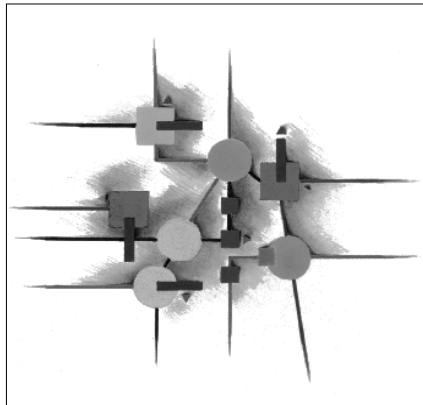


# SAP Business Connector Cooperative Development Guide



## SAP System

Release 4.8



**SAP AG - Dietmar-Hopp-Allee 16 - D69190 Walldorf**



## Copyright

©Copyright 2008 SAP AG. All rights reserved.

No part of this description of functions may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, EXCEL®, NT® and SQL-Server® are registered trademarks of Microsoft Corporation.

IBM®, OS/2®, DB2/6000®, AIX®, OS/400® and AS/400® are registered trademarks of IBM Corporation.

OSF/Motif® is a registered trademark of Open Software Foundation.

ORACLE®, is a registered trademark of ORACLE Corporation, California, USA.

webMethods® is a registered trademark of webMethods Incorporated, Virginia, USA.

INFORMIX®-OnLine for SAP is a registered trademark of Informix Software Incorporated.

UNIX ® and X/Open® are registered trademarks of SCO Santa Cruz Operation.

SAP®, R/2®, R/3®, RIVA®, ABAP/4®, SAPaccess®, SAPmai@l, SAPoffice®, SAP-EDI®, SAP Business Workflow®, SAP Early Watch®, SAP Archive Link®, R/3 Retail®, ALE/WEB®, SAPTRONIC® are registered trademarks of SAP AG.

All rights reserved.

# Contents

<b>Chapter 1. Introduction</b> .....	<b>3</b>
Welcome! .....	4
Typographical Conventions .....	4
Program Code Conventions .....	5
Related Documentation .....	6
Viewing this Document .....	7
Printing this Guide .....	7
<b>Chapter 2. Locking and Unlocking Elements</b> .....	<b>9</b>
Basic Concepts .....	10
What Is a Lock? .....	10
How Do I Know Who Has an Element Locked? .....	11
When Do I Lock an Element? .....	11
When Do I Unlock an Element? .....	11
Locking Elements .....	11
Locking an Element .....	11
Locking All Elements In a Folder or Package .....	12
Locking Java, C/C++, and WebTap Services .....	13
Locking Templates .....	14
System Locking Elements .....	14
Viewing the Status of a Locked Element .....	15
Unlocking Elements .....	15
Unlocking an Element Using Developer .....	16
Unlocking More Than One Element .....	16
Unlocking an Element Using the Server Administrator .....	18
<b>Unlocking a System Locked Element</b> .....	<b>20</b>
Viewing an Element's Corresponding Server Files .....	20
Automatically Unlocking Elements After Saving .....	22
Copying, Moving, and Deleting Locked Elements .....	22
<b>Chapter 3. Using Source Control with the SAP BC Server</b> .....	<b>25</b>
Basic Concepts .....	26
What Is a Third-Party Source Control Application? .....	26
Is Source Control the Same as Locking? .....	26
Methods of Using Source Control .....	26
Components of a SAP BC server Solution .....	27
Package Components .....	27

Location of a Package .....	27
Contents of a Package .....	28
Global Components .....	33
Adding Components to Source Control .....	34
Choosing a Development Methodology for Java Services .....	34
Checking Components Into Source Control .....	35
Checking In the ns Directory .....	36
Checking Components Out of Source Control .....	36
Updating Files That Are in Source Control .....	37
<b>Chapter 4. Troubleshooting .....</b>	<b>39</b>
Introduction .....	40
Lock/Unlock Problems .....	40
Package Management Problems .....	41
Save Problems .....	41
Other Problems .....	41
Frequently Asked Questions .....	41
<b>Chapter 5. Administration and Best Practices .....</b>	<b>43</b>
Introduction .....	44
Disabling and Re-Enabling Locking .....	44
Before You Begin .....	44
Procedure .....	44
Best Practices .....	46
Remote Server Configuration .....	46
Server Usernames .....	46
Package Replication and Publishing .....	47
Package and Folder Organization .....	48
Source Control .....	48
Upgrading SAP BC server .....	48
<b>Index.....</b>	<b>49</b>

# *1*

## **Introduction**

■ Welcome! .....	4
■ Typographical Conventions .....	4
■ Related Documentation .....	6
■ Viewing this Document .....	7

## Welcome!

This guide describes how to manage cooperative development of elements in SAP BC server®. It contains information for developers who want to prevent others from overwriting their work on the SAP BC Server. It also contains information for administrators on how to manage users and overall cooperative development.




To use this guide effectively, you should:

- Understand the basic concepts of source control and change management.
- Know how to develop services and other elements with SAP BC Developer.
- Be familiar with the SAP BC Server.

## Typographical Conventions

This document uses the following typographical conventions:

Convention	Example
Procedures are designated by a blue box in the left column. Procedures are presented as a series of numbered steps.	1 On the <b>Activity</b> menu, click <b>File</b> .
Terms that identify elements, options, selections, and commands on the screen are shown in bold.	The <b>Service</b> field on the <b>Properties</b> tab specifies the name of the requested service.
Characters that you must type exactly are shown in a typewriter font.	Type: <code>setup</code> and then press ENTER.
Variable information that you must type based on your specific situation or environment is shown in italics.	Type: <code>&lt;sapbc&gt;\setup</code> and then press ENTER.
Keyboard keys are shown in uppercase.	Press ENTER; then press TAB.
Keys that you must press simultaneously are joined with the "+" symbol.	Press CTRL+ALT+M.

Convention	Example
Directory paths are shown with the “\” directory delimiter unless the subject is UNIX-specific. In these cases, the “/” is used. If you are working in a UNIX environment, substitute a “/” for the “\” shown in the procedures in this book.	<code>&lt;sapbc&gt;\server\packages \Default</code>
Information that you must read before beginning a procedure or that alerts you to negative consequences of certain actions is denoted using this notation.	 <b>Important!</b> If the folder is not already open in the Service Browser, open it before you start the following procedure.
Notes that provide related, but non-critical, information are denoted using this notation.	 <b>Note:</b> When you start SAP BC Developer, you are prompted to log on to a SAP BC Server.
Helpful information such as shortcuts and alternatives.	 <b>Tip!</b> You can also use CTRL+C to copy an object.

## Program Code Conventions

For programming code and command syntax, this document uses the following typographical conventions:

Convention	Example
Keywords and values that you must type exactly as printed are shown in typewriter font.	<code>%CoSymbol%</code>
Variable values or parameters that you must supply are shown in italics.	<code>%VarName%</code>
Keywords or values that are optional are enclosed in [ ]. Do not type the [ ] symbols in your own code.	<code>%loop LoopVar [null=NULLValue]%</code>

## Related Documentation

The following documents are companions to this guide.

Refer to this book...	For...
<i>SAP BC Developer Guide</i>	Information about creating and testing services and client applications. This book for is for solution developers.  You will find this book at:  <sapbc>\developer\doc\SAPBCDeveloperGuide.pdf
<i>SAP BC Administration Guide</i>	Information about using the Server Administrator to configure, monitor, and control the SAP BC Server. This book is for server administrators.  You will find this book at:  <sapbc>\server\doc\SAPBCAdministrationGuide.pdf
<i>SAP BC Built-In Services Guide</i>	Descriptions of services that are installed on your SAP BC Server. This book for is for solution developers.  You will find this book at:  <sapbc>\developer\doc\SAPBCBuiltInServicesGuide.pdf
<i>Building Output Templates and DSPs</i>	Information about creating output templates and Dynamic Server Pages (DSPs). This reference is for solution developers.  You will find this book at:  <sapbc>\developer\doc\SAPBCTemplatesAndDSPs.pdf
<i>SAP BC Developer Tutorial</i>	Information that orients you to SAP BC server and shows you how to create a simple application. It includes basic conceptual information about SAP BC server. This book is for users of SAP BC server.  You will find this book at:  <sapbc>\developer\doc\SAPBCDevTutorial.pdf



## Viewing this Document

To view this document, which is in PDF format, you must have Acrobat Reader™ 4.0 or later installed on your system. If you have an earlier version of Acrobat Reader, you will receive the following error message when you open this document and Acrobat Reader will not display the images in this document:

```
Could not find the ColorSpace named 'Cs8.'
```

If you do not have this software or you do not have the correct version, you can download a free copy from:

<http://www.adobe.com/downloads>.

## Printing this Guide

To produce a hard copy of this guide, print this document from Acrobat Reader.



# 2

## Locking and Unlocking Elements

■ Basic Concepts . . . . .	10
■ Locking Elements . . . . .	11
■ Unlocking Elements . . . . .	15
■ Copying, Moving, and Deleting Locked Elements . . . . .	22

## Basic Concepts

In SAP BC server, you can manage changes to elements in development by locking them. This prevents two different users from editing an element at the same time. You can lock elements such as flow services, Java services, schemas, and specifications.





All elements in the Developer's Service Browser are read-only until you lock them. In other words, you cannot edit an element until you have locked it. However, you can use and run a service regardless of its lock status.

### What Is a Lock?

A lock on an element prevents another user from editing that element. There are two types of locks: *user locks* and *system locks*. When an element is locked by you, you have a *user lock*. The element is read-only to all other users on the SAP BC Server. Another user cannot edit the element until you unlock it.

When an element's server-side files are marked read-only on the SAP BC Server, the element is *system locked*. A system lock denotes that the element's corresponding server files (node.xml, for example) are read-only on the SAP BC Server. For example, the server administrator may have checked those files into a third-party source control application, which marked those files on the server as read-only. You cannot edit the element until the server administrator makes the element's server-side files writable, and you reload the package in which the element resides.

Elements are shown in the following ways in Developer:

Element	Status	Can I edit?	How do I gain rights to edit?
 ftp	Not locked	No	Click Edit ► Lock.
 ftp	Locked by you	Yes	N/A
 ftp	Locked by another user	No	Contact the user to unlock.
 ftp	Locked by the system	No	Contact the server administrator to unlock.

## How Do I Know Who Has an Element Locked?

On every element in the Service Browser, you can view the lock status by using the **Lock Status** command. This command provides information about the element such as the username of the person who owns the lock and when they locked it. If an element is system locked, you can use the **View Server Files** command to obtain the names of the server files that are read-only on the server. For details, see “Viewing an Element’s Corresponding Server Files” on page 20.

## When Do I Lock an Element?

You lock an element when you want to make changes to the element. For details, see “Locking an Element” on page 11.

## When Do I Unlock an Element?

You unlock an element after making your changes and saving those changes to the server. It is important to unlock the elements you are done with so that other users on the server can access them. For details, see “Unlocking Elements” on page 15.

If you want to automatically unlock an element after saving it, you can enable a setting on the **Preferences** dialog box. For details, see “Automatically Unlocking Elements After Saving” on page 22.

# Locking Elements

Before you edit an element, you must lock it. This ensures that you are the only person working on a particular element at a time, preventing the loss of changes. Elements can only be locked by one user at a time. If the element you need is already locked, request that the current owner of the lock release it. If the element is system locked, request that the Server Administrator release it by making the corresponding server files writable.

## Locking an Element

Elements are locked by SAP BC server username (the name you use to log in to the SAP BC Server). Because of this, it is important that you use a distinct username to log in to the server. If you change usernames, you will be unable to edit or unlock items that you locked using your old username.

When you create a new element, it is locked automatically for you. Elements generated by a service (including an Adapter service) will not be locked automatically. For example, if your service generates a record from a document, the record will not be locked. You must manually lock the record after you generate it if you want to edit it.



**Note:** When you lock an element, you lock the latest version of the element on the SAP BC Server. That is, upon locking an element, your Service Browser may refresh to show the latest version of that element that is saved to the server.

---

#### To lock an element

- 1 In the Service Browser, select the element that you want to lock. (You cannot multiselect elements in the Service Browser.)
- 2 Right-click the element and select **Lock**.

—OR—

Press CTRL + L.

—OR—

On the **Edit** menu, select **Lock**. If the element was successfully locked, a green check mark appears on the icon in the Service Browser.

---



#### **Note:**

—When you lock a Java/C/Webtap service, all Java/C/Webtap services in that folder are locked. For details, see “Locking Java, C/C++, and WebTap Services” on page 13.

—You cannot lock folders or packages; however, you can lock the contents of them. For details, see the following section.

---

## Locking All Elements In a Folder or Package

In a folder or package, you can automatically lock each element that is available to be locked. Some elements may not be available to be locked because they may be locked by another user or system locked.

Keep in mind that when you lock the contents of a folder or package, you only lock the existing contents. Any user can still create new elements in that folder or package, unless it is a Java, C/C++, or WebTap service. To create those types of services, you must have the existing Java, C/C++, and WebTap services unlocked in the folder in which you want to create them. For details, see “Locking Java, C/C++, and WebTap Services” on page 13.

#### To lock the contents of a folder or package

- 1 In the Service Browser, select the folder or package whose contents you want to lock.
- 2 Right-click the folder or package and select **Lock Contents**.

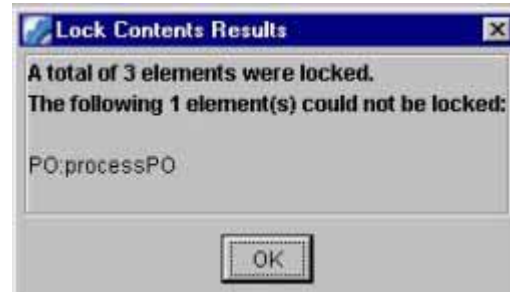
—OR—

Click CTRL+H.

—OR—

On the **Edit** menu, select **Lock Contents**. The following dialog box appears. Elements that are system locked or locked by another user are not locked.

Lock Contents Results Dialog Box



## Locking Java, C/C++, and WebTap Services

When you lock Java, C/C++, and WebTap services, there are special considerations to keep in mind.

- Locking and unlocking operations on Java, C/C++, and WebTap services are folder-wide. All Java, C/C++, and WebTap services in a folder share the same .java and .class files on the SAP BC Server. These files, located in the \code subdirectory of a package, correspond to all services (except flow services) in a folder. Therefore, when you lock a Java/C/Webtap service, all Java/C/Webtap services in that folder are locked.

For example, if you lock a Java service in a folder A, all Java, C/C++, and WebTap services in folder A are locked by you. Similarly, if another user has locked a Java service in folder B, you cannot add, edit, move, or delete any Java, C/C++, or WebTap services in folder B.

- The jcode development environment operates independently of locking. If you use jcode to develop Java services, you do not have the locking functionality that is available in the SAP BC server. When you use jcode, you may compile a service that is locked by another user, overwriting that user's changes to the service. Therefore, if you use jcode, do not use the locking features in the SAP BC server.
- Before you save a Java, C/C++, or WebTap service, multiple corresponding files must be writable on the server. A single Java, C/C++, or WebTap service corresponds to the following files:

```
.java
.class
.ndf
.frag (may not be present)
```

Before you save a Java, C/C++, or WebTap service, all of the preceding files must be writable. Therefore, make sure that all system locks are removed from those files before saving. If you use a third-party source control application, make sure that those files are open for editing in the source control application.

## Locking Templates

A SAP BC server template can be used with one or more services on the SAP BC Server. Currently, you cannot lock a template as an entity, only the service to which it is attached. Following are considerations for working with templates in a cooperative development environment.

- To create or edit a template for a service, you must have the service locked.
- The template for a service can change without your knowledge. Since a template can be attached to one or more services, keep in mind that a shared template can change without your knowledge. For example, if your template is attached to a service that another user locks and edits, that user can change your template.

## System Locking Elements

If you are a server administrator, you can system lock an element by making its corresponding server files read-only. If you do not know the names of the files that correspond to a particular element, use the **View Server Files** command. For details, see “Viewing an Element’s Corresponding Server Files” on page 20.

Typically, elements become system locked when you check them into a third-party source control application. For details, see the Using Source Control with the Integration Server chapter.

Usually, a system lock is not reflected in SAP BC Developer or the Server Administrator until you reload the package in which the element resides.



**Important!** Before you system lock an element, always verify that it is not locked by a user on the SAP BC Server. If an element becomes system locked while a user is editing it, the user will not know until he or she tries to save changes to the element. If this occurs, make the element’s corresponding files writable on the server. After this is done, the user can save his or her changes to the element.

---



## Viewing the Status of a Locked Element

The lock status of an element tells you if an element is available for locking, and if not, who owns the lock and when they locked it.

To view lock status

- 1 In the Service Browser, select the element for which you want to view the status.
- 2 Right-click the element and select **Lock Status**. The following dialog box appears if the element is locked by someone else. A similar dialog appears if the element is system locked or locked by you.

Locking Status Dialog Box



### Note:

- If “localhost” is shown for the Host, it means that it is the machine on which the server is currently running.
- If the element has been system locked since you last reloaded the package, the system lock status will **not** be shown in the Locking Status dialog box. You must reload the package to view the system-locked status.
- When a user unlocks an element, you must refresh the Service Browser to reflect the unlocked status. Similarly, when the server administrator removes a system lock from an element, you must reload the package in which the element resides to reflect the updated status.

## Unlocking Elements

After you edit an element and save changes to the server, you should unlock it to make it available to other users. There are several ways to unlock an element, depending on whether you are a member of the Developer ACL or the Administrator ACL. If you are a developer, you can unlock an element in Developer. If you are an administrator, you can unlock an element in the Server Administrator as well as in Developer.

## Unlocking an Element Using Developer

Elements must be unlocked explicitly. Disconnecting from the server does not unlock your element(s), since your locks are maintained from session to session.

Unlocking a Java/C/Webtap service causes all Java/C/Webtap services in that folder to be unlocked. For details, see “Locking Java, C/C++, and WebTap Services” on page 13.



---

**Important!** If any Java/C/Webtap service in a folder has unsaved changes, you will not be able to unlock other Java/C/Webtap services within that folder. Save the changes, and then unlock the services.

---

**To unlock an element using Developer**

- 1 In the Service Browser, select the element that you want to unlock. Make sure that the element has been saved to the server before you unlock it. If you attempt to unlock an element that has not been saved, you are prompted to save or discard changes.



---

**Note:** You cannot multiselect in the Service Browser. To unlock more than one element at a time, see “Unlocking More Than One Element” on page 16.

---

- 2 Right-click the element and select **Unlock**.

—OR—

Click CTRL +U.

—OR—

On the **Edit** menu, select **Unlock**. The Service Browser refreshes and the green checkmark next to the element disappears.

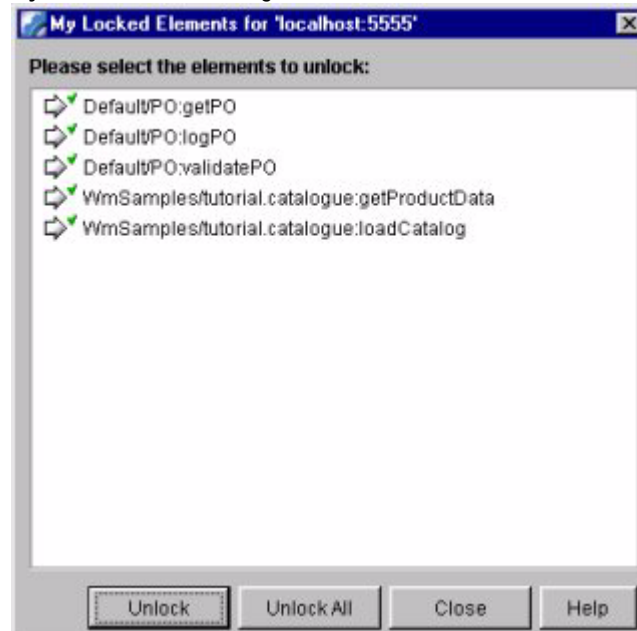
## Unlocking More Than One Element

You can unlock a set of elements or all elements at once by using the **My Locked Elements** command. See the following procedure.

**To unlock more than one element**

- 1 On the View menu, select My Locked Elements. The My Locked Elements dialog box appears.

My Locked Elements Dialog Box



- 2 Use CTRL+click to select the elements to unlock and then click **Unlock**. If you want to unlock all elements, click **Unlock All**.



**Important!** If any Java/C/Webtap service in a folder has unsaved changes, you will not be able to unlock other Java/C/Webtap services within that folder. Save the changes, and then unlock the services.



**Note:**

— If you have a WIDL locked, you will see multiple elements in the My Locked Elements dialog box. This is because a WIDL can correspond to record references and other elements. To unlock the WIDL, make sure that you unlock it and the related elements shown in the My Locked Elements dialog box.

— Unlocking a single Java/C/Webtap service causes all Java/C/Webtap services in that folder to be unlocked. For details, see “Locking Java, C/C++, and WebTap Services” on page 13.

## Unlocking an Element Using the Server Administrator



**Important!** Exercise caution when you remove user locks, since a user can lose changes. If you unlock an element while a user is editing it, the user will not know until he or she tries to save changes to the element and the operation fails. Always confirm with the user before removing his or her lock on an element.

To unlock an element using the Server Administrator

- 1 In the Server Administrator, under Packages, click Management.
- 2 Click View Locked Elements. The following screen appears, showing all elements that have user locks and system locks.

Locked Elements Screen

Packages > Management > Locked Elements

- [Return to Package Management](#)
- [Unlock Elements](#)
- [Sync to Name Space](#)

"localhost" means the machine on which the server is running

Locked by System			
Locked Element	Host	Date Locked	
No Elements locked by System			

Locked by Current User (Administrator)			
Locked Element	Host	Date Locked	
<a href="#">pub.sap.idoc.routing:inbound</a>	localhost		
<a href="#">pub.sap.idoc.routing:outbound</a>	localhost		

Locked by Other Users			
Locked Element	User	Host	Date Locked
No Elements locked by other users			

- 3 Click **Unlock Elements**. The following screen appears.

Unlock Selected Elements Screen

Packages > Management > Unlock Elements

- Return to Locked Elements

Locked by System				
Locked Element	Host	Date Locked		
No Elements locked by System				

Locked by Current User (Administrator)				
Unlock	Locked Element	Host	Date Locked	
<input checked="" type="checkbox"/>	pub.sap.idoc.routing:inbound	localhost		
<input type="checkbox"/>	pub.sap.idoc.routing:outbound	localhost		

Locked by Other Users				
Unlock	Locked Element	User	Host	Date Locked
No Elements locked by other users				

Unlock Selected Elements

- **Locked by System.** Lists elements whose corresponding files are marked read-only on the server file system. You cannot remove a system lock via the Server Administrator. On the server's file system, you must make the element's files writable and reload the package. For details, see "Unlocking a System Locked Element" on page 20.
  - **Locked by Current User.** Lists elements that are locked by you, the server administrator (or the username with which you logged into the Server Administrator). Before you unlock an item, make sure that you have saved all changes to the server.
  - **Locked by Other Users.** Lists elements that are locked by other users on the server. Before you remove a user's lock, make sure that the user has saved all changes to the server. If not, the user will lose all changes that they made to the element since they last saved it to the server.
- 4 Select the elements that you want to unlock (after informing users if necessary) and click **Unlock Selected Elements**.

Developers using SAP BC Developer should refresh their Service Browser to update their view of the lock status of all elements.



**Important!** If you receive a “failed to unlock” message, it means that the server files for a locked element were deleted from the server. Use the **Sync to Name Space** command to update the Server Administrator’s view of locked elements. For details, see the Locking Administration and Best Practices chapter.

---

## Unlocking a System Locked Element

If you are a server administrator, you can remove a system lock from an element using the server’s file system. After you remove the system lock, you must reload the package in which the element resides to reflect the element’s updated status. If you use Developer, you must refresh the Service Browser (after the package is reloaded) to reflect the element’s updated status.

If you use a third-party source control application... Removing a system lock from an element may cause a conflict, since many source control applications make files read-only after they have been checked in to an archive. Verify that the file properties are appropriate in the source control application before you remove a system lock. For details on using third-party source control applications with SAP BC server, see the Using Source Control with the Integration Server chapter.

### To remove a system lock from an element

- 1 If you do not know the names of the server files that correspond to the element, use the **View Server Files** command in Developer. For details, see “Viewing an Element’s Corresponding Server Files” on page 20.
- 2 On the server’s file system, remove the read-only properties from the files that correspond to the element to make the files writable.
- 3 Reload the package on the SAP BC Server that contains the element. The updated status is reflected in the Server Administrator. Refresh the Service Browser in Developer to view the updated status.



**Important!** If you accidentally system-locked an element while a user had a lock on it, remove the system lock but **DO NOT** have the user reload the package in SAP BC Developer. (Reloading the package in Developer will discard their edits.) The user can then save the element without losing the changes he or she made to it while you had the element system-locked.

---

## Viewing an Element’s Corresponding Server Files

You can view the names of the server files associated with every SAP BC server element. This is convenient when an element is system locked and you need to convey the element’s filenames to the server administrator. See the following procedure.

To view server files for an element

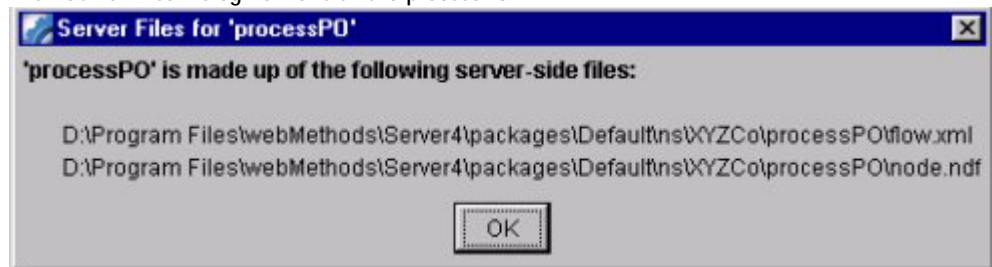
- 1 In the Service Browser, select the element for which you want to view the server filenames.
- 2 Right-click the element and select View Server Files.

—OR—

On the **Edit** menu, select **View Server Files**.

The following dialog box shows the server files associated with a flow service named processPO.

View Server Files Dialog Box for element 'processPO'



The following dialog box shows the server files associated with a system locked flow service named processPO. These files are not writable on the server.

View Server Files Dialog Box for element 'processPO'



**Note:** After a server administrator removes a system lock from an element, you must reload the package in which the element resides to reflect the unlocked status.

## Automatically Unlocking Elements After Saving

You can choose to automatically unlock flow services, records, and specifications after you save changes to them. This prevents you from forgetting to unlock them; however, it may not be the best option if you save periodically while editing an element.

To automatically unlock flow services, records, and specifications after saving

- 1 On the **Edit** menu, select **Preferences**. The Preferences dialog box appears.
- 2 On the **General** tab, check the **Automatically unlock upon save (not applicable for Java/C/WebTap services)** checkbox.

Preferences Dialog Box



- 3 Click **OK**.

## Copying, Moving, and Deleting Locked Elements

You can copy a locked element to another folder or package. However, you cannot move, rename, or delete an element unless it is locked by you or unlocked.



### Special Considerations for Folders and Packages

If a folder or package contains...	Then...
Only elements that are locked by you	You can delete the folder or package.
An element that is locked by another user	You cannot delete the folder or delete the package.
An element that is system-locked	You cannot delete the folder or delete the package.



## Using Source Control with the SAP BC Server

■ Basic Concepts . . . . .	26
■ Methods of Using Source Control . . . . .	26
■ Components of a SAP BC server Solution . . . . .	27
■ Adding Components to Source Control . . . . .	34
■ Checking Components Out of Source Control . . . . .	36
■ Updating Files That Are in Source Control . . . . .	37

## Basic Concepts

Often, the development environment includes a third-party source control application that co-exists with SAP BC server. You can use source control with SAP BC server by identifying the components that comprise a SAP BC server solution, adding them to source control, and managing the process by which users check elements in and out of source control. Following are some basic concepts to help you understand the procedures in this chapter.

### What Is a Third-Party Source Control Application?

A third-party source control application is a product designed to help an administrator archive versions of files in a development environment. Such products include PVCS, CVS, Visual Source Safe, Subversion and Perforce.

### Is Source Control the Same as Locking?

No. Locking is one component of a full source management solution. Source control is designed to allow the administrator to archive and maintain multiple file revisions. Many source control applications also allow the administrator to tag revisions, run revision history reports, and other functions. Lock functionality allows the user or administrator to control who is working on a particular element at a particular time.

## Methods of Using Source Control

There are a number of ways to use source control with the SAP BC Server. Two of the most prevalent ways are:

- **Package level.** You only put the package and global components in source control. For example, the only file in source control might be XYZCo.zip. Often, this entails assigning a single package per developer. To update the package, the developer:
  - 1 Checks the package out of source control.
  - 2 Locks all package elements in Developer.
  - 3 Modifies the elements and saves changes.
  - 4 Replicates the package to generate an updated .zip, such as XYZCo.zip.
  - 5 Checks the .zip back into source control.
- **Component level.** You put the package's components and global components in source control. For example, the files in source control might include node.ndf, getPO.java, POfolder.class, etc. To update an element, such as a flow service, the developer:
  - 1 Checks the flow service files out of source control (node.ndf and flow.xml).
  - 2 Locks the flow service in Developer.

- 3 Modifies the flow service and saves changes.
- 4 Checks the node.ndf and flow.xml files back into source control. If the source control application makes the files read-only when checked in, then the corresponding flow service will appear system locked in Developer.

The sections that follow address both methods of using source control, with emphasis on the component level, as it is the most detailed.

## Components of a SAP BC server Solution

A solution that you develop for the SAP BC server consists of a set of services and supporting components that reside in one or more packages on the SAP BC Server. In addition to components that reside in a package, a solution might also encompass certain global components—configuration files, access control lists, log files, and so forth—whose settings establish the run-time environment necessary for the package components to operate successfully.

### Package Components

On a SAP BC Server, a package is a container for a set of services and their supporting components (e.g., record, specifications, output templates, Dynamic Server Pages (DSPs), and so forth). It is an organizational device that you can use to group related components together and manage them as a single unit. Typically, you place all the components for a single solution in one package—for example, if you develop a set of services that interact with a human resources system, you might place all of its components in a package called HR.



**Note:** You are not required to put all components for a solution in the same package. For example, you might choose to organize your components by function—putting data-retrieval components in one package, data-transformation components in another package, utility components in another package, and so forth. You can determine what type of package arrangement makes the most sense for your particular solution.

### Location of a Package

A package is represented by a single subdirectory on the SAP BC Server. This subdirectory has the same name as the package it represents and resides in the following location on the server:

```
<sapbc>\server\packages\packageName
```

All services and related components that belong to a package, reside in that package's subdirectory.

## Contents of a Package

The component table shown below describes the contents of a package. Note that each component is labeled *STATIC*, *DYNAMIC*, or *DERIVED*, indicating whether it is a good candidate for source control, as follows:

<b>Component Type</b>	<b>Description</b>
<b>STATIC</b>	<p>Components that are not modified by Developer or the Server Administrator. These include components such as HTML pages, DSPs, and programming source code that you build using tools other than those provided by SAP.</p> <p>STATIC components <i>are</i> candidates for source control.</p>
<b>DYNAMIC</b>	<p>Components that are created and/or modified by the Developer and/or the Server Administrator. These include flow services, records, output templates, access control files, and so forth.</p> <p>DYNAMIC components <i>are</i> candidates for source control.</p>
<b>DERIVED</b>	<p>Temporary components that the SAP BC Server generates and consumes internally. These include the class files created when a Java service is compiled with the Developer IDE or the jcode utility.</p> <p>DERIVED components <i>are not</i> good candidates for source control.</p>

Components in a SAP BC server Package Directory

Component	Type	Description	Affected by
manifest.v3	DYNAMIC	Contains information about the package, including its status, version information, dependencies on other packages, and the names of its startup and shutdown services.	Operations you perform on the package with the Server Administrator.
pub\	STATIC	Contains HTML pages, DSPs, and related image files that are served to HTTP clients. (This is the WWW root directory for the package.)	Tools you use to create and edit HTML documents and/or DSPs. The contents of this directory is never updated by SAP BC server.
widl\	STATIC	Contains web-automation services (WIDLs) produced by earlier versions of SAP BC server.	Tools you use to create and edit WIDL files. The contents of this directory are never updated by the Developer or the Server Administrator.

Component	Type	Description	Affected by
templates\	DYNAMIC	Contains output templates that have been created for the services in this package.	Editing operations that you perform on the template files using the Server Administrator, the Developer, and/or your own text-editing tools.
code\	--	Contains subdirectories that hold the source and executable code for coded services (i.e., Java, C/C++, and COM services).	--
code\jars	STATIC	Contains external Java libraries used by the services in the package.	Tools you use to install or produce the external libraries. The content of this directory is never updated by SAP BC server.
code\libs	See Description	<p>Contains all executable C/C++ and/or COM program files that belong to the package. These files can be:</p> <ul style="list-style-type: none"> <li>■ Third-party C/C++ or COM libraries used by services in this package. Files of this type are STATIC.</li> </ul> <p>–OR–</p> <ul style="list-style-type: none"> <li>■ The libraries and/or executable files for services that you have developed using C/C++ and/or COM. Files of this type are DERIVED.</li> </ul>	<p>STATIC files are updated by the tools you use to install or produce them.</p> <p>DERIVED files are updated by the tools that you use to build the libraries and/or program files from their underlying source code.</p>
code\classes	DERIVED	Contains the Java class files for services that belong to the package.	Saving or compiling a Java service with Developer or running the jcode make command.

Component	Type	Description	Affected by
code\source	See Description	<p>Contains the source files for coded services (Java services, WebTap services, C/C++ services, and COM services) that belong to the package.</p> <p>If the package contains Java services, this directory will have one source file for each folder that contains Java services. This file holds the source code (as a single class) for all of the Java services (coded as methods of the class) in the folder.</p> <ul style="list-style-type: none"> <li>■ If you maintain your Java services with Developer, this source file is DERIVED.</li> <li>■ If you maintain Java services with your own IDE, this source file is STATIC.</li> </ul> <p>For each C/C++ service in the package, this directory will contain the C/C++ source code and supporting build files. These files are STATIC.</p>	<p>STATIC files are updated by the tools you use to create and/or modify them.</p> <p>DERIVED files are updated when you save or compile a Java service with the Developer.</p>
ns\	--	<p>Contains a subdirectory for each:</p> <ul style="list-style-type: none"> <li>■ Folder</li> <li>■ Service</li> <li>■ Record</li> <li>■ Specification</li> </ul> <p>that resides in the package.</p> <p>Each subdirectory has the same name as the component it represents (e.g., a folder named ACK is contained in a subdirectory named ACK; a service named ACK:shipNotice is contained in the subdirectory ACK\shipNotice).</p> <p>For additional information about the contents of the \ns directory, see “Checking In the ns Directory” on page 36.</p>	--



Component	Type	Description	Affected by						
ns\FolderName	See Description	<p>Contains the file node.idf, which holds Java code that is shared among the Java services residing in the folder.</p> <ul style="list-style-type: none"> <li>■ If you maintain Java services using Developer, this file is DYNAMIC.</li> <li>■ If you maintain Java services using your own IDE, this file is DERIVED.</li> </ul> <p>This directory also contains a subdirectory for each service, record, specification, and/or folder that resides within it.</p>	<p>This file is updated when you modify the folder's shared Java code with the Developer.</p> <p>This file is rewritten when you run the jcode frag command on the folder's Java source file.</p>						
ns\FlowServiceName	DYNAMIC	<p>Contains the following files:</p> <table border="1" data-bbox="614 780 1061 1060"> <thead> <tr> <th data-bbox="614 780 763 832">File Name</th> <th data-bbox="763 780 1061 832">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="614 832 763 981">node.ndf</td> <td data-bbox="763 832 1061 981">Contains the service's input/output parameters and run-time settings.</td> </tr> <tr> <td data-bbox="614 981 763 1060">flow.xml</td> <td data-bbox="763 981 1061 1060">Contains the service's logic.</td> </tr> </tbody> </table>	File Name	Description	node.ndf	Contains the service's input/output parameters and run-time settings.	flow.xml	Contains the service's logic.	<p>Operations that you perform on the flow service with Developer.</p> <p>Operations that you perform on the service's run-time settings using the Server Administrator or Developer.</p>
File Name	Description								
node.ndf	Contains the service's input/output parameters and run-time settings.								
flow.xml	Contains the service's logic.								

Component	Type	Description	Affected by						
ns\CodedServiceName	See Description	<p>Contains the following files for coded services.</p> <table border="1"> <thead> <tr> <th>File Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>node.ndf</td> <td> <p>Contains the service's input/output parameters and run-time settings.</p> <p>This file is DYNAMIC.</p> </td> </tr> <tr> <td>java.frag</td> <td> <p>Contains the logic for the service.</p> <ul style="list-style-type: none"> <li>■ If you maintain Java services with Developer, this file is DYNAMIC.</li> <li>■ If you maintain Java services with your own IDE, this file is DERIVED.</li> </ul> </td> </tr> </tbody> </table>	File Name	Description	node.ndf	<p>Contains the service's input/output parameters and run-time settings.</p> <p>This file is DYNAMIC.</p>	java.frag	<p>Contains the logic for the service.</p> <ul style="list-style-type: none"> <li>■ If you maintain Java services with Developer, this file is DYNAMIC.</li> <li>■ If you maintain Java services with your own IDE, this file is DERIVED.</li> </ul>	<p>The node.ndf file is updated when you edit the service's run-time settings using the Server Administrator or Developer.</p> <p>The java.frag file is updated when you modify the source code for the Java service with Developer.</p> <p>The java.frag file is replaced when you run the jcode frag command on the Java source file for this folder.</p>
File Name	Description								
node.ndf	<p>Contains the service's input/output parameters and run-time settings.</p> <p>This file is DYNAMIC.</p>								
java.frag	<p>Contains the logic for the service.</p> <ul style="list-style-type: none"> <li>■ If you maintain Java services with Developer, this file is DYNAMIC.</li> <li>■ If you maintain Java services with your own IDE, this file is DERIVED.</li> </ul>								
ns\RecordName	DYNAMIC	Contains the file node.ndf, which holds the record.	Operations that you perform on the record with the Developer.						

Component	Type	Description	Affected by								
<i>ns\SpecName</i>	DYNAMIC	Contains the file <i>node.ndf</i> , which holds the specification.	Operations that you perform on the specification with the Developer.								
<i>ns\WebTapServiceName</i>	DYNAMIC	Contains the following files: <table border="1" data-bbox="612 452 1061 904"> <thead> <tr> <th>File Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><i>node.ndf</i></td> <td>Contains the service's input/output parameters and runtime settings.</td> </tr> <tr> <td><i>java.frag</i></td> <td>Contains the java code for the WebTap service.</td> </tr> <tr> <td><i>webtap.rul</i></td> <td>Contains the service's WebTap rules.</td> </tr> </tbody> </table>	File Name	Description	<i>node.ndf</i>	Contains the service's input/output parameters and runtime settings.	<i>java.frag</i>	Contains the java code for the WebTap service.	<i>webtap.rul</i>	Contains the service's WebTap rules.	Operations that you perform on the WebTap service with the Developer.
File Name	Description										
<i>node.ndf</i>	Contains the service's input/output parameters and runtime settings.										
<i>java.frag</i>	Contains the java code for the WebTap service.										
<i>webtap.rul</i>	Contains the service's WebTap rules.										

## Global Components

In addition to package components, a solution that you build for SAP BC server also relies on certain global components—configuration files, access control lists, log files, and so forth—to operate successfully.

The following table contains global components that can affect the operation of the services in a package. It also identifies each component's type (STATIC, DYNAMIC, or DERIVED) with respect to source control. (For a definition of these terms, see page 28.)

Components Containing Global Parameters

Component	Type	Description	Updated by
<code>&lt;sapbc&gt;\server\bin server.bat</code> (Windows systems) —OR— <code>&lt;sapbc&gt;/server/bin /server.sh</code> (UNIX systems)	STATIC	Contains the script that starts the server and establishes certain operating parameters under which the server runs.  This file includes the server's CLASSPATH, which you must update when you want to make an external library (for example, a database driver) available to the server.	Editing operations that you perform on the file using your own text-editing tool. The content of this file is never modified by SAP BC server.
<code>&lt;sapbc&gt;\server\config\*.*</code>	DYNAMIC	Contains a set of configuration files—including access control lists, user account information, and server operating parameters—that define the server's operating environment.	Various operations (e.g., creating user accounts, editing access control lists, setting server properties, and so forth) that you perform with the Server Administrator.

## Adding Components to Source Control

To put a solution that you have developed in source control, you check in all of its components that are STATIC or DYNAMIC (do not check in DERIVED files) as outlined in the following sections. (To determine whether a component is STATIC or DYNAMIC, refer to the component lists on page 28 and page 33.)



**Tip!** An alternative to checking in components is to use the package replication and publishing feature and check in a single *packagename.zip* to source control. See “Methods of Using Source Control” on page 26.

## Choosing a Development Methodology for Java Services

If your solution includes Java services, you need to decide which development methodology your organization will use to maintain those services from this point forward. This decision will determine (1) which set of files you put in source control, and (2) the steps you must take after checking out a folder containing Java services to regenerate its DERIVED components.

- If you want to use Developer to maintain the Java services in a folder, your “source files” are the individual java.frag files that belong to the folder. These are the files you will place in source control. After you check out these files, you must run the “jcode comp” and “jcode make” commands to generate (i.e., derive) a new Java source file (in the code\source directory) from the java.frag files in the folder. These commands will also incorporate the shared code from the folder’s .idf file into the Java source file.
- If you want to use another IDE to maintain the Java services in a folder, your “source file” is the single Java file that resides in the *packageName*\code\source directory and contains the source code for all Java services in that folder. This is the file that you will put in source control. When you check out this file and edit it with the IDE, you must run the “jcode frag” and “jcode make” commands to generate (i.e., derive) new java.frag files in the namespace and update the shared code in the folder’s .idf file.



**Note:** The jcode utility is independent from locking in SAP BC Developer. For example, you can use jcode to generate a new java.frag file without having the service locked (although recommended).

## Checking Components Into Source Control

Check in the following components from each package that is a member of the solution. (Note that your package might not have all the components in the following list. For example, if your package does not contain widl files, it will not have a *packageName*\widls subdirectory.)

From <sapbc>\server\packages\...	Check In...
<i>packageName</i> \	manifest.v3
<i>packageName</i> \pub\	All files
<i>packageName</i> \widls\	All files
<i>packageName</i> \templates\	All files
<i>packageName</i> \code\jars\	All files
<i>packageName</i> \code\libs\	Third-party libraries only. (Do not check in the executable program files for C/C++ or COM services.)
<i>packageName</i> \code\source\	C/C++ and COM source code and build files. If you use your own IDE to maintain Java services (not Developer), check in the Java source file for each folder.
<i>packageName</i> \ns	See “Checking In the ns Directory” below.

In addition to package components, check in the following global components if your solution requires them.

- `<sapbc>\server\bin\server.bat` (Windows)
- `<sapbc>\server\bin\server.sh` (UNIX)
- `<sapbc>\server\config\*.cnf`

## Checking In the ns Directory

If your solution does not contain any coded services (Java, C/C++, and/or WebTap services) check in all files in `PackageName\ns`. However, if your solution includes Java services (or Java wrappers for C/C++ services), you need to know which development methodology your organization uses to maintain Java services in order to check the proper set of files from the ns subdirectory. (For additional information about development methodology for Java services, see “Choosing a Development Methodology for Java Services” on page 34.)

- If your organization maintains Java services using Developer, check in all files in `packageName\ns`.
- If your organization maintains Java services using another IDE (and then registers the completed code on the server with the jcode utility), check in the following files:

From `<sapbc>\server\packages\...`  
`packageName\source`  
`packageName\ns`

Check In...

All \*.java files.

All files except the following:

- node.idf in those folders that contain Java services
- java.frag files and the node.ndf files located in the same directory as the corresponding java.frag.

## Checking Components Out of Source Control

To check a solution out of source control, perform the following general steps:

- 1 Check out any of the following global components that it uses:
  - `<sapbc>\server\bin\server.bat` (Windows)
  - `<sapbc>\server\bin\server.sh` (UNIX)
  - `<sapbc>\server\config\*.cnf`
- 2 Check out the package(s) or package components with which you want to work.

- 3 If the package contains Java, C/C++ and/or WebTap services, do one of the following:
  - If you maintain Java services with Developer, run the following jcode commands to generate a new source file from the java.frag files in *packageName* \ns:
 

```
jcode comp packageName folderName.subfolderName
jcode make packageName folderName.subfolderName
```
  - If you maintain Java services using another IDE, run the following jcode commands to generate new java.frag files from the Java source files in *packageName* \code \source:
 

```
jcode frag packageName folderName.subfolderName
jcode make packageName folderName.subfolderName
```
- 4 If your package contains C/C++ services, rebuild those services (e.g., recompile the program to generate the DLL).

## Updating Files That Are in Source Control

When you update a SAP BC server solution that you have checked out from source control, keep the following points in mind:

- Make sure to check in any STATIC components (e.g., HTML pages, DSPs, C/C++ source code, and so forth) that you modified with your own tools. (This includes Java sources files that you updated with your own IDE.)
- Make sure to check in any DYNAMIC components (flow services, records, output templates, and so forth) that you modified with the Server Administrator or Developer.
- The following table shows common editing tasks that you might perform, and identifies the files they affect.

If you change...	You change the contents of...
Package settings, including: <ul style="list-style-type: none"> <li>■ Version ID</li> <li>■ Dependencies</li> <li>■ Startup/shutdown services</li> </ul>	manifest.v3
Run-time settings for a service (e.g., caching, state, prefetch)	The node.ndf file for that service
Input/output parameters for a service	The node.ndf file for that service
Logic for a flow service	The flow.xml file for that service
Output template for a service	<i>packageName</i> \templates \ <i>templateName</i>

- If you update a Java service with Developer, make sure to check in the components that you updated. The following table shows common editing tasks that you might perform, and identifies the files they affect.

If you change...	You change the contents of...
Logic of a Java service with Developer	The java.frag and .java files for that service
Shared code in a Java service with Developer	The node.idf file for that folder
Rules for a WebTap service	The webtap.rul file for that service



## Troubleshooting

■ Introduction . . . . .	40
■ Lock/Unlock Problems . . . . .	40
■ Package Management Problems . . . . .	41
■ Save Problems . . . . .	41
■ Other Problems . . . . .	41
■ Frequently Asked Questions . . . . .	41

## Introduction

This chapter addresses common problems that may arise when implementing cooperative development in the SAP BC server.

## Lock/Unlock Problems

**I can't lock an element.**

Check the version of the SAP BC Server that you are connected to. It must be Version 4.6 or later for locking functionality to be enabled.

**When I try to lock an element, I get an exception message.**

The element is system-locked by the administrator (marked read-only on the server). Reload the package in the Service Browser to show the system lock status.

**I can't unlock a Java, C, or WebTap service.**

If there is another Java, C, or WebTap service that is locked by another user or system locked in the same folder, then you cannot unlock any Java, C, or WebTap services in that folder. This is because those services share the same .java and .class files on the SAP BC Server.

**I can't unlock elements since I changed my username.**

You can only unlock elements that you have locked with your current username for the session. If you have changed usernames, log back in to the server with your old username, then unlock elements.

If the administrator has deleted your username, contact him or her to unlock the elements on the server. You can assist the administrator by using the **View Server Files** command to identify the names of the files on the server that need to be unlocked.

**Another user unlocked an element, but it still shows as locked in my Service Browser.**

If it is a Java, C, or WebTap service, reload the package in the Service Browser. If it is any other element, use the **Refresh** command to refresh the Service Browser.

**I receive an "element failed to unlock" message when I try to unlock elements in the Server Administrator.**

This indicates that the server files for the locked element were deleted from the server. You need to update the Server Administrator's list of unlocked elements by clicking **Sync to Name Space** on the **Unlock Selected Elements** screen. Note that the **Sync to Name Space** command is automatically run when the server is started or restarted.

---

## Package Management Problems

I can't preserve locking information when I replicate and publish a package.

This is expected behavior and is part of the feature's design. You can, however, preserve system locks (read-only file attributes).

When I disable a package, it does not preserve locking information.

This is expected behavior and prevents conflicts if another package with the same folder and element names gets installed.

## Save Problems

When I try to save an element, I get a "system locked" message, even though I have the element locked.

During the time that you had the lock, the administrator system-locked one or all of the files that pertain to that element on the server. Contact your administrator to remove the system lock. After this is done, you can save the element and your changes will be incorporated.

When I try to save a template, I get an error message.

The template file on the server is read-only. Contact your server administrator to make the file writable.

## Other Problems

I can't create a new Java, C, or WebTap service.

Another Java, C, or WebTap service is locked in the folder in which you want to create the new service. All of them must be unlocked or locked by you to add a new Java, C, or WebTap service to the same folder. See "Lock/Unlock Problems" on page 40.

I can't delete a package in the Service Browser.

One of the elements in that package is locked by another user or system-locked (read-only). Contact your administrator.

The SAP BC Server went down while I was locking or unlocking an element.

The operation may or may not have completed, depending on the exact moment at which the server ceased operation. When the server is back up, restore your session and look at the current status of the element.

## Frequently Asked Questions

What is the difference between a system locked element and a read-only element?

None. "System lock" is a term used in SAP BC server to denote an element that has read-only files on the SAP BC Server. The server administrator or third-party source control administrator usually "system locks" files (makes them read-only).

**Can I multi-select elements to lock or unlock in the Service Browser?**

No. However, you can use the **Lock Contents** and **My Locked Elements** commands to lock and unlock multiple elements, respectively.

**I only save elements after I'm completely done. Remembering to unlock elements after saving them is tedious. Is there a shortcut for this task?**

Yes. Use the **Preferences** command to view the Preferences dialog box. On the **General** tab, click the **Automatically unlock upon save (not applicable for Java/C/WebTap services)** checkbox.

**What happens to the locks on elements when I upgrade SAP BC server?**

Locking information is not preserved between versions of SAP BC server. Make sure that you have saved, unlocked, and archived elements in source control before upgrading to another version of the SAP BC server.

**In SAP BC server, where is the lock information stored (such as names of elements that are locked, when they were locked, etc.)?**

The information is stored in the SAP BC Repository version 2.



**Important!** It is not recommended that you use Cooperative Development functionality in a Remote Repository configuration. Locking information for elements could be inadvertently shared with another SAP BC Server sharing the Remote Repository. Also, if the Remote Repository Server goes down or a connection is lost, development will be stalled until the Remote Repository Server is back online. Using the local SAP BC Repository while developing will eliminate these Cooperative Development problems.

---

**Should I archive derived files?**

Generally, you should not archive derived files such as the .class file that is generated when you compile a Java service. For details, see the **Using Source Control with the Integration Server** chapter in this guide.

**What happens to the locks on elements when I replicate a package?**

Locking information is not preserved when you replicate and publish a package. This is expected behavior and is part of the feature's design. You can, however, preserve system locks (read-only file attributes).

# 5

## Administration and Best Practices

■ Introduction . . . . .	44
■ Disabling and Re-Enabling Locking . . . . .	44
■ Best Practices . . . . .	44

## Introduction

This chapter contains information intended for the Server Administrator and users who regularly replicate and publish packages as part of the production process.

## Disabling and Re-Enabling Locking

There may be times in which you do not want to implement locking on the SAP BC Server. If you are a server administrator, you can disable and re-enable locking by editing the configuration parameters in `\server\config\server.cnf`.

### Before You Begin

- Make sure that all users have completed development on the server and unlocked all elements.
- Close all Developer sessions. After you change the extended settings in the following procedure, users will need to open a new Developer session.

### Procedure

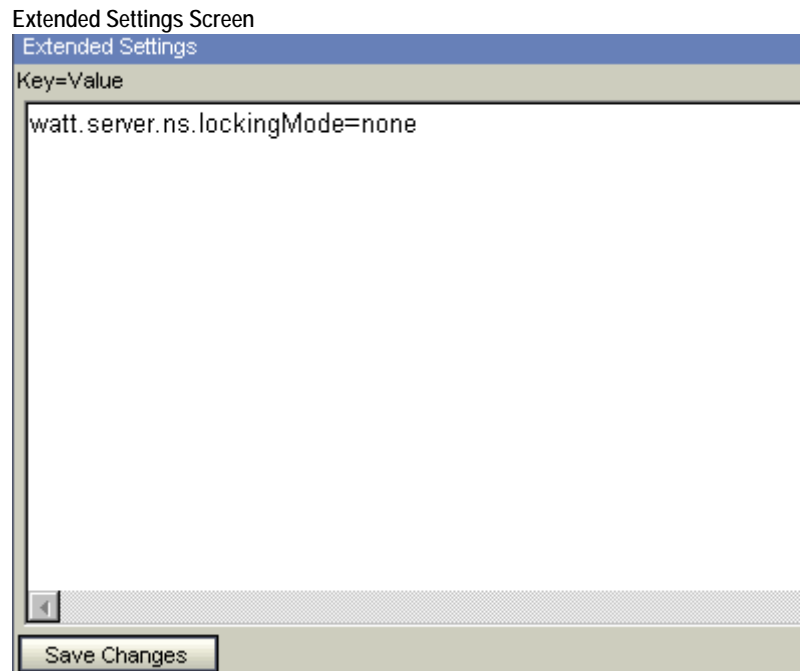
To disable or reenable locking, you use the Server Administrator or manually edit `server.cnf`. The following procedure describes the Server Administrator procedure.

Make sure that you only use this method of changing the settings. Later, if you change the settings by editing `server.cnf`, conflicts can occur.

#### To disable locking on the SAP BC Server

- 1 Complete the tasks in “Before You Begin”.
- 2 In the Server Administrator, under **Settings**, click **Extended**.
- 3 Click **Edit Extended Settings**.
- 4 In the Extended Settings box, type a key and value according to the following table.

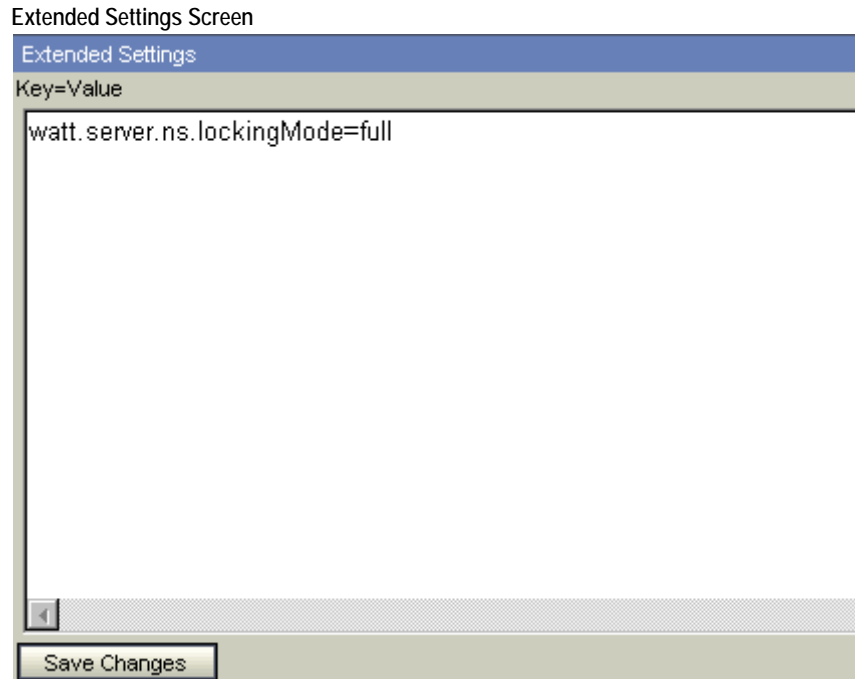
If you want to...	Type this...
Disable user locking and show no locks	<code>watt.server.ns.lockingMode=none</code>
Disable user locking but show system locks	<code>watt.server.ns.lockingMode=system</code>



- 5 Click Save Changes. The information is saved to `\server\config\server.cnf`.
- 6 Restart the SAP BC Server. The updated settings are now in effect.

#### To re-enable locking on the SAP BC Server

- 1 Complete the tasks in "Before You Begin".
- 2 In the Server Administrator, under Settings, click Extended.
- 3 Click Edit Extended Settings.
- 4 In the Extended Settings box, set the value of `watt.server.ns.lockingMode` to **full**.



- 5 Click Save Changes. The information is saved to `\server\config\server.cnf`.
- 6 Restart the SAP BC Server for the changes to take effect.

## Best Practices

### Remote Server Configuration

- It is not recommended that you use Cooperative Development functionality in a Remote Repository configuration. Locking information for elements could be inadvertently shared with another SAP BC Server sharing the Remote Repository. Also, if the Remote Repository Server goes down or a connection is lost, development will be stalled until the Remote Repository Server is back online. Using the local SAP BC Repository while developing will eliminate these Cooperative Development problems.

### Server Usernames

- When logging in to the SAP BC Server, use a distinct username. Locking is based on your username, so it is important that each user log in to the server with a unique username (not “Administrator” or “Developer”).



## Package Replication and Publishing

- If you are not using third-party source control, always back up your packages every day/night using package replication and publishing. Since locking information does not travel with packages (or partial packages) when they are replicated, it is recommended that you version each package according to date. Do not replace/overwrite packages--delete the old package entirely and install the new package.




---

**Note:** If you do replace/overwrite packages—SAP BC server takes the intersection of elements in the Service Browser. It will also move the existing package to the Salvaged folder.

---

- When you replicate and publish a package, the locking information is not preserved. This is expected behavior and is part of the feature's design. You can, however, preserve system locks (read-only file attributes).
- Before you publish a package, keep in mind that user locks are not preserved.
- When you disable a package, the locking information is not preserved. This is to prevent conflicts if another package with the same folder and element names gets installed.
- When you salvage a deleted package, lock information is not preserved. Before you salvage or delete a package, make sure that all locks are removed from the destination package.

It is not recommended that you use system or user locking on packages that are frequently replicated and/or partially replicated. For example, when sending frequently updated packages to partners.

- If an SAP specific package is being replicated from one SAP BC server to another, it is recommended to use an appropriate SAP system alias, e.g. <customer namespace>\_<application>. It is easier to create such an alias in the target system than to search for this value in all services of a package.
- ACLs and authorizations are not being replicated when transferring the packages from one SAP BC server to another. They must be newly created on the target server according to its specific requirements.
- Make sure that ports are not being replicated inadvertently when transferring the package to the target server. This might cause security gaps.
- Routing Rules and Inbound Maps are not being replicated when transferring a package, i.e. the corresponding services will be replicated without being specified as a Routing Rule or an Inbound Map. E.g. they won't figure in the Routing Rule list.
- To avoid unexpected behavior of a package you should use the option of defining a package dependency as specific as possible, e.g. requiring a specified version of the

SAP package. In this case you will get a 'non-released' message when starting a package with a different version.

## Package and Folder Organization

- Use a single package or folder per developer or per Java/C/WebTap service.

## Source Control

- If there has been a significant change to the source code, always reload the package to reflect the latest system locks.
- Before adding service code to source control, or checking service code in, make sure that no user has a lock on the service in SAP BC server.

## Upgrading SAP BC server

- When you upgrade the SAP BC Server to a new version, you lose all lock information. Therefore, before upgrading, make sure that all locks are removed and all changes are saved.

# Index

## A

### API

related documentation 6

## C

### C/C++ services

checking out of source control 36  
source control of 32, 36

### checking

components into source control 35  
components out of source control 36

### class files

location of 29  
source control 28, 29

### code directory 29

### code directoryclasses 29

### coded services

checking out of source control 36  
source control of 32, 36

### COM services

checking out of source control 36  
source control of 36

### components

checking into source control 35  
checking out of source control 36  
DERIVED 28  
DYNAMIC 28  
global 33  
package 27  
STATIC 28

### configuration files 34, 36

### configuration management

*See also* source control  
candidates for source control 28, 33  
checking components into source control 35  
checking components out of source control 36  
components in a solution 27  
global components 33  
operations that affect package components 37

package contents 27

package location 27

updating files that are under source control 37

conventions used in this document 4, 5

## D

### DERIVED components

definition of 28  
global components 33  
package components 28

### Developer

unlocking elements 16

### directories

code 29, 35, 36  
code jars 29  
code libs 29  
code source 35  
codeclasses 29  
code€libs 35  
codejars 35  
codesource 30  
ns 30, 35, 36  
packages 27  
pub 28, 35  
templates 28, 35, 37  
widls 28, 35

### documentation

conventions used 4  
printing 7  
related manuals 6  
using effectively 4  
viewing 7

### DSPs, source control of 28

### DYNAMIC components

checking in 34, 37  
definition of 28  
global components 33  
package components 28

## E

- element lock
  - copying, moving, and deleting 22
  - removing. See unlocking elements
  - system lock 10
  - user lock 10
- element locks
  - troubleshooting 40

## F

- files
  - affected by Developer operations 28, 37
  - affected by Server Administrator operations 28, 37
  - associated with services 28, 33
  - containing global parameters 33, 34, 36
  - flow.xml 31, 37
  - java.frag 32, 33, 38
  - manifest.v3 28, 35, 37
  - node.idf, interface 31, 38
  - node.ndf
    - coded service 32, 37
    - flow services 31, 37
    - Java service 32, 37
    - records 32
    - specification 33
    - WebTap service 33
  - package members 28
  - server.bat 33, 34, 36
  - server.sh 33, 34, 36
  - webtap.rul 33, 38
- flow services 31
  - file system representation 31
  - source control of 26, 28
- flow.xml 31, 37

## G

- global parameter files 33

## H

- HTML pages, source control 28

## I

- IDE
  - source control ramifications 36
- interfaces
  - file system representation 30
  - node.idf 31, 38

## J

- jars directory 29, 35
- Java services
  - checking out of source control 36
  - location of jar files 29
  - location of library files 29
  - source control of 26, 32, 36
- java.frag
  - for coded services 32, 38
  - for WebTap services 33
- jcode utility
  - using with source control 36

## L

- libs directory 29, 35
- locking elements
  - corresponding server files 21
  - description 10
  - folder or package 12
  - Java, C/C++, WebTap 13
  - single element 12
  - system locking 14
  - template service 14
  - unlocking. See unlocking elements
  - viewing lock status 11, 15

## M

- managing components of an SAP solution 26
- manifest.v3 28, 35, 37

## N

- node.idf, for interface definition 31, 38
- node.ndf
  - for coded services 32, 37
  - for flow service 31, 37

- for records 32
  - for specification 33
  - for WebTap services 33
- ns directory 30, 35, 36
  - checking components out from source control 36
  - flow services 31
  - interfaces 30
  - specifications 32
- O**
- output templates
  - source control of 28
- P**
- packages
  - contents of 27, 28
  - directories 27
  - location on server 27
  - source control candidates 28
- PDF, viewing 7
- printing
  - this guide 7
- pub directory 28, 35
- R**
- records
  - file system representation 30
  - source control of 28
- S**
- SAP BC Server
  - configuration files 33
  - package directory structure 27
  - property files affecting services 33
  - source control of 33
- Server Administrator
  - unlocking elements 18
- server files
  - viewing for locked elements 21
- server.bat 34, 36
- server.sh 34, 36
- services
  - checking out of source control 36
  - source control of 26, 32, 36
- source code
  - location of 30
  - source control of 30
- source control
  - \*.cnf files 34, 36
  - C/C++ services 32, 36
  - candidates 28, 33
  - checking components in 35
  - checking components out 36
  - classes directory files 29
  - coded services 32, 36
  - COM services 36
  - DERIVED components 28, 33
  - DYNAMIC components 28, 33, 34, 37
  - global parameter files 26, 33, 35
  - interface files 30
  - jars directory files 29, 35
  - Java services 32, 36
  - libs directory files 29, 35
  - managing components 26
  - manifest.v3 28
  - ns directory files 30, 35, 36
  - operations that affect package components 37
  - package files 26, 28, 35
  - pub directory files 28, 35
  - records 32
  - running jcode after check out 36
  - server configuration files 33
  - server.bat 34, 36
  - server.sh 34, 36
  - source directory files 29, 35, 36
  - specifications 32
  - STATIC components 28, 33, 34, 37
  - template directory files 28, 35, 37
  - updating files that are under source control 37
  - widl directory files 28, 35
- source directory 29, 35, 36
- specifications
  - file system representation 32
  - source control of 28
- STATIC components

- checking in 34, 37
- definition of 28
- global components 33
- package components 28
- system locking an element 14

## T

- templates
  - See also* output templates
- templates directory 28, 35, 37
- To 12
- typographical conventions 4, 5

## U

- unlocking elements
  - more than one 16
  - system locked 20
  - using Developer 16
  - using Server Administrator 18
- updating files that are under source control 37

## V

- viewing
  - documentation in PDF format 7
  - status of a locked element 11, 15
- viewing this document in PDF format 7

## W

- WebTap
  - webtap.rul 33, 38
- WIDL
  - source control of 28
- widls directory 28, 35
- WQL
  - See also* webMethods Query Language

## X

- XQL
  - See also* XML Query Language