000) (p. 4-5)</i>	Display the base VRC combinations.
<i>Base VRC Combinations (ttpmc0111m100) (p. 4-6)</i>	Maintain the base VRC combinations.
<i>Base VRCs by Base VRC Combination (p. 4-7)</i>	Link the base VRCs to a base VRC combination.
<i>Solutions (ttpmc1100m000) (p. 4-8)</i>	Maintain solutions.
<i>Collections (ttpmc1101s000) (p. 4-33)</i>	Maintain <u>collections</u> .
<i>Password by Patch (ttpmc1105s000) (p. 4-42)</i>	Enter a password for a <u>patch</u> . A patch is meant to be installed as a whole. If you want to break up a patch

	to install the solutions it contains separately, a password is required.
<i>Patch Header (ttpmc1106m000) (p. 4-44)</i>	Maintain patch headers.
<i>Packages by Patch Header (ttpmc1107m000) (p. 4-50)</i>	Link packages to a patch header.
<i>Defects by Solution (ttpmc1110m000) (p. 4-13)</i>	Specify which defects are solved by a solution.
<i>Component by Solution (ttpmc1120s000) (p. 4-16)</i>	Link a component to a solution.
<i>Sessions by Component (ttpmc1125m000)</i>	Display the sessions that use a component that is linked to a solution.
<i>Dependencies (ttpmc1140m000) (p. 4-26)</i>	Define the dependencies between solutions or <u>patches</u> .
<i>Solutions by Collection (ttpmc1151m000) (p. 4-34)</i>	Add solutions to a <u>collection</u> .
<i>Export Solution/Patch (ttpmc1200s000) (p. 4-30)</i>	Create the export dump for a solution or a <u>patch</u> .
<i>Export Solution Multi-level (ttpmc1202s000) (p. 4-31)</i>	Create a multilevel export dump for a released solution.
<i>Export Collection (ttpmc1210s000) (p. 4-36)</i>	Export a <u>collection</u> , which means that a dump of the collection is made.
<i>Connect Components to PMC Solution (ttpmc1221m000) (p. 4-17)</i>	Connect software components in the <u>export VRC</u> to a PMC solution.
<i>Generate Dependencies (ttpmc1240s000) (p. 4-28)</i>	Generate the dependency relations between solutions which must be installed together at a customer system.
<i>Generate Patch (ttpmc1250s000) (p. 4-41)</i>	Generate a <u>patch</u> .
<i>Generate Collection (ttpmc1251s000) (p. 4-34)</i>	Add a range of solutions to a <u>collection</u> .

<i>Generate Obsolete Solutions</i> (ttpmc1252s000) (p. 4-52)	Generate obsolete solutions.
<i>Generate Patch for New Base VRC</i> (ttpmc1253s000) (p. 4-53)	Generate patch for New Base VRC.
Select "In Progress" Solution (ttpmc1299m000)	Link a component to an In progress solution.
<i>Print Solution/Patch</i> (ttpmc1400m000) (p. 4-12)	Print solutions and <u>patches</u> .
<i>Validate Patch</i> (ttpmc1401s000) (p. 4-43)	Validate a <u>patch</u> .
<i>Validate Collection</i> (ttpmc1402s000) (p. 4-35)	Validate a <u>collection</u> .
<i>Print Collection</i> (ttpmc1403m000) (p. 4-37)	Print the contents of a <u>collection</u> .
<i>Print Components by Solution</i> (ttpmc1420m000) (p. 4-25)	Print components connected to a solution.
<i>Print Dependencies (Multilevel)</i> (ttpmc1440m000) (p. 4-29)	Print the dependencies.
<i>Print Maintenance History</i> (ttpmc1460m000) (p. 4-59)	Print the history of changes for a solution or <u>patch</u> .
<i>Patches</i> (ttpmc1501m000) (p. 4-38)	Maintain <u>patches</u> .
Patches (ttpmc1501m100)	Maintain patches
<i>Collections</i> (ttpmc1503m000) (p. 4-31)	Maintain <u>collections</u> .
<i>Components by Solution</i> (ttpmc1520m000) (p. 4-14)	Connect the modified components to a solution.
Components by solution (ttpmc1521m000)	View the components that are linked to a solution.
Dependencies (ttpmc1541m000)	View the dependencies between solutions or <u>patches</u> .
Solutions by Collection (ttpmc1551m000)	View the solutions that are part of a <u>collection</u> .

<i>Maintenance History (ttpmc1560m000)</i> (p. 4-56)	Display the status changes of a solution or <u>patch</u> for a <u>base VRC</u> .
<i>General Maintenance History</i> (ttpmc1561m000) (p. 4-58)	Display the status changes of a all solutions and <u>patches</u> .
Specify Another Solution (ttpmc1821s000)	Specify another solution.

Appendix B

Glossary

B

additional file

A generic component, such as an XML schema file, a GIF image, and so on. From SSA ERP 6.0a on, additional files are stored in a specific package, module, and package VRC.

base VRC

A means in PMC to identify products in a unique way. Updates at the distributor side are provided with the base VRC identifier. A base VRC can contain the code of the physical VRC in which the related master product is installed, for example, B61_a, but can also be a code not related to a physical VRC, for example, 7.6_a_tt. At the recipient side, every update VRC is linked to a base VRC identifier. The installation process checks if the base VRC identifier of the update matches with the base VRC identifier of the update VRC. If not, you cannot install the update in that update VRC.

base VRC combination

A base VRC Combination is defined at the PMC distributor side and consists of a set of related base VRCs. A base VRC combination controls the creation of co-requisites between base VRCs. You can only define co-requisites between base VRCs that are part of the same base VRC combination. Base VRC combinations prevent the unwanted creation of co-requisites between base VRCs.

Business Object

A Business Object is an object understandable by the business, such as a purchase order or an organizational unit. A Business Object has information stored in the Business Object attributes, such as the purchase order number or the organizational unit name. A Business Object also contains a set of actions, known as Business Object methods, that can manipulate the Business Object attributes, such as create purchase order and list organizational units.

From a development perspective, a Business Object is a collection of tables and functions that manipulate these tables that are implemented simultaneously as one group during the development phase. A Business Object is identified by the combination of a package code, module code, and Business Object code.

chart

A graphic or diagram that displays data or the relationships between sets of data in pictorial rather than numeric form. The data can be presented in a graph, a line, or a pie, and can include titles, legends, and footnotes.

check in

A process that releases the checked-out software component and copies the software component from the *development VRC* (p. B-4) to the original VRC. This process also stores a historical version.

A revision text for the changed software component is required for a check-in process.

check out

A process that locks the software component for other developers. During the check-out phase, other related components are locked as well while the component can be updated and tested. A copy of the component is made from the original VRC to the development VRC.

collection

In PMC, a collection is a group of individual solutions. At the PMC distributor side, you can perform grouping in various ways, for example, manual grouping based on a functional topic or grouping based on solutions created in a particular period etc. You cannot define dependencies between collections. At the recipient side, the entity collection is not available. When a collection is scanned, the individual solutions are added to the PMC registry and can be processed individually.

co-requisite

In general, co-requisites are defined between solutions of a standard product and derived products. Co-requisites guarantee that related products are updated simultaneously under the condition that the update VRCs of the related products are linked to the same VRC combination. The order of installation is not relevant. The solutions can have the same base VRC, or different base VRCs.

customization

A derived product, meant to fulfill the requirements of a specific customer or group of customers.

dependency

In PMC, the relation between solutions. Dependencies are defined at the PMC distributor side and are part of the meta data of a PMC solution and guarantee that PMC solutions are installed in the correct configuration and sequence at the PMC recipient side.

The following values indicate the dependency type between solutions.

Three dependency types are available:

- Pre-requisites
- Co-requisites
- Post-requisites

You can only install solutions that are dependent on other solutions if the other solutions are already present, or are also installed.

The same dependency types exist between patches. However, to keep the descriptions readable, only solutions are mentioned, but patches are meant as well. One exception applies: the post-requisite type is not applicable to patches.

derived product

A product that is derived from a parent master product. The parent product can, for example, be a main ERP version such as SSA ERP LN 6.1. In general, the derived product contains a subset of components from the parent product that are changed and possibly a number of new software components.

There are three categories of derived products:

- Localizations
- Extensions
- Customizations

details session

A dialog box that shows all the details (fields) of the line (record) selected in the associated overview session. Use a details session to view, enter, or change the data of one record.

A details session can contain a number of tabs to group related fields.

development VRC

In PMC a physical VRC, derived from the export VRC, in which checked-out software components are temporarily stored during a change process.

domain

A domain describes the properties of table fields. The main property of a field is the data type, e.g. string, long, double, date, UTC. Other properties are, for example, the length of a string, the alignment of a string, and the date and time format.

In the Chart Manager, a range with an initial value, an end value, and a step size. Domains indicate the valid values for an attribute and are used to define the scale division of the axes or to verify data.

export VRC

The physical VRC from which components that belong to a PMC solution must be exported at the PMC distributor side. Each base VRC has an export VRC linked, so components for different products are exported from different physical VRCs.

extension

A derived product meant to fulfill requirements of a particular line of business.

form

A screen that appears when a session is started. A form interacts with the database, and provides the user interface that is used to manipulate the data on the form.

function

A piece of program code that makes up part of a program script.

A function is a self-contained software routine that can perform a task for the program in which the function is written, or for another program.

integrated session

The session and the session's form are integrated into one object. The form is a subcomponent of the session.

When you perform an operation, for example, copy, delete, check-in, or checkout, on an integrated session, you also perform the operation on the integrated session's form.

A non-integrated session's form is a separate object.

label

A code that is used instead of language-dependent text in forms, reports, and menus. A label consists of a name and a content description. The content of a label can differ by language, but the label name remains the same for all languages.

menu

A list of options from which a user can make a selection to perform a desired action, such as starting sessions, other menus, and queries. A start-up menu is defined for each user. Using this start-up menu, the user can access all sub-menus attached to the start-up menu tree.

message

A notification that informs you about something. More specifically, a message attends you to an event, error, warning, and so on. Messages usually appear in a message window or logged in a file. If displayed in a window, a message requires a confirmation: Click OK. Messages are distinguished from questions, which always require a choice response.

obsolete solution

Obsolete solutions are an administrative aid to manage the synchronization of updates at the PMC recipient side when you install a Service Pack. An obsolete solution does not contain software components.

package

A set of related modules that implements a complete part of the functionality, for example, Enterprise Planning, Financials, or Warehouse Management. Packages are designed to function as independent as possible, to enable a customer to implement only particular packages.

A package code consists of two characters. For example, it is the code of the SSA ERP Tools package.

Each package has a unique version structure, the VRC structure.

package VRC

See: *version - release - customer* (p. B-13)

patch

In PMC, a patch is a collection of solutions. In general a patch contains solutions created in a larger time period. The patch entity is both known at the PMC distributor and PMC recipient side. Patches are an indivisible set of solutions. You cannot install or uninstall individual solutions that belong to a patch at the PMC recipient. You can only install or uninstall patches as a whole. You can define dependencies between patches. Patches leave the base VRC that is linked to the update VRC at the PMC recipient unchanged. The existing PMC registry will remain and will be extended with data of the newly installed patch. Patches only permit the most recent version of software components to be maintained. Patches in general mainly contain corrective solutions.

Note

In PMC versions earlier than SSA ERP LN 6.1, the synonym Service Packs was often used for patches.

PMC

See: *product maintenance and control* (p. B-8)

PMC distributor

The functional part of PMC that manages the creation of updates. PMC Distributor is especially used by software vendors who create updates.

PMC recipient

The functional part of PMC that manages the installation of updates. Customers, who install updates in particular use PMC recipient.

PMC version

The PMC version is linked to every update and is intended to guarantee that various formats of PMC solution dumps are handled by the right version of the PMC software.

- PMC version 0 : does not support Service Packs
- PMC version 1 : supports Service Packs

Note

Dumps created for a higher PMC version cannot be processed at the recipient side if PMC recipient has not been upgraded to that PMC version.

Dumps of older PMC versions can always be processed.

post-requisite

Post-requisites are mainly meant to prevent the installation of bad solutions. In general, a post-requisite is a link from an earlier, bad solution to a more recent, correct solution.

pre-requisite

Pre-requisites mainly steer the sequence in which solutions are installed. In general a pre-requisite is the link from a more recent solution to a predecesing solution. Pre-requisites are the most common type of dependencies. A pre-requisite dependency exists between two solutions if one solution must have been installed before the other solution is installed. In that case, the first solution is a pre-requisite for the other solution. Typically, pre-requisite dependencies exist between a solution and a previous solution, if these solutions have one or more components in common. Pre-requisite dependencies can only be created to solutions in the same base VRC.

product maintenance and control

Product maintenance and control (PMC) is a tool that helps a customer manage the updates of the SSA ERP LN system.

With the PMC tool, you can check all patches against the customer's SSA ERP LN system to verify their completeness, check any potential interference with the customization, and detect dependencies.

These capabilities ensure the complete and accurate installation of each software patch and Service Pack. In addition, using the PMC tool enhances the quality of the support being available to customers.

PMC consists of a PMC distributor part and a PMC recipient part.

Acronym: PMC

program script

A sequence of instructions that is used to program a number of actions that must take place in addition to the standard program. Two different program scripts are available, namely 3GL scripts and 4GL scripts.

Synonym: UI script

PVRC

See: *version - release - customer* (p. B-13)

question

A notification that requires a choice response. For example, a question can prompt you to confirm or cancel a delete action. If you do not respond to a question, the process that prompted the question cannot continue. Questions are distinguished from *messages* (p. B-5), which offer no choice and do not necessarily require a response.

report

A report is used to present data from the database, usually on paper. The report can be sent to a device, such as a physical printer, a display, or to a file.

SCM

See: *software configuration management* (p. B-10)

Service Pack

In PMC, a Service Pack is a collection of solutions. In general, a Service Pack contains solutions created in a larger time period. In PMC the term 'patch' is also applied for Service Packs. The patch entity is both known at the PMC distributor and PMC recipient side. A property in the patch entry makes the difference between patches and Service Packs. Service Packs are an indivisible set of solutions. You can install or uninstall solutions that belong to a Service Pack at the PMC recipient. You can only install or uninstall Service Packs as a whole. You can define dependencies between Service Packs. Service Packs are intended to enable you to maintain multiple base VRCs in parallel. Service Packs change the base VRC that is linked to the update VRC at the PMC recipient. The existing PMC registry for the update VRC will be moved to history and a new registry will be started for the update VRC. This type of patch in general contains a significant amount of functional changes.

Note

Service Packs as described in the preceding definition do not exist in PMC versions earlier than SSA ERP LN 6.1.

Acronym: SP

session

An elementary part of SSA ERP the user can start to run an application's functionality. Usually, a session is linked to a main database table and a program script. In addition, a session uses zero or more forms, reports, and charts.

The code of a session consists of a package code, a module code, four digits that indicate the main table number and the session type, an m or an s, and three additional digits, for example, Countries (tcmcs0510m000).

software component

The SSA ERP software exists of the following separate software components:

- message
- report
- label
- function
- business object
- help components
- chart
- integrated session
- additional file
- question
- session
- domain
- table
- menu
- form
- program script
- library

software configuration management

With software configuration management, a developer can modify and test their own revision of a software component. Using a check-out and check-in functionality, a software component is locked for others developers. This method guarantees that no more than one developer can modify the same software component at the same time.

Acronym: SCM

solution

In PMC, the smallest, indivisible type of update. A solution is identified both at the distributor and recipient side by a unique solution code. The term individual solution is also frequently used and has the same meaning.

Note

In the PMC software the term solution is often used as an alternative for the term update. A solution can then be an individual solution, which is the smallest, indivisible type of an update, or a patch.

solution status distributor

The following statuses describe the progress of the maintenance of solutions, collections, and patches. To keep the descriptions readable, only solutions are described. However, in each case, collections and patches are intended as well, unless explicitly excepted.

The status of the solution is only used at the distributor side. A history of status changes is recorded in the **Maintenance History** table.

To distinguish maintained solutions at the distributor side from imported solutions at the recipient side, the status is cleared when the solution is imported. This also applies to patches, but not to collections. The reason is that a collection is not known as such at the recipient side. Only the solutions contained in the collection are known at the recipient side.

- **In progress**
The initial status of the solution upon creation.
- **Dependencies defined**
The dependencies between the various solutions are defined, and the solution is ready to be exported. You must select this status manually, because you might want to create dependencies that cannot be created automatically.
- **Exported**
The solution is exported. The system handles this status. You cannot set the status manually. However, you can set a solution with the status Exported back to In Progress, Solved, Completed, or Dependencies Defined. If you change anything in an exported solution, the status is always set back to Dependencies Defined, and you must export the solution again. Before you can release a solution, the status must be Exported.
- **Released**
The solution is released, which means that the solution is frozen. You can no longer change anything in the solution. To release a solution, you must change the status from Exported to Released, and you must save the solution with this status.

SP

See: *Service Pack* (p. B-9)

table

A data structure that is used to store data that consists of a list of records, each entry being identified by a unique key and containing a set of related values. A table contains a number of table fields that belong to a specific domain.

A table code consists of a package code, module code, and three digits.

Example

Table: tc mcs010 **Countries**

Code	Label	Length	Data Type
ccty	Country	3	String
dsca	Description	30	Multibyte String
meec	EU Member State	5	Enumerated
...			

Table fields

UI script

See: *program script* (p. B-8)

undo check-out

A command of *software configuration management* (p. B-10) that deletes the checked-out software component from the *development VRC* (p. B-4).

update

In PMC, an update is a set of changed software components, including PMC metadata, which is required to install the update in a safe and correct way. An update can contain corrective changes or functional enhancements.

Updates can be delivered in four different configurations:

- Solutions
- Collections
- Patches
- Service Packs

update VRC

A physical VRC at the PMC recipient side in which updates are installed. Every update VRC has a base VRC linked.

version

An upgrade of a package, for example, B61. Each change in the functionality of a package results in a new version. A version can contain several releases.

version - release - customer

The version - release - customer (VRC) code is an identification of a stage in the development of the SSA ERP software. An example of a VRC is *B61_a_ams*.

A VRC code consists of:

- **Version**
A stage in the development in which a major part of the software is modified.
- **Release**
A stage in the development in which a minor part of the software is modified.
- **Customer**
An extension, localization, or customization of the software for a single customer or a small group of customers.

A VRC can be derived from a preceding VRC. Every software component that is contained in the preceding VRC and not explicitly modified or set to expired in the current VRC will also be available in the current VRC.

Acronym:	VRC
Acronym:	PVRC
Synonym:	package VRC

VRC

See: *version - release - customer* (p. B-13)

Index

- additional file**, B-1
- architecture**, 2-2
- base VRC**, 3-3, 4-4, 4-7, B-1
- base VRC combination**, 3-3, 4-5, 4-6, 4-7, B-1
- benefits**, 1-2
- Business Object**, B-2
- chart**, B-2
- check in**, B-2
- check out**, B-2
- collection**, 2-5, 4-33, 4-34, B-2
 - create, 3-6
 - export, 4-36
 - generate, 4-34
 - maintain, 4-31
 - print, 4-37
 - validate, 4-35
- co-requisite**, B-3
- customization**, B-3
- dependency**, 2-4, 4-26, B-3
 - generate, 4-28
 - print multi-level, 4-29
- derived product**, B-3
- details session**, B-4
- development VRC**, 4-4, B-4
- distributor**
 - parameters, 4-1
- domain**, B-4
- export VRC**, 4-4, B-4
- extension**, B-4
- form**, B-4
- function**, B-4
- integrated session**, B-5
- introduction**, 1-1
- label**, B-5
- menu**, B-5
- message**, B-5
- multi-level**
 - print dependencies, 4-29
- obsolete solution**, 4-52, B-5
- overview**, 2-1
- package**, B-5
- package VRC**, B-13
- patch**, 2-9, B-6
 - create, 3-7
 - export, 4-30
 - generate, 4-41
 - header, 4-44, 4-50
 - history, 4-56, 4-58, 4-59
 - maintain, 4-38
 - password, 4-42
 - print, 4-12
 - validate, 4-43
- PMC**, B-8
 - architecture, 2-2
 - base VRC combination, 3-3, 4-5, 4-6, 4-7
 - base VRC, 3-3, 4-4, 4-7
 - benefits, 1-2
 - collection, 2-5, 4-33, 4-34, 4-34
 - connect components to PMC solution, 4-17
 - create collections, 3-6
 - create patches, 3-7
 - create service packs, 3-8
 - create solutions, 3-4
 - defect, 4-13

- dependency, 2-4, 4-26, 4-28
- development VRC, 4-4
- distributor functionality, 2-10
- distributor parameters, 3-2
- distributor policy, 2-9
- distributor session summary, A-1
- distributor setup procedure, 3-2
- export collection, 4-36
- export patch, 4-30
- export solution multi-level, 4-31
- export solution, 4-30
- export VRC, 4-4
- generate service pack, 4-53
- history, 4-56, 4-58
- individual solution, 2-5
- introduction, 1-1
- maintain collections, 4-31
- maintain patches, 4-38
- obsolete solution, 4-8, 4-52
- overview, 2-1
- parameters, 4-1
- password for patch, 4-42
- patch header, 4-44, 4-50
- patch, 2-9, 4-41
- PMC version, 4-5, 4-6
- print collection, 4-37
- print components, 4-25
- print dependencies (multi-level), 4-29
- print maintenance history, 4-59
- print solution/patch, 4-12
- recipient functionality, 2-11
- service pack, 2-7
- service packs and patches, 2-6
- software component, 4-14, 4-16
- solution, 4-8
- validate collection, 4-35
- validate patch, 4-43
- where to find the PMC module, 2-2

PMC distributor, B-6

- functionality, 2-10
- parameters, 3-2
- policy, 2-9
- session summary, A-1
- setup procedure, 3-2

PMC recipient, B-6

- functionality, 2-11

PMC version, 4-5, 4-6, B-7

post-requisite, B-7

pre-requisite, B-7

product maintenance and control, B-8

program script, B-8

PVRC, B-13

question, B-8

recipient

- parameters, 4-1

report, B-8

SCM, B-10

service pack, 2-7

- create, 3-8
- generate, 4-53
- header, 4-44

Service Pack, B-9

service packs and patches, 2-6

session, B-9

software component, B-10

software configuration management, B-10

solution, 4-8, B-10

- connect components, 4-17
- create, 3-4
- defect, 4-13
- export multi-level, 4-31
- export, 4-30
- history, 4-56, 4-58, 4-59
- individual solution, 2-5
- obsolete solution, 4-8, 4-52
- print components, 4-25
- print, 4-12
- software component, 4-14, 4-16

solution status distributor, B-11

SP, B-9

table, B-12

UI script, B-8

undo check-out, B-12

update, B-12

update VRC, B-13

version, B-13

version - release - customer, B-13

VRC, B-13