SAP Business Connector
Administration 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®, and 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®, SAPma®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

Chapter 1. Introduction .............................................................. 13
   Welcome! ............................................................................. 14
   Typographical Conventions .................................................. 14
   Program Code Conventions .................................................. 15
   Related Documentation ....................................................... 16
   Viewing this Document ....................................................... 17
   Printing this Guide .............................................................. 18

Chapter 2. The Role of the Administrator ..................................... 19
   What Does an Administrator Do? ......................................... 20
   Typical Administrative Responsibilities ................................ 20
   The Server Administrator ................................................... 20
   Receiving Administrative Messages from the Server ............... 21
   The Administrator User ....................................................... 21
      The Administrator’s Password ........................................... 21
   Adding Backup Administrators .......................................... 21

Chapter 3. An Overview of the Server ....................................... 23
   The Role of the Server ......................................................... 24
   Architecture ....................................................................... 24
      Services ........................................................................... 25
      Retrieving Data for Services .......................................... 27
   How the Server Executes Services ...................................... 27
   Security Features ................................................................ 28
   Audit-Trail Logging ............................................................ 29
   Caching ............................................................................. 29

Chapter 4. Starting and Stopping the Server .............................. 31
   Starting the SAP BC Server ................................................ 32
      Starting the Server from the Command Line ...................... 32
   What Happens When You Start the Server? .......................... 34
      How to Tell if the Server is Running Correctly ................... 35
   Shutting Down the SAP BC Server .................................... 35
      Viewing Active Sessions ............................................... 35
   Restarting the SAP BC Server ............................................ 36
   Server Recovery ............................................................... 36
      Site-Specific Services .................................................... 37
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Checklist for Using Client Certificates</td>
<td>137</td>
</tr>
<tr>
<td>Items You Need Before Configuring Ports to Request Client Certificates</td>
<td>138</td>
</tr>
<tr>
<td>Importing a Client Certificate and Mapping It to a User</td>
<td>138</td>
</tr>
<tr>
<td>Configuring How Ports Handle Client Certificates</td>
<td>139</td>
</tr>
<tr>
<td>Basic Authentication (User Names and Passwords)</td>
<td>141</td>
</tr>
<tr>
<td>Customizing Authentication</td>
<td>141</td>
</tr>
<tr>
<td>Overview of Steps</td>
<td>143</td>
</tr>
<tr>
<td>Protecting Your Internal SAP BC Server with Reverse Invoke</td>
<td>147</td>
</tr>
<tr>
<td>How Reverse Invoke Works</td>
<td>148</td>
</tr>
<tr>
<td>Setting Up Reverse Invoke</td>
<td>149</td>
</tr>
<tr>
<td>Configure an SAP BC Server in the DMZ to Be a Reverse Invoke SAP BC Server</td>
<td>150</td>
</tr>
<tr>
<td>Setting Up the Registration Port</td>
<td>152</td>
</tr>
<tr>
<td>Setting Up the Proxy Port</td>
<td>155</td>
</tr>
<tr>
<td>Setting Up the Filtering Service</td>
<td>158</td>
</tr>
<tr>
<td>Monitoring Traffic Between Your Reverse Invoke SAP BC Server and the Internal SAP BC Server</td>
<td>159</td>
</tr>
<tr>
<td>Setting Up Your Internal SAP BC Server to Connect to a Reverse Invoke SAP BC Server</td>
<td>160</td>
</tr>
<tr>
<td>Specifying How the Internal Server Connects to the Reverse Invoke SAP BC Server</td>
<td>160</td>
</tr>
<tr>
<td>Specifying Client Authentication Mode</td>
<td>163</td>
</tr>
<tr>
<td>Setting a Trusted Authorities Directory</td>
<td>164</td>
</tr>
<tr>
<td>Setting the Access Mode for Your Internal Server</td>
<td>164</td>
</tr>
<tr>
<td>Viewing and Resetting Connections</td>
<td>165</td>
</tr>
<tr>
<td>Frequently Asked Questions About Reverse Invoke</td>
<td>166</td>
</tr>
<tr>
<td>Responding to NT Challenge/Response Authentication</td>
<td>168</td>
</tr>
<tr>
<td>User Name, Password, and Domain Name</td>
<td>169</td>
</tr>
<tr>
<td>Activating NT Challenge/Response</td>
<td>169</td>
</tr>
<tr>
<td>Using Tokens to Counter XSRF Attacks</td>
<td>170</td>
</tr>
<tr>
<td>Overview</td>
<td>170</td>
</tr>
<tr>
<td>Enabling token usage</td>
<td>170</td>
</tr>
<tr>
<td>Defining entry pages</td>
<td>171</td>
</tr>
<tr>
<td>Avoiding pitfalls of token usage</td>
<td>172</td>
</tr>
<tr>
<td>Chapter 10. Using an External Directory (LDAP or NIS)</td>
<td>175</td>
</tr>
<tr>
<td>Overview of Using External Directories</td>
<td>176</td>
</tr>
<tr>
<td>How the Server Uses Externally-Defined Information</td>
<td>176</td>
</tr>
<tr>
<td>When the Server Accesses Externally-Defined Information</td>
<td>177</td>
</tr>
<tr>
<td>Configuring the Server to Use an External Directory</td>
<td>177</td>
</tr>
<tr>
<td>Configuring the Server to Use LDAP</td>
<td>177</td>
</tr>
<tr>
<td>Configuring the Server to Use NIS</td>
<td>180</td>
</tr>
<tr>
<td>Stopping Use of an External Directory</td>
<td>181</td>
</tr>
</tbody>
</table>
Considerations for User Accounts and Groups .............................................. 181
Authenticating External Clients ................................................................. 183
Granting Administrator Privileges to External Users .................................... 183
Granting Developer Privileges to External Users ......................................... 184
Granting Access to Services and Files to External Users ............................. 185

Chapter 11. Managing Packages ................................................................. 187
Using Packages .......................................................................................... 188
Predefined Packages ................................................................................... 188
How the Server Stores Package Information .............................................. 189
Manifest File .............................................................................................. 191
Finding Information about Your Packages ............................................... 191
Viewing the Packages that Reside on Your Server ................................... 192
Determing Whether the Server Successfully Loaded the Package .............. 192
Determining Whether the Package Is Enabled or Disabled ....................... 193
Displaying Information about a Package .................................................. 193
Displaying Information about Services and Folders in a Package .............. 196
Displaying Documentation for a Package .................................................. 196
Working with Packages ............................................................................. 197
Creating a Package ....................................................................................... 198
Activating a Package .................................................................................... 198
Reloading a Package .................................................................................... 198
Enabling a Package ..................................................................................... 199
Disabling a Package ..................................................................................... 199
Deleting a Package ...................................................................................... 200
Recovering a Package .................................................................................. 200
Archiving a Package .................................................................................... 201
Copying Packages from One Server to Another ...................................... 201
Overview of Package Replication ............................................................... 201
Version Checking ......................................................................................... 206
Who Can Subscribe? ..................................................................................... 207
Guidelines for Using Package Replication ............................................... 207
The Publishing Server .................................................................................. 208
Displaying Subscribers ............................................................................... 208
Adding Subscribers from a Publishing Server ......................................... 208
Updating Subscriber Information ............................................................... 210
Removing Subscribers for a Package .......................................................... 211
Publishing a Package ................................................................................... 212
Specifying File and Version Information for a Release or Archive ............ 213
The Subscribing Server ................................................................................. 217
Displaying Packages That Your Server Subscribes To ............................... 218
Manually Pulling a Package .................................................. 218
Subscribing to a Package from a Subscribing Server ................. 219
Updating Your Subscription Information ................................. 222
Canceling a Subscription ...................................................... 224
Installing a Package Published by Another Server ..................... 225

Chapter 12. Managing Services .......................................... 227
About Services ................................................................. 228
Fully-Qualified Service Names ............................................. 228
Package Names and Service Names ....................................... 229
Finding Information about Services and Folders ....................... 229
Listing Folders and Services ................................................ 229
Displaying Information about a Service .................................. 230
Working with Services ....................................................... 231
Manually Adding a Service to the Server ................................. 231
Testing Services ............................................................... 231
Running Services When Packages Are Loaded, Unloaded, or Replicated .................................................. 232
What Is a Startup Service? .................................................. 232
What Is a Shutdown Service? ............................................... 232
What Is a Replication Service? ............................................ 233
Guidelines for Using Startup, Shutdown, and Replication Services .................................................. 233
Running Services in Response to Specific Events ...................... 233
Scheduling Services to Execute at Specified Times ..................... 234
Scheduling a User Task ....................................................... 234
    Using the Once Option .................................................. 235
    Using the Repeating Option .......................................... 235
    Using the Complex Option ............................................ 235
Viewing Scheduled User Tasks ............................................. 241
Updating Scheduled User Tasks ........................................... 241
Suspending Scheduled User Tasks ........................................ 241
Resuming Suspended Scheduled User Tasks ............................. 242
Canceling Scheduled User Tasks .......................................... 242
Viewing the Scheduled System Tasks .................................... 242

Chapter 13. Configuring Guaranteed Delivery ......................... 245
About Guaranteed Delivery .................................................. 246
Configuring the Server for Guaranteed Delivery ....................... 247
Settings Shared by Both Inbound and Outbound Transactions .......... 247
Settings for Inbound Transactions ....................................... 247
Settings for Outbound Transactions .................................... 249
Administering Guaranteed Delivery ...................................... 251
10

Contents

Shutting Down Guaranteed Delivery ...................................................... 251
Reinitializing Guaranteed Delivery ....................................................... 251
  Inbound Transactions ................................................................. 252
  Outbound Transactions .............................................................. 252
  Testing Guaranteed Delivery ......................................................... 253
Specifying an E-Mail Address and SMTP Server for Error Messages ............ 254

Chapter 14. Configuring Access to Database Systems ............................. 255
  Database Connection Information .................................................. 256
  Pre-Loading JDBC Drivers at Server Initialization .............................. 256
  Making the Java Classes for JDBC Drivers Accessible .......................... 257
  Identifying Database Aliases ....................................................... 257
  Database Connection Pooling ....................................................... 261
  Updating Information for Database Aliases ..................................... 261
  Deleting Database Aliases ........................................................... 262
  Monitoring Connection Pools ....................................................... 262

Appendix A. SAP BC Server Deployment Checklist ................................. 265
  Introduction .................................................................................... 266
  STAGE 1: Installation ........................................................................ 266
  STAGE 2: Basic Configuration .......................................................... 266
  STAGE 3: Setting Up Users, Groups, and ACLs .................................... 268
  STAGE 4: Publishing Packages ......................................................... 269
  STAGE 5: Configuring Database Connections ..................................... 270
  STAGE 6: Installing Run-Time Classes .............................................. 270
  STAGE 7: Setting Up Security .......................................................... 271
  STAGE 8: Startup and Test .............................................................. 272
  STAGE 9: Archive Sources .............................................................. 272

Appendix B. Server Configuration Parameters ....................................... 275
  Introduction .................................................................................... 276
  watt.debug ....................................................................................... 276
  watt.debug2 ..................................................................................... 276
  watt.mail ......................................................................................... 276
  watt.net .......................................................................................... 277
  watt.repo ......................................................................................... 279
  watt.security .................................................................................... 279
  watt.server ..................................................................................... 280
  watt.tx ............................................................................................ 290
## Contents

### Appendix C. Server Log Files and JVM Information

- The Audit Log ................................................................. 294
- The Error Log ................................................................. 294
- The Server Log ............................................................... 295
- The Session Log .............................................................. 297
- The Transaction Logs ......................................................... 298
- JVM Information .............................................................. 299
  - Trace Information (Standard JVM) ....................................... 299
  - Trace Information (SAP JVM) ............................................ 299
- Memory Information ......................................................... 300
- VM Information ............................................................... 300
- Thread Information .......................................................... 300
- Debugger Control ............................................................. 300

### Appendix D. Wireless Communication with the SAP BC Server

- How Does the SAP BC Server Communicate with Wireless Devices? ......................................................... 302
- Using URLs for Wireless Access to the SAP BC Server ................................................................. 304
  - Invoking a Service with a URL ........................................... 304
  - Requesting a WML or HDML Page with a URL ...................... 305
- WML and HDML Samples .................................................... 306

### Index ................................................................. 309
Introduction

- Welcome! ................................................................. 14
- Typographical Conventions ........................................ 14
- Related Documentation .............................................. 16
- Viewing this Document ............................................. 17
- Printing this Guide .................................................. 18
Welcome!

This guide is for the administrator of a SAP BC Server. It provides an overview of how the server operates and explains common administrative tasks such as:

- Starting and stopping the server
- Configuring the server
- Setting up user accounts and security
- Managing packages and services

Typographical Conventions

This document uses the following typographical conventions:

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

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

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

The following documents are companions to this guide. Some documents are in PDF format and others are in HTML.

<table>
<thead>
<tr>
<th>Refer to this book...</th>
<th>For...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAP BC Developer Guide</strong></td>
<td>Information about creating and testing services and client applications. This book is for solution developers. You will find this book at: &lt;sapbc&gt;\developer\doc\SAPBCDeveloperGuide.pdf</td>
</tr>
<tr>
<td><strong>SAP BC Cooperative Development Guide</strong></td>
<td>Information about locking and unlocking elements, as well as integration with third-party source control applications. This book is for solution developers. You will find this book at: &lt;sapbc&gt;\server\doc\SAPBCCoopDevGuide.pdf</td>
</tr>
<tr>
<td><strong>SAP BC Built-In Services Reference Guide</strong></td>
<td>Descriptions of services that are installed on your SAP BC Server. This book is for solution developers. You will find this book at: &lt;sapbc&gt;\developer\doc\SAPBCBuiltInServices.pdf</td>
</tr>
<tr>
<td><strong>SAP BC Output Templates and DSPs</strong></td>
<td>Information about creating output templates and Dynamic Server Pages (DSPs). This reference is for solution developers. You will find this book at: &lt;sapbc&gt;\developer\doc\SAPBCTemplatesAndDSPs.pdf</td>
</tr>
<tr>
<td><strong>SAP BC Java API Reference</strong></td>
<td>Descriptions of the Java classes you use to create services. This reference is for developers who build services using Java. You will find this book at: &lt;sapbc&gt;\server\doc\api\Java\index.html</td>
</tr>
<tr>
<td><strong>SAP BC SOAP Programming Guide</strong></td>
<td>Information about using the SAP BC Server to exchange SOAP messages. This book is for developers. You will find this book at: &lt;sapbc&gt;\developer\doc\SAPBCSoapProgrammingGuide.pdf</td>
</tr>
</tbody>
</table>
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 the Adobe web site:

http://www.adobe.com/downloads

**Printing this Guide**

To produce a hard copy of this guide, print this document from Acrobat Reader.
The Role of the Administrator

- What Does an Administrator Do? ................................................................. 20
- Typical Administrative Responsibilities .................................................... 20
- The Server Administrator .......................................................................... 20
- Receiving Administrative Messages from the Server .................................. 21
- The Administrator User ............................................................................ 21
- Adding Backup Administrators ................................................................. 21
What Does an Administrator Do?

In an SAP BC environment, the administrator is responsible for installing, configuring, and maintaining the SAP BC server. He or she is also responsible for ensuring the server is secure, available to clients, and running at peak performance. Usually, one person is appointed as the administrator, although most sites identify at least one other person to act as a backup.

Typical Administrative Responsibilities

If you are the SAP BC server Administrator for your site, you might be involved in some or all of the following activities.

- **Installing and upgrading the SAP BC server**, which includes tasks such as equipping the server computer with appropriate hardware and software, downloading and installing the server program, and implementing upgrades as needed.

- **Starting and stopping the server**, which includes shutting down the server when necessary (e.g., for routine maintenance or reconfiguration) and restarting it afterwards. It also includes performing your site's standard recovery procedures following a hardware or software failure of the server computer. For information about these activities, see “Starting and Stopping the Server” on page 31.

- **Configuring server settings**, which includes setting basic operating parameters such as the maximum session limits, log file options, and port assignments. For information about these activities, see “Configuring the Server” on page 63.

- **Administering users and groups**, which includes defining user names and passwords for authorized users and assigning them to groups. For information about this task, see “Managing Users and Groups” on page 45. Alternatively, you can configure the server to acquire user and group information from an external system (e.g., LDAP or NIS). For more information, see “Using an External Directory (LDAP or NIS)” on page 175.

- **Administering server security**, which includes identifying other administrators, assigning access controls to individual services, and configuring the server's use of digital certificates. For more information about this task, see “Managing Server Security” on page 107.

- **Managing packages and services**, which includes tasks such as activating/deactivating services, copying packages, and updating services and/or packages as necessary. For more information about this task, see “Managing Packages” on page 187 and “Managing Services” on page 227.

The Server Administrator

The “Server Administrator” is the utility you use to accomplish administrative tasks. You use it to monitor server activity, examine log information, add users, enable/disable
services, and adjust the server’s performance features. For information about the Server Administrator, see “Using the Server Administrator” on page 39.

Receiving Administrative Messages from the Server

The SAP BC server issues e-mail messages for a variety of failure conditions (for example, internal errors, binding errors, and transaction manager errors). As an administrator, you are the one who should receive these messages and take appropriate action when errors occur.

To ensure that you (or an appropriate alternate) receive messages from the server, you must set the Mail parameters using the Server Administrator as described in “Working with Log Files” on page 89.

The Administrator User

Every SAP BC Server is installed with a predefined user account called “Administrator.” By default, this user is the only one who can perform administrative tasks with the Server Administrator.

The Administrator’s Password

The predefined password assigned to the Administrator user account is “manage”.

Important! The predefined password for the Administrator account is “manage”. The predefined password for the Developer account is “isdev”. The predefined password for the Replicator account is “iscopy”.

Change all of these passwords immediately after installing the SAP BC server. Otherwise, your server will be vulnerable to anyone who knows the default passwords that SAP installs on its servers. When assigning a password, make it something that is difficult to guess. For example, make it a mixture of upper- and lowercase letters, numbers, and special characters. Do not use a name, a phone number, your license plate, your social security number, or other generally available information. Do not write passwords down. Do not tell anyone the password unless you are sure of that person’s identity.

To learn how to change passwords, see “Changing Passwords and Password Requirements” on page 50.

Adding Backup Administrators

It is a good idea to designate at least one individual as a “backup administrator,” who can administer the SAP BC Server when you are not available.
To add a backup administrator to your server, create a regular user account for the user (if he or she does not already have one); then add that user account to the “Administrators” group.

Only members of the “Administrators” group can use the Server Administrator. For information about creating user accounts and adding them to groups, see “Managing Users and Groups” on page 45.

**Note:** If you use an external directory for user and group information, see “Granting Administrator Privileges to External Users” on page 183 for information about adding administrators.
An Overview of the Server

- The Role of the Server ......................................................... 24
- Architecture ................................................................. 24
- How the Server Executes Services ................................. 27
- Security Features ........................................................... 28
- Audit-Trail Logging .......................................................... 29
- Caching ................................................................. 29
The Role of the Server

The SAP BC server hosts packages that contain services and related files. The server comes with several packages. For example, it includes packages that contain built-in services that your developers might want to invoke from their services or client applications and services that demonstrate some of the features of the SAP BC Server. You can create additional packages to hold the services that your developers create. Your developers can create services that perform functions, such as, integrating your business systems with those of your partners, retrieving data from legacy systems, and accessing and updating databases.

The SAP BC server provides an environment for the orderly, efficient, and secure, execution of services. It decodes client requests, identifies the requested services, invokes the services, passes data to them in the expected format, encodes the output produced by the services, and returns output to the clients.

Additionally, the server authenticates clients, verifies that they are authorized to execute the requested service, maintains audit-trail logs, and promotes throughput using facilities such as service result caching.

Architecture

The SAP BC server listens for client requests on one or more ports. You can associate the type of protocol that the server uses for each port. The server supports HTTP, HTTPS, FTP, and e-mail ports.

When applications are built around the SAP BC Server thin client, the application uses an HTTP or HTTPS port for communication with the server. When using HTTP or HTTPS ports, the clients communicate using the webMethods Remote Procedure Call (RPC). Because the server supports both HTTP and HTTPS, it can listen on an HTTP port for non-secure client requests and an HTTPS port for secure requests.

Note: Unlike HTTP, FTP, and e-mail, HTTPS provides for secure data transmission. It does this through encryption and certificates. Without HTTPS, unauthorized users might be able to capture or modify data, use IP spoofing to attack servers, access unauthorized services, or capture passwords. If you must pass passwords, make sure the back-end application has minimal privileges.

To interact with the server without using the webMethods RPC, use an FTP port. A typical use for an FTP port is to get a directory listing, change to the directory that contains the service you want to invoke, put a file that contains input to the service, and run the service. The server returns the output from the service to the directory in which the service resides. Use an e-mail port to receive requests through an e-mail server, such as POP3 or IMAP.

You can define as many ports as you want. When you initially install the server, it has an HTTP port at 5555.
The Server Listens for Requests on Ports that You Specify

There may be times when you want to use the standard port numbers used by Web servers: port 80 for HTTP requests and port 443 for HTTPS requests. If your SAP BC server runs on a Windows NT system, this is not a problem. However, if your SAP BC server runs on a Unix system, using a port number below 1024 requires that the server run as “root.” For security reasons, SAP discourages this practice. Instead, run your SAP BC Server using an unprivileged user ID on a high number port (for example 1024 or above) and use the port remapping capabilities present in most firewalls to move requests to the higher numbered ports.

Services

Client requests involve executing one or more services. The server maintains successfully loaded services as runnable objects within the server’s program space.

When you initialize the server, the server loads the services that are contained in enabled packages into memory. When you or another administrator enable a disabled package, the server loads services that are in that package.
When a client invokes a service, that service runs as a thread within the SAP BC Server program. Depending on what function the service is to accomplish, it can also create additional threads to perform tasks simultaneously.
Retrieving Data for Services

Tasks that services perform often include retrieving data from data sources. The server can retrieve data (for example, XML and HTML data) from local data sources or by issuing HTTP or HTTPS requests to resources such as Web servers and JDBC-enabled databases. When the server accesses data from external data sources, you can optionally route either protocol (HTTP or HTTPS) through a proxy server.

The Server Gets Data from Local Resources or Resources on the Internet

How the Server Executes Services

When the SAP BC Server receives a request from a client, it performs the following actions:

1. The server authenticates the client.
2. If a session already exists for the client, the server uses the existing session. If one does not exist, the server creates a session.
3. The server determines the content-type of the service request so it can prepare data for the requested service.
4. The server uses the supplied service name to look up the service.
CHAPTER 3 An Overview of the Server

5 The server determines whether access to the requested service is being controlled based on the port on which the request came in. If there is no restriction, the server continues with the execution of the service.

6 The server looks up the Access Control List (ACL) for the service and, using the ACL, determines whether the client is to be granted access to the service. If the ACL indicates that the client is allowed to access the service, the server continues with the execution of the service.

7 The server adds an entry to the Audit Log to mark the start of the request.

8 The server starts gathering service statistics for the service.

9 The server checks to see if the results for this service are cached. If they are, the server returns the cached results. If they are not, the server invokes the service. If the service is a flow service, which can consist of several services, it invokes each service in the flow.

Note: For each service in a flow, the server performs steps through 11.

10 The server ends the gathering of server statistics for the service.

11 The server adds an entry to the Audit Log to mark the end of the request.

12 The server encodes the service results as specified by the content type.

13 The server returns the results to the client.

Security Features

The SAP BC Server has several built-in security mechanisms to protect services from unauthorized access, prevent unauthorized administration of the SAP BC Server, and to prevent data from being intercepted during transmission.

- It requires clients to present valid credentials (i.e., user name and password or a client certificate) in order to connect to the server.

- It controls access to individual services by user groups. This mechanism is provided through the use of Access Control Lists (ACLs) that you associate with a service. For the greatest security, associate all services with an ACL.

- It allows you to control access to services based on the port on which a service request is received.

- It requires clients to present valid user names (with passwords) that have Administrator privileges before allowing access to the SAP BC Server Administrator functions.

- It hashes user passwords before storing them.

- It supports encrypted conversations through Secure Sockets Layer (SSL).
Audit-Trail Logging

- It allows your SAP BC Server to present different client certificates to different SSL servers.

For additional information about the server’s security features, refer to “Managing Server Security” on page 107.

The security of the SAP BC Server depends on the security of the underlying operating system. Make sure you do the following:

- Follow all vendor recommendations for tight configuration
- Remove any unnecessary network services, such as telnet or mail, in case they contain security flaws.
- Regularly check for and install patches from the vendor that might affect security.

See your operating system’s documentation for instructions on accomplishing these tasks.

Audit-Trail Logging

The SAP BC Server keeps several log files to maintain an audit-trail of the services that execute and errors that occur. It also maintains other log files to support guaranteed delivery and maintain performance statistics.

You can specify the level of detail you want to maintain in a log and how much historical information you want to keep.

For additional information about working with the log files, see “Working with Log Files” on page 89.

Caching

Caching is an optimization feature that can improve the performance of services. You activate it on a service-by-service basis. When you enable caching, the server saves the service invocation results in a local cache for a specified period of time. While the results are in cache, rather than re-invoking the service, the server can quickly retrieve the service results for subsequent clients’ requests for the service.

Caching can significantly improve response time of services that retrieve information from busy data sources such as high-traffic commercial Web servers or databases.

For additional information about using cache, see “Caching Service Results” on page 103.
Starting and Stopping the Server

- Starting the SAP BC Server ................................................................. 32
- What Happens When You Start the Server? ........................................... 34
- Shutting Down the SAP BC Server ......................................................... 35
- Restarting the SAP BC Server ............................................................... 36
- Server Recovery .................................................................................. 36
Starting the SAP BC Server

The SAP BC Server must be running in order for clients to execute services. If you are using the server in a development environment, it must be running in order for your developers to build, update, and test services using the SAP BC Developer.

**To start the SAP BC Server on Windows NT**

1. Click Start.
2. In the Program menu point to the webMethods folder, then point to the Servers folder.
3. Click the SAP BC Server icon.

**To start the SAP BC Server on UNIX**

1. Locate the server.sh script file that you modified for your environment when you installed the server.
2. Execute this script.

**Note:** Run this script when logged in as a non-root user. Running the script as root might reduce the security of your system.

Starting the Server from the Command Line

There are times when it is useful to start the server from the command line. Starting the server this way allows you to override certain settings in the configuration file. It also lets you start the server in “debug” mode, so you can record or display server activity.

1. At a command line, type the following command to switch to the server’s home directory:
   ```bash
cd <sapbc>/server
   ```
2. Type the following command to start the server:
   ```bash
   For Windows:  bin\server.bat -switch -switch ...
   For UNIX:     bin/server.sh -switch -switch ...
   ```
   where `switch` is any of the following:
### switch Description

<table>
<thead>
<tr>
<th>switch</th>
<th>Description</th>
</tr>
</thead>
</table>
| `-port portNumber` | Specifies the port on which the server listens for HTTP requests.  
  `portNumber` specifies the TCP/IP port number  
  **Example:** `-port 8080`  
  This switch overrides the value assigned to watt.server.port.  
  **Note:** To use port 80 (the standard for HTTP) or port 443 (the standard for HTTPS), UNIX users must be running as “root.” For security reasons, a better method is to use a higher number port (5555 for HTTP and 8080 for HTTPS), and if necessary have the firewall remap port 80 to the desired port. See “Architecture” on page 24 for a discussion of remapping ports. |
| `-home directoryName` | Specifies the server’s home directory.  
  `directoryName` specifies the complete path for the home directory.  
  **Example:** `-home D:\wmtest\server`  
  This switch overrides the value assigned to watt.server.home. |
| `-debug level` | Specifies the level of detail you want the server to maintain in its server log.  
  `level` is a number from 1 to 10 that indicates the level of detail you want to record in the log.  
  **Specify:**  
  
<table>
<thead>
<tr>
<th>Specify</th>
<th>To record:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Critical messages only</td>
</tr>
<tr>
<td>2</td>
<td>Error and critical messages</td>
</tr>
<tr>
<td>3</td>
<td>Warning, error, and critical messages</td>
</tr>
<tr>
<td>4</td>
<td>Debug, warning, error, and critical messages</td>
</tr>
<tr>
<td>5-10</td>
<td>Informational, debug, warning, error, and critical messages</td>
</tr>
</tbody>
</table>
  
  The server records more levels of informational messages the higher you set the number. |
What Happens When You Start the Server?

When you start the SAP BC Server, it performs a series of initialization steps to make itself ready for client requests. The server:

1. Establishes the operating environment by using the configuration parameters located in the configuration file (<sapbc>\server\config\server.cnf).
2. Initializes processes that perform internal management.
3. Loads information about all the enabled packages and their services that reside in the <sapbc>\server\packages directory. If a package depends on other packages, the server loads the prerequisite packages first. The server does not load disabled packages.
4. Executes the startup services for each loaded package.
5. Initializes the guaranteed delivery engine. The server checks the job store for pending guaranteed delivery transactions. It retries the pending transactions as the guaranteed delivery configuration settings specify. For more information, refer to “Configuring Guaranteed Delivery” on page 245.
6. Schedules internal system tasks, such as log rotation.
How to Tell if the Server is Running Correctly

SAP BC server To determine whether your server is running, start your browser and point it to the SAP BC Server. (See “Starting the Server Administrator” on page 40 if you need instructions for this step.)

If the server is running, you will be prompted for a name and password.

If the server is not running, your browser will issue an error message similar to the following:

“Cannot open the Internet site http://localhost:5555.”
“A connection with the server could not be established.”

Shutting Down the SAP BC Server

Shut down the server to stop the SAP BC Server and all active sessions.

To shut down the server

1. Open the Server Administrator if it is not already open.
2. In the upper left corner of any Server Administrator screen, click **Shutdown and Restart**.
3. Select whether you want the server to wait before shutting down or to shut down immediately.
   - **Delay number minutes or until all client sessions are complete.** Specify the number of minutes you want the SAP BC Server to wait before shutting down. It then begins monitoring user activity and automatically shuts down when all non-administrator sessions complete or when the time you specify elapses (whichever comes first).
   - **Perform action immediately.** The server and all active sessions terminate immediately.
4. For instructions on how to view the active sessions, refer to “Viewing Active Sessions” on page 35.
5. Click **Shutdown**.

Viewing Active Sessions

Before you shut down or restart the server, you can view the sessions that are currently active.

To view active sessions
1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Statistics.
3. Click on the current number of sessions.

**Restarting the SAP BC Server**

Restart the server when you need to stop and reload the SAP BC Server. You should restart the server when:

- **You make certain configuration changes.** Some configuration changes require the server to be restarted before they take effect. This document indicates when you are required to restart the server for configuration changes.
- **You want to incorporate updated services that cannot be dynamically reloaded.** This typically occurs for non-Java services.

**To restart the server**

1. Open the Server Administrator if it is not already open.
2. In the upper left corner of any Server Administrator screen, click **Shutdown and Restart.**
3. Select whether you want the server to wait before restarting or to restart immediately.

   - **Delay number minutes or until all client sessions are complete.** Specify the number of minutes you want the SAP BC Server to wait before restarting. It then begins monitoring user activity and automatically restarts when all non-administrator sessions complete or when the time you specify elapses (whichever comes first).
   - **Perform action immediately.** The server and all active sessions terminate immediately. Then the server restarts.

   For instructions on how to view the active sessions, refer to “Viewing Active Sessions” on page 35.
4. Click **Restart.**

**Server Recovery**

If a hardware or software problem causes the SAP BC Server to fail, restart the server using the normal start-up procedure. The server will automatically perform clean-up and initialization processes to reset the operating environment.

As part of the recovery process, the server automatically:

- Reloads the cache environment to its pre-failure state.
Restores the transaction manager’s guaranteed delivery queues. See “Configuring Guaranteed Delivery” on page 245 for additional information about guaranteed delivery recovery options.

**Site-Specific Services**

Although the server itself does not require you to take any special steps after a server failure, services that your site has created might have their own unique recovery requirements. Consult with your developers for information about these requirements.

**Recovery and the Log File**

If the SAP BC Server fails before midnight and is not restarted until after midnight, the daily log file will not be closed as it should. Instead, the server will append the new day’s information to the previous day’s log file, and the file will contain two day’s worth of information.

If you want to prevent this from happening, you must manually close the previous day’s logs before you restart the server. To close the log file, rename or move all the .log files using the naming conventions described in “Working with Log Files” on page 89.
Using the Server Administrator

- What Is the Server Administrator? .................................................. 40
- Starting the Server Administrator .............................................. 40
- Basic Operation ................................................................. 41
- Logging Out of the Server Administrator ................................. 42
- The Configuration File ......................................................... 42
What Is the Server Administrator?

The Server Administrator is an HTML-based utility you use to administer the SAP BC server. It allows you to monitor server activity, manage user accounts, make performance adjustments, and set operating parameters.

You can run the Server Administrator from any browser-equipped workstation on your network. (The Server Administrator is a browser-based application that uses services to accomplish its work.)

Starting the Server Administrator

To use the Server Administrator, simply open your browser and point it to the port on the host machine where the SAP BC Server is running.

![Important!](http://localhost:5555)
The SAP BC Server must be running in order to use this utility. If the server is not running, your browser will issue an error similar to the following:

“Cannot open the Internet site http://localhost:5555.”
“A connection with the server could not be established.”

To start the Server Administrator

1. Start your browser.
2. Point your browser to the host and port where the SAP BC Server is running.

Examples

If the server were running on the default port on the same machine where you are running the Server Administrator, you would type:

```
http://localhost:5555
```

If the server were running on port 4040 on a machine called QUICKSILVER, you would type:

```
http://QUICKSILVER:4040
```

3. Log on to the server with a user name and password that has administrator privileges.

If you just installed the SAP BC Server, you can use the following default values:

User Name: Administrator
Password: manage
Important! Use the exact combination of upper and lower-case characters shown above—user names and passwords are case sensitive.

If you change the password, be sure to select one that is difficult to guess. For example, use a mixture of upper- and lower-case letters, numbers, and special characters. Do not use a name, phone number, social security number, license plate or other generally available information.

Basic Operation

When you start the Server Administrator, your browser displays the Statistics screen.

The Server Administrator Screen

The navigation area on the left side of the screen displays the names of menus from which you can select a task. To start a task, click a task name in the navigation area. The server displays a screen that corresponds to the task you select.
CHAPTER 5 Using the Server Administrator

Getting Help

You can obtain information about the Server Administrator by clicking the Help link in the top menu of any Server Administrator screen. The help system displays a description of the parameters for the screen and a list of procedures you can perform from the screen. From this window, click Show Navigation Area to view the help system’s table of contents from which you can search for a specific procedure or screen description.

Logging Out of the Server Administrator

Log out of the Server Administrator when you no longer need to continue your current session. When you log out, the server cleans up your session.

If you are going to be away from your PC, you might also want to log out of the Server Administrator to prevent someone from accessing the Server Administrator with your user name from your PC.

To log out of the Server Administrator

1. Click Log Off in the top menu of any Server Administrator screen.
   The SAP BC Server displays a dialog box to ensure you want to log out.
2. Click OK to log out of the Server Administrator.
   The browser displays the following screen:

   ![Session terminated]
   
   To resume use of the Server Administrator, click here.

The Configuration File

Configuration settings for the SAP BC Server are stored in the server configuration file (server.cnf). This file resides in the <sapbc>/server/config directory and contains parameters that determine how the server operates.

Typically, you will use the Server Administrator to set parameters in the server.cnf file, but there may be times when you need to edit the file directly with a text editor.
For a list of parameters in the server.cnf file and their default values, see “Server Configuration Parameters” on page 275.
Managing Users and Groups

- Users and Groups ................................................................. 46
- Defining a User Account ...................................................... 46
- Disabling and Enabling Users .............................................. 53
- Defining Groups ................................................................. 55
Users and Groups

Use the Server Administrator to define user and group information to the server. The definition for a user contains the user name, password, and group membership. The definition for a group contains the group name and a list of users in the group. The server stores and maintains the information.

Alternatively, you can set up the SAP BC server to access the information from an external directory if your site uses one of the following external directories for user and group information:

- Lightweight Directory Access Protocol (LDAP)
- Network Information Service (NIS)

This chapter describes only how the SAP BC server works when user and group information is defined internally. For information about using an external directory with the SAP BC server, see “Using an External Directory (LDAP or NIS)” on page 175.

Purpose of Users and Groups

The server uses user and group information to authenticate clients and determine the server resources that a client is allowed to access.

If the server is using basic authentication (user names and passwords) to authenticate a client, it uses the user names and passwords defined in user accounts to validate the credentials a client supplies.

After a client is authenticated (whether through basic authentication or client certificates), the server uses the group membership to determine if a client is authorized for the requested action, such as, using the Server Administrator or invoking a service.

Access to the server’s resources is controlled at the group level. By setting up users and groups, you can control who can:

- **Configure and manage the server.** Only users that are members of the Administrators group (administrator privilege) can access the Server Administrator.

- **Create, modify, and delete services that reside on the server.** Only users that are members of the Developers group (developer privileges) can connect to the server from the SAP BC Developer.

- **Access services and files that reside on the server.** Access to services and files is protected at the group level.

Defining a User Account

When you create a user account on the SAP BC server, you specify a user name, password, and group membership.
Defining a User Account

- **User name.** A user name is a unique name that identifies a client. You can specify a user name that represents an actual person (e.g., “JDSmith” for John D. Smith), or you can specify a user name to represent applications, job functions, or organizations. For example, you might set up generically-named user names such as “MktgPurchAgent,” “MktgTimeKeeper,” and so forth, to represent job functions.

- **Password.** A password is an arbitrary string of characters that you associate with a user name. The server uses the password when authenticating a client who has submitted a valid user name. For more information about authentication, see “Authenticating Clients” on page 135.

  A password is meant to be a secret code shared only by the server, the server administrator, and the owner of the user account. Its purpose is to give the server added assurance that a request is coming from a legitimate user. Only administrators can assign a password to a user name and change a password for an existing account. For additional security, the server hashes passwords before storing them.

- **Group membership.** The group membership identifies the groups to which a user belongs. Access to the server’s resources is controlled at the group level:

  Only users that are members of the Administrators group can configure and manage the server using the Server Administrator. For more information about controlling access to the Server Administrator, see “Controlling Who Can Configure and Manage the Server” on page 108.

  Only users that are members of the Developers group can connect to the server from the SAP BC Developer to create, modify, and delete services. For information, see “Controlling Who Can Create, Modify, and Delete Services” on page 110.

  The server protects access to services and files using Access Control Lists (ACLs). You set up ACLs that identify groups that are allowed or not allowed to access a resource. For more information about protecting services and files, see “Controlling Access to Services and Files with ACLs” on page 123.

Predefined User Accounts

The server has the following predefined user accounts:

<table>
<thead>
<tr>
<th>User</th>
<th>Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Everybody Administrators Replicators</td>
<td>A user account that has administrator privileges. You can use the Administrator user account to access the Server Administrator to configure and manage the server.</td>
</tr>
<tr>
<td>Default</td>
<td>Everybody Anonymous</td>
<td>The server uses the information defined for the Default user when the client does not supply a user name and password.</td>
</tr>
</tbody>
</table>
Adding User Accounts

Use the following procedure to add a user account for a user.

**To add a user account to the server**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Users and Groups**.
3. Click **Add and Remove Users**.
In the **Create Users** section of the screen, specify the following information:

<table>
<thead>
<tr>
<th><strong>For this parameter...</strong></th>
<th><strong>Specify...</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Names</strong></td>
<td>A unique user name made up of a combination of letters, numbers, or symbols. You can specify one user name or one <code>username;password</code> combination per line. Press Enter to separate the lines.</td>
</tr>
</tbody>
</table>

**Important!** User names are case sensitive. When you create a user account, type it *exactly* as you want the client to enter it.

<table>
<thead>
<tr>
<th><strong>Password</strong></th>
<th>A password made up of a combination of letters, numbers, or symbols. You can specify the password in the <strong>User Names</strong> field by entering <code>username;password</code> or you can enter the password in this field. If you do not specify a password in the <strong>User Names</strong> field, the server uses the password specified in this field for the user. If you specify multiple users without passwords in the <strong>User Names</strong> field, the server uses the password in the <strong>Password</strong> field as the password for those users.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A password is required.</td>
</tr>
</tbody>
</table>

**Important!** Passwords are case sensitive. Type these values *exactly* as you want the client to enter it.

Be sure to select passwords that are difficult to guess. For example, use a mixture of upper- and lower-case letters, numbers, and special characters. Do not use a name, phone number, social security number, license plate or other generally available information.

**Re-Enter Password** | The same password again to make sure you typed it correctly.

5 Click **Create Users**.

**Removing User Accounts**

Use the following procedure to delete a user account when it is no longer needed.

**Note:** The server will not allow you to remove the following built-in user accounts: Administrator, Default, Developer, and Replicator.
To delete a user account from the server

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Users and Groups.
3. Click Add and Remove Users.
4. In the Remove Users section of the screen, select the user names for the user accounts you want to delete.
5. Click Remove Users. The server issues a prompt to verify that you want to delete the user account. Click OK to remove the user account.

**Important!** When you delete a user, the user is automatically removed from the members lists of all groups to which it was assigned.

### Changing Passwords and Password Requirements

You can change the password for your user account. In addition, you can control whether users are allowed to change their passwords through the Developer.

**Important!** Do **not** change a password if you are outside of the corporate firewall and you did not use SSL to connect to the SAP BC server.

**Note:** You cannot use the Server Administrator or the SAP BC Developer to administer users or groups stored in an external directory. This restriction includes changing the passwords of these users.

### Password Requirements

For security purposes, the SAP BC Server places length and character restrictions on passwords for non-administrators. The SAP BC Server contains a default set of password requirements; however, you can change these with the Server Administrator. A non-administrator must observe these restrictions when changing a password. An administrator user receives a warning if he or she changes a password to one that does not meet these restrictions.

The default password requirements provided by SAP are as follows:
Use the following procedure to change the password associated with a user name.

**Important!** Be sure to select passwords that are difficult to guess. For example, use a mixture of upper- and lower-case letters, numbers, and special characters. Do not use a name, phone number, social security number, license plate or other generally available information; the security of your system depends on it.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum number of characters (alphabetic characters, digits, and special characters combined) the password must contain.</td>
<td>8</td>
</tr>
<tr>
<td>Minimum number of upper case alphabetic characters the password must contain.</td>
<td>2</td>
</tr>
<tr>
<td>Minimum number of lower case alphabetic characters the password must contain.</td>
<td>2</td>
</tr>
<tr>
<td>Minimum number of digits the password must contain.</td>
<td>1</td>
</tr>
<tr>
<td>Minimum number of special characters, such as asterisk (*), period (.), question mark (?), and ampersand (&amp;) the password must contain.</td>
<td>1</td>
</tr>
</tbody>
</table>

To change a user's password

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Users and Groups**.
3. In the **Users** section of the screen, select the user name for the user whose password you want to change and click **change password**.
4. Enter the following information:
For this parameter... | Specify...
---|---
New Password | The new password, made up of any combination of letters, numbers, or symbols.

**Important!** Passwords are case sensitive. Type this value exactly as you want the client to enter it.

Be sure to select passwords that are difficult to guess. For example, use a mixture of upper- and lower-case letters, numbers, and special characters. Do not use a name, phone number, social security number, license plate or other generally available information.

Confirm Password | The same password again to make sure you typed it correctly.

5 Click **Save Password**.
Disabling and Enabling Users

There may be times when you need to disable a user. Doing so makes password cracking attacks harder by eliminating well-known user names. When you disable a user, login attempts with that user name will fail authentication and be rejected. For example, you might disable the user account of a developer who is on vacation, or the account of a trading partner whose trading privileges are suspended. Because the user has been disabled rather than deleted, you can later reinstate the account without changing the password or resetting permissions.

For deployment, you should disable the Administrator user to prevent someone from trying to guess the password and gain access to your system. Before disabling the Administrator user, you must first create another user, for example SmithAdmin, and add

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Password Change?</td>
<td>Whether users are allowed to change their passwords. These users must have developer privileges.</td>
<td>Yes</td>
</tr>
<tr>
<td>Minimum Password Length</td>
<td>Minimum number of characters (alphabetic characters, digits, and special characters combined) the password must contain.</td>
<td>8</td>
</tr>
<tr>
<td>Minimum Number of Upper Case Characters</td>
<td>Minimum number of upper case alphabetic characters the password must contain.</td>
<td>2</td>
</tr>
<tr>
<td>Minimum Number of Lower Case Characters</td>
<td>Minimum number of lower case alphabetic characters the password must contain.</td>
<td>2</td>
</tr>
<tr>
<td>Minimum Number of Digits</td>
<td>Minimum number of digits the password must contain.</td>
<td>1</td>
</tr>
<tr>
<td>Minimum Number of Special Characters (neither alphabetic nor digits)</td>
<td>Minimum number of special characters, such as asterisk (*), period (.), question mark (?), and ampersand (&amp;) the password must contain.</td>
<td>1</td>
</tr>
</tbody>
</table>
it to the Administrators, Developers, and Replicators groups. Then disable the Administrator user. (Internal server functions that run as the Administrator user, such as start up and shut down services, will still be able to run as Administrator.) Then you can use the SmithAdmin user to administer your SAP BC server.

**Disabling a User**

Use the following procedure to disable a user.

> **Important!** Before you disable the Administrator user, make sure you have defined another user with administrator privileges so you are not locked out of the server.

**To disable a user**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Users and Groups**.
3. Click **Enable and Disable Users**.
4. In the **Enabled Users** list select (highlight) the user or users you want to disable.
   
   To select additional users without deselecting currently selected users, press the CTRL key while you click on the users you want to select. To deselect a user, press the CTRL key while you click the currently selected entry.
5. Click **Save Changes**.

**Enabling a User**

Use the following procedure to enable a user. The only time you will need to enable a user is if the system administrator explicitly disabled it.

**To enable a user**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Users and Groups**.
3. Click **Enable and Disable Users**.
4. In the **Disabled Users** list select (highlight) the user or users you want to enable.
   
   To select additional users without deselecting currently selected users, press the CTRL key while you click on the users you want to select. To deselect a user, press the CTRL key while you click the currently selected entry.
5. Click **Save Changes**.
Defining Groups

A group is a named collection of users that share privileges. The privileges can be:

- Administrator privileges
- Replicator privileges
- Developer privileges
- Privileges to invoke a service
- Privileges to allow the server to serve files

Privileges to invoke a service or access files are granted and denied by Access Control Lists (ACLs) that you set up. When an administrator creates ACLs, he or she identifies groups that are allowed to access services and files and groups that are denied access to services and files.

Administrator, replicator, and developer privileges are typically granted by adding a user to the Administrators, Replicators, or Developers group, respectively. Alternatively, you can create new groups and add them to the allow lists of the Administrators, Replicators, or Developers ACLs.

Create groups that identify groups of users that will share the same privileges. When you create a group definition, you specify a group name and the members of the group.

- **Group name.** A group name is a unique name that identifies the group. You can use any name, for example, a name that defines a department (Marketing) or job function (Programmers).
- **Members.** List of user names that are members of the group.

Predefined Groups

The server is installed with the following predefined groups.

<table>
<thead>
<tr>
<th>Group Name</th>
<th>Members</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators</td>
<td>Administrator</td>
<td>This group identifies users that have administrator privileges. A user must have administrator privileges to configure and manage the server.</td>
</tr>
</tbody>
</table>

**Important!** Membership in this group gives substantial power to affect the configuration of the SAP BC server. Use caution in assigning membership in this group to individuals who can be trusted to use the privilege carefully.
### Group Name | Members | Description
--- | --- | ---
Anonymous | Default | This group identifies users that have not been authenticated.

Developers | Developer | This group identifies users that have developer privileges. A user must have developer privileges to connect to the server from the Developer.

**Important!** Membership in this group gives substantial power to affect the configuration of the SAP BC server. Use caution in assigning membership in this group to individuals who can be trusted to use the privilege carefully.

Everybody | Administrator Default Developer Replicator | All users are a member of this group. Every new user is automatically added to the Everybody group.

Replicators | Administrator Replicator | This group identifies users that have replicator privileges. The Replicators group gives its members the authority to perform package replication. (By default, the server uses members of the Replicators group for package replication.)

Users do not have to be members of the Replicators group to perform package replication. As long as user is a member of a group that is assigned to the Replicators ACL, it can perform package replication.

For more information about package replication, see “Copying Packages from One Server to Another” on page 201.

Membership in this group gives substantial power to affect the configuration of the SAP BC server. Use caution in assigning membership in this group to individuals who can be trusted to use the privilege carefully.
Adding Groups

Use the following procedure to add groups.

To add a new group to the server

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Users and Groups.
3. Click Add and Remove Groups.
4. In the Create Groups area of the screen, type a unique group name made up of a combination of letters, numbers, or symbols. You can add more than one group at a time by specifying multiple lines, one group to a line. Press Enter to separate lines.

Important! Group names are case sensitive.

5. Click Create Groups.

Adding Users to a Group

Use the following procedure to add users to a group.

Note: You cannot change the membership of the Everybody group.

To add users to a group

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Users and Groups.
The server displays the following screen.

The **Groups** area of the screen (on the right) contains two lists. **Users in this Group** is a list of users currently in the group. **Remaining Users** is a list of users *not* currently in the group.

3. From the pull-down list of groups, select the group to which you want to add a user.

4. In the **Remaining Users** list select (highlight) the user or users you want to add to the group.
   
   To select additional users without deselecting currently selected users, press the **CTRL** key while you click on the users you want to select. To deselect a user, press the **CTRL** key while you click the currently selected entry.

5. After you have selected all the users you want to add to the group, click **<**.
   
   The server moves the selected users to the **Users Currently in this Group** list.

6. Click **Save Changes**.

**Removing Users from a Group**

Use the following procedure to remove users from a group.

**Note:** You cannot change the membership of the Everybody group.
To remove a user from a group

1. Open the Server Administrator if it is not already open.

2. In the Security menu of the navigation area, click Users and Groups.
   The server displays the following screen.

   ![Server Administrator interface](image)

   The Groups area of the screen (on the right) contains two lists. Users in this Group is a list of users currently in the group. Remaining Users is a list of users not currently in the group.

3. From the pull-down list of groups, select the group from which you want to remove a user.

4. In the Users in this Group area of the screen, select (highlight) users that you want to remove from the group.
   To select additional users without deselecting currently selected users, press the CTRL key while you click on the users you want to select. To deselect a user, press the CTRL key while you click the currently selected entry.

5. At the bottom of the Users in this Group area of the screen click the arrow button.
   The server moves the selected users to the Remaining Users area of the screen.
Viewing Group Membership

Use the following procedure to view the members or a group or change the members in a group.

To view group membership for a group

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area click Users and Groups.
   The server displays the following screen.

   ![Screen showing Users and Groups](image)

   The Groups area of the screen (on the right) contains two lists. Users Currently in this Group is a list of users currently in the selected group. Remaining Users is a list of users not currently in the selected group.

3. From the pull-down list of groups, select the group for which you want to view membership.
4. The server displays the users in the Users in this Group list.

Removing Groups

Use the following procedure to remove groups that you no longer need.

**Note:** You cannot delete any of the following groups: Administrators, Developers, Replicators, Anonymous, and Everybody.
To delete a group from the server
1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Users and Groups.
3. Click Add and Remove Groups.
4. In the Remove Groups area of the screen, select the groups you want to remove.
5. Click Remove Groups.
CHAPTER 7
Configuring the Server

- Viewing and Changing Licensing Information ........................................ 64
- Configuring Ports .................................................................................. 66
- Allowing and Denying Inbound Connections to the Server ..................... 78
- Specifying a Third-Party Proxy Server for Outbound Requests ............... 83
- Setting Up Aliases for Remote SAP BC servers .................................... 85
- Working with Log Files .......................................................................... 89
- Setting the Session Timeout Limit ...................................................... 96
- Managing Server Thread Pools ............................................................ 98
CHAPTER 7 Configuring the Server

Viewing and Changing Licensing Information

When you purchase a SAP BC server, your organization is granted a license to use it with a specified number of concurrent users (simultaneous sessions). The license expires after a time period specified by your particular purchase agreement.

The License Key

When you install the server, the setup program asks you to enter your “key,” which is a special code associated with your license. After you enter this code, it is assigned to the watt.server.key parameter in the server.cnf file in the <sapbc>\server\config directory. If the watt.server.key parameter is inadvertently changed or deleted or if your license expires, your server reverts to demo mode. In this mode, there are only two licensed sessions and the server automatically shuts down 30 minutes after it is started.

Viewing or Changing the License Key

To view or change the license key for your SAP BC server, use the Licensing screen in the Server Administrator.

To view the License Key

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Licensing.
3. Click Edit License Key.
4. The server displays the License Key screen.

To change the License Key

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Licensing.
3. Click Edit License Key.
4. The server displays the License Key screen.
5. Type the new license key in the License Key field.
6. Click Submit.

Note: The SAP BC server updates the expiration date automatically after you click Submit.
Renewal Reminders
Approximately 30 days before your license expires, the SAP BC server sends an e-mail to the administrative-message recipient, reminding him or her to renew the license. In addition, the server displays the following message at the top of the License screen of the Server Administrator:

License key expires in about days days ... contact SAP for a new key.

Renewing a Key
If you need to obtain a new key or renew your license, contact your SAP sales representative.

Licensed Sessions
Your license allows a specified number of users to have sessions in the SAP BC server concurrently. The SAP BC server creates a session when a developer connects to the server from the SAP BC Developer or a server client connects to the server to execute services. If a user attempts to access the server while the maximum number of sessions are in use, the server rejects the request and returns the following error to the user:

Server has reached client limit.

You can view the current number of active sessions and the licensed session limit using the Statistics screen in the Server Administrator. This value is permanently associated with your license key and can only be changed by obtaining a new license.

Any connection made to the server by a non-Administrator user (that is, a user that is not part of the Administrators group) consumes a licensed session. The session exists until it times out (based on the server’s Session Timeout setting) or the requester stops the session by invoking the wm.server:disconnect service.

If a user invokes a stateless service and a session does not already exist for the user, the server creates a session. If the user is a non-Administrator, the user consumes a licensed session. After the service completes, the server removes the session and reduces the number of licensed sessions in use.

To view the current number of active sessions and the licensed sessions limit

1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Statistics.

The server displays the current number of active sessions in use in the Total Sessions field. The server displays the maximum number of licensed sessions your license allows in the Licensed Sessions fields.

For detailed information about the active sessions, click the number in the Total Sessions field.
CHAPTER 7 Configuring the Server

Configuring Ports

The SAP BC server listens for requests on ports that you specify. Each port is associated with a specific type of protocol—HTTP, HTTPS, e-mail, or FTP.

Typically, most client requests use an HTTP port, or for secure requests, an HTTPS port. Use an FTP port for an alternate way to request services and a convenient way for moving files to and from the server. Use an e-mail port to receive requests through an e-mail server, such as POP3 or IMAP.

In addition, if you deploy with reverse invoke, there are two special protocols you will use: SOCK and SSLSOCK. In the reverse invoke configuration, an SAP BC server sits in your company’s DMZ while your main or internal SAP BC server sits behind your inner firewall. The server in the DMZ intercepts requests and passes them to your internal server. For more information about Reverse Invoke, see “Protecting Your Internal SAP BC Server with Reverse Invoke” on page 147.

Note: Unlike HTTP, FTP and e-mail, HTTPS provides for secure data transmission. It does this through encryption and certificates. Without HTTPS, unauthorized users might be able to capture or modify data, use IP spoofing to attack servers, access unauthorized services, or capture passwords.

All ports are associated with a package. By default, they are associated with WmRoot. You can associate a port with an application package so that when you replicate the package, it continues to use a port with the same number on the new server. This feature is useful if you create an application that expects input on a specific port. The application will continue to work after it is replicated to another server.

Important! Be careful when setting up a port that is associated with a package. When copied to the target server, the new port might decrease security on that system. For example, suppose you replicate a package that is associated with an HTTP port at 5556. The replication process creates an HTTP port at 5556 on the target server. If the target server normally uses only HTTPS ports because of their greater security, then the new port presents a possible security hole on that server.

For security reasons, by default, all ports except 5555 are configured to deny access to all services, except services specified in an allow list. However, you can configure individual ports to allow access to more services as needed.

About E-mail Ports

By setting up one or more e-mail ports on your SAP BC server, you can receive client requests through an e-mail server (POP3 or IMAP).

The client builds an e-mail that contains the name of the service to run and parameters to pass to the service. The e-mail can also contain userid and password information.
The different parts of the e-mail message contain the information needed for service invocation, as shown below.

### Area of e-mail | What It Contains
---|---
**Subject** | Name of the service to execute on the SAP BC server. Specify the service in the form `folder:service`, for example, `wm.server:reflect`.
**Body** | Parameters to pass to the service. These parameters are URL encoded. The body area can also contain the userid and password of the SAP BC Server userid under which the service is to run.
**Attachment** | Parameters, in a file, for example an XML file, to pass to the service. The SAP BC server executes the service once for each attachment. When it receives an attachment, the port runs the appropriate content handler to parse the contents of the file and pass them to the service.

Here is an example of an e-mail message containing service invocation information.

```
Subject: myFolder:myFlow

one=1&two=2&three=3&$user=Administrator&$pass=manage

<?xml version="1.0"?>
<PurchaseOrder>
  <OrderNumber>1001</OrderNumber>
  <Status>Pending</Status>
  <Company>webMethods, Inc.</Company>
  <LineItem>
    <SKU>45669</SKU>
    <Quantity>1</Quantity>
    <Description>PalmPilot Professional w/PalmPilot Modem</Description>
    <Price>449.95</Price>
  </LineItem>
  <LineItem>
    <SKU>22257</SKU>
    <Quantity>1</Quantity>
    <Description>PalmPilot Modem Cable</Description>
    <Price>17.95</Price>
  </LineItem>
  <LineItem>
    <SKU>47839</SKU>
    <Quantity>1</Quantity>
    <Description>CardScan Plus 300 Parallel Port Scanner W/ CardScan Software</Description>
    <Price>264.95</Price>
  </LineItem>
</PurchaseOrder>
```

You can write a flow to send information obtained from the e-mail, such as protocol, sender, recipient, date sent, and date received, into the pipeline using the...
pub.flow:getTransportInfo service. The service returns the pub.flow:transportInfo NS record. If this information is of use to you, code your service to read this information.

Here is a sample service that reads this information:

```java
public static final void myService(IData in)
  throws ServiceException
{
  // --- <<B2B-START(myService)>> ---
  String subject = null;
  String ctype = null;
  String protocol = null;
  String filename = null;
  String sentdate = null;
  String recvdate = null;
  InputStream is = null;

  IDataCursor idc = in.getCursor();
  if (idc.first("transport")) {
    IData transport = (IData)idc.getValue();
    idc.destroy();
    idc = transport.getCursor();
    if (idc.first("protocol")) {
      protocol = (String)idc.getValue();
      if (protocol.equals("email")) {
        if (idc.first("email")) {
          IData email = (IData)idc.getValue();
          idc.destroy();
          idc = email.getCursor();
          if (idc.first("subject")) {
            subject = (String)idc.getValue();
          }
          if (idc.first("ctype")) {
            ctype = (String)idc.getValue();
          }
          if (idc.first("filename")) {
            filename = (String)idc.getValue();
          }
          if (idc.first("sentdate")) {
            sentdate = (String)idc.getValue();
          }
          if (idc.first("recvdate")) {
            recvdate = (String)idc.getValue();
          }
          if (idc.first("content")) {
            is = (InputStream)idc.getValue();
          }
          idc.destroy();
        }
      }
    }
  }
  // --- <<B2B-END(myService)>> ---
}
```
Adding Ports

By default, the server is pre-configured with an HTTP port at 5555. In addition, you can configure one or more additional ports. You can associate an HTTP, HTTPS, FTP, or e-mail protocol with the additional ports. You might add additional ports:

- If you have applications that require a specific port number
- If you want to support multiple types of listening protocols
- If you want to open several ports for the same protocol
- If you want to deploy your server in a reverse invoke configuration, in which a reverse invoke SAP BC server sits in your DMZ and intercepts requests before passing them to the server behind your inner firewall. For instructions on adding reverse invoke ports, see “Protecting Your Internal SAP BC Server with Reverse Invoke” on page 147.

**Important!** For security purposes, when you add a new port, the server defines the port to deny access to all services except those specified in an allow list. Therefore, after adding a port, you might need to perform additional steps to make more services available through the port. These steps are described in “Controlling Access to Services by Port” on page 132.

---

**To add an additional port**

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Click Add New Port.
4. In the Add Port area of the screen, select the type of port to add.
5. Click Go to Step 2. The SAP BC server displays a screen requesting information about the port.
For an HTTP port, enter the following information:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>The number you want to use for the port. Use a number that is not already in use.</td>
</tr>
<tr>
<td><strong>Package name</strong></td>
<td>The package associated with this port. When you enable the package, the server enables the port. When you disable the package, the server disables the port. If you replicate this package, the SAP BC server creates a port with this number and the same settings on the target server. If a port with this number already exists on the target server, its settings remain intact. This feature is useful if you create an application that expects input on a specific port. The application will continue to work after it is replicated to another server.</td>
</tr>
<tr>
<td><strong>Bind Address (optional)</strong></td>
<td>IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you.</td>
</tr>
</tbody>
</table>

For an HTTPS port, enter the following information:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>The number you want to use for the port. Use a number that is not already in use.</td>
</tr>
<tr>
<td><strong>Client Authentication</strong></td>
<td>The type of client authentication you want the SAP BC server to perform. See “Authenticating Clients” on page 135 for more information.</td>
</tr>
<tr>
<td><strong>None</strong></td>
<td>the server will not request client certificates but will prompt for a user and password instead.</td>
</tr>
<tr>
<td><strong>Request Client Certificates</strong></td>
<td>the server will request client certificates for all requests that come in on this HTTPS port. If the client does not present a certificate, the request proceeds anyway, prompting the user for a userid and password.</td>
</tr>
<tr>
<td><strong>Require Client Certificates</strong></td>
<td>the server requires client certificates for all request that come in on this HTTPS port. If a client does not supply a client certificate, the request fails.</td>
</tr>
</tbody>
</table>
### Configuring Ports

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Package name</strong></td>
<td>Package associated with this port. When you enable the package, the server enables the port. When you disable the package, the server disables the port. If you replicate this package, the SAP BC server creates a port with this number and the same settings on the target server. If a port with this number already exists on the target server, its settings remain intact. This feature is useful if you create an application that expects input on a specific port. The application will continue to work after it is replicated to another server.</td>
</tr>
<tr>
<td><strong>Bind Address</strong></td>
<td>IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you.</td>
</tr>
<tr>
<td>(optional)</td>
<td></td>
</tr>
<tr>
<td><strong>Certificate Chain</strong></td>
<td>Optional. Comma-separated list of path and file names of the SAP BC server's digital certificate. List the files containing the certificates in the following order: server certificate, intermediate certificate 1,..., intermediate certificate n, root certificate. Specify a value here only if you want this port to present a different server certificate from the one specified on the Certificates screen.</td>
</tr>
<tr>
<td>(comma-separated)</td>
<td></td>
</tr>
<tr>
<td><strong>Private Key</strong></td>
<td>Optional. Path and file name of the file that contains the private key of the private/public key pair associated with the SAP BC server's digital certificate. If you leave this field blank, the SAP BC server uses the private key specified on the Certificates screen.</td>
</tr>
<tr>
<td><strong>Trusted Authority</strong></td>
<td>Optional. Name of the directory (relative to the server home) that contains the digital certificates of certificate authorities trusted by this server, for example <code>config\cas</code>. If you leave this field blank, the SAP BC server uses the trusted authority directory specified on the Certificates screen. If the trusted authority field is blank on the Certificates screen as well, the server trusts no certificates.</td>
</tr>
<tr>
<td><strong>Directory</strong></td>
<td></td>
</tr>
</tbody>
</table>
For an FTP port, enter the following information:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>The number you want to use for the FTP port. Use a number that is not already in use.</td>
</tr>
<tr>
<td><strong>Package name</strong></td>
<td>Package associated with this port. When you enable the package, the server enables the port. When you disable the package, the server disables the port. If you replicate this package, the SAP BC server creates a port with this number and the same settings on the target server. If a port with this number already exists on the target server, its settings remain intact. This feature is useful if you create an application that expects input on a specific port. The application will continue to work after it is replicated to another server.</td>
</tr>
<tr>
<td><strong>Bind Address</strong></td>
<td>IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you.</td>
</tr>
</tbody>
</table>

For an e-mail port, enter the following information:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Package Name</strong></td>
<td>Package associated with this port. When you enable the package, the server enables the port. When you disable the package, the server disables the port. If you replicate this package, the SAP BC server creates a port with this number and the same settings on the target server. If a port with this number already exists on the target server, its settings remain intact. This feature is useful if you create an application that expects input on a specific port. The application will continue to work after it is replicated to another server.</td>
</tr>
</tbody>
</table>
For this parameter Specify...

Server Information

Host Name—name of the machine on which the POP3 or IMAP server is running

Type—type of mail server. Select POP3 or IMAP

User Name—user name that identifies you to the mail server

Password—password associated with the user name that identifies you to the mail server

**Note:** Passing a userid and password in an e-mail presents a possible security exposure. While the e-mail resides on the POP3 or IMAP server, someone might be able to access this information.

Time Interval—how often (in seconds) the e-mail port is to check for incoming e-mails on the POP3 or IMAP server.

Port—port to use for the mail server. The default for POP3 is 110; the default for IMAP is 143.

Log out after each mail check—for use with IMAP and multithreading only. If you select Yes, the SAP BC server logs out a read-only thread to the IMAP mail server after checking for mail on that thread. The main read/write thread to the IMAP server remains intact. If you select No, all the read-only threads remain intact. Select Yes if your IMAP server restricts the number of connections it will allow to remain logged in.

Security

Run services as user—If you selected Yes in the Require authorization within message field, the Run services as user field remains blank because the SAP BC server expects the user name and password to be in the e-mail. If you selected No in the Require authorization within message field, you must enter the user under which the service is to run on the SAP BC server.

Require authorization within message—if you select Yes, the SAP BC server checks for $user and $pass parameters in the Subject line of the e-mail. The user name is the user under which the service is to run on the SAP BC server. If you select No, you must specify the user in the Run services as user field above.
<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Processing</td>
<td><strong>Global Service</strong>—service to be executed on the SAP BC server. This field overrides a service specified in the Subject line of the e-mail.</td>
</tr>
<tr>
<td></td>
<td><strong>Default Service</strong>—service to be executed if the e-mail does not provide a valid service in the Subject line and the Global Service field is blank.</td>
</tr>
<tr>
<td></td>
<td><strong>Send reply email with service output</strong>—tells the SAP BC server to send any output generated by the service to the original sender in an e-mail attachment. If the original e-mail contained multiple attachments, the reply contains an equal number of attachments.</td>
</tr>
<tr>
<td></td>
<td><strong>Send reply email on error</strong>—tells the SAP BC server to report any errors that occurred during service execution to the original sender in the Body portion of an e-mail.</td>
</tr>
</tbody>
</table>
For this parameter | Specify...
---|---
Delete valid messages (IMAP only)—tells the SAP BC server to delete a valid e-mail from the IMAP server once the SAP BC server has successfully received the e-mail. This setting helps prevent e-mails from accumulating on the IMAP server, possibly affecting disk space and performance. The SAP BC server always deletes e-mails on a POP3 server.

Delete invalid messages (IMAP only)—tells the SAP BC server to delete invalid e-mails from the IMAP server. Invalid e-mails are those that experienced errors during processing. This setting helps prevent invalid e-mails from accumulating on the IMAP server, possibly affecting disk space and performance. The SAP BC server always deletes e-mails on a POP3 server.

Reprocess invalid messages (IMAP only)—tells the SAP BC server to reprocess invalid messages. This can be useful if messages have been qualified ‘invalid’ because e.g. the receiving system has been down at sending time.

Set processed messages to be unseen (IMAP only)—tells the SAP BC server wether you want to set processed messages to be unseen or not. If you choose ‘yes’, messages processed by SAP BC will still appear as unseen in the mailbox.

Multithreaded processing (IMAP only)—tells the SAP BC server to use multiple threads for this port. This setting allows the port to handle multiple requests at once and avoid a bottleneck.

Number of threads if multithreading is turned on—tells the SAP BC server the number of threads to use for this port. The default is 3.

Invoke service for each part of a multipart message—tells the SAP BC to execute the services included in the subject of each part of a multipart message (or the corresponding global/default services respectively).

Include email headers when passing message to content handler—specifies wether you want to pass the email header information to the content handler.

Email body contains URL encoded input parameters—specifies wether you want the email body text to be decoded or not. If you choose ‘No’, SAP BC preserves the original body text without decoding it.

Click Save Changes.
On the **Ports** screen, check the list of ports to ensure that the status in the **Enabled** column is ✔️ (enabled). If it is not, click the icon to enable the port.

The server will deny access to all services through this new port. To allow access to services through this port, see “Allow Access to Specified Services (Deny All Others)” on page 133.

If your server runs on AS/400, limit the size of the port queue that is available to the TCP/IP stack. The port queue is the number of outstanding inbound connections that are queued in the TCP/IP stack. Navigate to the `<sapbc>/server/config` directory and add this line:

```
watt.server.portQueue=511.
```

Restart the server to pick up the changes.

### Changing the Primary Port

By default, the server designates an HTTP port at 5555 as the primary port.

The primary port is an HTTP or HTTPS port that you designate as the server’s main listening port. The server does not use this port any differently than other HTTP or HTTPS ports, except that it doesn’t allow a primary port to be deleted. The primary port number is also the number that is reflected in the server’s `watt.server.port` setting. This is the number that clients will receive when they query your server’s `watt.server.port` property.

There may be times when you need to change the primary port. For example, if you decide you want only secure connections to the server, you must change the protocol for the primary port to HTTPS.

If the port you want to be the primary port has not yet been defined, add it, following the directions on page 8, then follow the directions here to make the new port the primary port.

**To change the primary port**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Ports**.
3. Click **Change Primary Port**.
4. In the **Select New Primary Port** area of the screen, select the port you want to make the primary port from the pull-down menu and click **Update**.
5. On the **Ports** screen, check the list of ports to ensure that the status in the **Enabled** column is ✔️ (enabled). If it is not, click the icon to enable the port.
Deleting a Port

If you no longer need a port, you can delete it.

**Important!** You cannot delete the primary port defined for the server.

**To delete a port**

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click **Ports**.
3. Locate the port in the **Port List**, and click the ✗ icon in the **Delete** column to delete the port. The server displays a dialog box that prompts you to verify your action. Click **OK** to verify you want to delete the port.

Enabling/Disabling a Port

If you want to temporarily prevent the server from accepting requests on one of its ports, you can disable that port. This action blocks incoming requests from reaching the server. When a port is disabled, clients receive an error message when they issue requests to it. Later, you can enable the port. If you shut down and restart the server, the port remains disabled until an administrator enables it. Disabling a port is a convenient way to eliminate developer access to an SAP BC server once it goes into production.

Another way to enable or disable a port is to enable or disable the package associated with the port. You can associate a package with a specific port so that when you replicate the package, it continues to use a port with the same number on the new server. When a package is associated with a port, enabling the package enables the port and disabling the package disables the port. For more information about associating a package with a port see “Configuring Ports” on page 66.

**Important!** You must leave at least the primary port enabled.

**To disable a port**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Ports**.
3. Locate the port in the **Listening Ports** list, and click the ✅ icon in the **Enabled** column to disable the port. The server displays a dialog box that prompts you to verify your action. Click **OK** to verify you want to disable the port.

   The server replaces the ✅ icon with **No** to indicate that the port is now disabled.
To enable a port

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Locate the port in the Port List, and click No in the Enabled column to enable the port. The server displays a dialog box that prompts you to verify your action. Click OK to verify you want to enable the port.

The server replaces the No with the ✔️ icon to indicate that the port is now enabled.

Allowing and Denying Inbound Connections to the Server

One of the many ways of protecting your SAP BC server is by controlling which hosts can access it. You control this access by configuring the server’s ports to allow or deny inbound requests from that host.

You can set up all ports to accept the same hosts or you can set up different ports to accept different hosts.

For any given port, you can specify IP access one of two ways:

- Set up the port to deny requests from all hosts except for ones you explicitly allow. Use this approach if you want to deny most hosts and allow a few.
- Set up the port to allow requests from all hosts except for ones you explicitly deny. Use this approach if you want to allow most hosts and deny a few.

Controlling the Global IP Access for Ports

This section describes how to specify the global IP access setting for ports. The server uses this setting to determine IP access for ports that do not have a custom IP access setting. The default global setting is Allow by Default.

When you create a port, you can customize IP access for it, or you can specify that it use the global IP access setting for the server. If you use the global IP access setting and later change it, the server uses the new global setting for the port. For example, as shipped, the server uses Allow by Default as the global IP access setting (with no hosts explicitly denied). If you create a new port 6666 and do not customize IP access for it, the server uses Allow by Default for port 6666. If you later change the global IP access to Deny by Default, the server then uses Deny by Default for port 6666. If you later customize IP access to port 6666, subsequent changes to the global setting will have no effect on port 6666.

To customize IP access for individual ports, see “Controlling IP Access to Individual Ports” on page 81.
Allow Inbound Connections from Specified Hosts (Deny all Others)

The following procedure describes how to change the global IP access setting to Deny by Default and specify some hosts to allow.

With this setting in effect, the server denies most hosts and allows some.

**Important!** Before you switch your global setting to Deny by Default, make sure you have at least one port that does not rely on the global setting and allows at least one host. If you inadvertently lock all hosts out of the server, you can correct the problem by manually updating the appropriate configuration file, as shown below:

**Before updating these configuration files, be sure to shut down the SAP BC server.**

To reset the global setting, update the `watt.server.hostAllow` parameter in the `server.cnf` file. For example:

```plaintext
watt.server.hostAllow=132.906.19.22
```

To reset an individual port, update the following parameter in the `config\listeners.cnf` file in the package for which the port is defined:

```xml
<array name="hostAllow" type="value" depth="1">
  <value>132.906.19.22</value>
</array>
```

To allow inbound requests from only specified hosts

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Click Change Global IP Access Restrictions.
4. Click Change IP Access Mode to Deny by Default.
   
   The server changes the access mode and displays a screen from which you can add hosts to the Allow List. Notice that the server has already included the host name and IP address of the machine from which you are using the Server Administrator so that you are not locked out of the server.
5. Click Add Hosts to Allow List.
6. Specify the host names (e.g., `workstation5.sap.com`) or IP addresses (e.g., `132.906.19.22`) of hosts from which the server is to accept inbound requests.
Separate your entries with commas, for example: *.allowme.com,

**Note:** IP addresses are harder to spoof, and therefore more secure.

You can use the following pattern-matching characters to identify several clients with similar host names or IP addresses.

<table>
<thead>
<tr>
<th>Char</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches any number of characters</td>
<td>r*.sap.com</td>
</tr>
<tr>
<td>?</td>
<td>Matches any single character</td>
<td>workstation?.sap.com</td>
</tr>
</tbody>
</table>

7 Click Add.

**Deny Inbound Connections from Specified Hosts (Allow All Others)**

The following procedure describes how to change the global IP access setting to Allow by Default and specify some hosts to deny.

With this setting in effect, the server allows most hosts and denies some.

**To deny inbound requests from specified hosts**

1 Open the Server Administrator if it is not already open.

2 In the **Security** menu of the navigation area, click **Ports**.

3 Click **Change IP Access Mode to Allow by Default**.

   The server changes the access mode and displays a screen from which you can add hosts to the Deny List.

4 Click **Add Hosts to Deny List**.

5 Specify the host names (e.g., workstation5.sap.com) or IP addresses (e.g., 132.906.19.22) of hosts from which the server is to deny inbound requests. Separate your entries with commas, for example: *.denyme.com, *.denyme2.com.

   **Note:** IP addresses are harder to spoof, and therefore more secure.

You can use the following pattern-matching characters to identify several clients with similar host names or IP addresses.
Controlling IP Access to Individual Ports

This section describes how to change the IP access settings for individual ports.

Allow Inbound Requests from Specified Hosts (Deny All Others)

The following procedure describes how to change the IP access settings for an individual port to Deny by Default and allow some hosts.

With this setting in effect, the server denies most hosts and allows some through this port.

Important! Before you switch your global setting to Deny by Default, make sure you have at least one port that does not rely on the global setting and allows at least one host. If you inadvertently lock all hosts out of the server, you can correct the problem by manually updating the appropriate configuration file, as shown below.

Before updating these configuration files, be sure to shut down the SAP BC server.

To reset the global setting, update the `watt.server.hostAllow` parameter in the `server.cnf` file. For example:

```
watt.server.hostAllow=132.906.19.22
```

To reset an individual port, update the following parameter in the `config/listeners.cnf` file in the package for which the port is defined:

```
<array name="hostAllow" type="value" depth="1">
  <value>132.906.19.22</value>
</array>
```

To allow inbound requests from only specified hosts

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Locate the port in the Port List and click Edit in the IP access field.
CHAPTER 7 Configuring the Server

4 Click **Change IP Access Mode to Deny by Default**.

   The server changes the access mode and displays a screen from which you can add hosts to the Allow List. Notice that the server has already included the host name and IP address of the machine from which you are using the Server Administrator so that you are not locked out of the server.

5 **Click Add Hosts to Allow List**.

6 **Specify the host names or IP addresses of clients from which the server is to accept inbound requests** (e.g., workstation5.sap.com). **Separate your entries with commas, for example:** *.allowme.com, *.allowme2.com.

   You can use the following pattern-matching characters to identify several clients with similar host names or IP addresses.

<table>
<thead>
<tr>
<th>Char</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches any number of characters</td>
<td>r*.sap.com</td>
</tr>
<tr>
<td>?</td>
<td>Matches any single character</td>
<td>workstation?.sap.com</td>
</tr>
</tbody>
</table>

7 **Click Add Hosts**.

**Deny Inbound Requests from Specified Hosts (Allow All Others)**

The following procedure describes how to change the IP access settings for an individual port to Allow by Default and deny some hosts.

With this setting in effect, the server allows most hosts and denies some through this port.

---

**To deny inbound requests from only specified hosts**

1 **Open the Server Administrator if it is not already open.**

2 **In the Security menu of the navigation area, click Ports.**

3 **Locate the port in the Port List and click Edit in the IP access field.**

4 **Click Change IP Access Mode to Allow by Default.**

5 **Click Add Hosts to Deny List.**

6 **Specify the host names or IP addresses of hosts from which the server is to deny inbound requests** (e.g., workstation5.sap.com). **Separate your entries with commas, for example:** *.denyme.com, *.denyme2.com.

   You can use the following pattern-matching characters to identify several clients with similar host names or IP addresses.
Specifying a Third-Party Proxy Server for Outbound Requests

When the SAP BC server executes a request against a remote server (for example a remote invoke against another SAP BC server), or a Load Document service, it issues an HTTP or HTTPS request to the specified target server. If your SAP BC server sits behind a firewall and must route these requests through a third party proxy server, you must define the address of the proxy server using the Server Administrator.

If you want to isolate your SAP BC server behind an internal firewall, you can provide greater security by running a special reverse invoke SAP BC server in your DMZ and disallowing all inbound connections to your internal server. For information about setting up a reverse invoke SAP BC server see “Protecting Your Internal SAP BC Server with Reverse Invoke” on page 147.

The SAP BC server allows you to define different proxies to use for outbound HTTP/HTTPS requests and FTP.

To specify a Proxy Server

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Proxy Servers.
3. Click Edit Proxy Settings.
4. Set the Proxy (HTTP) and Secure Proxy (HTTPS) and FTP Proxy (FTP) parameters as follows. (If you use a proxy server for only one request type, complete the parameters for that type, and leave the parameters for the other types empty.)

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Host</td>
<td>The name of the proxy server.</td>
</tr>
<tr>
<td>Proxy Port</td>
<td>The port on which the proxy server listens for HTTP and/or HTTPS requests.</td>
</tr>
<tr>
<td>Proxy User</td>
<td>The user name the SAP BC server must use when accessing this proxy server (if one is required).</td>
</tr>
<tr>
<td>Proxy Password</td>
<td>The password the SAP BC server must use to access this proxy server (if one is required).</td>
</tr>
</tbody>
</table>

Specifying a Third-Party Proxy Server for Outbound Requests

<table>
<thead>
<tr>
<th>Char</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Matches any number of characters</td>
<td>r*.sap.com</td>
</tr>
<tr>
<td>?</td>
<td>Matches any single character</td>
<td>workstation?.sap.com</td>
</tr>
</tbody>
</table>
For this parameter... Specify...
Proxy Type (for FTP only)
Type of FTP proxy server to connect to using the pub:client.ftp built-in service. The proxy server always requires the FTP server name, FTP user name, and the FTP user password. The method you use to send this information to the FTP proxy server depends on the type of proxy server you have. The SAP BC server supports the following proxy server types:

0. No proxy
Do not use an FTP proxy server. This is the default.

1. user@host no proxy auth.
Connect to the proxy server, but do not log into it.
Then send the following:

```
USER ftp_user@real_ftp_hostname
PASS ftp_password
```

2. user@host proxy auth
Connect to the proxy server, and log into it with:

```
USER proxy_user
PASS proxy_password
```
Then send the following:

```
USER ftp_user@real_hostname
PASS ftp_password
```

3. site command
Connect to the proxy server and log into it with:

```
USER proxy_user
PASS proxy_password
```
Then send the following:

```
SITE real_ftp_hostname
USER ftp_user
PASS ftp_password
```

4. open command
Connect to the proxy server and log into with:

```
USER proxy_user
PASS proxy_password
```
Then send the following:

```
OPEN real_ftp_hostname
USER ftp_user
PASS ftp_password
```

5. real_user@proxy_user@real_host
Connect to the proxy server and log into it, then send the following:

```
USER ftp_user@proxy_user@real_hostname
PASS ftp_password@proxy_password
```
Setting Up Aliases for Remote SAP BC servers

5 Click Save Changes.

Bypassing a Proxy Server

If you are using a proxy server for outbound HTTP and/or HTTPS, you can optionally route selected requests directly to their targets, bypassing the proxy.

To do this, you use the Server Administrator to define the list of domains to which you want the SAP BC server to issue requests directly.

To bypass a proxy server

1 Open the Server Administrator if it is not already open.
2 In the Settings menu of the navigation area, click Proxy Servers.
3 Click Edit Proxy Settings.
4 In the Bypass Proxy List, type the fully-qualified host and domain name of each server to which you want the SAP BC server to issue requests directly. Type the host name and the domain name exactly as it will appear in the URLs that the server uses. To enter multiple names, separate each with commas.

You can use the asterisk (*) to identify several servers with similar names. The asterisk matches any number of characters. For example, if you wanted to bypass requests made to localhost, www.yahoo.com, home.microsoft.com, and all hosts whose names begin with NYC, you would type:

localhost,www.yahoo.com,home.microsoft.com, NYC.*

5 Click Save Changes.

Setting Up Aliases for Remote SAP BC servers

You can set up aliases for remote servers. Communication through the alias is optimized, making transactions with the remote server faster. In addition, using an alias is more convenient because it saves you from specifying connection information each time you communicate with the remote server.

Use a remote alias when:

- **Invoking services on other SAP BC servers.** After you establish aliases, you can use the pub.remote:invoke and pub.remote.gd:* services to invoke services on remote servers by identifying the remote servers by their aliases.

- **Presenting multiple client certificates.** The SAP BC server can present a single client certificate to all servers or it can present different client certificates to different SSL servers. In addition, the SAP BC server can present certificates provided for this purpose by other organizations. Setting up remote aliases for these SSL servers makes
it easier to present different certificates to them. See “Presenting Multiple Client Certificates” on page 113 for more information.

- **Performing package replication.** For a subscriber to set up a subscription with a publisher or pull a package from the publisher, you must define the publishing server as a remote server to the subscriber. The alias tells the subscribing server how to connect to the publishing server to set up the subscription or pull the package. See “The Subscribing Server” on page 217 for more information.

- **Using a reverse invoke SAP BC server.** If you use a reverse invoke SAP BC server in your DMZ to protect your internal SAP BC server, you must define the reverse invoke SAP BC server as a remote server to your internal server.

  The definition for an alias contains the connection information the server requires to connect to a remote server. It identifies the host name or IP address of the remote server and indicates whether the server should use an HTTP or HTTPS connection to connect to the remote server.

  The alias also identifies a user name and password that the server supplies to the remote server. The remote server uses the user name and password to authenticate the client and to determine if the client is authorized to execute the requested service.

  In effect, the alias grants access to a remote service by allowing the user to impersonate an authorized user on the remote server. Therefore, to prevent unauthorized users from accessing services on remote servers, the alias also contains access control information. You specify an ACL that protects the use of the alias. If a client that is authorized to use the alias makes a request, the server will request the service on the remote server. If a client that is not authorized to use the alias makes a request, the server rejects the request and does not invoke the service on the remote server.
Adding an Alias

Use the following procedure to add an alias for a remote SAP BC server.

To add an alias for a remote server

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Remote Servers.
3. Click Create Remote Server Alias.
4. Set the Remote Server Alias Properties as follows:

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alias</strong></td>
<td>Name that you want to use for the alias. You can give the remote server any alias name you want. There is no restriction on the characters you can specify or a limit to the length of the alias.</td>
</tr>
<tr>
<td><strong>Host Name or IP Address</strong></td>
<td>Host name or IP address of the remote server for which you are creating an alias (e.g., workstation5.sap.com).</td>
</tr>
<tr>
<td><strong>Port Number</strong></td>
<td>Port number on which the remote server listens for incoming requests from your server (e.g., 5555).</td>
</tr>
<tr>
<td><strong>User Name</strong></td>
<td>User name for a user account on the remote server. When you invoke a service using this alias, the remote server uses this user account for authentication and access control. Specify a user name that has access to the services you want to invoke on the remote server.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Password identified in the user account for User Name.</td>
</tr>
<tr>
<td><strong>Access Control</strong></td>
<td>ACL that governs which user groups on your server can use this alias for the remote server. Select an ACL from the drop down list. By default, only members of groups governed by the Internal ACL can use this alias.</td>
</tr>
<tr>
<td><strong>Idle Timeout</strong> (in minutes) <strong>0=none</strong></td>
<td>Number of minutes that the server maintains an idle connection to the remote server. If you specify 0, there is no timeout limit; the server maintains the connection until your local server is shut down or the sessions that are using the alias expire.</td>
</tr>
</tbody>
</table>
CHAPTER 7 Configuring the Server

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use SSL</td>
<td>Whether you want your server to connect to the remote server using Secure Sockets Layer (SSL). If you want to use SSL, select yes; otherwise, select no.</td>
</tr>
</tbody>
</table>

**Important!** If you select yes, the remote server must be configured to listen for incoming HTTPS requests.

<table>
<thead>
<tr>
<th>Private Key</th>
<th>Specifies the name of the file containing the RSA private key associated with this server’s digital certificate.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificates</td>
<td>Certificate you want to present to this remote server. You must specify the entire certificate chain using this format. Subject, Intermediate1, Intermediate2, …, Root</td>
</tr>
<tr>
<td></td>
<td>Subject is the local SAP BC server’s certificate, that is, the certificate you want to present to the remote server.</td>
</tr>
<tr>
<td></td>
<td>Intermediate1 and Intermediate2 are optional intermediate certificates in the certificate chain.</td>
</tr>
<tr>
<td></td>
<td>Root is the root CA certificate of the certificate chain.</td>
</tr>
<tr>
<td></td>
<td>Specify the path and file name for each element of the chain. For example: config\cert.der, config\intermedcert.der, config\cacert.der</td>
</tr>
</tbody>
</table>

| Retry Server | Host name or IP address (e.g., workstation6.sap.com) of a remote server you want your local Integration Server to connect to if the primary remote server is unavailable. Specify this parameter only if you are using the remote:invoke and pub.remote.gd:* services to invoke services on a remote server that is part of a cluster. |

5. Click **Save Changes**.

**Testing the Connection to a Remote Server**

After you add an alias, you can test the connection to the remote server to ensure that the host name (or IP address) and port number specified for the alias identifies an SAP BC server that is currently running. Use the following procedure to test the connection:

**To test the connection to a remote server**

1. Open the Server Administrator if it is not already open.
2. In the **Settings** menu of the navigation area, click **Remote Servers**.
3. Click the icon in the Test column for the alias you want to test.

The server displays a status line that indicates whether the connection is successful or not. The status line is displayed above the list of existing aliases.

### Editing an Alias

If you need to update the information for an alias, you can edit it to make your changes. Use the following procedure to edit an alias.

**To edit an alias for a remote server**

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Remote Servers.
3. Locate the alias you want to edit and click on the alias name.
4. Update the information for the alias.
5. Click Save Changes.

### Deleting an Alias

If you no longer need an alias for a remote server, you can delete it. Use the following procedure to delete an alias.

**To delete an alias for a remote server**

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Remote Servers.
3. Locate the alias you want to delete and click the icon in the Delete field. The server displays a dialog box that prompts you to verify your action. Click OK to verify that you want to delete the alias.

### Working with Log Files

The SAP BC server uses log files to maintain an audit-trail of services it executes and errors that occur. These files contain a historical trace you can use to diagnose application failures, analyze server performance, and trace server-related events.
What Kind of Log Files Does the Server Maintain?

The server maintains the following log files. Each file contains information for a particular purpose or process.

<table>
<thead>
<tr>
<th>Log</th>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit</td>
<td>audit.log</td>
<td>Maintains an audit trail for all services that the server executes. For each service, the server adds two records that contain information about the service. It adds one when it begins execution; this record includes the timestamp of when execution began. It adds the second record when the service terminates; this record includes the timestamp of when the service terminated. You can use this log to determine all services requested from the server and which client requested each service.</td>
</tr>
<tr>
<td>Error</td>
<td>error.log</td>
<td>Maintains information about all errors and exceptions that have occurred in the server system. You can use this log for debugging problems with services.</td>
</tr>
<tr>
<td>Server</td>
<td>server.log</td>
<td>Maintains information about the version of the SAP BC server that is running and the operations that the server has performed. You can use this information for general debugging purposes.</td>
</tr>
<tr>
<td>Session</td>
<td>session.log</td>
<td>Maintains information about all sessions in the server system. The SAP BC server creates a session when a developer connects to the server from the Developer or a server client connects to the server to execute services. Each record in the Session Log includes a timestamp indicating when a user established a connection with the SAP BC server.</td>
</tr>
<tr>
<td>Transaction</td>
<td>txin.log</td>
<td>txout.log</td>
</tr>
</tbody>
</table>

For a sample of the contents of each log, see “Server Log Files and JVM Information” on page 293.
Where Are the Log Files?

By default, the SAP BC server maintains the log files in <sapbc>\server\log. However, using the Developer you can redirect audit and exception data to another log, such as the Event Log on a Windows NT/2000 system, or to a database system. See the SAP BC Developer Guide for more information about this topic.

How Much Information Is Recorded?

Most log files contain brief, standard descriptions of key events. The server log (server.log) is an exception. Its contents can be configured.

For the server.log file, you can specify the type of events you want the server to record and the amount of detail you want to see. This flexibility is useful for troubleshooting. For example, you might temporarily increase the level of detail written to this log to help uncover the cause of a server error or performance problem, and return to a lower level once the problem is resolved.

To configure the contents of the server.log file

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Logging.
3. Click Edit Logging Settings.
4. In the Level of Logging field, type a value from 1 to 10 to specify the amount of information you want the server to record for the selected facilities:

   Specify: To record:
   1. Critical messages only
   2. Error and critical messages
   3. Warning, error, and critical messages
   4. Debug, warning, error, and critical messages
   5-10. Informational, debug, warning, error, and critical messages

   Note: Recording more information consumes more system resources.

5. For Facilities, select the type of information you want the server to include in the log.
6. For Email Notification, specify the SMTP server you want the SAP BC server to use and where the server is to direct the different types of error messages.
When critical errors occur, the server automatically issues a message to a specified e-mail address. To ensure that this information reaches the appropriate person, you must specify where you want the server to deliver these messages.

For this parameter... Specify...

SMTP Server Domain name (e.g., purple.sap.com) or IP address (e.g., 132.906.19.22) of the SMTP server that you want the SAP BC server to use.

Internal Email E-mail address of the user you want notified when an internal server error occurs. Typically, this is the e-mail address of the administrator for the SAP BC server.

The server issues this type of error when a critical problem occurs with the server program or a service. For example, when the server encounters a null pointer exception or other exception it cannot handle. Most sites route this type of message to the server administrator or an appropriate alternate.

Service Email The e-mail address of the person the server will notify when an output binding fails. Typically, this is the e-mail address for someone that is familiar with the services on your server, such as a lead developer.

The server issues this type of error when an output binding fails—i.e. the structure of a requested document does not match the expectations of the service requesting it. In these cases, your developers may need to update the service to match the document's new structure. Most sites route this type of message to someone in their development organization.

Transaction Email The e-mail address of the person the server will notify when guaranteed delivery fails.

The server issues this type of error when the server's guaranteed delivery capabilities become disabled. Route this type of message to a server administrator. When the administrator receives an e-mail notification of an error, the administrator should correct the problem; then reinitialize guaranteed delivery capabilities.

Click Save Changes.
Specifying the Date Format to Use in Log Files

You can control how the date is displayed in log files. The default date format is the hexadecimal representation of the system time; however, you can configure a more readable date format.

Configuring the Date Format

You use the following procedure to configure an alternate format.

To configure the date format in log files

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click **Logging**.
3. Click **Edit Logging Settings**.
4. In the **Log Date Format** field, set the format to one that is supported by the Java class `java.text.SimpleDateFormat`, for example, `yyyy-MM-dd hh:mm:ss:SSSS`.
5. Click **Save Changes**.

How Long Does the Server Maintain Its Logs?

The server regularly closes log files and opens new ones. This process is referred to as “rotating the log files.” You can configure:

- **When the first rotation of the log files occurs.** You specify how long the server is to collect log information from the time the server is started to the time it rotates the log files for the first time.

- **How long subsequent rotation intervals are.** You specify an interval of time that indicates how often (after the first log file rotation) the server continues to rotate the log files.

- **How the server stores the closed log files.** You specify whether the server is to maintain all the log files in a single directory (rename) or create a new directory each time it rotates the log files (move).

To have the server store all log files in a single directory, use the **rename** setting. This setting indicates that the server is to rename the log files to include the date and time the log files were closed.
CHAPTER 7 Configuring the Server

Renamed log files

- SAP BC
- Server
  - log
    - audit.20010318.000024.log
    - errors.20010318.000026.log
    - server.20010318.000026.log
    - sessions.20010318.000032.log
    - stats.20010318.000034.log
    - txin.20010318.000036.log
    - txout.20010318.000036.log

To have the server create a new directory for each set of closed log files, use the **move** setting. This setting indicates that the server is to create a new directory and move the closed log files to the new directory. The server gives the new directory a name that identifies when it closed the log file.

Moved log files

- SAP BC
- Server
  - logs
    - 20010318.000024
    - 20010318.000012
      - audit.log
      - error.log
      - server.log
      - sessions.log
      - stats.log
      - bin.log
      - txout.log

By default, the server maintains log files that hold information for a single day. The first rotation occurs at 00:00 (midnight). The rotation interval is 24 hours, so it continues to close the log files every day at 00:00 (midnight). When the server closes the previous day’s log files, it stores them in a directory that contains just that day’s log files.

**Configuring Log File Rotation**

To configure log file rotation, you must update several properties in the server.cnf file. You can update this file by using the **Settings>Extended** screen of the Server Administrator, as described below.
To configure log file rotation

1. Open the Server Administrator if it is not already open.

2. In the Settings menu of the navigation area, click Extended.

3. The server displays a screen that lists configuration parameters specified in the server.cnf file.

4. To add or change a parameter setting, click Edit Extended Settings and add or change the following parameters:

<table>
<thead>
<tr>
<th>Set this parameter:</th>
<th>To specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>watt.server.logRotateFirstInterval</td>
<td>The number of milliseconds after the server is started when the server rotates the log files for the first time.</td>
</tr>
<tr>
<td>watt.server.logRotateInterval</td>
<td>The number of milliseconds after the first rotation when the server will continue to rotate the log files.</td>
</tr>
</tbody>
</table>

5. To set how the rotation will occur, locate the watt.server.logRotate parameter. If this parameter does not exist, add it. Set the watt.server.logRotate parameter to one of the following:

<table>
<thead>
<tr>
<th>Set this value...</th>
<th>To...</th>
</tr>
</thead>
<tbody>
<tr>
<td>rename</td>
<td>Rename the log files to include the date and time the server closes the log files and keep them in the same directory.</td>
</tr>
<tr>
<td>move</td>
<td>Create a new directory with a name that represents when the server closes the log files and move the log files to the new directory. This is the default.</td>
</tr>
</tbody>
</table>

6. Click Save Changes.

7. Restart the server for the changes to take effect.

   1. In the upper left corner of any Server Administrator screen, click Shutdown and Restart.

   2. Select whether you want the server to wait before restarting or to restart immediately.

   3. Click Restart.
CHAPTER 7 Configuring the Server

Viewing the Log Files

Use the following procedure to view the Session Log, Server Log, or Errors Log.

To view the log files

1. Open the Server Administrator if it is not already open.
2. In the Log menu, click the type of log you want to view.

<table>
<thead>
<tr>
<th>Click</th>
<th>To view...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit</td>
<td>An audit trail for all services that the server executes.</td>
</tr>
<tr>
<td>Error</td>
<td>Information about all errors and exceptions that have occurred in the server system.</td>
</tr>
<tr>
<td>Server</td>
<td>Information about the version of the SAP BC server that is running and the operations that the server has performed.</td>
</tr>
<tr>
<td>Session</td>
<td>Information about all sessions in the server system. The SAP BC server creates a session when a developer connects to the server from the Developer or a server client connects to the server to execute a service.</td>
</tr>
<tr>
<td>Guaranteed Delivery</td>
<td>Information about all critical operations the server performs for guaranteed delivery transactions.</td>
</tr>
<tr>
<td></td>
<td>The server maintains one log for inbound guaranteed delivery transactions (txin.log) and a separate log for outbound transactions (txout.log).</td>
</tr>
</tbody>
</table>

Setting the Session Timeout Limit

When a remote client connects to the SAP BC server, the server starts a session for that client. That session remains active until the client application specifically issues a disconnect instruction to the server (which forces an immediate termination) or the session “times-out” due to inactivity, whichever comes first.

If a session is idle (inactive) for a long period of time, it usually means that the client is no longer active or the connection between client and the server has been lost. The server constantly monitors for inactive sessions, and terminates sessions that are idle for more than the allowed period of time. (If the server did not take steps to clear out such sessions, they would remain active indefinitely, wasting valuable server resources.)

You use the Session Timeout parameter to specify the length of time you will allow an idle session to remain active (in other words, how long you want the server to wait before terminating an idle session). To set the Session Timeout parameter appropriately, you must be familiar with the clients that use your server.
If your clients are all Java programs, you can usually reduce the timeout value to six or seven minutes. You may need to experiment with this setting to find the appropriate value for your site. By default, the server uses a timeout limit of 10 minutes. This is an appropriate value for most sites. However, you may have to increase this value if your clients normally have lengthy delays (greater than 10 minutes) between successive requests.

To set the Session Timeout Limit

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Resources.
3. Click Edit Resource Settings.
4. In the Session Timeout field, type the number of minutes you want the server to wait before terminating an idle session.
5. Click Save Changes.
Managing Server Thread Pools

To better tune your server’s performance, you can configure the minimum and maximum number of threads the server uses to run services.

To manage server thread pools

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Resources.
3. Click Edit Resource Settings.
4. Update the server thread pool settings, as follows:

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Thread Pool</td>
<td><strong>Maximum Threads</strong>&lt;br&gt;The maximum number of threads the server maintains in the thread pool that it uses to run services. If this maximum number is reached, the server waits until services complete and return threads to the pool before running more services. The default is 75.</td>
</tr>
<tr>
<td></td>
<td><strong>Minimum Threads</strong>&lt;br&gt;The minimum number of threads the server maintains in the thread pool that it uses to run services. When the server starts, the thread pool initially contains this minimum number of threads. The server adds threads to the pool as needed until it reaches the maximum allowed, which is specified in the <strong>Maximum Threads</strong> field. The default is 10.</td>
</tr>
<tr>
<td>Scheduler Thread Pool</td>
<td><strong>Minimum Threads</strong>&lt;br&gt;The maximum number of threads in the thread pool that the server uses for the scheduler. This number controls the maximum number of jobs that can run concurrently on the server. If this maximum number is reached, the server waits until jobs complete and return threads to the pool before running more jobs. The default is 10.</td>
</tr>
</tbody>
</table>

Configuring Outbound HTTP Settings

The Outbound HTTP parameters control how the server presents and processes outbound HTTP and HTTPS requests (i.e., requests that the SAP BC server issues on behalf of a client). The parameters control behavior such as how long the server waits for a response, how many times it retries a failed request, and so forth.
A developer can override some of the server’s outbound HTTP setting defaults at run time. The following descriptions note when this is allowed.

**User Agent**
The User Agent parameter specifies the value the server uses in the HTTP User Agent request header that it sends when requesting a Web document. The User Agent header tells a Web server what type of browser is making the request. In the case of the SAP BC server, the User Agent header indicates the type of browser that the SAP BC server appears to be to the Web server. Some Web servers examine this header to determine a client’s capabilities so they can tailor their responses accordingly.

When you install the SAP BC server, the User Agent parameter is set to Mozilla/4.0 [en] (WinNT; I). You can change this value as you need; however, the value you set should satisfy the majority of services that your server executes.

Be sure your developers know the User Agent value your server uses. If their applications require a different User Agent, they can override the server’s default at run time by including an HTTP User Agent header with their request.

**Max Redirects**
The Max Redirects parameter specifies the number of times that the SAP BC server allows a request to be redirected (i.e., automatically sent to another URL by the target server). If a request exceeds the specified number of redirections, the SAP BC server immediately returns an I/O exception to the client.

When you install the SAP BC server, Max Redirects is set to five. You will need to increase this value if the targets that you access typically redirect their requests more than this.

**Timeout**
The Timeout parameter specifies the length of time the server waits for a response from a target server. If the SAP BC server does not receive a response in the allotted time, it retries the request up to the number of times specified by the Retries parameter. When it exceeds the number of allowed retries, it returns an exception.

When you install the SAP BC server, the Timeout parameter is set to three minutes. For most sites this is a reasonable setting; however, you may need to adjust this value if you work with targets that have longer response times than this (e.g., large commercial Web sites or databases during peak periods).

**Retries**
The Retries parameter specifies the number of times the server reissues a request that has timed out (i.e., one from which it did not receive a response within the time period specified by the Timeout parameter).

When you install the SAP BC server, Retries is set to zero. This means that the server automatically returns an exception if it does not get a response within the allotted time. Set Retries to a value greater than zero if you want the server to retry (reissue) timed-out requests. The server will retry the request the number of times you specify.

Be sure your developers know what Retries value your server uses. If they need to use a different value, they can explicitly assign a Retries value to their service.
 CHAPTER 7 Configuring the Server

Specifying Outbound HTTP Settings

Use the following procedure to set the Outbound HTTP Settings Defaults.

To set the Outbound HTTP Settings Defaults

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Resources.
3. Click Edit Resource Settings.
4. Set the Outbound HTTP Settings Defaults as follows:

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Agent</td>
<td>The string that you want the server to supply in the HTTP User Agent header if the client does not specify a value. Type the string exactly as you want it to appear in the HTTP header, including spaces, symbols, and punctuation.</td>
</tr>
<tr>
<td>Max Redirects</td>
<td>An integer that indicates the number of times to allow a request to be redirected before the server returns an I/O exception to the client.</td>
</tr>
<tr>
<td>Timeout</td>
<td>An integer that indicates the number of seconds the server waits for a response from the target server before retrying the service or returning a timeout error to the client.</td>
</tr>
<tr>
<td>Retries</td>
<td>An integer that indicates the number of times the server retries a service that has timed out before returning an exception to the client.</td>
</tr>
</tbody>
</table>

5. Click Save Changes.

Working with Extended Configuration Settings

There may be times when you want to view special server property settings. These properties are specified in the server.cnf file, however you can view them and edit them using the Server Administrator. Typically you do not need to change these settings unless directed to by SAP documentation or Technical Support.

Important! Typically, you will use the Server Administrator to set properties in the server.cnf file, but there may be times when you need to edit the file directly with a text editor. Before updating this file directly, be sure to shut down the SAP BC server.
To view and edit extended configuration settings

1. Open the Server Administrator if it is not already open.

2. In the **Settings** menu of the navigation area, click **Extended**.
   
The server displays a screen that lists configuration properties specified in the server.cnf file.

3. By default, no properties are shown. If the properties you want to view are shown, skip this step. To select properties to be displayed, click **Show and Hide Keys**.
   
The server displays a list of all properties included in the server.cnf file (their values are not shown.) Check the box to the left of each property you want the server to display and click **Save Changes**. The server displays the **Extended Settings** screen again, this time with the selected properties and their values displayed.

4. To add, delete, or change a property setting, click **Edit Extended Settings** and type your changes.

   **Important!** Remember: Any change you make here will be reflected in the server.cnf file.

5. Click **Save Changes**.

   Any properties you added will automatically display a check mark ✓ in the Show and Hide Keys list and will be displayed, with their values, in the Extended Settings list.

6. Restart the server for the changes to take effect.
   1. In the upper left corner of any Server Administrator screen, click **Shutdown and Restart**.
   2. Select whether you want the server to wait before restarting or to restart immediately.
   3. Click **Restart**.
Caching Service Results

- What Is Caching? ................................................................. 104
- When Are Cached Results Returned? .............................. 104
- Resetting the Cache ......................................................... 105
- Viewing Service Statistics ............................................. 106
CHAPTER 8  Caching Service Results

What Is Caching?

Caching is an optimization feature that can improve the performance of services. You indicate the services for which you want to use caching from the SAP BC Developer. When you enable caching for a service, the server saves the results from invoking the service in a local cache for the period of time that you specify. When the server receives subsequent requests for a service with the same set of input values, it returns the cached result to the client rather than invoking the service again.

Caching can significantly improve response time of services. For example, services that retrieve information from busy data sources such as high-traffic commercial Web servers could benefit from caching. The server can cache the results for all types of services—flows, Java services, and C/C++ services.

The goal for caching is to strike the right balance between data currency and memory usage. To gauge the effectiveness of your cache, you can monitor its performance by viewing service statistics from the Server Administrator and adjust your caching values accordingly.

You set the controls for caching a service from the Developer. See the SAP BC Developer Guide for more information.

When Are Cached Results Returned?

When you enable caching for a service in the SAP BC Developer, the SAP BC Server handles the cached results differently, depending on whether the service has input parameters. It is recommended that a cached service has input parameters.

- **Service with input parameters.** When a cached service has input parameters, at run-time the SAP BC Server scopes the pipeline down to only the input parameters of the service. The scoped-down inputs are compared to the previously stored copy of inputs. If they exist and match, the cached results from the previous service invocation are used.
Pipeline Inputs Are Compared to the Cached Copy at Run-Time

At run-time, the BC Server scopes the pipeline down to only the input parameters of the service... and compares them to the cached copy. If they match in name, dimension, and value, then the cached output from the previous service invocation is used.

**Service without input parameters.** When a cached service does not have input parameters (for example, a date/time service) and previous results do not exist in the cache, at run-time the SAP BC Server executes the service and stores the results. When the service executes again, the cached copy is used. In other words, the pipeline is not used; you will always receive cached results until the cache expires.

When variables that are defined in the cached service’s input parameters are missing from the pipeline, the SAP BC Server extracts any variables that exist in the pipeline that match the cached service’s input parameters. If no required variables exist in the pipeline, the SAP BC Server ignores the pipeline and essentially considers that no input parameters were provided.

**Important!** If you edit a cached service by changing the inputs (not the pipeline), you must click **Reset Cache** on the **Settings** tab in the Developer, or in **Service Usage** in the Server Administrator. If you do not reset the Server cache, the old cached input will be used at run-time.

**Resetting the Cache**

You can reset the cache for all services or you can reset the cache for a specific service. When the server resets the cache, it removes all cached service results from memory.
CHAPTER 8 Caching Service Results

To reset the cache for all services
1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Service Usage.
3. Click in the Reset Server Cache.

To reset the cache for a specific service
1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click the name of the package that contains the service for which you want to reset the cache.
4. Click Browse Services in packagename.
5. Click the name of the service for which you want to reset the cache.
6. Click Reset Service Cache.

Viewing Service Statistics

Use the following procedure to monitor the performance of your cache.

To monitor the performance of your cache
1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Service Usage.

   The Service Usage screen displays the current results of your cache control settings for each cache-controlled service.
CHAPTER 9

Managing Server Security

- Overview of Security ................................................................. 108
- Controlling Who Can Configure and Manage the Server ............ 108
- Controlling Who Can Create, Modify, and Delete Services ........ 110
- Securing Communications with the Server .................................. 111
- Enabling and Disabling Users .................................................... 122
- Controlling Access to Services and Files with ACLs .................. 123
- Controlling Server SSL Security Level by Port ......................... 132
- Controlling Access to Services by Port ...................................... 132
- Authenticating Clients .............................................................. 135
- Protecting Your Internal SAP BC Server with Reverse Invoke .... 147
- Responding to NT Challenge/Response Authentication ............... 168
- Using Tokens to Counter XSRF Attacks ..................................... 170
Overview of Security

To secure access to your server and the data that resides on the server, you can:

- **Control who can configure and manage the server.** You can restrict access to the Server Administrator.
- **Control who can use the SAP BC Developer to connect to the server.** You can specify who is authorized to create, edit, and delete the services that reside on the server.
- **Secure the transmission of data between server clients and the server.** You can configure a port for SSL communications.
- **Digitally sign documents and verify digital signatures.** You can code your services to invoke a built-in service (pub.security.pkcs7:sign) to digitally sign a document. Similarly, you can invoke another built-in service (pub.security.pkcs7:verify) to ensure a document has not been altered since it was digitally signed. For more information about the built-in services, refer to the *SAP BC Built-In Services Guide.*
- **Control access to services and the files that reside on the server.** You can create Access Control Lists (ACLs) that control access to individual services and files, and you can restrict access to services based on the port on which an incoming request is received.
- **Specify how you want the server to authenticate clients.** This allows you to authenticate a client based on client certificates or user name/password authentication.
- **Use different certificates for different connections.** This allows you to specify different certificates (and associated private keys) depending on the host with which the server is communicating.
- **Isolate your SAP BC server behind an inner firewall.** You can use the reverse invoke feature to place a reverse invoke SAP BC Server in your DMZ to intercept requests from external clients before passing the requests to your internal server. See “Protecting Your Internal SAP BC Server with Reverse Invoke” on page 147 for more information.

In addition, the SAP BC Server also supports NT Challenge/Response authentication when the server acts as a Web client to access information from a server. Microsoft’s Internet Information Server is an example of a server that supports Microsoft NT’s built-in authentication mechanism.

**Controlling Who Can Configure and Manage the Server**

Use the Server Administrator to configure and manage the server. Before the server allows access to the Server Administrator, it ensures the user has administrator privileges.

A user has administrator privileges if he or she belongs to the Administrators group or to any other group added to the Allow List of the Administrators ACL. To determine if a user has administrator privileges, the server authenticates the user to obtain his or her user name. (For information about how the server determines the user name, see...
“Authenticating Clients” on page 135.) After determining the user name, the server determines if the user belongs to a group that is allowed and does not belong to any group that is denied access by the Administrators ACL. If so, the server allows access to the Server Administrator.

To grant administrator privileges to a user, you must assign that user to the Administrators group or to a group you have added to the Allow list of the Administrators ACL. In addition, you must make sure the user is not a member of a group that is denied access by the Administrators ACL.

**Important!** The user to whom you want to grant administrative privileges must already have a user account on the SAP BC Server. If the user does not already have a user account, create one for him or her before you perform the following steps.

---

**To grant administrative privileges to a user**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **Users and Groups**.
   
   The server displays the following screen.

   ![Server Administrator Screen](image)

   The **Groups** area of the screen (on the right) contains two lists. **Users in this Group** is a list of users currently in the selected group. **Remaining Users** is a list of users not currently in the selected group.

3. In the **Groups** area of the screen, select the Administrator group from the pull-down list of groups.
4 In the Remaining Users list, select (highlight) the user or users to whom you want to grant administrator privileges.

To select additional users without deselecting currently selected users, press the CTRL key while you click on the users you want to select. To deselect a user, press the CTRL key while you click the currently selected entry.

5 After you have selected all the users you want to add to the group, click . The server moves the selected users to the Users in this Group list.

6 Click Save Changes.

Note: Alternatively, you can create a new group such as LocalAdministrators, add that group to the Administrators ACL’s allow list, and add the user to that group.

Controlling Who Can Create, Modify, and Delete Services

A developer can use the SAP BC Developer to create, modify, and delete services that reside on the server. Before the server allows a connection from the Developer, it ensures the user has developer privileges.

A user has developer privileges if he or she belongs to the Developers group or to any other group added to the Allow List of the Developers ACL. To determine if a user has developer privileges, the server authenticates the user to obtain his or her user name. (For information about how the server determines the user name, see “Authenticating Clients” on page 135.) After determining the user name, the server determines if the user belongs to a group that is allowed and does not belong to any group that is denied access by the Developers ACL. If so, the server allows the connection between the Developer and the server to be established.

To grant developer privileges to a user, you must assign that user to the Developers group or to a group you have added to the Allow list of the Developers ACL. In addition, you must make sure the user is not a member of a group that is denied access by the Developer ACL.

Important! The user to whom you want to grant developer privileges must already have a user account on the SAP BC Server. If the user does not already have a user account, create one for him or her before you perform the following steps.

To grant developer privileges to a user

1 Open the Server Administrator if it is not already open.

2 In the Security menu of the navigation, area click Users and Groups.
Securing Communications with the Server

The server displays the following screen.

The **Groups** area of the screen (on the right) contains two lists. **Users in this Group** is a list of users currently in the selected group. **Remaining Users** is a list of users *not* currently in the selected group.

3. In the **Groups** area of the screen, select the Developer group from the pull-down list of groups.

4. In the **Remaining Users** list, select (highlight) the user or users to whom you want to grant developer privileges.

   To select additional users without deselecting currently selected users, press the CTRL key while you click on the users you want to select. To deselect a user, press the CTRL key while you click the currently selected entry.

5. After you have selected all the users you want to add to the group, click **→**. The server moves the selected users to the **Users Currently in this Group** list.

6. Click **Save Changes**.

**Note:** Alternatively, you can create a new group such as LocalDevelopers, add that group to the Developers ACL’s allow list, and add the user to that group.

Securing Communications with the Server

An administrator can configure the SAP BC Server to use Secure Sockets Layer (SSL) to provide secure communications with the server. Use SSL to ensure that data is transmitted privately and that the content of the data is not altered during transit.
Background About SSL

In an SSL transaction, there is an SSL client and an SSL server. The SSL client initiates an SSL transaction. At the beginning of an SSL transaction, the client and server perform what is called an SSL handshake:

1. The server sends its digital certificate to the client. The client uses this certificate to authenticate the server, which assures the client that it is communicating with the organization that the certificate identifies.

2. Optionally, the server can request a client certificate from the client. The server can use the client certificate to authenticate the client.

3. The client and the server negotiate how they will securely transmit data.

SSL and the SAP BC Server

Depending on the situation, the SAP BC Server can be either a client or a server in an SSL transaction.

When the SAP BC Server Is an SSL Server

When a server client communicates via HTTPS with the SAP BC Server, the server client is the SSL client and the SAP BC Server is the SSL server.

If it is configured to do so, the SAP BC Server will ask the client for a certificate. See “Authenticating Clients” on page 135 for more information about how the SAP BC Server authenticates clients.

When the SAP BC Server Is an SSL Client

When a service on the SAP BC Server submits an HTTPS request to another resource on the Internet, the SAP BC Server is the SSL client and the target system to which it is communicating is the SSL server.
When the server acts as an SSL client, as part of the SSL handshake it receives the digital certificate of the Internet Resource to which it is connecting. The digital certificate is usually part of a digital certificate chain. The chain contains the certificates of one or more certificate authorities (CAs) and the digital certificate of the Internet Resource.

At times, one or more CA certificates in the chain might be expired. When a Web browser connects to the Internet resource, it might accept the connection even if it receives an expired CA certificate. The Web browser accepts the connection if it has on file a valid certificate for the CA whose certificate is expired. In contrast, the SAP BC Server does not accept a connection when one of the CA certificates in the chain is expired unless you specifically configure the SAP BC Server to do so.

If you want the SAP BC Server to accept a connection when one or more of the CA certificates in the chain are expired, you must update the `watt.security.ssl.ignoreExpiredChains` property in the server configuration file (server.cnf) to `true`. This setting will cause the server to ignore expired CA certificates in the chain. To change this setting, use the `Settings>Extended` screen of the Server Administrator, as described in “Working with Extended Configuration Settings” on page 100. Remember to restart the server after changing the setting.

**Presenting Multiple Client Certificates**

**Note:** It is less secure to ignore the expired certificates than to deny the connection due to expired certificates.

The SAP BC Server can present a single client certificate to all servers or it can present different client certificates to different SSL servers. In addition, the SAP BC Server can present certificates provided for this purpose by other organizations. (Some organizations prefer to provide certificates signed by their own CAs for clients to use, rather than accept the client's certificate.) You control which certificate the SAP BC Server presents to an SSL...
server by using remote server aliases or special public services. See “Configuring the Server to Present Multiple Client Certificates” on page 120 for more information.

**Checklist for Using SSL**

<table>
<thead>
<tr>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the SAP BC Certificate Toolkit to create a private key and a certificate signing request (CSR) for a digital certificate and send it to a certificate authority.</td>
<td>Refer to “Items You Need Before Configuring Ports to Request Client Certificates” on page 138 and to the SAP BC Certificate Toolkit Guide.</td>
</tr>
<tr>
<td>Wait for your signed certificate. Periodically check the status of your request.</td>
<td>Refer to the SAP BC Certificate Toolkit Guide.</td>
</tr>
<tr>
<td>Obtain your digital certificate and the certificate of the certificate authority that signed your digital certificate. Use the SAP BC Certificate Toolkit to make the certificates available to the SAP BC Server and convert them to DER format if necessary.</td>
<td>Refer to the SAP BC Certificate Toolkit Guide.</td>
</tr>
<tr>
<td>If the SAP BC Server will act as an SSL client, obtain the digital certificates of the certificate authorities that signed the certificates for the Internet resources that you will connect to. Place each certificate in a separate file. Place the files in the directory you use to store digital certificates of certificate authorities.</td>
<td>Refer to “Items You Need Before Configuring Ports to Request Client Certificates” on page 138 and “Obtaining the Certificate of the CA that Signed an Internet Resource’s Certificate” on page 116.</td>
</tr>
<tr>
<td>Configure the SAP BC Server to use SSL.</td>
<td>Refer to “Configuring the Server to Use SSL” on page 117.</td>
</tr>
<tr>
<td>Add an HTTPS port if none are defined. If you want to allow only secure connections to the server, ensure that the primary port uses an HTTPS port and delete all other non-HTTPS ports. Add as many additional HTTPS ports as you want.</td>
<td>Refer to “Configuring Ports” on page 66.</td>
</tr>
</tbody>
</table>
### Securing Communications with the Server

<table>
<thead>
<tr>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you want to authenticate using client certificates but will allow clients without certificates to authenticate using passwords, configure the server to request client certificates.</td>
<td>Refer to “Authenticating Clients” on page 1357.</td>
</tr>
<tr>
<td>If you want to authenticate using client certificates and will not allow clients to authenticate using passwords, configure the server to require client certificates.</td>
<td></td>
</tr>
</tbody>
</table>

**Items You Need Before Configuring SSL**

Before the SAP BC Server can act as an SSL server or SSL client, you must obtain items that are required for an SSL transaction. To obtain most of these items, you can use the SAP BC Certificate Toolkit. For instructions on using the SAP BC Certificate Toolkit, see the *SAP BC Certificate Toolkit Guide* at `<sapbc>\developer\certkit\doc`.

- **Private/Public Key.** The SSL server and SSL client use public key encryption (also known as asymmetric encryption) during the SSL handshake. This type of encryption requires a key pair that is made up of a public key and a private key. The data that is encrypted with one of the keys can only be decrypted using the other key in the pair.

  You use the SAP BC Certificate Toolkit to create the private/public key pair. You place the private key of the key pair in a file. The toolkit uses the private key to create the public key then places the public key in the certificate signing request. The key then becomes part of the digital certificate for the SAP BC Server.

  The party with which the SAP BC Server is communicating obtains the public key from the SAP BC Server's certificate. To communicate securely, the other party can encrypt information with the public key before sending it to the SAP BC Server, which decrypts the information with its private key.

  Refer to the *SAP BC Certificate Toolkit Guide* for instructions on creating the private key.

- **Digital Certificate for the SAP BC Server.** A digital certificate attests to the identity of the SAP BC Server. You can use the SAP BC Certificate Toolkit to create a Certificate Signing Request (CSR) for a digital certificate and to make the certificate available on your server.

  After creating the CSR, the SAP BC Certificate Toolkit takes you to Verisign's website so that you can submit your request to them. Request the certificate in DER format. If you receive a certificate in PEM format (or any format other than DER), use the SAP BC Certificate Toolkit to convert it to DER format.

  When the SAP BC Server acts as an SSL server, it uses this certificate in the SSL handshake to identify itself to the client. When the SAP BC Server acts as an SSL client
and the SSL server requests a client certificate, the SAP BC Server presents this certificate as its client certificate.

The SAP BC Server can present its own client certificate or certificates provided by other organizations. For example, some organizations prefer to provide certificates signed by their own CAs for clients to use, rather than accept the client's certificate. You can set up the SAP BC Server to present client certificates from multiple organizations. This involves obtaining the certificates and setting them up on your server, then using remote aliases or special public services to control which certificate is being presented.

Refer to the SAP BC Certificate Toolkit Guide for instructions on obtaining a certificate for the SAP BC Server. Refer to “Configuring the Server to Present Multiple Client Certificates” on page 120 for more information about sending different certificates to different SSL servers.

- **Certificate of the CA that signed the SAP BC Server’s Server certificate.** The signing CA's certificate attests to the identity of the CA that signed the digital certificate for the SAP BC Server. The CA should send this certificate to you when it sends you the digital certificate for the SAP BC Server. If it is not in DER format, you can use the SAP BC Certificate Toolkit to convert it to DER format.

  When the SAP BC Server acts as an SSL client and the SSL server requests a client certificate, the SAP BC Server presents this certificate along with its client certificate.

  If the certificate authority does not send you its certificate, refer to the SAP BC Certificate Toolkit Guide for instructions on obtaining it.

- **Certificate of the CA that signed an Internet resource’s certificate.** If your SAP BC Server will run services that submit HTTPS requests to other resources on the Internet, the SAP BC Server will be acting as a client and will receive certificates from these resources. In order for these transactions to work, your SAP BC Server must have on file copies of the CA certificates of the Internet resources. For example, if your SAP BC Server runs a service that requests services from Molly Manufacturing, your SAP BC Server must have on file a copy of the certificate of the CA that signed Molly Manufacturing’s certificate. Refer to “Obtaining the Certificate of the CA that Signed an Internet Resource’s Certificate” below.

**Obtaining the Certificate of the CA that Signed an Internet Resource’s Certificate**

You may be able to obtain the certificate of the CA that signed the certificate of an Internet Resource you want to use by importing if from your browser. Browsers typically contain the digital certificates of many certificate authorities. The method you use to obtain the certificate depends on your browser. Another method is to copy the certificate from the CA’s website.
Using Hardware Accelerators to Improve SSL Efficiency

SSL provides for excellent protection of network messages, however encryption processing is very CPU intensive. To improve the efficiency of setting up encrypted connections, you can off load this work to a dedicated device, called a hardware accelerator.

First you must purchase a hardware accelerator and install it using the instructions provided by the vendor. See the readme.txt file in the server’s installation directory for information about supported hardware accelerators.

Once you have installed the hardware accelerator and its associated software, you must configure your SAP BC Server to use it. Follow these steps:

To configure the server to use a hardware accelerator

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Certificates.
3. Click Edit Certificates Settings.
4. In the Hardware SSL Acceleration area of the screen, select the name of the vendor whose hardware accelerator you want to use from the pull-down list.

Important! You must have previously installed this hardware accelerator on your host.

Configuring the Server to Use SSL

Before you configure your SAP BC Server to use SSL, make sure you have read “Items You Need Before Configuring Ports to Request Client Certificates” on page 138.

The following procedure describes how to set up the SAP BC Server to use SSL for secure transmission of data. If you want to set up the server to request or require client certificates for authenticating clients, see “Authenticating Clients” on page 135.

To configure the server to use SSL for secure communications

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Certificates.
3. Click Edit Certificates Settings.
4. Set the Outbound SSL Certificate parameters as follows:
## For this parameter

**Server's Signed Certificate**
- Specify...
  - Path and file name of the file that contains the SAP BC Server's digital certificate.

**Signing Certificates (comma-separated)**
- Comma-separated list of path and file names of the SAP BC server's digital certificate. List the files containing the certificates in the following order: intermediate certificate 1,..., intermediate certificate n, root certificate.

**Server's Private Key**
- Path and file name of the file that contains the private key of the private/public key pair associated with the SAP BC Server's digital certificate.

---

**Note:** The SAP BC Server uses the certificate information on this screen for SSL communications through a port unless you have specified different certificate information for that port. See “Configuring Ports” on page 66 for more information about configuring ports and “Configuring the Server to Present Multiple Client Certificates” on page 120.

The hardware accelerator vendor information always applies server-wide; you cannot specify different hardware accelerators for different ports.

5. In the **Trusted Certificates** area of the screen, in the **CA Certificate Directory** field, type the name of the directory (relative to the server home) that contains the digital certificates of certificate authorities trusted by this server, for example `config\cas`. 


Securing Communications with the Server

Note: Most of the time you will want to specify a trusted certificates directory; however, there may be times when you want to leave it blank. For example, you might want to trust all certificate authorities on outbound requests and trust specific CAs on different ports for incoming requests. For outbound requests (a certificate the server receives from a server that it submits a request to), if you leave this field blank or specify a directory that does not contain certificates for CAs, the server trusts all certificate authorities. If you do not want the server to trust all certificate authorities, be sure to specify a trusted certificates directory that contains certificates of CAs your server should trust.

For inbound requests, you can specify a trusted certificates directory at the server level (on the Security Certificates screen) or at the port level (on the Edit HTTPS Port Configuration screen). If you omit a trusted authorities directory (or specify a directory that does not contain CA certificates) from both the server level and the port level, the server will trust no certificate authorities. If you specify a trusted authorities directory at the server level and at the port level, the server uses the directory specified at the port level for determining trust on connections being made to that port. If you specify a trusted authorities directory at just the port level, the server uses the port-level setting for requests being made to the port.

For S/MIME signature trust validation, if you leave this field blank or specify a directory that does not contain the certificates of trusted CAs, the server will trust all signatures on S/MIME messages.

6 In the Hardware SSL Acceleration area of the screen, in the Hardware Vendor field, select the name of the vendor whose hardware accelerator you want to use. If you are not using a hardware accelerator, leave this field set to None.

Important! You must have previously installed this hardware accelerator on your host.

7 Add an HTTPS port if one does not already exist. Follow the instructions in “Adding Ports” on page 69. Specify HTTPS for the type of port. Make sure no other applications are listening on the port you want to use.

For HTTPS protocol, the standard port is 443.

Note: If your SAP BC Server runs on a Unix system, using a port number below 1024 requires that the server run as “root.” For security reasons, SAP discourages this practice. Instead, run your SAP BC Server using an unprivileged user ID on a high number port (for example 1024 or above) and use the port remapping capabilities present in most firewalls to move requests to the higher numbered ports.
Test whether your server is listening to https requests on the port you specified. Bring up your browser and type in https://localhost:<port>. If the port is working properly, you will see the logon screen for the Server Administrator.

If the Server Administrator does not display, check the following:

— If you used the SAP BC Certificate Toolkit to create this certificate, make sure the key you specified on the Convert and Save Certificates for use with SAP Software screen is the same as the key you sent with your CSR. If the keys do not match and the correct one is the one you sent with the CSR, then go to the Convert and Save Certificates for use with SAP Software screen and perform the conversion again, this time specifying the correct key.

— Check to see if a service running on the machine is listening to the same port.

If you want the server to ignore expired CA certificates that it receives from an Internet resource (i.e., a Web server, another SAP BC Server), update the watt.security.ssl.ignoreExpiredChains property to be true. For information about this setting, see “When the SAP BC Server Is an SSL Client” on page 112. To change this setting, use the Settings>Extended screen of the Server Administrator, as described in “Working with Extended Configuration Settings” on page 100. Remember to restart the server after changing the setting.

Configuring the Server to Present Multiple Client Certificates

The SAP BC Server can present a single client certificate to all SSL servers it communicates with or it can present different client certificates to different SSL servers.

The client certificates the SAP BC Server presents can be its own or certificates provided by other organizations. Some organizations prefer to provide certificates signed by their own CAs for clients to use, rather than accept the client's certificate. For example, suppose company A wants to exchange information with company B, but Company B does not trust client certificates unless they are signed by their own CA. Therefore, in order to do business with B, A must obtain a certificate from company B and present it when connecting to company A's server.
In the following diagram, Company B and Company C require that certificates signed by their CA be presented by the client. Company D accepts Company A’s client certificate.

**Checklist for Presenting Multiple Client Certificates**

<table>
<thead>
<tr>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain a copy of the certificate you want to use</td>
<td>You can use an existing one, create one, or obtain one from the SSL server with which you want to communicate. See “Obtaining Certificates” below for instructions.</td>
</tr>
<tr>
<td>Set up a remote alias</td>
<td>Although not required, using a remote server alias is a convenient way of directing particular certificates to particular SSL servers. See “Setting Up a Remote Server Alias” below for instructions.</td>
</tr>
<tr>
<td>Code your flows</td>
<td>How you code your flows depends on whether or not you have defined a remote server alias for the remote server. See “Coding Your Flows” below for more information.</td>
</tr>
</tbody>
</table>

**Obtaining Certificates**

Make the certificate you want to use available to your SAP BC Server. If you do not already have the certificate you want to use, you can create it using the SAP BC Certificate Toolkit. Refer to the *SAP BC Certificate Toolkit Guide* for instructions on using the toolkit. If you are going to use a certificate provided by the SSL Server with which you want to communicate, obtain the certificate from that organization.
Place the certificate in a location that is easily accessible to the SAP BC Server. A good place is the server’s config directory. For example, you could put the client certificate to use with Company B in `<sapbc>/config/certs/companyB`.

**Setting Up a Remote Server Alias**

Using a remote server is a convenient way of presenting different certificates to different SSL servers. Communication through the alias is optimized, making transactions with the remote server faster. In addition, using an alias is more convenient because it saves you from specifying connection information each time you communicate with the remote server.

Assign a remote server alias to the SSL server to which you want to present a special certificate. See the instructions in “Setting Up Aliases for Remote SAP BC servers” on page 85. It is the alias that controls which certificate is presented to the remote server. If you do not use the alias, you must control which client certificate the SAP BC Server presents by using built-in services. These services, the `pub.security:setKeyAndChain` and `pub.security:clearAndChain` services, are described below and in more detail in the *SAP BC Built-In Services Guide*.

**Coding Your Flows**

How you code your flows depends on whether or not you have defined a remote server alias for the SSL server you want to communicate with. If you are using a remote server alias, the alias controls which certificate is presented. With a remote server alias defined, you can use the `pub.remote:invoke` services in your flows to run services on the remote server.

If you have not defined a remote server alias, you must code your flows to handle switching from one certificate to another using special public services provided by SAP. The `pub.security:setKeyAndChain` service tells your SAP BC Server which client certificate to present. The `pub.security:clearAndChain` service tells your SAP BC Server to revert back to the default certificate. The `pub.remote:invoke`, `pub.security:setKeyAndChain`, and `pub.security:clearAndChain` services are documented in the *SAP BC Built-In Services Guide*.

**Enabling and Disabling Users**

When you are ready to deploy the SAP BC Server, it may be advisable, for security reasons, to disable the well-known built-in user accounts such as Administrator, Developer, and Replicator. For example, you might create a new administrator account SmithAdmin, and then disable Administrator.

See “Disabling and Enabling Users” on page 53 for more information about disabling users.
Controlling Access to Services and Files with ACLs

To control access to services that clients can invoke and the files that the server can serve, assign appropriate Access Control Lists (ACLs) to services and files.

About ACLs

ACLs control access to services and files at the group level. An ACL identifies groups that are allowed to access a service or file (Allowed Groups) and/or groups that are not allowed to access a service or file (Denied Groups). When identifying Allowed Groups and Denied Groups, you select from groups that you have previously defined.

When a client requests that the server invoke a service, the server checks the ACL associated with the service.

When a client requests that the server serve a file, the server determines whether the file is protected by an ACL. If the file is not explicitly protected by an ACL, the server applies the Default ACL. If the file is explicitly protected, the server checks the designated ACL. If the file resides in a folder that is protected by an ACL, the file inherits that protection.

The following table summarizes how the server handles access for a user that is a member of a single group:

<table>
<thead>
<tr>
<th>If a client is a member of a group that is:</th>
<th>Access to the Service is</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td>Denied</td>
<td>Denied</td>
</tr>
<tr>
<td>Not-specified</td>
<td>Denied</td>
</tr>
</tbody>
</table>

The server uses the following rules to determine access for a user that is a member of more than one group: if the user belongs to any group that is allowed, and to no group that is denied, the user is allowed. Otherwise the user is denied. The following table summarizes this approach for a user that is a member of both Group1 and Group2:

<table>
<thead>
<tr>
<th>Group1's Access to the File, Service, or Folder</th>
<th>Allowed</th>
<th>Denied</th>
<th>Not specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed</td>
<td>User Allowed</td>
<td>User Denied</td>
<td>User Allowed</td>
</tr>
<tr>
<td>Denied</td>
<td>User Denied</td>
<td>User Denied</td>
<td>User Denied</td>
</tr>
<tr>
<td>Not specified</td>
<td>User Allowed</td>
<td>User Denied</td>
<td>User Denied</td>
</tr>
</tbody>
</table>
Predefined ACLs

The server comes with the following predefined ACLs. You cannot delete these ACLs.

- **Administrators.** Allows only users in the Administrators group access to a service or file and denies all other users.

- **Anonymous.** Provides access to unauthenticated users (those that did not specify a valid userid).

- **Default.** Allows all authenticated users access to a service or file. When a service is not specifically assigned an ACL or does not inherit an ACL from containing folders, the server uses the Default ACL. If the ACL assigned to a service or file is deleted, the server uses the Default ACL. The Default ACL authorizes authenticated users only. Unauthenticated users (those that did not specify a valid userid) are authorized by the Anonymous ACL.

- **Developers.** Allows only users in the Developers group access to a service and denies all other users.

- **Internal.** Allows only users in the Administrators and Developers groups access to a service and denies all other users. The server assigns this ACL to built-in utility services shipped with the server, such as those in the wmroot and wmpublic folders. You should never need to use this ACL.

- **Replicators.** Allows the Replicator user replication privileges.

- **SAPUsers.** Protects all Inbound Maps and Inbound Processes of the transports related to an SAP system against unauthorized access.

- **WmPartnersUsers.** Restricts user access to the transaction manager and allows editing routing rules.

When Does the Server Perform ACL Checking?

By default, the SAP BC Server performs ACL checking against externally invoked services only. Externally invoked services are those that are directly invoked by a client or DSP. A client can be a browser user, another SAP BC Server, a server client (using the server client API), or a custom HTTP client.

For example, suppose a client invokes the OrderParts service on server A. After checking port access, server A checks its ACLs to make sure the requesting user is allowed to run the OrderParts service. Calls from OrderParts to other services do not result in ACL checks.

You can tell the SAP BC Server to perform ACL checks against internally invoked services, not just externally-invoked services, by selecting the Enforce Internally option associated with the service's ACL setting on the settings screen in the SAP BC Developer.
Creating ACLs

When creating an ACL, you select groups to use for the Allowed Groups and Denied Groups from previously defined groups.

To create an ACL

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click ACLs.
3. Click Add and Remove ACLs.
4. Specify one ACL name per line. Press Enter to separate the lines.
5. Click Create ACLs.

Updating ACLs

After an ACL is created, you can edit it to add and/or remove groups that are identified as Allowed Groups and Denied Groups.

To update an ACL

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click ACLs.
The server displays a screen similar to the following:

Groups in the **Allowed** list have been explicitly allowed to access the files, services, or folders associated with this ACL.

Groups in the **Not Specified** list have not been explicitly allowed or denied access to the file, services, and folders associated with this ACL.

Groups in the **Denied** list have been explicitly denied access to the files, services, or folders associated with this ACL.

To help you determine which users are allowed access by this ACL, the server displays a list of allowed users in the **Resulting users in this ACL with access** area of the screen. The server builds this list by looking at the groups to which the user belongs and comparing that to the groups allowed by the ACL. See “About ACLs” on page 123 for more information about how the server determines access.

3 From the pull-down list of ACLs, select the ACL with which you want to associate a group.
To move a user from one list to another select (highlight) the group or groups you want to move.

After you have selected all the users you want to move from this list, click the appropriate `<->` or `->` button.

The server moves the selected groups to the desired list.

Click **Save Changes**.

**Deleting ACLs**

You can delete any ACL except the predefined ACLs: Anonymous, Administrators, Default, Developers, Internal, and Replicators. You can delete ACLs that are currently assigned to services or files. When a client attempts to access a service or file that is assigned to a deleted ACL, the server denies access.

When you delete an ACL that is assigned to a service, the service retains information about the deleted ACL. As a result, when you view the service information, the server displays the name of the deleted ACL in the **ACL** field. For information about how to assign a different ACL to a service, see “Assigning ACLs to Services and Folders” on page 130.

When you delete an ACL that is assigned to a file, update the `.access` file to assign a different ACL to the file. For more information about assigning ACLs to files, see “Assigning ACLs to Files the Server Can Serve” on page 131.

**To delete an ACL**

1. Open the Server Administrator if it is not already open.
2. In the **Security** menu of the navigation area, click **ACLs**.
3. Click **Add and Remove ACLs**.
4. In the **Remove ACLs** area of the screen, select the ACL or ACLs you want to remove.
5. Click **Remove ACLs**. The server issues a prompt to verify that you want to delete the ACL. Click OK to delete the ACL.

**How ACLs Work with Folders and Services**

This section describes which ACL the server uses to control access to a service based on whether there is:

- No ACL assigned to a service
- An ACL assigned to a folder for a service
- An ACL assigned to a service
Default Configuration—Internal ACL

By default, top-level folders have the Internal ACL assigned to them.

<table>
<thead>
<tr>
<th>Folder List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folders</td>
</tr>
<tr>
<td>[ ] Eitterroot</td>
</tr>
</tbody>
</table>

When you create a nested folder or a service, it inherits the ACL from the containing folder.

No ACL Assigned—Default ACL

If you remove an ACL assignment from a folder or service, the server displays <None> in the ACL field of the folder and service information screens. When a client attempts to invoke a service that does not have an ACL (and has not inherited one from the containing folder), the server uses the Default ACL.

<table>
<thead>
<tr>
<th>Service List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
</tr>
<tr>
<td>[ ] RFCFunctionSearch</td>
</tr>
</tbody>
</table>

**Important!** The Default ACL identifies the Everybody group as an Allowed group and Anonymous as a denied group. This means that if a service has no ACL, then all clients, except unauthenticated ones, can access the service. To avoid inadvertent access to resources, assign an appropriate replacement for the Default ACL.

When You Assign an ACL to a Folder

When you assign an ACL to a folder, it affects the subfolders and services in the folder. The subfolders and services that do not have an assigned ACL inherit the ACL that you assign to the folder. Services in the folder to which you specifically assign an ACL are not affected. When a folder has an assigned ACL, the server displays the name of the ACL in the ACL field of the folder information.

<table>
<thead>
<tr>
<th>Folder List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folders</td>
</tr>
<tr>
<td>[ ] Eitterroot</td>
</tr>
</tbody>
</table>
When a subfolder or service inherits the ACL of its folder, the server displays the name of the ACL of the folder followed by an asterisk (*) in the **ACL** field of the service information. When the client attempts to invoke a service, the server uses the inherited ACL.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>ACL</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCagPo</td>
<td>Developers *</td>
<td></td>
</tr>
<tr>
<td>ProcessCagPo</td>
<td>Developers *</td>
<td></td>
</tr>
<tr>
<td>ProcessOrder</td>
<td>Developers *</td>
<td></td>
</tr>
<tr>
<td>SubmitCagPo</td>
<td>Developers *</td>
<td></td>
</tr>
<tr>
<td>bindPO</td>
<td>Developers *</td>
<td></td>
</tr>
</tbody>
</table>

* ACL setting propagated from parent.

### When You Assign an ACL to a Service

You can specifically assign an ACL to a service. When a service has an assigned ACL, the server displays the name of the assigned ACL in the **ACL** field of the service information. When the client attempts to invoke the service, the server uses the assigned ACL.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>ACL</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCagPo</td>
<td>Developers *</td>
<td></td>
</tr>
</tbody>
</table>

You can change a service’s ACL assignment from inherited to specific. For example, as shown below, the bindPO service inherits its ACL assignment from its containing folder, as reflected by Developers* in the ACL field.

<table>
<thead>
<tr>
<th>Service Name</th>
<th>ACL</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>bindPO</td>
<td>Developers *</td>
<td></td>
</tr>
</tbody>
</table>

* ACL setting propagated from parent.

If you later assign the DBUsers ACL directly to the SearchTable service, the ACL field for this service will show DBUsers instead of DBUsers*. If you subsequently change the ACL assignment for the folder to a different ACL, the SearchTable service keeps DBUsers as its ACL, whereas services without specific ACL assignments inherit the new ACL setting.

You can change a service’s ACL assignment from specific to inherited by changing the service’s ACL assignment to <None>. If the service is contained in a folder that is assigned to an ACL, the service will inherit the containing folder’s ACL assignment.
Summary of ACL Inheritance

The following table summarizes which ACL the server uses based on whether you have assigned an ACL to a service, its immediate parent folder, or neither.

<table>
<thead>
<tr>
<th>ACL is assigned to the service</th>
<th>If ACL is assigned to the folder</th>
<th>The server uses the:</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>ACL assigned to the service</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Inherited ACL that is assigned to the folder</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>ACL assigned to the service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default ACL</td>
</tr>
</tbody>
</table>

Assigning ACLs to Services and Folders

You can assign an ACL to an individual service or to all the subfolders and services in a folder. Use the following procedure to assign a new or different ACL to a service or folder.

**To assign an ACL to a Service or Folder**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click Browse Folders.
4. If the current screen does not list the folder or service to which you want to assign an ACL, click the name of the parent folder until the server displays a screen that lists the folder or service with which you want to work.
5. Click in the ACL field.

   The server displays the ACL Information screen. Use the pull-down list to select the ACL you want to assign to the folder or service and click Save Changes.

**To remove an ACL from a Service or Folder**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click Browse Folders.
4. If the current screen does not list the folder or service to which you want to assign an ACL, click the name of the parent folder until the server displays a screen that lists the folder or service with which you want to work.
5 Click in the **ACL** field.

The server displays the **ACL Information** screen. Select **<None>** from the pull-down menu of ACL names and click **Save Changes**.

### Assigning ACLs to Files the Server Can Serve

The server can serve files that reside in the pub directory for a package or a subdirectory of the pub directory. For more information about how to serve files from the SAP BC Server, refer to the **SAP BC Developer Guide**.

To control file protection, place a .access file in the directory that contains files you want to protect. If the directory contains subdirectories, they will not inherit the protection, so you must provide a .access file in each directory.

For each file in the directory that you want to protect, place a line in the .access file to identify the file and the ACL you want to use to protect the file.

**Note:** Use an operating system tool of your choice to edit the .access file.

For example, assume you have a directory that contains three files (adminpage.dsp, home.dsp, and index.htm). You want to protect the adminpage.dsp file with the Administrators ACL so that only administrators can access this file. You want to protect the home.dsp file with the Developers ACL so only developers can access this file. You also want to assign the Default ACL to the index.htm file so all users can access it. To accomplish this, you would place the following records in the .access file:

```
adminpage.dsp Administrators
home.dsp Developers
index.htm Default
```

**Note:** In the above example, because you want all users to be able to access the index.htm file, you could omit the index.htm Default from the .access file. The server uses the Default ACL for files that are not identified in a .access file or all files in a directory without a .access file.

**Important!** The SAP BC Server loads .access files when a package is loaded; therefore, if you want the changes you make to take effect immediately, reload the package.

### Rules for Using .access Files

When making entries in .access files, observe the following rules:

- Specify the file name only, such as adminpage.dsp followed by the ACL name. If you specify a relative path, the file will not be protected. For example, suppose file
home.dsp is in subdirectory docs in directory pub (pub\docs\home.dsp). If you add the following entry to the .access file on directory pub, the file will not be protected:

docs\home.dsp Developers

Instead, add the following entry to the .access file on directory pub\docs:

home.dsp Developers

The case in which you enter the name depends on how your file system handles case. Suppose you have a file named index.dsp. If you use a case-insensitive system such as Windows, you can enter the file name in any case. Therefore Index.dsp, INDEX.DSP, and so on are all acceptable. However, if you use a case-sensitive system such as Unix, you must enter index.dsp.

Removing ACL Protection from a File

Use the following procedure to remove ACL protection from a file.

To remove ACL protection from a file

1. Shut down the server. For instructions, see “Shutting Down the SAP BC Server” on page 35.
2. Edit the .access file and delete the line that specifies the file whose ACL protection you want to remove.
3. Restart the server. For instructions, see “Restarting the SAP BC Server” on page 36.

Controlling Server SSL Security Level by Port

You can configure your SAP BC Server to present different server certificates with different ports. One reason to do this is so that different ports can provide different SSL security levels. You determine the security level of a certificate during the certificate signing process. You tell the certificate authority which class of certificate you need and it creates a certificate with those attributes. Later, when you configure your ports, you specify the certificate with the security level you want to associate with that port. See “Configuring Ports” on page 66 for instructions on configuring your ports.

Controlling Access to Services by Port

By default, the SAP BC Server provides a single port that allows access to all service requests that come in on that port (unless prohibited by an ACL). Although this port is ideal for initial SAP BC Server installation and configuration, as well as many development environments, for deployment, you should replace this port with ports that limit access to services you intend to make available to your partners and users.
There are two types of port access:

- **Deny By Default.** This is the default type for newly created ports. Use this type to deny access to all services except those you specify in a list that is associated with the port. You might use a Deny By Default port to restrict access so only the set of services that a single application uses are accessible through the port. Set the port to Deny By Default and specify the services for the application in the list associated with the port. Then, clients using the application can only access the specific services for the application. All ports, except 5555, are initially set to Deny By Default with a limited list of services available.

- **Allow By Default.** Select this type if you intend to allow access to all services except those you explicitly deny in a list that is associated with the port.

**Note:** Another way to control access to services through a port is to restrict access to clients that present particular client certificates. See “Authenticating Clients” on page 135 for more information.

### Allow Access to Specified Services (Deny All Others)

With this setting in effect, the server denies access to most services and allows access to some. This is the default setting.

**Important!** When performing the following procedure, do not log into the server through the port you want to change. The procedure involves temporarily denying access to all services through the port. If you log in on the port you want to change and then deny access to all services through it, you will be locked out of the server. Instead, log in through a different existing port or create a new port to log in on.

#### To allow access to specified services

1. Open the Server Administrator if it is not already open.
2. In the Security menu in the navigation area, click Ports.
3. Click Edit in the Access Mode field with which you want to work.
4. Click Set Access Mode to Deny by Default.
5. Click Add Folders and Services to Allow List.
6. Build a list of folders and services for the server to allow from this port.

You can build the list by entering one folder or service at a time, entering sets of folders or services, or doing a combination of the two. Typically, you will group the services you want to expose to your partners in one or more folders. It is then a simple matter of adding those folders to the list.
To enter folders or services one at a time, enter the folder or service name in the area provided on the left, and press Enter. Repeat until you have added all the folders and services you want to add.

Alternatively, you might want to allow all services associated with a specific ACL. For example, to create a custom Administrator port, you can expose all services protected by the administrators ACL. To enter a set of services or folders associated with an ACL, use the pull-down menu on the right of the screen to select an ACL. The server displays a list of the folders and services protected by the ACL. Initially, all these items are selected. If you do not want to add all of them to the list, deselect the ones you do not want. (Use Ctrl-Click to deselect a selected item.) To move these entries to the list of folders and services that will be accessible through the port, click Append Selected. The server appends the selected entries to the existing list.

Continue the process of adding individual items and/or sets of items until you have built the list of folders and services you want to make available from this port.

7 Click Save Additions to return to the previous screen.

Deny Access to Specified Services (Allow All Others)

With this setting in effect, the server allows access to most services and denies access to some.

To deny access to specified services

1 Open the Server Administrator if it is not already open.
2 In the Security menu in the navigation area, click Ports.
3 Click Edit in the Access Mode field with which you want to work.
4 Click Set Access Mode to Allow by Default.
5 Click Add Folders and Services to Deny List.
6 Build a list of folders and services for the server to deny from this port.

You can build the list by entering one folder or service at a time, entering sets of folders or services, or doing a combination of the two.

To enter folders or services one at a time, enter the folder or service name in the area provided on the left, and press Enter. Repeat until you have added all the folders and services you want to add.

To enter a set of services or folders, use the pull-down menu on the right of the screen to select an ACL. The server displays a list of the folders and services protected by the ACL. Initially, all of these items are selected. If you do not want to add all of them to the list, deselect the ones you do not want. (Use Ctrl-Click to deselect a selected item.) To move these entries to the list of folders and services that will be accessible through
the port, click Append Selected. The server appends the selected entries to the existing list.

Continue the process of adding individual items and/or sets of items until you have built the list of folders and services you want to deny access to from this port.

7 Click Save Additions to return to the previous screen.

To reset a port to the default

1 Open the Server Administrator if it is not already open.
2 In the Security menu in the navigation area, click Ports.
3 Click the port with which you want to work.
4 Click Reset to Default Access Settings.

The SAP BC Server changes the type to Deny By Default and creates a default list of allowed services. These include the standard services required to connect to and authenticate to the server.

**Authenticating Clients**

This section describes how the SAP BC Server processes requests from clients attempting to communicate with it. For information about how the SAP BC Server behaves when it is the client, see “When the SAP BC Server Is an SSL Client” on page 112 and “Presenting Multiple Client Certificates” on page 113.

Authentication is determining who a client is. When the server performs authentication, it determines the user name of a client.

Authentication works with access control. After the server determines the user name of a client, it can then determine whether the client should be granted access to the requested resource. The server uses the client’s group membership to control access to the server resources.

<table>
<thead>
<tr>
<th>The server authenticates when a client attempts to...</th>
<th>The server controls access to the requested resource by determining whether...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke a service</td>
<td>The client is a member of a group listed among the Allowed Groups or Denied Groups in the ACL that is associated with the service.</td>
</tr>
<tr>
<td>Access a file that is protected by an ACL</td>
<td>The client is a member of a group listed among the Allowed Groups or Denied Groups in the ACL that is associated with the file.</td>
</tr>
</tbody>
</table>
**Client Certificates**

A client certificate is a digital certificate that identifies a client. The server attempts to authenticate using client certificates only if the incoming request is an HTTPS request.

If a port is configured to request (or require) client certificates, the server requests the client certificate during the SSL handshake that the client and server perform when initializing an SSL transaction.

After the SSL handshake is complete, the server tries to authenticate the client using the client certificate. What happens next depends on how your server is configured.

If a port is configured to **Request** client certificates, a client can connect to that port with or without a certificate. If the certificate exactly matches one on file on the server, the client is automatically logged in as the user that has been previously mapped to that certificate. If the server does not find a matching certificate, it performs Basic Authentication (prompts the client for user and password information). For more information about mapping a user to a certificate, see “Importing a Client Certificate and Mapping It to a User” on page 138. For more information about Basic Authentication, see “Basic Authentication (User Names and Passwords)” on page 141.

If a port is configured to **Require** client certificates, a client can connect to that port only if the certificate the client presents was signed by a trusted CA and exactly matches a client certificate on file on your server. Anything less will cause the request to fail. If the request succeeds, the client is automatically logged in as the user to which the certificate has been previously mapped.

If you do not configure the port to request or require client certificates, the server will not request a client certificate, but will process one if the client presents it. After this point, processing is the same as if you had configured the port to **Request** client certificates.
The following diagram illustrates the differences between requesting and requiring a client certificate.

The following lists reasons why the server cannot authenticate a client using client certificates:

- The server is not configured to use SSL.
- The incoming request is not an HTTPS request.
- Your server is configured to **Require** certificates and the client does not supply a client certificate or the CA certificate of the CA that signed the client's certificate is not in the CA certificate directory, or the client certificate has expired.
- Your server is configured to **Require** certificates, the client's certificate is signed by a trusted authority, but there is no certificate mapping for that certificate.

**Checklist for Using Client Certificates**

<table>
<thead>
<tr>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure the server to use SSL.</td>
<td>Refer to “Securing Communications with the Server” on page 111.</td>
</tr>
<tr>
<td>Obtain the certificates of the Certificate Authorities that you</td>
<td>Place each certificate in a separate file. Place all the files in a</td>
</tr>
<tr>
<td>want the server to use to validate client certificates.</td>
<td>directory to which the SAP BC Server has access. For more information,</td>
</tr>
<tr>
<td></td>
<td>see “Items You Need Before Configuring Ports to Request Client</td>
</tr>
<tr>
<td></td>
<td>Certificates” below.</td>
</tr>
</tbody>
</table>
## Task Notes

- Configure the port to request client certificates. Refer to “Configuring How Ports Handle Client Certificates” on page 139.
- Import client certificates and map to specific user. Refer to “Importing a Client Certificate and Mapping It to a User” on page 138.

### Items You Need Before Configuring Ports to Request Client Certificates

Before configuring ports to request client certificates, you must configure the server to use SSL and obtain the certificates that the server uses to validate client certificates.

- **Configure the server to use SSL.** For information about configuring the server to use SSL, see “Securing Communications with the Server” on page 111.

- **CA certificates.** These are the certificates that the server uses to validate client certificates. One way to obtain these certificates is to extract them from a Web browser. Most Web browsers that support SSL are shipped with the certificates of well-known certificate authorities. If the certificates are not in DER format, use the SAP BC Certificate Toolkit to convert them to DER format.

Place each certificate in a separate file. Place all the files in the same directory.

### Importing a Client Certificate and Mapping It to a User

You can import client certificates and keep them on file and associate each certificate with a particular user. This mapping allows you to control which userid a client logs in as based on the certificate it presents. For example, you might want only the user that presents a particular certificate to log in as user FINANCE.

For ports configured to **Require** certificates, your server will search this store of client certificates for a match. If the server finds a match, the client is automatically logged in as the user that is mapped to that certificate. If no match is found, the request fails.

For ports configured to **Request** certificates, the server will search the store of client certificates for a match. If the server finds a match, the client is automatically logged in as the user that is mapped to that certificate. If no match is found, the server prompts the user to enter userid and password information.

If you are going to configure one or more ports to **Require** client certificates, you must import the client certificates you will accept and map them to SAP BC Server userids you want the clients to log in as.

Even if you do not configure any ports to **Require** client certificates, you might want to import client certificates and map them to users so that clients presenting these certificates can automatically log in as those users.
Authenticating Clients

**Important!** Be careful when mapping a userid to particular client certificate. Make sure the user you specify does not have more authority than you want it to.

---

**To import a client certificate and map it to a user**

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Certificates.
3. Click Client Certificates.
4. In the Certificate Path field, enter the path and file name of the file that contains the certificate you want to import.
5. Click Import Certificate.
6. To map the certificate to a user, find the Current Certificates area of the screen, and click the name in the Subject CN field for the certificate.
7. Click Change User Mapping.
8. From the pull-down list, select a user to which you want to map this certificate and click Save Changes.

---

**Configuring How Ports Handle Client Certificates**

This section describes how to use the Server Administrator to view or change how a port handles client certificates. For instructions on adding a port, see “Adding Ports” on page 69.

---

**To view or change how a port handles client certificates**

1. Open the Server Administrator if it is not already open.
2. In the navigation area of the screen, on the Security menu, click Ports.
3. Locate the port whose client certificate settings you want to view or change and disable it if it is not already disabled. (To disable a port, click the ✔ icon in the Enabled column. The server replaces the ✔ icon with No to indicate that the port is now disabled.) Then click the port number.
4. Click Edit HTTPS Port Configuration.
5. Update the information in the following fields, as necessary:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Authentication</td>
<td>Type of client authentication you want the SAP BC Server to perform for this port. Select one of the following authentication types from the pull-down list in the field.</td>
</tr>
<tr>
<td></td>
<td><strong>None</strong>—The server will not request client certificates. If the client presents a certificate anyway, the SAP BC Server processes it. If the certificate matches exactly a client certificate on file on the server, the client is logged in as the user pre-mapped to the certificate. Otherwise, the server prompts the client for a userid and password.</td>
</tr>
<tr>
<td></td>
<td><strong>Request Client Certificates</strong>—The server will request client certificates for all requests that come in on this HTTPS port. If the client does not provide a certificate, the request proceeds anyway. If the certificate matches exactly a client certificate on file, the client is logged in as the user to which the certificate is pre-mapped. Otherwise, the server prompts the client for a userid and password.</td>
</tr>
<tr>
<td></td>
<td><strong>Require Client Certificates</strong>—The server requires client certificates for all requests that come in on this HTTPS port. For the request to succeed, the client must present a certificate that was signed by a trusted authority and that matches exactly a client certificate on file on the SAP BC Server. If the certificate matches a client certificate on file, the client is logged in as the user to which the certificate was pre-mapped.</td>
</tr>
</tbody>
</table>

In all cases, if the certificate presented has not been signed by a trusted authority, the SAP BC Server does not use it.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate Chain</td>
<td>Optional. Comma-separated list of path and file names of the SAP BC server’s digital certificate. List the files containing the certificates in the following order: server certificate, intermediate certificate 1,…, intermediate certificate n, root certificate. Specify a value here only if you want this port to present a different server certificate from the one specified on the <strong>Certificates</strong> screen.</td>
</tr>
<tr>
<td>Private Key</td>
<td>Optional. Path and file name of the file that contains the private key of the private/public key pair associated with the SAP BC Server’s digital certificate. If you leave this field blank, the SAP BC Server uses the private key specified on the <strong>Certificates</strong> screen.</td>
</tr>
<tr>
<td>Trusted Authority</td>
<td>Optional. Name of the directory (relative to the server home) that contains the digital certificates of certificate authorities trusted by this server, for example config\cas. If you leave this field blank, the server uses the trusted authority directory specified on the <strong>Certificates</strong> screen. If this field is blank on the <strong>Certificates</strong> screen as well, the server trusts no certificates.</td>
</tr>
<tr>
<td>Directory</td>
<td></td>
</tr>
</tbody>
</table>
6 Click **Save Changes**.

7 Enable the port by clicking **No** in the **Enabled** column.

### Basic Authentication (User Names and Passwords)

When the server uses basic authentication, it prompts the client for a user name and password. If a user account is found for the supplied user name, the server authenticates the user name by comparing the supplied password to the password in the user account for the supplied user name. If the password is correct, the server proceeds with the request. If the password is not correct, the server rejects the request. If a user account for the supplied user name is not found, the server rejects the request.

For security reasons, the SAP BC Server (starting with Release 4.0) stores passwords in hashed format. Once passwords are hashed, you cannot convert them back to an unhashed format. Therefore, once you have installed and run with Release 4.0 or later of the SAP BC Server, you cannot go back to an earlier release unless you have backed up your server.

If the client does not supply a user name or password, the server uses the Default user account for the client.

<table>
<thead>
<tr>
<th>Client supplied a user name/password?</th>
<th>User Name found?</th>
<th>Password correct?</th>
<th>Request...</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>proceeds</td>
</tr>
<tr>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>is rejected</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
<td>n/a</td>
<td>is rejected</td>
</tr>
<tr>
<td>NO</td>
<td>n/a</td>
<td>n/a</td>
<td>proceeds using the Default user account</td>
</tr>
</tbody>
</table>

For more information on setting up user accounts, see “Defining a User Account” on page 46. You can also use externally defined user accounts; for more information on how to use external directories and how basic authentication works when using external users, see “Using an External Directory (LDAP or NIS)” on page 175.

### Customizing Authentication

There may be times when you need to perform customized authentication. For example, if you use an external directory such as LDAP or NIS to store and manage users and passwords, the passwords might be unavailable to the SAP BC Server because they are encoded in an unsupported format or because they are stored in an authentication system such as Kerberos.
To access these users and passwords, you can write your own pluggable module to take over authentication processing. The server calls this module when the standard method of authentication cannot provide the necessary information.

The pluggable module is deployed in a package on the SAP BC Server and consists of at least a factory class and an authentication module.

- Factory class—passes the client-provided userid and password to the authentication module.
- Authentication module—performs the actual authentication processing.

To make the pluggable module available to the server, you must register the factory class with the server. This registration occurs during execution of a start up service that you write.
The following sections describe how to set up a pluggable module for your SAP BC Server.

Note: There is a feature of the SAP BC Server that allows you to map client certificates to particular users. This mapping allows a user who presents a particular certificate to log in automatically as the corresponding pre-mapped user. To use this feature you must create and maintain a store of client certificates on the SAP BC Server. If you use an external directory to manage users and passwords and the directory contains certificate information, you can write a pluggable module to obtain certificate information directly from the external directory. This approach saves you from maintaining two certificate stores and allows you to customize certificate authentication.

For more information about mapping a user to a certificate, see “Importing a Client Certificate and Mapping It to a User” on page 138. For more information about Basic Authentication, see “Basic Authentication (User Names and Passwords)” on page 141.

Overview of Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create the factory class.</td>
</tr>
<tr>
<td>2</td>
<td>Create the authentication module.</td>
</tr>
<tr>
<td>3</td>
<td>Create startup and shutdown services to register and unregister the factory class.</td>
</tr>
<tr>
<td>4</td>
<td>Place the factory class, authentication module, and startup and shutdown services in a package.</td>
</tr>
<tr>
<td>5</td>
<td>Enable the package.</td>
</tr>
</tbody>
</table>

Step 1: Creating the Factory Class

The factory class instantiates a new instance of the authentication module for the SAP BC Server and passes the user name and password supplied by the client to the module. The
factory class must implement the `com.wm.security.auth.ModuleFactory` interface. Here is a simple example.

```java
public static class TestModuleFactory
implements ModuleFactory
{
    protected TestModule _module;
    public TestModuleFactory()
    {
        _module = new TestModule();
    }
    public Module getInstance()
    {
        return _module;
    }
    public static String getMechanismName()
    {
        return BasicModule.MECHANISM_NAME;
    }
}
```

### Creating the Authentication Module

The authentication module performs the actual authentication of the user name and password supplied by the client.

In the simple example below, the `processToken(Token token)` method verifies that the supplied user's name is `bob` and that the supplied password is `123`. If the user name and password are correct, the method returns the user name as a string to the SAP BC Server.
The server then checks to make sure this user exists in its list of users. (This list consists of users defined locally and in external directories.)

```java
public static class TestModule
    implements Module
{
    public TestModule()
    {
    }
    public String processToken(Token token)
    {
        if (token == null) {
            return null;
        }
        String id = null;

        try {
            BasicToken bt = (BasicToken) token;
            String name = bt.getName();
            if (name == null) {
                return null;
            }
            if (name.equals("bob") &&
                bt.getPassword().equals("123") &&
                UserManager.getUser(name) != null) {
                id = name;
            }
        } catch (ClassCastException cce) {
        }
        return id;
    }
    public String getMechanism()
    {return "basic";
    }
}
```

This example is very simple. Typically, rather than checking for a hard coded user name and password, your `processToken` method will perform authentication checking in another system, such as LDAP or NIS, or in a proprietary or third party system.

**Step 3  Creating Startup and Shutdown Services to Register and Unregister the Factory Class**

To make your pluggable module available to the server, you must register the factory class with the server. Use the `AuthenticationManager.registerMechanism` method from a startup service to register the class. A startup service runs each time its associated package is enabled.

When you enable the package that contains your pluggable module, the startup service executes and registers the factory class, making the pluggable module the server's alternate authentication processor. This means that if the server cannot perform authentication using the default SAP BC authentication, the server turns processing over to the pluggable module.
Here is a sample startup service that registers the factory class with the server:

```java
public static final void registerAuth (IData pipeline)
    throws ServiceException
{
    AuthenticationManager.registerMechanism(TestModuleFactory.getMechanismName(), new TestModuleFactory());
}
```

You must unregister the factory class when the package containing the pluggable module is disabled. You can do so by executing the `AuthenticationManager.unregisterMechanism` method from the package's shutdown service. A shutdown service is one that executes each time the package is disabled.

Here is a sample shutdown service that unregisters the factory class from the server:

```java
public static final void unregisterAuth (IData pipeline)
    throws ServiceException
{
    AuthenticationManager.unregisterMechanism(BasicModule.MECHANISM_NAME);
}
```

For information about setting up startup and shutdown services for a package, see “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232.

**Step 4** Place the Factory Class, Authentication Module, and Startup and Shutdown Services in a Package

Place the factory class, authentication module, and startup and shutdown services in a package. By placing related files in a package, you can easily manage all the services and files in the package as a unit. For example, you can make them all available, disable them, refresh them, or delete them with one action. Additionally, if you have more than one SAP BC Server installed, you can use the package replication feature to copy all the services and files in a package to another server.

Most likely you will want to keep files and services related to your pluggable authentication module in a package separate from other applications. This way you can disable those packages for maintenance without affecting authentication on your server.

**Step 5** Enable the Package

To make your pluggable module available to the server, you must enable the package in which the module resides.

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click No in the Enabled column for the package you want to enable. The server issues a prompt to verify that you want to enable the package. Click OK to enable the package.
When the package is enabled, the server displays a ✅ icon and Yes in the Enabled column.

For more information about enabling a package, see “Enabling a Package” on page 199.

Protecting Your Internal SAP BC Server with Reverse Invoke

To allow your internal SAP BC Server to operate in environments where connections from a DMZ to the internal network are prohibited, you can configure it to communicate with another SAP BC Server in a reverse invoke configuration. In this configuration, your internal server remains behind your inner firewall, where external clients cannot access it. You place another SAP BC Server in your DMZ to act as a reverse invoke SAP BC Server. The reverse invoke SAP BC Server acts as an intermediary between the Internet and your internal server.

All user validation and transaction processing is performed on the internal server.

External clients send requests to the reverse invoke SAP BC Server (1), which in turn passes the requests to the internal SAP BC Server (2). After processing the requests, the internal server sends the response to the reverse invoke SAP BC Server (3), which in turn passes them back to the client (4).

With reverse invoke, there is no need to open a port through the internal firewall to allow a connection from the DMZ to the internal network. For reverse invoke to work, the internal firewall must still allow a connection from the internal server to the DMZ (that is, an outbound connection). By limiting the connections to just those established by the internal server, the reverse invoke facility makes it more difficult for an attacker to directly penetrate your internal network, even if they subvert a system in the DMZ. However, like any other security mechanism, it is not foolproof; the information still flows from the DMZ to the internal network over the connection established from the inside.

In addition to passing requests from and to the external clients, you can also use the reverse invoke SAP BC Server to perform user-defined functions before passing requests
to the internal server. For example, you might create a package that performs XML validation.

The reverse invoke server is transparent to the client and, unlike some third-party proxy servers, requires no modifications to the client.

Reverse invoke supports nearly all requests that a regular SAP BC Server handles, including guaranteed delivery. The exception is services that require direct access to HTTP/S request and response headers. These headers are not available to the internal server because the reverse invoke server extracts authorization information and data from the external client and transports it to the internal server using a proprietary asynchronous message-based protocol.

⚠️ **Important!** To get the maximum benefit from the reverse invoke configuration, SAP highly recommends that you configure your inner firewall to deny all inbound connections.

### How Reverse Invoke Works

For an SAP BC Server to function as a reverse invoke SAP BC Server, it must have a *proxy* port to listen for requests from external clients (partners) and a *registration* port through which it maintains its connection to the internal server. For security purposes, the internal server initiates the connections to the reverse invoke SAP BC Server.

The internal server uses a proprietary, asynchronous communication protocol to connect to the reverse invoke SAP BC Server. This special protocol uses two transports: SOCK, which is unencrypted and uses plain TCP sockets, and SSLSOCK, which is encrypted and uses SSL sockets. These connections are persistent.

When selecting a connection to the internal server, the reverse invoke SAP BC Server selects registered connections in a round robin manner, provided they satisfy the threshold criteria. Specifically, it selects the next connection that does not exceed the threshold for the number of pending requests. The following steps summarize how an external client request is handled in a reverse invoke scenario:

- The external client sends a request to the reverse invoke SAP BC Server.
- (Optional) The reverse invoke SAP BC Server performs additional processing, such as user-defined XML validation.
- The reverse invoke SAP BC Server converts the request to a message and sends it to the internal server on the previously-established connection between the reverse invoke SAP BC Server and the internal server.
- The internal server processes the request then sends a response to the reverse invoke SAP BC Server.
- The reverse invoke SAP BC Server sends a response to the external client.
The following diagram shows where the proxy and registration ports fit into the reverse invoke configuration.

![Diagram showing reverse invoke configuration]

**Advantages to Reverse Invoke vs. Traditional Third-Party Proxy Servers**

A reverse invoke configuration offers a number of advantages over traditional third-party proxy servers:

- **Reverse Invoke uses persistent SSL socket connections.** These connections eliminate the huge overhead of establishing SSL connections, while providing all the benefits of encryption.
- **With Reverse Invoke, you can configure your inner firewall to deny all inbound connections,** isolating the internal server from the DMZ.
- **Reverse Invoke requires no changes to the external client.**
- **A reverse invoke SAP BC Server can handle both HTTP and HTTPS requests.** Typically third-party proxy servers can handle only one or the other.
- **If desired, you can perform additional processing,** such as user-defined XML validation, on the reverse invoke SAP BC Server before sending the request to the internal server.

**Setting Up Reverse Invoke**

The two main steps to setting up reverse invoke are:

- Configuring an SAP BC Server in the DMZ to be a reverse invoke SAP BC Server.
Configuring your internal SAP BC Server to connect to the reverse invoke SAP BC Server

Configure an SAP BC Server in the DMZ to Be a Reverse Invoke SAP BC Server

The following checklist summarizes the tasks involved in setting up a reverse invoke SAP BC Server:

<table>
<thead>
<tr>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install an SAP BC Server in your DMZ to be your reverse invoke SAP</td>
<td>Install it and follow the instructions below. Any external client on the internet can access your reverse invoke SAP BC Server; therefore, be very security conscious about the services you make available and the users you define. Do not use the reverse invoke SAP BC Server for development work. Do not perform development work there and do not set up users or groups on it. In a later step you will set up IP access controls to prevent anyone except the internal SAP BC Server from connecting to the reverse invoke SAP BC Server.</td>
</tr>
<tr>
<td>Server</td>
<td></td>
</tr>
<tr>
<td>Disable the Developer and Replicator users</td>
<td>You will not need these users on a reverse invoke SAP BC Server. Disabling them prevents someone from gaining access to your reverse invoke server through them. See “Disabling and Enabling Users” on page 53.</td>
</tr>
<tr>
<td>Make sure an administrator user exists on the reverse invoke SAP</td>
<td>An administrator user account (Administrator by default) must exist on the reverse invoke SAP BC Server and it must have a password. The user and password are required because the internal server defines this server as a remote server and must specify a user and password to connect to it. For security reasons, you should disable the default Administrator user and create a new one with the proper authorities. See “Disabling and Enabling Users” on page 53 for instructions.</td>
</tr>
<tr>
<td>BC Server</td>
<td></td>
</tr>
</tbody>
</table>
Protecting Your Internal SAP BC Server with Reverse Invoke

<table>
<thead>
<tr>
<th>Task</th>
<th>Notes</th>
</tr>
</thead>
</table>
| Set up the registration port | This is the port through which the reverse invoke SAP BC Server maintains its connection to the internal server. See “Setting Up the Registration Port” on page 152 for instructions on setting up this port.  
If you are going to set up an encrypted connection between the internal server and the reverse invoke SAP BC Server, you can optionally store a certificate for the administrator user (defined above) on the reverse invoke SAP BC Server. See “Importing a Client Certificate and Mapping It to a User” on page 138 for more information. |
| Optional (but strongly recommended). Set up IP address filtering on the registration port so that only the internal SAP BC Server can connect to your reverse invoke SAP BC Server. This step provides an additional layer of protection to supplement the IP address filtering performed by your firewall and the user authentication. | **Note:** Even if your external firewall filters out connections to the reverse invoke registration port, IP address filtering is a good idea because it will stop insiders from connecting to the reverse invoke SAP BC Server.  
See “Allowing and Denying Inbound Connections to the Server” on page 78 for more information. |
Setting Up the Registration Port

This section explains how to set up the registration port on a reverse invoke SAP BC Server. The registration port is the port through which the reverse invoke SAP BC Server maintains its connection to the internal server.

Note: Most likely you will need only one registration port, however, if you do set up more than one, make sure they are all SOCK or all SSLSOCK.

To set up the registration port

1. Open the Server Administrator for the reverse invoke SAP BC Server if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3 Click **Reverse Invoke Settings**.

4 Click **Configure this Server as a Proxy**.

5 Click **Add Registration Listeners**.

6 Select SOCK or SSLSOCK then click **Go to Step 2**.

    For a SOCK port, enter the following information:

**Note:** When using HTTP, there is no choice of client authentication methods (username and password is the only supported mechanism), therefore no configuration is required.

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>The number you want to use for the registration port. Use a number that is not already in use. It is best <em>not</em> to use a standard port such as 80 (the standard port for HTTP) or 443 (the standard port for HTTPS); since the external firewall will allow access to those ports from the outside world.</td>
</tr>
</tbody>
</table>

| Package name       | This field associates a package with a port. Typically you won’t need to work with packages from a reverse invoke SAP BC Server; therefore you can leave this field with the default setting. |

| Bind Address (optional) | IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you. |

For an SSLSOCK port, enter the following information:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port</td>
<td>The number you want to use for the registration port. Use a number that is not already in use. It is best <em>not</em> to use a standard port such as 80 (the standard port for HTTP) or 443 (the standard port for HTTPS); since the external firewall will allow access to those ports from the outside world.</td>
</tr>
</tbody>
</table>
### For this parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Client Authentication</strong></td>
<td>The type of client authentication to perform when the internal server establishes a persistent connection to the reverse invoke SAP BC Server. See “Authenticating Clients” on page 135 for more information.</td>
</tr>
<tr>
<td><strong>None</strong></td>
<td>the reverse invoke SAP BC Server will not request client certificates, but rather looks for user and password information in the request header.</td>
</tr>
<tr>
<td><strong>Request Client Certificates</strong></td>
<td>the reverse invoke SAP BC Server will request a client certificate from the internal server. If the internal server does not present a certificate, the request proceeds anyway. The reverse invoke SAP BC Server uses user and password information it obtains from the request header.</td>
</tr>
<tr>
<td><strong>Require Client Certificates</strong></td>
<td>the reverse invoke SAP BC Server requires client certificates from the internal server. If the internal server does not supply a client certificate, the request fails. In addition, if the certificate is not mapped to a user with Administrator privileges, the request fails.</td>
</tr>
<tr>
<td><strong>Package name</strong></td>
<td>This field associates a package with a port. Typically you won’t need to work with packages from a reverse invoke SAP BC Server; therefore you can leave this field with the default setting.</td>
</tr>
<tr>
<td><strong>Bind Address (optional)</strong></td>
<td>IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you.</td>
</tr>
<tr>
<td><strong>Certificate Chain (comma-separated)</strong></td>
<td>Optional. Comma-separated list of path and file names of the SAP BC server’s digital certificate. List the files containing the certificates in the following order: server certificate, intermediate certificate 1,..., intermediate certificate n, root certificate. Specify a value here only if you want this port to present a different server certificate from the one specified on the Certificates screen.</td>
</tr>
<tr>
<td><strong>Private Key</strong></td>
<td>Optional. Path and file name of the file that contains the private key of the private/public key pair associated with the SAP BC Server’s digital certificate specified in the Server’s Certificate field.</td>
</tr>
<tr>
<td></td>
<td>If you leave this field blank, the reverse invoke SAP BC Server uses the private key specified on the Certificates screen.</td>
</tr>
</tbody>
</table>
7 Click Save Changes.

8 Locate the port in the Port List, and click No in the Enabled column to enable the port. The server displays a dialog box that prompts you to verify your action. Click OK to verify you want to enable the port.

The server replaces the No with the ✔️ icon to indicate that the port is now enabled.

Setting Up the Proxy Port

This procedure describes how to set up a proxy port on your reverse invoke SAP BC Server. The proxy port is the port through which the reverse invoke SAP BC Server listens for requests from external clients. An SAP BC Server is not considered a reverse invoke SAP BC Server unless it has an enabled proxy port.

For this parameter Specify...

<table>
<thead>
<tr>
<th>Trusted Authority Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional. Name of the directory (relative to the server home) that contains the digital certificates of certificate authorities trusted by this server, for example config\cas.</td>
</tr>
</tbody>
</table>

If the internal server presents a client certificate, the reverse invoke SAP BC Server looks in this directory to see if the client certificate was signed by an authority the reverse invoke SAP BC Server trusts.

If you leave this field blank, the reverse invoke SAP BC Server uses the trusted authority directory specified on the Certificates screen. If this field is blank on the Certificates screen as well, the server trusts no certificates.

To set up the proxy port

1 Open the Server Administrator for the reverse invoke SAP BC Server if it is not already open.

2 In the navigation area of the screen, on the Security menu, click Ports.

3 Click Reverse Invoke Settings.

4 Click Configure this Server as a Proxy.
5 Click **Add Proxy Listeners**.

6 Select HTTP or HTTPS then click **Go to Step 2**.

For an HTTP port, enter the following information:

**Note:** When using HTTP, there is no choice of client authentication methods (username and password is the only supported mechanism), therefore you do not need to configure certificate information.

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>The number you want to use for the proxy port. Use a number that is not already in use. This is the port that will be allowed through your outer firewall.</td>
</tr>
<tr>
<td><strong>Package name</strong></td>
<td>This field associates a package with a port. Typically you won’t need to work with packages from a reverse invoke SAP BC Server; therefore you can leave this field with the default setting.</td>
</tr>
</tbody>
</table>
| **Bind Address**   | IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you. | *(optional)*

For an HTTPS port, enter the following information:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Port</strong></td>
<td>The number you want to use for the proxy port. Use a number that is not already in use.</td>
</tr>
<tr>
<td><strong>Client Authentication</strong></td>
<td>The type of client authentication to perform for requests coming in on the proxy port (in other words, from the external client). See “Authenticating Clients” on page 135 for more information.</td>
</tr>
</tbody>
</table>

**Note:** Although the reverse invoke SAP BC Server asks for authentication information, it does not perform the authentication. Instead, it passes this information to the internal server, which performs the authentication.

**None**—the reverse invoke SAP BC Server will not request client certificates. Instead it looks for user and password information in the request header.
Protecting Your Internal SAP BC Server with Reverse Invoke

For this parameter  Specify...

Request Client Certificates — the reverse invoke SAP BC Server will request client certificates for all requests that come in on this HTTPS port (in other words, from the external client). If the client does not present a certificate, the request proceeds anyway. The reverse invoke SAP BC Server obtains user and password information from the request header.

Require Client Certificates — the reverse invoke SAP BC Server requires client certificates for all requests that come in on this HTTPS port (in other words, from the external client). If the client does not supply a certificate, the request fails.

**Important!** Use the same authentication mode here as you use for the internal server. For example, suppose you specify authentication mode Required on the internal server. Specifying Required on the external port of the reverse invoke SAP BC Server ensures that the request passed to the internal server includes a certificate.

If you have both an HTTP and an HTTPS proxy port on the reverse invoke SAP BC Server, set the authentication mode on the internal server to Request Client Certificates. Otherwise, the internal server will deny all requests coming in on the HTTP proxy port.

Package name  This field associates a package with a port. Typically you won't need to work with packages from a reverse invoke SAP BC Server; therefore you can leave this field with the default setting.

Bind Address (optional)  IP address to which to bind this port. Specify a bind address if your machine has multiple IP addresses and you want the port to use this specific address. If you do not specify a bind address, the server picks one for you.

Certificate Chain (comma-separated)  Optional. Comma-separated list of path and file names of the SAP BC server's digital certificate. List the files containing the certificates in the following order: server certificate, intermediate certificate 1,..., intermediate certificate n, root certificate. Specify a value here only if you want this port to present a different server certificate from the one specified on the Certificates screen.
CHAPTER 9  Managing Server Security

For this parameter Specify...

Private Key Optional. Path and file name of the file that contains the private key of the private/public key pair associated with the SAP BC Server’s digital certificate specified in the Server’s Certificate field.

If you leave this field blank, the server uses the private key specified on the Certificates screen.

Trusted Authority Directory Optional. Name of the directory (relative to the server home) that contains the digital certificates of certificate authorities trusted by this server, for example config\cas.

If the external server presents a client certificate, the reverse invoke SAP BC Server looks in this directory to see if the client certificate was signed by an authority the reverse invoke server trusts.

If you leave this field blank, the reverse invoke SAP BC Server uses the trusted authority directory specified on the Certificates screen. If the trusted authority field is blank on the Certificates screen as well, the reverse invoke SAP BC Server trusts no certificates.

7  Click Save Changes.

8  Locate the port in the Port List, and click No in the Enabled column to enable the port. The server displays a dialog box that prompts you to verify your action. Click OK to verify you want to enable the port.

   The server replaces the No with the ✔ icon to indicate that the port is now enabled.

Setting Up the Filtering Service

You might want to perform additional processing of requests on the reverse invoke SAP BC Server before sending them on to your internal server. For example, you might want to run a service that performs XML validation, so that the internal server is protected from malformed XML requests, which are likely to come from unauthorized users.

To set up a filtering service you must:

- Create the service. See “Working with Services” on page 231 for more information about this topic.

- Configure the reverse invoke SAP BC Server to run the service. You must tell the reverse invoke SAP BC Server to run this service when requests are received. In addition, you must make the service available from the reverse invoke SAP BC Server’s proxy port and set up the appropriate protection for the service.
To configure the reverse invoke SAP BC Server to run a filtering service

To tell the reverse invoke SAP BC Server to run the filtering server, you must change some server properties in the server.cnf file. Use the Settings>Extended screen of the Server Administrator, as described on page “Working with Extended Configuration Settings” on page 100.

1. Change these properties:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>What It Specifies</th>
</tr>
</thead>
<tbody>
<tr>
<td>watt.server.revInvoke.filterSvcName=</td>
<td>Fully qualified name of the filtering service (folder.folder2:servicename).</td>
</tr>
<tr>
<td>watt.server.revInvoke.errorSvcName=</td>
<td>Fully qualified name of the service to run if your filtering service encounters errors (folder.folder2:servicename).</td>
</tr>
<tr>
<td>watt.server.revInvoke.filterSvcContentTypes=</td>
<td>Name of content types for which you want the filtering service to run. If the content type specified by the client matches any in the list, the reverse invoke SAP BC Server runs the filtering service. You can specify multiple content types separated by commas.</td>
</tr>
<tr>
<td>text/xml, text/html</td>
<td></td>
</tr>
</tbody>
</table>

2. Make the service available from the proxy port of the reverse invoke SAP BC Server by adding the service name to the list of allowed services for the port. See “Controlling Access to Services by Port” on page 132 and “Allow Access to Specified Services (Deny All Others)” on page 133.

3. Make sure the service is protected by the Anonymous ACL. See “Assigning ACLs to Services and Folders” on page 130 for instructions.

4. Restart the server.

**Monitoring Traffic Between Your Reverse Invoke SAP BC Server and the Internal SAP BC Server**

You can configure your reverse invoke SAP BC Server to keep track of the number of requests sent to, but not yet executed, by the internal server, and to take action when the number exceeds a threshold you specify. (Too many pending requests suggests the internal server is not handling the request load and could benefit from more connections to the reverse invoke SAP BC Server.)

When the number of pending requests exceeds the threshold, the reverse invoke SAP BC Server fires an alarm event, causing an event handler to execute. You might code the event handler to search for the string “All Reverse Connections are Busy” and send an e-mail to the administrator stating that someone should register more connections from the internal server. Make sure your event handler does not subscribe to any other service that
produces an alarm event. See the SAP BC Developer Guide for information about setting up an event handler.

To specify a threshold for pending requests

1. Open the Server Administrator for the reverse invoke SAP BC Server if it is not already open.
2. In the Security menu in the navigation area, click Ports.
3. Click Reverse Invoke Settings.
4. Click Edit Pending Request Alarm Trigger.
5. Enter the number of pending requests after which the reverse invoke SAP BC Server triggers an alarm event.
6. Click Save Threshold.

Setting Up Your Internal SAP BC Server to Connect to a Reverse Invoke SAP BC Server

In this step, you tell the internal server how to connect to the reverse invoke SAP BC Server. You do this by defining an alias for the reverse invoke SAP BC Server, which specifies connection information, and by specifying the number of connections to extend to the SAP BC reverse invoke SAP BC Server.

Note: When selecting a connection to the internal server, the reverse invoke SAP BC Server uses a round robin approach and selects the next connection it finds that meets the selection criterion. Specifically, it selects the next connection that does not exceed the threshold for the number of pending requests.

Specifying How the Internal Server Connects to the Reverse Invoke SAP BC Server

In this step you specify the alias of the reverse invoke SAP BC Server, select a communications protocol (SOCK or SSLSOCK), and specify the number of connections to exist between the reverse invoke SAP BC Server and your internal server.
To specify how the internal server connects to the reverse invoke SAP BC Server

1. Open the Server Administrator for the internal server if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Click Reverse Invoke Settings.
4. Click Register Reverse Connections.
5. Enter connection information in the following fields:
### Field | What It Means
--- | ---
Reverse Proxy Alias | Alias assigned to the reverse invoke SAP BC Server. The alias contains connection information such as host name or IP address and communications protocol. If you have not already defined an alias for this reverse invoke SAP BC Server, click the link to go to the **Remote Servers** screen. From this screen you can set up an alias for the server. See “Setting Up Aliases for Remote SAP BC servers” on page 85 for more information.

When specifying an alias, use the protocol that matches the protocol specified for the reverse invoke server’s registration port. If the registration port specifies **SOCK**, select **No** for **Use SSL**? if the registration port specifies **SSLSOCK**, specify **Yes** for **Use SSL**.

Number of Connections | Number of connections the internal server is to establish in addition to the number of files specified in the **server.cnf** file.

For these connections to work, the reverse invoke SAP BC Server must be running and a registration port (SOCK for non-SSL or SSLSOCK for SSL) must be available on it.

Minimum Number to keep alive. | This value specifies the minimum number of connections to keep alive if the established connections have been cut e.g. by the firewall.

**Note:** With the default value ‘-1’ you accept the minimum number of connections specified in the **server.cnf** file. If you enter ‘0’ or any positive value, the minimum number of connections specified in the **server.cnf** file will be overwritten by this value.

6. Click **Register Connections**.

7. By default, the access mode for the internal server is set to Deny by default with no allow list, meaning no services will be available from the internal server. See “Setting the Access Mode for Your Internal Server” on page 164 for instructions on changing the access mode.
Specifying Client Authentication Mode

In this procedure you specify the type of client authentication you want the internal server to perform against external clients.

To set client authentication mode for your internal server

1. Open the Server Administrator for the internal server if it is not already open.
2. In the Security menu in the navigation area, click Ports.
3. Click Reverse Invoke Settings.
4. In the Configure Authentication Mode for Inbound Messages area in the Client Authentication field, select an authentication mode from the pull-down menu:

<table>
<thead>
<tr>
<th>For this parameter</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Authentication</td>
<td>The type of client authentication you want the internal server to perform for requests coming from the reverse invoke SAP BC Server.</td>
</tr>
<tr>
<td></td>
<td><strong>User/Password</strong> — the server will not request client certificates. Instead, it will use user and password information. The reverse invoke SAP BC Server includes this information in the message it sends to the internal server.</td>
</tr>
<tr>
<td></td>
<td><strong>Request Client Certificates</strong> — the internal server will look for client certificates for all requests from the client. If the client did not present a certificate, the request proceeds anyway. The internal server uses user and password information instead.</td>
</tr>
<tr>
<td></td>
<td><strong>Require Client Certificates</strong> — the internal server requires client certificates for all requests from the client. If the client does not supply a certificate, the request fails.</td>
</tr>
</tbody>
</table>

**Important!** Use the same authentication mode here as you used for the proxy listener on the reverse invoke SAP BC Server. For example, suppose you specify authentication mode **Required** on the internal server. Specifying **Required** on the external port of the reverse invoke SAP BC Server ensures that the request passed to the internal server includes a certificate.

If you have an HTTP and an HTTPS proxy listener on the reverse invoke server, set the authentication mode on the internal server to Request Client Certificates. Otherwise, the internal server will deny all requests coming in on the HTTP proxy port.
5 Click Save Changes.

Setting a Trusted Authorities Directory

In this procedure you specify a directory that contains a list of certificate authorities the internal server trusts. The server uses this directory when checking certificates submitted by external client.

[Tip]
To set a trusted authorities directory for your internal server

1 Open the Server Administrator for the internal server if it is not already open.
2 In the Security menu of the navigation area, click Ports.
3 Click Reverse Invoke Settings.
4 In the Internal Server Specific Credentials area in the Trusted Authority Directory field, select a directory that contains the digital certificates of certificate authorities trusted by this server, for example config\cas. If you leave this field blank, the SAP BC Server uses the trusted authority directory specified on the Certificates screen. If the trusted authority field is blank on the Certificates screen as well, the internal server trusts no certificates.

Setting the Access Mode for Your Internal Server

The access mode controls which services on the internal server are available to external clients sending requests through the reverse invoke SAP BC Server.

[Warning]
Important! By default, the port is set to Deny by Default with no service list, which means no services are available, so be sure to change it.

[Tip]
To set the access mode for your internal server

1 Open the Server Administrator for the internal server if it is not already open.
2 In the navigation area of the screen, on the Security menu, click Ports.
3 Click Reverse Invoke Settings.
4 In the Access Mode for Inbound Messages area in the Access Mode field, select an access mode.

There are two ways to set up the restrictions for a port:

Deny By Default. Use this type to deny access to all services except those you specify in a list.

Allow By Default. Use this type to allow access to all services except those you specify in a list that is associated with the port.
Refer to “Controlling Access to Services by Port” on page 132 for more information.

**Viewing and Resetting Connections**

There may be times when you need to establish or break connections between the internal server and the reverse invoke SAP BC Server. For example, if you bring up the internal server before the reverse invoke SAP BC Server is available, the internal server will not be able to connect to the reverse invoke SAP BC Server. When the reverse invoke SAP BC Server becomes available, you must log on the internal server as Administrator and establish the connections.

**Note:** When the internal server is shutdown and restarted, it remembers the connections that existed at shutdown and tries to re-establish them.

**To establish connections**

1. Open the Server Administrator for the internal server if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Click Reverse Invoke Settings.
4. Click Register Reverse Connections.
5. Select the alias that corresponds to your reverse invoke SAP BC Server from the pull-down list of remote server aliases.
6. Enter the **Number of Connections** the server is to establish in addition to the number of connections specified in the server.cnf file.
7. Enter the **Minimum Number to keep alive.** This value specifies the minimum number of connections to keep alive if the established connections have been cut e.g. by the firewall.

**Note:** With the default value ‘-1’ you accept the minimum number of connections specified in the server.cnf file. If you enter ‘0’ or any positive value, the minimum number of connections specified in the server.cnf file will be overwritten by this value.

8. Click Register Connections.

**To delete connections**

1. Open the Server Administrator for the internal server if it is not already open.
2. In the Security menu of the navigation area, click Ports.
3. Click Reverse Invoke Settings.
4 In the Connections from Servers section of the screen, click ✗ in the Delete column to remove the connection.

### Frequently Asked Questions About Reverse Invoke

This section provides answers to some frequently asked questions about Reverse Invoke.

1 **By default, is my Integration Server an internal server or a reverse invoke server?**

   An internal server. To find out whether you are logged into a reverse invoke or internal server, open the Server Administrator, go to the Security menu, click Ports, and click Reverse Invoke Settings. The screen name reflects whether the server is reverse invoke or internal, for example Security > Ports > Reverse Invoke > Internal Server.

   **Note:** A server is not a reverse invoke server until its proxy port has been defined and enabled.

2 **If I define the proxy port on the reverse invoke server to use HTTPS, do I need to define my registration port to be an SSLSOCK port?**

   No. The proxy port and the registration port operate independently.

3 **How many reverse connections should I register between the reverse invoke server and the internal server?**

   That depends on the expected load and the size of the transactions. A reverse connection between the reverse invoke and internal server is available except when a request is being written to the Internal server. In other words, reverse invoke connection utilization is I/O bound. Therefore, if you expect large, simultaneous transactions, increase the number of registered connections accordingly.

4 **Is there persistence with the reverse invoke server?**

   No. The reverse invoke server is just a network hop for the incoming request.

5 **I want to authenticate the SSL credentials of external clients. Where do I set up certificates?**
What exactly does the reverse invoke server do with the incoming request from the external client?

The reverse invoke server does the following:

1. Performs trust checking against HTTPS certificates presented by the client (if applicable).
2. Checks IP access rules.
3. Parses the request as an HTTP 1.0/1.1 based request.
4. Executes a filtering service (if applicable).
5. Rewrites the request as a message (not a stream) in a webMethods proprietary protocol called SOCK (for HTTP requests) or SSLSOCK (for HTTPS) requests.
6. Sends the request to the internal server across one of the registered connections.

Is there a way to perform a “keep alive” on the reverse connection to avoid a time-out at the firewall? If a connection is severed due to an exception, transient network outage, firewall intervention or other problem, will it automatically reconnect?

Starting with Release 4.6 of the SAP BC Server, the internal server schedules a task that runs every 30 seconds (by default) and executes a keep-alive service on all the registered connections, thus preventing timeouts at the firewall. You can change the 30 second interval using the Settings > Resources screen of the Server Administrator.
If a connection is severed, it will not be automatically reconnected unless you write custom code to do so. Use the Security > Ports > Reverse Invoke > Internal Server > Register Reverse Invoke Connections screen of the Server Administrator to reregister the connection.

8 Can I use the reverse invoke server as my outbound proxy server as well?

No. The only requests that go through the reverse invoke server are inbound requests from the external client destined for the internal server and responses to those requests from the internal server back to the external client. Any non-solicited requests from the internal server go directly to the external client.

9 Which components does reverse invoke support?

Trading Networks and webMethods eStandards modules (including EDI, ebXML, RosettaNet and CIDX).

10 What authentication mode should I use for the reverse invoke server and the internal server?

As a general rule, specify the same authentication mode on both servers. If this is not practical, at a minimum, make sure that if your internal server requires certificates, you configure your reverse invoke server to require certificates as well.

If you specify authentication mode None or Request on the proxy port of the reverse invoke server, the reverse invoke server will accept requests that do not supply certificates. If your reverse invoke server forwards such a request to an internal server that requires a certificate, the internal server will reject the request.

Responding to NT Challenge/Response Authentication

Note: This section refers to the Microsoft Internet Information Server (IIS); however, the information applies to any Web server that supports NT Challenge/Response authentication.

When the SAP BC Server executes services that access Web pages, the server acts as a Web client requesting information from a Web Server. That is, in these situations, the server functions much like a Web browser. In many circumstances, a Web Server authenticates a Web client.

Most Web servers use Basic Authentication and client certificates to authenticate Web clients. The Microsoft Internet Information Server (IIS) supports another type of authentication. IIS supports Microsoft Windows NT Challenge/Response authentication. NT Challenge/Response authenticates a user without requiring the transmission of actual passwords or sensitive account information across the network. For IIS to use NT Challenge/Response to authenticate a Web client, the Web client must also support NT Challenge/Response. The SAP BC Server contains support for NT Challenge/Response.
The SAP BC Server supports NT Challenge/Response authentication on the connection from the SAP BC Server to a proxy. The SAP BC Server does not support NT Challenge/Response authentication on the connection from a proxy to the Internet.

**User Name, Password, and Domain Name**

The goal of the NT Challenge/Response is the same as for any authentication; that is, to determine the user name for the client and ensure that the client is who he or she claims to be.

If the SAP BC Server is running as an NT service, it uses the local system rights for authentication when responding to an NT Challenge/Response. If you log in as a user, the SAP BC Server uses the credentials associated with that logon session when responding to an NT Challenge/Response.

**Activating NT Challenge/Response**

You must activate NT Challenge/Response before the SAP BC Server can participate in an NT Challenge/Response. Once activated, the SAP BC Server automatically responds to NT Challenge/Response requests.

---

**Note:** NT Challenge/Response is only available when the SAP BC Server is running on NT or some newer Microsoft OS.

---

**To activate NT Challenge/Response**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. If the WmWin32 package is not already enabled, enable it by clicking No in the Enabled column for this package.
4. In the list of packages, click WmWin32.
5. Click Browse Services in WmWin32.
6. In the list of services, click \w\m\ntlm\reg.
7. Click Test reg. The server displays the test screen for the win32.ntlm.reg service.
8. Click Test (without inputs). The server activates NT Challenge/Response.

---

**Note:** If you want NT Challenge/Response available whenever the SAP BC Server is running, make the win32.ntlm.reg service a startup service for the Win32 package. For instructions, see the SAP BC Developer Guide.
To deactivate NT Challenge/Response
1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. In the list of packages, click WmWin32.
4. Click Browse services in WmWin32.
5. In the list of services, click wm.ntlm:unreg.
6. Click Test unreg. The server displays the test screen for the win32.ntlm.unreg service.
7. Click Test (without inputs). The server deactivates NT Challenge/Response.

Using Tokens to Counter XSRF Attacks

Overview
SAP BC offers a specific mechanism to protect your server from cross-site request forgeries (XSRF).
This mechanism is based on a token that is associated with a particular user session.
In the following section you can learn more about how to use this token.
For background information on XSRF attacks and token usage, please refer to SAP BC Security Best Practices Guide.

Enabling token usage

To enable token usage
1. Enable token usage by setting the parameter watt.server.session.enableRequestToken=true in the server configuration file server.cnf or via Extended Settings.

Note: By default the value is false, and hence token usage is generally switched off in order to not break existing scenarios.

When using tokens, you can also configure regular HTTP(S) ports and reverse invoke ports to be protected by a token as well as an ACL. With the given ACL, a port can deny
access based on the user and the absence of the correct token. A more detailed discussion of ACLs in the context of tokens is given in the section on Avoiding pitfalls of token usage.

Defining entry pages

The only exceptions to that rule are the so-called entry pages: As soon as token usage is switched on, the UI shows an additional menu item Entry Pages in the Security section. This menu item allows the user to configure certain DSPs as entry points that are accessible without a token, even though the port may require a token.

Entry pages are pivotal in starting an administration session, since the first contact with a server cannot present a token. Therefore, by default, all pages involved in the initial screen of the administration UI are marked as entry pages. Although XSRF attacks can exploit these pages, this does not present a security risk since these pages execute services that do not alter data or have any critical effect on the server. Caution should be exercised when designating entry pages, since each entry page is a door to the BC server that is open to XSRF attacks. It is definitely not recommendable to have entry pages that entail write access to critical data, or any data for that matter. The pre-selected set is considered safe and comfortable enough for administration purposes. For your own UI centric scenarios choose a safe entry page with caution.

To define an entry page

2. Select a package from the Package List.
3. Select the DSPs that you want to specify as entry pages.

Note: As there can be only a single configuration with regard to ACLs and tokens, you should have separated setups for B2B and UI related scenarios, if you are using reverse invoke.
Avoiding pitfalls of token usage

In the context of tokens, every regular HTTP(S) port is protected by an ACL that determines the users that are granted access to the BC server through that port. Without an ACL, a port does not check authorization and admits any user, relying on the services that are invoked during the request to deal with authorization. Hence an unprotected port (for B2B scenarios, for instance) is wide open to XSRF attacks, since invoking critical services is very often (and must be) permitted for administrators as well as B2B clients. Therefore, ports need to be able to preselect users in order to effectively protect the server.

However, the new level of safety introduced through tokens and ACLs is still rendered null and void, if unprotected ports and protected ports share users. Consider the following scenario: A user X is in the ACL of both ports A and B, where A is a protected port and B is not a protected port. (Both are regular HTTP(S) ports.) If user X works with the administration UI through port A, an XSRF attack is nonetheless possible via port B, since X is permitted to send requests through port B. All the attacker needs to do, is to explicitly mention the unprotected port. The browser adds the credentials, since it does not differentiate between requests on the basis of ports.

Therefore, protected and unprotected ports must not share users in order for XSRF protection to be effective. For this reason there is an additional column ‘XSRF’ on the ports screen that shows the security state of each port with a red, yellow, or green ball.

<table>
<thead>
<tr>
<th>Port</th>
<th>XSRF</th>
<th>Protocol</th>
<th>Type</th>
<th>Package</th>
<th>Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>4451</td>
<td>Green</td>
<td>Email</td>
<td>Regular</td>
<td>WmRoot</td>
<td>Yes</td>
</tr>
<tr>
<td>4802</td>
<td>Green</td>
<td>HTTP</td>
<td>Proxy</td>
<td>WmRoot</td>
<td>No</td>
</tr>
<tr>
<td>4806</td>
<td>Green</td>
<td>SSL/SSLv3</td>
<td>Registration</td>
<td>WmRoot</td>
<td>No</td>
</tr>
<tr>
<td>5558</td>
<td>Green</td>
<td>HTTP</td>
<td>Regular</td>
<td>WmRoot</td>
<td>Yes</td>
</tr>
<tr>
<td>5555</td>
<td>Green</td>
<td>HTTP</td>
<td>Regular</td>
<td>WmRoot</td>
<td>Yes</td>
</tr>
<tr>
<td>4801</td>
<td>Yellow</td>
<td>SOCK</td>
<td>Registration</td>
<td>WmRoot</td>
<td>No</td>
</tr>
<tr>
<td>4713</td>
<td>Yellow</td>
<td>HTTP</td>
<td>Regular</td>
<td>WmRoot</td>
<td>Yes</td>
</tr>
<tr>
<td>4711</td>
<td>Yellow</td>
<td>HTTPS</td>
<td>Regular</td>
<td>WmRoot</td>
<td>No</td>
</tr>
<tr>
<td>4712</td>
<td>Yellow</td>
<td>FTP</td>
<td>Regular</td>
<td>WmDB</td>
<td>No</td>
</tr>
</tbody>
</table>

All ports that are not susceptible to XSRF attacks have the green ball. These are the ports that are disabled or are not regular HTTP(S) ports. In the remainder of this chapter we shall only refer to enabled regular HTTP(S) ports and shall not mention “enabled regular HTTP(S)” explicitly anymore, implicitly assuming a port to be an enabled and regular HTTP(S) port.

An unprotected port has the yellow ball, indicating that browser access through such a port carries the danger of an XSRF attack. A protected port has a green ball, if it does not share any users with any other unprotected port. The red ball is given to all ports that are protected, but share users with unprotected ports. The more conspicuous rating ‘red’ is assigned to the latter kind of ports, since they seem secure at first sight, but it is in fact this deceptiveness that makes them less safe than the openly unprotected ports.
If there are ports in XSRF status ‘red’, an additional detail screen below the ports screen explicitly lists the problematic ports and their shared users (this detail screen is hidden by default, but can be made visible and hidden at the user’s discretion).

The risk details can assist an administrator in identifying the shared users, and that way in improving ACLs and their assignment to ports.

In the above example both ports 5556 and 5557 are protected ports, whereas ports 4713 and 5555 are unprotected ports. When using user Administrator in connection with port 5556, for instance, both ports 4713 and 5555 are open to a XSRF attack. For port 5557, working with user Developer or SAPUser leaves port 4713 vulnerable to XSRF attacks.

**Note:** Port 4711 is green only because it is disabled. If it were enabled it would be shown in status yellow (since it is unprotected) and the XSRF risk details would also include the same entries as for port 4713 since (in this example) both ports are protected by the same ACL.
Using an External Directory (LDAP or NIS)

- Overview of Using External Directories ........................................ 176
- Configuring the Server to Use an External Directory ........................... 177
- Considerations for User Accounts and Groups .................................. 181
- Authenticating External Clients ...................................................... 183
- Granting Administrator Privileges to External Users ........................... 183
- Granting Developer Privileges to External Users ............................... 184
- Granting Access to Services and Files to External Users ...................... 184
Overview of Using External Directories

If your site uses either Lightweight Directory Access Protocol (LDAP) or Network Information Service (NIS) for user and group information, you can configure the SAP BC Server to obtain user and group information from the external directory. You can use only one external directory with the SAP BC Server at a time, either LDAP or NIS.

LDAP and NIS are protocols designed to facilitate sharing information about resources on a network. Typically, they are used to store profile information (login ID, password, etc.). You can also use them to store additional information.

Using your existing LDAP or NIS information allows you to take advantage of a central repository of user and group information. System administrators can add and remove users from the central location. Users do not need to remember a separate password for SAP BC Server applications; they can use the same user names and passwords that they use for other applications.

Note: You cannot use the Server Administrator or the SAP BC Developer to administer users or groups stored in an external directory. This restriction includes changing the passwords of these users.

How the Server Uses Externally-Defined Information

The server can use externally-defined information for the same purposes it uses internally-defined user and group information:

- To authenticate clients using user names and passwords
- To control who can configure and manage the server
- To control who can create, modify, and delete services using the SAP BC Developer
- To control access to services and files that are controlled by the server
Externally-defined information does not replace ACLs. To control access to services and files, you still need to set up the ACLs that identify the groups that are allowed and denied access to specific services and files. However, you can assign externally-defined groups to an ACL.

When you configure the server to use an external directory, the server displays externally-defined user and group information on screens in the Server Administrator (in addition to internally-defined user and group information). For example, when you display the New ACL screen to create a new ACL, the server displays all groups that are defined internally and externally in the Allowed Groups and Denied Groups fields.

When the Server Accesses Externally-Defined Information

The server uses Java Naming and Directory Interface (JNDI) to access the user and group information in LDAP or NIS. The server obtains externally-defined information in the following circumstances:

- To authenticate clients
- To determine if an ACL allows or denies an action
- To display screens that contain externally-defined information

The server does not preload externally-defined user and group information. It accesses the information as it is needed to satisfy requests. When the server accesses information, it requests the minimal amount it can to satisfy the request. The server stores the externally-defined information that it retrieves in cache. The information remains in the cache for a period of time you specify when you configure the server to use the external directory. If the server receives subsequent requests that require the information it has in cache, it uses the cached information rather than accessing the external directory.

**Important!** Because the server must interact with an external directory, client requests that require the server to access an external directory take a longer time for the server to complete than those that do not. The server also takes a longer time to display the Access screens that display external information.

Configuring the Server to Use an External Directory

You can configure the server to use either LDAP or NIS.

**Configuring the Server to Use LDAP**

Use this procedure to configure the server to obtain user and group information from LDAP.
To configure the server to use LDAP

1. Open the Server Administrator if it is not already open.
2. In the Security menu of the navigation area, click Users and Groups.
3. Click JNDI Settings.
4. Click Edit JNDI Settings.
5. Select LDAP from the drop down list in the Provider field. The server issues a prompt to verify that you want to change the setting. Click OK.

The server redisplay the JNDI Settings screen listing configuration settings that are specific to LDAP.

6. Fill in the Settings parameters as follows:

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server URL</td>
<td>The complete URL of the LDAP server. The URL has the format ldap://hostname:portnumber, for example, ldap://blue:389.</td>
</tr>
<tr>
<td>Directory Root</td>
<td>The distinguished name (DN) that defines the root entry of your LDAP directory, for example, o=SAP, c=US.</td>
</tr>
<tr>
<td>Connection principal</td>
<td>The user ID the SAP BC Server should supply to connect to the LDAP server, for example, cn=DirectoryManager, o=SAP, c=US.</td>
</tr>
<tr>
<td>Connection credentials</td>
<td>The password the SAP BC Server should supply to connect to the LDAP server.</td>
</tr>
<tr>
<td>User Root</td>
<td>The name of the root where user entries are located on the LDAP system, relative to Directory Root, for example, ou=People.</td>
</tr>
</tbody>
</table>

Note: Some LDAP servers, for example Netscape’s, store passwords in the SHA-1 encryption format. The SAP BC Server does not support this format; therefore, you must configure your LDAP server to store passwords in the Unix crypt encryption format.
### Configuring the Server to Use an External Directory

#### For this parameter... Specify...

**User Filter**
A filter the SAP BC Server should use to narrow the selection of users within **User Root**, for example, *(objectclass=inetOrgPerson) or (uid=*)*.  
This setting is optional. If you specify a filter, when the SAP BC Server requests user information from the LDAP server, the LDAP server only searches user entries that match the filter criteria.

For more information about LDAP search filters, consult RFC 2254: *Search Filters for LDAP*.

**Group Root**
The name of the root where group entries are located on the LDAP system, relative to **Directory Root**, for example, `ou=Group`.

**Group Filter**
A filter the SAP BC Server should use to narrow the selection of groups within **Group Root**, for example, *(objectclass=groupOfNames) or (cn=*)*.  
This setting is optional. If you specify a filter, when the SAP BC Server requests group information from the LDAP server, the LDAP server only searches group entries that match the filter criteria.

For more information about LDAP search filters, consult RFC 2254: *Search Filters for LDAP*.

**User ID attribute**
The name of the attribute that contains each user’s SAP BC Server user ID, for example, `uid`.

**Password attribute**
The name of the attribute that contains each user’s SAP BC Server user password, for example, `userpassword`.

**Member attribute**
The name of the attribute that specifies the members of a group, for example, `member`.

**Addt’l info (string)**
A comma delimited list of additional string attributes you want retrieved from the LDAP server for each user. For example, if you want to retrieve the “phonenumber” and “mail” attributes for each user, specify `phonenumber, mail`.

---

For this parameter... Specify...

**User Filter**
A filter the SAP BC Server should use to narrow the selection of users within **User Root**, for example, *(objectclass=inetOrgPerson) or (uid=*)*.  
This setting is optional. If you specify a filter, when the SAP BC Server requests user information from the LDAP server, the LDAP server only searches user entries that match the filter criteria.

For more information about LDAP search filters, consult RFC 2254: *Search Filters for LDAP*.

**Group Root**
The name of the root where group entries are located on the LDAP system, relative to **Directory Root**, for example, `ou=Group`.

**Group Filter**
A filter the SAP BC Server should use to narrow the selection of groups within **Group Root**, for example, *(objectclass=groupOfNames) or (cn=*)*.  
This setting is optional. If you specify a filter, when the SAP BC Server requests group information from the LDAP server, the LDAP server only searches group entries that match the filter criteria.

For more information about LDAP search filters, consult RFC 2254: *Search Filters for LDAP*.

**User ID attribute**
The name of the attribute that contains each user’s SAP BC Server user ID, for example, `uid`.

**Password attribute**
The name of the attribute that contains each user’s SAP BC Server user password, for example, `userpassword`.

**Member attribute**
The name of the attribute that specifies the members of a group, for example, `member`.

**Addt’l info (string)**
A comma delimited list of additional string attributes you want retrieved from the LDAP server for each user. For example, if you want to retrieve the “phonenumber” and “mail” attributes for each user, specify `phonenumber, mail`.
CHAPTER 10  Using an External Directory (LDAP or NIS)

180

SAP BC Administration Guide  4.8

Click Save JNDI Settings.

8 Restart the server for the changes to take effect. For instructions, see “Restarting the SAP BC Server” on page 36.

Configuring the Server to Use NIS

Use this procedure to configure the server to obtain user and group information from NIS.

To configure the server to use NIS

1 Open the Server Administrator if it is not already open.
2 In the Security menu of the navigation area, click Users and Groups.
3 Click JNDI Settings.
4 Click Edit JNDI Settings.
5 Select NIS from the drop down list in the Provider field. The server issues a prompt to verify that you want to change the setting. Click OK.

The server redisplay the JNDI Settings screen listing configuration settings that are specific to NIS.
6 Fill in the Settings parameters as follows:

<table>
<thead>
<tr>
<th>For this parameter...</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add'l info (binary)</td>
<td>A comma delimited list of additional binary attributes you want retrieved from the LDAP server for each user. For example, if you want to retrieve the &quot;userphoto&quot; and &quot;certificate&quot; attributes for each user, specify userphoto, certificate.</td>
</tr>
<tr>
<td>Timeout (ms)</td>
<td>Number of milliseconds that you want the SAP BC Server to maintain user and group information that it retrieved from the LDAP server in cache. If the SAP BC Server receives subsequent requests that it can satisfy with the information in cache, it does so rather than accessing the LDAP server.</td>
</tr>
</tbody>
</table>

7 Click Save JNDI Settings.

8 Restart the server for the changes to take effect. For instructions, see “Restarting the SAP BC Server” on page 36.
Considerations for User Accounts and Groups

For this parameter... Specify...
Domain Name The name of the domain being served by the NIS server.
Timeout (ms) Number of milliseconds that you want the SAP BC Server to maintain user and group information that it retrieved from the NIS server in cache. If the SAP BC Server receives subsequent requests that it can satisfy with the information in cache, it does so rather than accessing the NIS server.

7 Click Save JNDI Settings.
8 Restart the server for the changes to take effect. For instructions, see “Restarting the SAP BC Server” on page 36.

Stopping Use of an External Directory

If you no longer want to use an external directory, you can update the configuration to remove the external directory configuration settings.

To stop using an external directory

1 Open the Server Administrator if it is not already open.
2 In the Security menu of the navigation area, click Users and Groups.
3 Click JNDI Settings.
4 Click Edit JNDI Settings.
5 Select <None> from the drop down list in the Provider field. The server issues a prompt to verify that you want to change the setting. Click OK.

The server redispalyes the JNDI Settings screen, which now indicates that you are not using an external provider of directory information.
6 Click Save JNDI Settings.
7 Restart the server for the change to take effect. For instructions, refer to “Restarting the SAP BC Server” on page 36.

Considerations for User Accounts and Groups

This section provides information about user accounts and groups that you should consider if you are using an external directory for user and group information.
You should keep internal and external user accounts and group names unique. It might get confusing if you have an external user account that has the same user name as an internal user account or an external group with the same group name as an internal group. If you do have identically named user names or group names, the server always uses the internally-defined information.

To avoid confusion, it is recommended that you do not set up user accounts or groups internally if you are using an external directory. The exception is the predefined user accounts Default, Administrator, Developer, and Replicator and the predefined groups Everybody, Administrators, Developers, Replicators, and Anonymous. You cannot delete these user accounts and groups; therefore, make sure the internal accounts and groups have the correct definitions.

Important! Although the SAP BC Server is distributed with a predefined Replicator account, you can use a different account for package replication. When publishing a package to another server, the publishing server uses the account specified by the subscription requester. For example, if the subscription requester (either the publisher or the subscriber) specified account DEPT01, the publisher will log into the subscriber server as DEPT01. DEPT01 must be a member of a group that is assigned to the Replicators ACL on the subscriber server.

Although the SAP BC Server is distributed with a predefined Replicators group, you can use a different group for package replication. As long as the subscription requester specifies an account that is a member of a group that is assigned to the Replicators ACL, that user can perform replication.

Refer to “Copying Packages from One Server to Another” on page 201 for more information about package replication.

You should only assign externally-defined users to externally-defined groups. Unpredictable situations can occur when an externally-defined user is included as a member of an internally-defined group. Only assign internally-defined users to internally-defined groups and externally-defined users to externally-defined groups.

An exception to this is that all users (internally- and externally-defined) are members of the internally-defined Everybody group.
You cannot use the Server Administrator to manage (i.e., create, edit, or delete) LDAP or NIS user and group information. To make changes to LDAP or NIS directories, follow your site’s standard directory update procedures.

### Authenticating External Clients

When the server is authenticating a client using user names and passwords, it first attempts to find the user name and password internally. If it finds an internally-defined user account for the supplied user name, the server authenticates the client using the internally-defined information. If the supplied password is correct, the server proceeds with the request. If the supplied password is not correct, the server rejects the request.

If the server cannot find an internally-defined user account for the supplied user name, the server accesses the external directory to obtain user name and password information for the client. If it finds an externally-defined user account, the server authenticates the client using the externally-defined information. If the supplied password is correct, the server proceeds with the request. If the supplied password is not correct, the server rejects the request.

**Note:** If the passwords in the external directory are in a format that is not supported by the SAP BC Server or if the passwords are contained in a different authentication system such as Kerberos, you must create your own pluggable module to obtain this information. See “Customizing Authentication” on page 141 for information about setting up a pluggable module.

If the server cannot find either an internally or externally defined user account for the user, the server rejects the request.

If the user does not supply a user name or password, the server uses the internally-defined Default user account.

### Granting Administrator Privileges to External Users

The Administrators ACL controls who has administrator privileges. Because you cannot assign externally-defined users to internally-defined groups, you cannot grant externally-defined users administrator privileges by assigning them to the internally-defined Administrators group. Instead, you need to set up an externally-defined group for administrators. Then, identify the externally-defined group of administrators to the Administrators ACL.
To grant administrator privileges to an externally-defined user

1. Set up an externally-defined user account for the user if one does not already exist.
2. Set up an externally-defined administrators group if one does not already exist.

**Important!** Do not name the externally-defined group “Administrators.” The name of the group must not be the same name as any internally-defined group.

3. Make the externally-defined user a member of the externally-defined administrators group.
4. Update the Administrators ACL to include the externally-defined administrators group in the Allowed list.

For instructions on how to update an ACL, refer to “Updating ACLs” on page 125. When the server displays the ACL Membership screen, it displays externally-defined groups in the Allowed, Not Specified, and Denied lists.

### Granting Developer Privileges to External Users

The Developers ACL controls who can connect to the SAP BC Server from the Developer to create, modify, and delete services that reside on the server. Because you cannot assign externally-defined users to internally-defined groups, you cannot grant externally-defined users developer privileges by assigning them to the internally-defined Developers group. Instead, you need to set up an externally-defined group for the SAP BC Developer. Then, identify the externally-defined group to the Developers ACL.
Granting Access to Services and Files to External Users

You create ACLs that control access to services and files and assign them to the specific services and files that you want to protect.

To grant access to a service or file, the server first uses internally-defined information to determine whether the client is a member of allowed or denied groups listed in the ACL. If the server cannot find the information internally, it obtains externally-defined information to determine if the ACL allows or denies access.

If you want to allow an externally-defined user access to a service or file, update the ACL that protects the service or file to identify the external user’s group as an Allowed group in the ACL. Similarly, if you want to explicitly deny an externally-defined user access to a

---

**Granting Access to Services and Files to External Users**

To grant developer privileges to an externally-defined user

1. Set up an externally-defined user account for the user if one does not already exist.
2. Set up an externally-defined developers group if one does not already exist.

**Important!** Do not name the externally-defined group “Developers.” The name of the group must not be the same name as any internally-defined group.

3. Make the externally-defined user a member of the externally-defined developers group.
4. Update the Developers ACL to include the externally-defined developers group in the Allowed list.

For instructions on how to update an ACL, refer to “Updating ACLs” on page 125. When the server displays the **ACL Membership** screen, it displays externally-defined groups in both the **Allowed** and **Denied** lists.

---

**SAP BC Server**

- Developers

**External Directory**

- Lindsay
- Rebecca

---

---

---
service or file, update the ACL that protects the service or file to identify the external user’s group as a Denied group in the ACL.

For information about working with ACLs, refer to “Updating ACLs” on page 125.
Managing Packages

- Using Packages ................................................................. 188
- How the Server Stores Package Information .......................... 189
- Finding Information about Your Packages ........................... 191
- Working with Packages ....................................................... 197
- Copying Packages from One Server to Another .................... 201
Using Packages

A package contains a set of services and related files, such as specifications, records, and output templates. When you add a service, specification, record, or output template to the SAP BC Server, you must add it to a package. Use a package to group services and related files.

By placing related files in a package, you can easily manage all the services and files in the package as a unit. For example, you can make them all available, disable them, refresh them, or delete them with one action. Additionally, if you have more than one SAP BC Server installed, you can use package management features to copy some or all services and files in a package to another server.

You can group your services using any package structure you choose, though most organizations group services into packages by function or application. For example, you might put all purchasing-related services in a package called “PurchaseOrderMgt” and all time-reporting services into “TimeCards.”

**Important!** Every service on the server must belong to a package. Before you can make a service available for execution, you must load the package to which it belongs.

You can associate a package with a specific port so that when you replicate the package, it continues to use a port with the same number on the new server. See “Configuring Ports” on page 66 for more information about associating a package with a port.

**Important!** Be careful when replicating a package that is associated with a port; the new port might decrease security on the target system. For example, suppose you replicate a package that is associated with an HTTP port at 5556. The replication process creates an HTTP port at 5556 on the target server. If the target server normally uses only HTTPS ports because of their greater security, then the new port presents a possible security hole on that server.

Predefined Packages

The SAP BC Server comes with the following predefined packages:

- **Default.** This package is empty. SAP provides it as a convenience so you can create services and store them here without first creating a package.
How the Server Stores Package Information

**Note:** This package is not enabled when you first install the SAP BC Server. For instructions on enabling and disabling packages, see “Enabling a Package” on page 199 and “Disabling a Package” on page 199.

Before you enable this package, make sure it is the latest version of the SAP BC Server Ariba Supplier OnRamp. When SAP releases a new version of the SAP BC Server, it includes the version of the SAP BC Server Ariba Supplier OnRamp that is generally available at that time. To obtain the most recent version of the SAP BC Server Ariba Supplier OnRamp, go to the SAP product download site or contact SAP Customer Care.

For more information about this product, see the *SAP BC Server Ariba Supplier OnRamp User Guide*.

- **WmDB.** This package contains services that access JDBC-enabled databases. *Do not alter or delete this package.*

- **WmPartners** This package contains services for Partner Manager. For information about the services in this package, see the *SAP BC SAP Adapter Guide*.

- **WmPublic.** This package contains services (i.e., utilities) that you can invoke from your client applications and services. For more information about the services in this package, see the *SAP BC Built-In Services Guide*.

- **WmRoot.** This package contains services that the server uses for core functionality and auxiliary files. *Do not alter or delete this package.*

- **WmSamples.** This package contains sample services. The tutorial folder contains services used in the *SAP BC Developer Tutorial*.

- **WmWin32.** This package contains services you can use to invoke methods on COM objects. These services are documented in the *SAP BC Developer Guide*. This package also contains Windows-specific samples, such as sample Visual Basic services.

**Important!** Never remove the WmRoot or WmDB packages. The SAP BC Server uses the services in these packages.

### How the Server Stores Package Information

The server physically stores package information in the `<sapbc>\server\packages` directory. The server creates a new subdirectory for each package. The name of the subdirectory is the name of the package. For example, if a package is named “TimeCards,” the server creates the `<sapbc>\server\packages\TimeCards` directory to hold the files for the package.

When you create a new package, the server creates the following subdirectories to hold all the files associated with the package:
The code subdirectory holds the Java and C/C++ services that belong to this package. Within the code subdirectory is the classes, jars, source, and lib subdirectories:

- The classes subdirectory is for Java classes for the Java and C/C++ services.
- The jars subdirectory is for Java classes that are packaged together in jar files.
- The source subdirectory is for the source of Java services.
- The libs subdirectory (not shown here) holds DLLs or specialized libraries that the Java and C/C++ services use.

For ease of administration, place services that use shared libraries in the same package.

- The doc subdirectory holds documentation for the package.
- The ns subdirectory holds flows, specifications, records, and code fragments for Java services.
- The pub subdirectory holds Web documents for the package. For instructions on how to access the Web documents for a package, see “Displaying Documentation for a Package” on page 196.
- The templates subdirectory holds output templates that are associated with this package.
- The widls subdirectory holds WIDL services that are associated with this package.

Note: The SAP BC Server does not automatically create the libs directory because the directory’s existence prevents you from reloading a package without restarting the server. You cannot reload a package that uses shared libraries; you must restart the server.
Manifest File

Each package has a manifest file. It contains:

- **Indication of whether the package is enabled or disabled.** The server does not load disabled packages at server initialization and you cannot execute services that reside in disabled packages.

- **List of startup, shutdown, and replication services, if any, for the package.** For more information about startup, shutdown, and replication services and how to identify them, see “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232.

- **Package description.** A brief description of the package.

- **Version information.** Package version and build number. Also included is the JVM version under which the package was published.

- **Patches applied.** A list of patches that have been applied to the package. These are names or numbers that are meaningful to your installation, possibly obtained from your problem tracking system.

- **Package dependencies, if any, for the package.** For a specific package, the developer can identify other packages that the server should load before it loads the services in a particular package. In other words, the developer can identify when one package depends on another. For information, see “Displaying Information about a Package” on page 193 and the SAP BC Developer Guide.

- **Target package name.** Name of the package.

- **Publishing server.** The SAP BC Server that published the package. If the package has not been published, this field contains None.

The manifest for a package is in the manifest.v3 file in the top directory for the package.

Finding Information about Your Packages

The server displays a variety of information about your packages. The section describes the information that is available and the procedures to use to display the information.

<table>
<thead>
<tr>
<th>Information</th>
<th>Refer to page:</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of all of the packages that reside on your server</td>
<td>192</td>
</tr>
<tr>
<td>Status of whether the server successfully loaded the package or not</td>
<td>192</td>
</tr>
<tr>
<td>Status of whether the package is enabled or disabled</td>
<td>193</td>
</tr>
<tr>
<td>Version number of the package</td>
<td>193</td>
</tr>
</tbody>
</table>
CHAPTER 11  Managing Packages

Viewing the Packages that Reside on Your Server

The Package List screen of the Server Administrator lists all packages that reside on your server. It also displays whether the server successfully loaded the package and whether the package is enabled.

To view the packages that reside on the server

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.

Determining Whether the Server Successfully Loaded the Package

The server displays a status in the Loaded? column of the Packages screen.

<table>
<thead>
<tr>
<th>Status</th>
<th>Indicates that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes 🟢</td>
<td>The server successfully loaded all services associated with the package. The services in the package are available for execution. The server also displays this status if the package is empty.</td>
</tr>
<tr>
<td>Partial 🟢</td>
<td>The server did not load one or more of the services associated with the package. The services that the server successfully loaded are available for execution. For instructions on how to determine which services the server did not successfully load and why, see “Displaying Information about a Package” on page 193.</td>
</tr>
<tr>
<td>No</td>
<td>The server did not load any of the services associated with the package. None of the services are available for execution. For instructions on how to determine why the server could not load the services, see “Displaying Information about a Package” on page 193.</td>
</tr>
</tbody>
</table>
When the server is started, it automatically loads into memory all services that are in enabled packages. If a package is disabled at startup, the server loads the services into memory when the package is enabled. You can manually reload a package into memory by reloading it. For instructions on reloading a package, see “Reloading a Package” on page 198.

**Determining Whether the Package Is Enabled or Disabled**

The server displays a status in the Enabled column of the **Packages** screen. The status indicates whether the package is enabled or disabled. A package must be enabled before the server allows clients access to the services in the package.

<table>
<thead>
<tr>
<th>Status</th>
<th>Indicates that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>The package is enabled and clients can invoke the services in the package.</td>
</tr>
<tr>
<td>No</td>
<td>The package is disabled and clients cannot invoke the services in the package.</td>
</tr>
<tr>
<td>Warnings</td>
<td>Some of the services in the package encountered warnings, but were loaded and are available for use. To learn which services caused warnings look at the Load Warnings list at the bottom of the screen.</td>
</tr>
</tbody>
</table>

For instructions on enabling and disabling packages, see “Enabling a Package” on page 199 and “Disabling a Package” on page 199.

**Displaying Information about a Package**

The **Packages Information** screen displays the following information about a package:

- The version number of the package. By default, the Developer assigns version 1.0 to a new package. You can assign a new version number to the package when you release it.

- The build number of the package. The build number is a kind of generation number a developer assigns to a package each time it is regenerated. For example, a developer might generate version 1.0 of the Ordering package 10 times and assign build numbers 1,2,3…10 to the different generations of the package.

- The minimum version of JVM required to run the package.

- A list of patches included in the package.

- A brief description supplied by the developer who created the package.

- How many services in the package are loaded in the server’s memory and access to the list of these services.
CHAPTER 11 Managing Packages

- How many services in the package are not loaded in the server’s memory, a list of these services, and the reason why the server could not load them.
- The list of startup, shutdown, and replication services in the package.
- Package dependencies (packages on which this package depends and packages that depend on this package).
- The list of subscribers to this package.
- Patch history.

**To display information about a package**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click on the name of the package for which you want to display information.
4. The server displays the Packages Management screen, which contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package Name</td>
<td>Name of the package.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number of the package.</td>
</tr>
<tr>
<td>Build</td>
<td>A number that a developer assigns to a package each time it is regenerated. For example, a developer might generate version 1.0 of the Ordering package 10 times and assign build numbers 1,2,3,…10. These build numbers are generally used to identify the generations of a package in a development environment.</td>
</tr>
<tr>
<td>Minimum Version of JVM</td>
<td>Minimum version of the Java Virtual Machine (JVM) required to run this package.</td>
</tr>
<tr>
<td>Patches Included</td>
<td>A list of patches that have been applied to this release of the package. These are numbers that are meaningful to your installation, possibly obtained from your problem tracking system.</td>
</tr>
<tr>
<td>Description</td>
<td>Package description.</td>
</tr>
<tr>
<td>Publisher</td>
<td>Name of the publisher.</td>
</tr>
<tr>
<td>Created On</td>
<td>Date of creation.</td>
</tr>
</tbody>
</table>
Finding Information about Your Packages

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services Loaded</td>
<td>Number of services that the server successfully loaded. If the server successfully loaded one or more services, the server displays the (show list) link. Click on this link to have the server list the services that it successfully loaded.</td>
</tr>
<tr>
<td>Services Not Loaded</td>
<td>Number of services that the server failed to load. If the server failed to load one or more services, the Load Errors section of the screen lists the services that it could not load, along with the reason.</td>
</tr>
<tr>
<td>Startup Services</td>
<td>List of the services that you or another administrator have identified as startup services. For more information about startup services, refer to “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232.</td>
</tr>
<tr>
<td>Shutdown Services</td>
<td>List of the services that you or another administrator have identified as shutdown services. For more information about shutdown services, see “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232.</td>
</tr>
<tr>
<td>Replication Services</td>
<td>List of the services that you or another administrator have identified as replication services. For more information about replication services, see “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232.</td>
</tr>
<tr>
<td>Packages on which this package depends</td>
<td>List of the packages the server must load before it loads this package. For more information about package dependencies, see the SAP BC Developer Guide.</td>
</tr>
<tr>
<td>Packages that depend on this package</td>
<td>List of packages that depend on this package. If you disable the package, these packages will be affected.</td>
</tr>
<tr>
<td>Subscribers</td>
<td>List of other SAP BC Servers that subscribe to this package. For information on how to copy packages from one server to another, how to subscribe to packages, and how to publish packages to another server, see “Copying Packages from One Server to Another” on page 201.</td>
</tr>
</tbody>
</table>
CHAPTER 11 Managing Packages

Displaying Information about Services and Folders in a Package

You can browse a list of services or folders in a package. See “Finding Information about Services and Folders” on page 229.

Displaying Documentation for a Package

You can document the function of a package and its services in Web documents that the SAP BC Server will serve. Place the Web documents in the pub subdirectory for a package.

Be sure to create an index.html file that holds the home page for the package and contains links to the other Web documents for the package.

Field | Description
--- | ---
Load Errors | Displays a list of services that generated errors and could not be loaded onto the server when the package was installed. When some services do not load, the load status for the package becomes Partial.
Load Warnings | Displays a list of services that generated warnings when the package was installed. The server was able to load the packages, despite the warnings. When package elements are loaded with warnings, the load status for the package becomes Warnings.

To access the home page for a package

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area click Management.
3. Click the home icon for the package.

To access any Web document for a package

Make sure the package is enabled. (See “Determining Whether the Package Is Enabled or Disabled” on page 193 for instructions.) Enter the URL for the Web document. The URLs for the Web documents have the following format:

http://host:port/PackageName/Docname
where:

- `host:port` is the server name and port address of the SAP BC Server
- `PackageName` is the name of the package in which the Web document resides
- `DocName` is the name of the Web document. If you do not specify `DocName`, the server displays the index.html file.

**Working with Packages**

You can perform the following tasks that act on all the files in a package as a unit:

<table>
<thead>
<tr>
<th>Use this function:</th>
<th>When you want to:</th>
<th>Refer to page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Create a new package. Developers create packages from the Developer. See the <em>SAP BC Developer Guide</em> for more information.</td>
<td>197</td>
</tr>
<tr>
<td>Activate</td>
<td>Use a package that you just created without having to restart the server.</td>
<td>198</td>
</tr>
<tr>
<td>Reload</td>
<td>Reload the services in the package into memory without having to restart the server.</td>
<td>198</td>
</tr>
<tr>
<td>Enable</td>
<td>Enable a package that you previously disabled.</td>
<td>199</td>
</tr>
<tr>
<td>Disable</td>
<td>Disable access to a package, but do not want to delete the package.</td>
<td>199</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete all services and related files in a package.</td>
<td>200</td>
</tr>
<tr>
<td>Recover</td>
<td>Recover the services and related files from a package that you previously deleted. You can only recover a deleted package if you had the server save a copy of the package before deleting it.</td>
<td>200</td>
</tr>
<tr>
<td>Archive</td>
<td>Make a working copy of a package without making it generally available to others through a release. You might use this copy as a backup.</td>
<td>201</td>
</tr>
<tr>
<td>Copy</td>
<td>Copy a package from one server to another.</td>
<td>201</td>
</tr>
</tbody>
</table>

**Note:** You can also manage packages by using a set of built-in services. See the *SAP BC Built-In Services Guide* for more information.
CHAPTER 11 Managing Packages

Creating a Package

When a developer wants to create a new grouping for services and related files, he or she creates a package. This creates an empty container package into which your developers can store services, specifications, records, and output templates. When a developer creates a package, the server builds the directory structure of the package as described in “How the Server Stores Package Information” on page 189. See the SAP BC Developer Guide for instructions on creating a Package.

Activating a Package

There may be times when a package is installed on your server but is not active. When a package is active, it is “officially recognized” by the server and displayed in the Package List on the Package Management screen. When a package is inactive, it exists in the Packages directory, but is not officially recognized by the server.

Possible reasons for a package being inactive are:

- You manually installed the package while the server was running.
- Another server published the package to your server, but the package requires a version of the JVM that is higher than the version on your server. A subscribing server will not activate a package under these circumstances.

The package will not be available until either you restart the server or you activate the package.

To activate a package

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click Activate Inactive Packages.
4. In the Load Inactive Packages area, select the package you want to activate from the pull-down menu and click Load.

Reloading a Package

If the server is running when a developer changes a Java service or flow service, you must reload the package in which the service is contained for the changes to take effect. Reloading the package invokes the VM class loader to reload the package’s Java services and relocations the flow services into memory. Developers can also reload a package from the Developer.

To reload a package
1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click the reload icon in the Reload column for the package.

The ✔️ icon in the Loaded? column indicates whether the server loaded the package successfully. For more information, see “Determining Whether the Server Successfully Loaded the Package” on page 192.

### Enabling a Package

To allow clients access to the services in a package, you must ensure the package is enabled. Before the server can execute a service in a package, the package must be enabled and the service must be loaded. By default, the packages are enabled.

When you enable a disabled package, the server loads the services in the package into memory.

#### To enable a package

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click No in the Enabled column for the package you want to enable. The server issues a prompt to verify that you want to enable the package. Click OK to enable the package.

   When the package is enabled, the server displays a ✔️ icon and Yes in the Enabled column.

### Disabling a Package

When you want to temporarily prohibit access to the services in a package, disable the package. When you disable a package, the server unloads all of its services from memory.

**Important!** Never disable the WmRoot package. The SAP BC Server uses the services in this package.

#### To disable a package

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area click the Management link.
3. Click the ✔️ icon in the Enabled column for the package you want to disable. The server issues a prompt to verify that you want to disable the package. Click OK to
disable the package. When the package is disabled, the server displays \textbf{No} in the \textit{Enabled} column.

\textbf{Note:} The server retains the access status of a package (enabled or disabled) across server restarts. When you start the server, the server does not load services in disabled packages.

\section*{Deleting a Package}

When you no longer need the services and files in a package, you can delete the package. When you delete a package, all the elements of the package (services, specifications, records) become unavailable.

When you delete a package, you can optionally select to save a copy of the package. If you save a copy, the server copies the package to the $<\text{sapbc}>\backslash \text{server} \backslash \text{replicate} \backslash \text{salvage}$ directory before deleting the package from the $<\text{sapbc}>\backslash \text{server} \backslash \text{packages}$ directory. If needed, you can recover the package at a later time. For instructions on recovering a deleted package, see “Recovering a Package” on page 200.

\textbf{Important!} Never delete the WmRoot package. The SAP BC Server uses the services in this package.

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area click \textbf{Management}.
3. If you want to save a copy of the package so you can recover it later if necessary, click the \includegraphics[width=0.1\textwidth]{icon.png} icon in the row that corresponds to the package you want to delete.
4. If you do \textit{not} want to save a copy, click the \includegraphics[width=0.1\textwidth]{icon.png} icon in the row that corresponds to the package you want to delete.

\section*{Recovering a Package}

If you deleted a package using the \textbf{Safe delete} option and you need the package again, you can recover the package.

\textbf{To recover a package}

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area click \textbf{Management}.
3. Click \textbf{Recover Packages}.
4. Select the package you want to recover from the pull down and click \textbf{Recover}.
Archiving a Package

There may be times when you want to make a copy of a package without making it generally available. For example, you might want to back it up or send it to someone with whom you do not have a publisher/subscriber relationship.

To archive a package

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Locate the package you want to archive in the Package List, and click the icon.
4. The server displays a screen from which you specify the files you want to archive, the type of archive (full or patch), and version information. See “Specifying File and Version Information for a Release or Archive” on page 213 for instructions on specifying this information.

Copying Packages from One Server to Another

Use package replication to copy (publish) packages from one SAP BC Server to another. It is a convenient way to distribute a package from one server to another anywhere on the Web.

Note: If you want to make a copy of package, for example to make a backup, without sending it to another server, see “Archiving a Package” on page 201.

Overview of Package Replication

During replication, a single SAP BC Server sends (publishes) a specified package to one or more recipient servers. The server on which the package originates is referred to as the publisher, and the recipients are referred to as subscribers.
Subscribing servers receive the package in their inbound directory
(<sapbc>\server\replicate\inbound). To activate the new package, an administrator on
the subscribing server must install the package after it arrives. (This procedure is
explained in “Installing a Package Published by Another Server” on page 225.)

Either a publisher or a subscriber can request a subscription. A publisher can send (push)
the package and the subscriber can request (pull) the package.

Before you send a package to another server, you must create a release. When you create a
release, the server creates a distribution file that contains the package and information
about the package, and makes the package available to subscribers.

You can have multiple releases for a given package. For example, you might have
separate releases for versions 1.0, 1.1, and 1.2 of a given package. Or, you might use
different releases to separate packages for different audiences. Each release must have a
unique name.

Important! If you have multiple releases of a given package and one or more subscribers
have specified the automatic pull feature, those subscribers will receive all releases of a
package when a new release of it becomes available. For more information about the
automatic pull feature, see “The Subscribing Server” on page 217.
A release can contain the complete package (a full release) or just patches to the package (a patch release). Typically you will publish a full release when you have made major changes to the package and use patches just to correct problems with a package.

With a full release, the new package entirely replaces the old package on the subscriber’s server. With a patch release, the files in the patch release replace the versions of those files in the target package; all other files in the target package remain intact.

In addition to specifying a full or patch release, you can select all files to go in the release or just some.

The following diagram illustrates how a patch release replaces files:

![Diagram showing how a patch release replaces files]

Select just Service B for replication
The following diagram illustrates the results if you selected a single service for replication and specified a full release instead.

Most often you will select all files and specify a full release, or select some files and specify a patch release. There might be times however when you want to select just some files and specify a full release. For example, there might be files in a package, such as internal documentation files, that a developer does not want released to others. Selecting all files except the extraneous ones and specifying a full release results in a package that contains just the desired files.

There might be other times when you want to replace some files, leave others intact, and delete others. To achieve this greater level of control, you can perform a patch release and specify files to copy and files to delete. Files that you do not specify for copying or deletion remain intact. In the following example, we want to leave Service A intact, replace Service B, and delete Service C from the target package.
The following shows what you must specify on the Specify Files for the Release screen to accomplish this task:

- **Select just Service B** for replication and select **Service C** for deletion.

Select these files. They will replace the versions in the target package.

Click Selected Files.

Type in these files. They will be deleted from the target package.
The SAP BC Server keeps track of package versions, SAP BC Server versions, and JVM versions so that during package installation the subscribing server can make sure the package being installed is compatible with the subscribing server’s environment. The type of version checking performed depends on whether the release is a full or patch release.

**Note:** If patch releases have been applied to a package, the developer can see the patch history when viewing the package from the Developer. However, when the publisher publishes a full release of the package, the patch history is removed.

### Version Checking

When the administrator on the subscribing server installs the package, the subscribing server performs some version checking:

**Target server verifies that...**

**Target JVM Version**

The target server is running the same or a later version of the JVM, as specified during release creation. If this requirement is not met, the subscribing server issues a warning and installs the package but does not activate it. See “Activating a Package” on page 198 for instructions on activating a package.

**Package Version**

<table>
<thead>
<tr>
<th>For a full release</th>
<th>For a patch release</th>
</tr>
</thead>
<tbody>
<tr>
<td>The version of the package on the target server is earlier than or the same as the package being installed. If this requirement is not met, package installation fails.</td>
<td>The version of the package on the target server exactly matches the version required by the release (as specified during release creation). If this requirement is not met, package installation fails.</td>
</tr>
</tbody>
</table>

For example, if you create a new release and specify that it contains Version 2.0 of the wmExample package, the wmExample package on the target system must be release 2.0 or earlier.

For example, if you create a new release that contains a patch for wmExample package version 2.0, and you specify that the target package must be version 2.0, package installation will fail if the target package is not version 2.0.

This restriction prevents you from inadvertently installing an old version of a package over a newer one.

This restriction gives you greater control over how and where patches are applied. This is useful because patches are typically release-dependent.
Who Can Subscribe?

Any SAP BC Server can subscribe to a package on another server if both servers allow it from a security perspective.

Security for package replication is accomplished a number of ways:

- **Userid and password.** In order to send a package to a subscriber, the publisher must login to the subscriber by specifying a userid and password that exist on the subscriber.

- **ACLs.** The userid the publisher uses to log in to the subscriber must be a member of a group that is assigned to the Replicators ACL or higher on the subscriber.

- **SSL.** You can specify that the servers involved in package replication connect to each other using SSL.

The publisher maintains a list of subscribing servers for each package.

Subscriptions can be added by the publisher or the subscriber:

- **Publisher.** The administrator of a publishing server can use the publisher functions to add (or remove) subscribers to any package that originates on the publishing server (i.e., one to which you do not subscribe).

- **Subscriber.** The administrator of a remote SAP BC Server (the subscriber) can submit a subscription request to the publisher. When the publisher receives this request, it automatically adds that server to the subscription list for the requested package as long as authentication was successful. Subscribers can also issue cancellation requests (i.e., cancel their subscriptions) for packages to which they subscribe.

Guidelines for Using Package Replication

Keep the following guidelines in mind when using the package replication facility:

- Publishers and participating subscribers must use SAP BC Server Version 2.0 or higher. For the for Automatic Pull feature to work, they must be running Version 4.0 or later. If you are running version 4.0 or later of the SAP BC Server and publish to an earlier release of the SAP BC Server, the subscriber cannot perform a manual or automatic pull of a package. Instead, the subscriber must wait for the publisher to send the package.

- Any SAP BC Server can publish a package.

- Any SAP BC Server can subscribe to a package on another SAP BC Server.

- An SAP BC Server can be both a publisher of packages and a subscriber of other packages; however, it cannot be both a publisher and a subscriber of the same package.

- After setting up a subscription, if you delete the user account with which the subscription was set up (the account on the subscribing server that the publishing
server uses to log in), the publisher will not be able to log into the subscribing server to send this package.

The Publishing Server

This section describes the tasks you perform when your server is participating in package replication as the publishing server:

<table>
<thead>
<tr>
<th>Task:</th>
<th>Refer to page:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displaying the list of subscribers for a package</td>
<td>208</td>
</tr>
<tr>
<td>Specifying subscribers for a package</td>
<td>208</td>
</tr>
<tr>
<td>Updating subscriber information</td>
<td>210</td>
</tr>
<tr>
<td>Removing subscribers for a package</td>
<td>211</td>
</tr>
<tr>
<td>Publishing a package to subscribing servers</td>
<td>212</td>
</tr>
<tr>
<td>Specifying File and Version Information for a Release or Archive</td>
<td>213</td>
</tr>
</tbody>
</table>

Displaying Subscribers

Use this procedure to display the list of subscribers for a specific package on your server.

**To display the subscribers for a single package**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click the name of the package for which you want to view subscribers.
   - The server lists the subscribers to the package in the Subscribers field.

**To display the subscribers for all packages**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Publishing.
   - The server displays a list of all packages, their subscribers, and releases.

Adding Subscribers from a Publishing Server

When you add a subscriber, you are identifying the SAP BC Servers that are to receive a package. You can have a different list of subscribers for each package on your server.
Specify the subscribers (recipients) of the package. (You only need to execute this task the first time you publish the package—from then on, you can simply modify or reuse the initial list.)

**Note:** The following procedure is for adding a subscriber from a publishing server. If you want to request a subscription from a subscribing server, see *Subscribing to a Package from a Subscribing Server* on page 213.

To add a subscriber

1. Open the Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Publishing**.
3. Click **Add a Subscriber**.
4. Select the package for which you want to identify subscribers from the drop down list in the **Package** field.
5. To identify a subscribing server, enter information in the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Name</td>
<td>Name of the machine on which the subscribing server is running</td>
</tr>
<tr>
<td>Host Port</td>
<td>Port number on which the subscribing server listens for this package to be published.</td>
</tr>
<tr>
<td>Transport</td>
<td>Method the publishing server uses to send the package to the subscribing server. Select HTTP or HTTPS. HTTP is the default.</td>
</tr>
<tr>
<td>Note: If you want the publisher to use SSL when sending the package to the subscriber, you must specify HTTPS here.</td>
<td></td>
</tr>
</tbody>
</table>

   When the publisher connects to the subscriber, the publisher uses its default certificate (specified on the publisher’s **Security Settings** screen).

<table>
<thead>
<tr>
<th>Remote User Name</th>
<th>User the publishing server uses to log into the subscriber server. This user must be a member of a group that is assigned to the Replicators ACL on the subscribing server.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Password</td>
<td>Password of the user that the publishing server uses to log into the subscribing server.</td>
</tr>
<tr>
<td>Notification Email</td>
<td>Email address of the administrator to notify when the publishing server releases a package.</td>
</tr>
</tbody>
</table>
Then, click Add Subscriber. The server adds the subscriber to the list in the Subscribers field.

Repeat this step for each server you want to identify as a subscriber to the package.

**Note:** To specify the automatic pull feature, you must create the subscription from the subscriber.

**Note:** The subscribing server must be running at the time you add the subscriber.

### Updating Subscriber Information

Use this procedure to update information about a subscriber, such as the package name.

**To update subscriber information**

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of in the navigation area, click Publishing.
3. Click Update and Remove Subscribers.
4. Locate this subscriber in the subscriber information list and click Edit in the Update column.
5. To change subscriber information, enter information in the appropriate fields below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packages</td>
<td>Packages to which the subscriber subscribes. You can change the subscription to be for another package. You can only select a package to which your server does not already subscribe because you cannot both publish and subscribe the same package.</td>
</tr>
<tr>
<td>Host Name</td>
<td>Name of the machine on which the subscribing server is running.</td>
</tr>
<tr>
<td>Host Port</td>
<td>Port number on which the subscribing server listens for this package to be published. The number you specify must correspond to a port that already exists and is enabled on the subscribing server. In addition, the publishing server must have replicator access or higher.</td>
</tr>
</tbody>
</table>
Then, click Submit Changes. The server adds the subscriber to the list in the Subscribers field. The server updates the information on both the subscribing and publishing servers.

Removing Subscribers for a Package

Use this procedure to remove a subscriber from a package that you publish.

Field Description

- **Transport**: Method the publishing server uses to send the package to the subscribing server. Select HTTP or HTTPS. HTTP is the default. The transport type must match the type defined for the host port on the subscribing server.

  **Note**: If you want the publisher to use SSL when sending the package to the subscriber, you must specify HTTPS here.

  When the publisher connects to the subscriber, the publisher uses its default certificate (specified on the publisher’s Security Settings screen).

- **Remote User Name**: User the publishing server uses to log into the subscriber server. This user must be a member of a group that is assigned to the Replicators ACL on the subscribing server.

- **Remote Password**: Password of the user that the publishing server uses to log into the subscribing server.

- **Notification Email**: Email address of the administrator to notify when the publishing server releases a package.

To remove subscribers

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of in the navigation area, click Publishing.
3. Click Update and Remove Subscribers.
4. Locate the package for which you want to remove subscribers and check the box in the Delete field.

**Note**: If a subscriber removes a subscription initiated by the publisher, the subscribing server removes the subscription from its subscriptions list, but the subscription is not immediately removed from the publisher’s list. Instead, the next time the publishing server tries to send the package to the subscriber, the publisher is notified of the removal and then deletes the subscription from the publisher’s list.
Note: If the subscriber is running when you remove it from the subscriber list, the publisher tells the subscriber it has been removed. However, if the subscriber is not running, the subscriber will not know the subscription has been canceled. In this case, you should manually delete the subscription from the subscriber server later when it is available.

Publishing a Package

Publishing a package to other SAP BC Servers involves two tasks:

- **Creating a release.** To publish a package, your server creates a distribution file that contains the information for the package.

  When you create the distribution file, you select what information to include in the file.

  You can select all files to send, or just some. In addition, you can request a **full release** or a **patch release**. With a full release, the new package entirely replaces the old package on the subscriber’s server. With a patch release, the files in the patch release replace the versions of those files in the target package; all other files in the target package remain intact. See “Overview of Package Replication” on page 201 for more information about how full and patch releases differ.

  After you indicate the files to include in the release, the server places all the selected files into a single, compressed file (a zip file). It places the zip file in the `<sapbc>\server\replicate\outbound` directory. If the outbound directory already contains a zip file for this package, the server overwrites the existing file.

- **Sending the release.** After you create the release, you can send it to the subscribing servers.

  A subscribing server receives the zip file containing the release in its inbound directory (`<sapbc>\server\replicate\inbound`). If a zip file for the package already exists in a subscribing server’s inbound directory, the server overwrites it. The zip file remains in the inbound directory on the subscribing server until the administrator of that server installs the package.

  A developer can set up the package to execute a service when you create the release. When you begin to create the release, this service executes before the list of files to be zipped is displayed. You can use this service to write state and configuration information for the package to a file. This file will be included with the other zipped files included in the release. See the *SAP BC Developer Guide* for instructions on setting up replication services.

**Important!** Before you can publish a package, you must specify the subscribers. For instructions, refer to “Adding Subscribers from a Publishing Server” on page 208.
To create a release

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Publishing.
3. Click Create and Delete Releases.
4. Locate the package for which you want to create a release, and click Create Release.
5. The server displays a screen from which you specify the files you want to include in the release, the type of release (full or patch), and version information. See “Specifying File and Version Information for a Release or Archive” on page 213 for instructions on specifying this information.

To send the release

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area click Publishing.
3. Locate the release of the package you want to send under Available Releases, and click Send Release.

Specifying File and Version Information for a Release or Archive

When you archive a package or create a release, the server displays a screen from which you can specify the files you want to archive or release, the type of archive or release (full or patch), and version information.

To specify file and version information
1 Identify the files that you want to include in the release/archive.

<table>
<thead>
<tr>
<th>If you want to include:</th>
<th>Do this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>All files</td>
<td>In the Files to include section, select All files.</td>
</tr>
<tr>
<td>Most, but not all, of the files</td>
<td>In the Files available in package, section, select the files you do NOT want to include in the archive or release. In the Files to include section, select all except selected files. If the developer added package dependencies or startup, shutdown, or replication services to the package since the last archive or release was created, be sure to include the manifest.v3 file. Otherwise these services will not be available in the resultant package. See “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232 for more information about startup, shutdown, and replication services.</td>
</tr>
<tr>
<td>Only a few of the files</td>
<td>In the Files available in package section, select the files you want to include in the archive or release. In the Files to include section, select Selected files. If the developer added package dependencies or startup, shutdown, or replication services to the package since the last archive or release was created, be sure to include the manifest.v3 file. Otherwise these services will not be available in the resultant package. See “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232 for more information about startup, shutdown, and replication services.</td>
</tr>
</tbody>
</table>
Identify files you want to delete from the target package by entering one file name per line. Separate each entry with a semicolon (;). When the subscribing server installs the package, these files will be deleted from the target package.

2 **Identify files you want to delete from the target package by entering one file name per line. Separate each entry with a semicolon (;). When the subscribing server installs the package, these files will be deleted from the target package.**
3 Specify package version information and description:

<table>
<thead>
<tr>
<th>Field</th>
<th>What It Means:</th>
</tr>
</thead>
</table>
| **Archive/Release Type**   | **Full:** All files in the package are written to the archive or release  
**Patch:** Selected files in the package are written to the archive or release. When the administrator on the target server installs a patch archive or release, the files contained in the patch archive or release replace the versions of those files in the target package; all other files in the target package remain intact. If the developer added package dependencies or startup, shutdown, or replication services to the package since the last archive or release was created, be sure to include the manifest.v3 file. Otherwise these services will not be available in the resultant package. See “Running Services When Packages Are Loaded, Unloaded, or Replicated” on page 232 for more information about startup, shutdown, and replication services. |
| **Archive/Release Name**   | A name you assign to the archive or release, for example Beta Release of WmExample Package.                                                                                                                                 |
| **Brief Description**      | A description you assign to the archive or release, for example “Dec release with patches to correct OrderProcess problem.”                                                                                                    |
| **Version**                | The version number you assign to the package you are archiving or releasing. This version might not be the same as the version of the package itself. When a developer first creates a package, the SAP BC Developer assigns version 1.0 to it.     |
| **Build Number**           | A number that a developer assigns to a package each time it is regenerated. For example, a developer might generate version 1.0 of the WmExample package 10 times, assigning build number 1,2,3...10.                   |
| **Patches Included**       | A list of patches that have been applied to this release of the package. These are numbers that are meaningful to your installation, possibly obtained from your problem tracking system.                                |
4 Specify subscriber settings:

<table>
<thead>
<tr>
<th>Field</th>
<th>What It Means:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP BC Server</td>
<td>Version of the SAP BC server that must be running on the target server. For more information about the version checking performed by the subscribing server, see “Version Checking” on page 206.</td>
</tr>
<tr>
<td>Minimum Version of JVM</td>
<td>Minimum version of the Java Virtual Machine (JVM) that the target SAP BC Server should be running when using this package. When the administrator installs the package, the server checks the version of the JVM it is running. If it is running a different version, the server installs the package but does not activate it. For more information about the version checking performed by the subscribing server, see “Version Checking” on page 206.</td>
</tr>
</tbody>
</table>

5 Specify version of target package (for patch releases only)

6 This is the version of the package the target server must be running. When the administrator installs the patch on the target server, the server checks to make sure the version of the target package is the same as the one specified here. If the target package is a different version, the server does not install the package. This restriction gives you greater control over how and where patches are applied. This is useful because patches are typically release-dependent.

**The Subscribing Server**

This section describes the tasks the subscribing server performs when participating in package replication as the subscribing server.

Subscribers can retrieve packages manually or automatically. To retrieve a package manually, an administrator on the subscribing server views a list of available subscriptions and retrieves the desired package. When automatic pull is in effect, the subscribing server automatically pulls a package from the publisher when a new release becomes available.

For a package to be retrieved automatically, the subscriber must specify the automatic pull feature when setting up the subscription. When a new release becomes available, the publishing server sends a service-invocation email to a designated email server. The service-invocation email contains a call to a service that runs on the subscribing server to retrieve packages. The subscribing server periodically checks the email server through an email port on the subscribing server. When it receives and processes the service-invocation email, the subscribing server automatically pulls the package from the
publisher and places it in the Inbound directory. The administrator on the subscribing server can then install the package.

<table>
<thead>
<tr>
<th>Task:</th>
<th>Refer to page:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Displaying packages to which your server subscribes</td>
<td>218</td>
</tr>
<tr>
<td>Manually Pulling a Package</td>
<td>218</td>
</tr>
<tr>
<td>Subscribing to a package from another server</td>
<td>219</td>
</tr>
<tr>
<td>Updating subscription information</td>
<td>222</td>
</tr>
<tr>
<td>Canceling a subscription to a package on another server</td>
<td>224</td>
</tr>
<tr>
<td>Installing a package that was published from another server</td>
<td>225</td>
</tr>
</tbody>
</table>

**Displaying Packages That Your Server Subscribes To**

You can view the subscriptions your server has to packages on other servers.

**To display the packages to which your server subscribes**

1. Open the Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area click **Subscribing**.
3. The server displays a list of your available subscriptions, organized by publisher, package, and release.

The server automatically updates this information when you (the subscriber) add, update, or remove a subscription; however, to see changes made by the publishers, you must click **Update All Subscription Details**.

**Manually Pulling a Package**

You can manually pull subscriptions to the inbound directory of your server.

**To pull a package you have already subscribed to**

1. Open the Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Subscribing**.
3. The server displays a list of your available subscriptions, organized by publisher, package, and release.
4. Find the release of the package you want to pull and in the **Retrieve** field, click the retrieval method you want to use.
### Copying Packages from One Server to Another

#### Install the package.

For instructions, see “Installing a Package Published by Another Server” on page 225.

### Subscribing to a Package from a Subscribing Server

When you subscribe to a package from the subscribing server, your server sends a subscription request to the publishing server. The publishing server adds your server to the subscription list for the package.

The remote server must have an alias defined on the local server. If the remote server does not already have an alias defined, you can define one ahead of time by going to the Settings menu of the navigation area and clicking Remote Servers or you can define one while creating the subscription.

When requesting a subscription, the subscriber must provide the following two-way connection information to the publisher:

- **Method the subscriber will use to connect to the publisher to make the subscription request.** The subscriber must supply a valid userid and password it can use to log in to the publishing server. You set up this and other connection information using a remote server alias for the publisher.

- **Method the publisher will use to connect to the subscriber when sending it a package.** The subscriber must supply a valid userid and password that the publisher can use to log in to the subscribing server. This userid must be a member of a group that is assigned

<table>
<thead>
<tr>
<th>Field</th>
<th>What It Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via Service Invocation</td>
<td>The publishing server sends the release using HTTP or HTTPS.</td>
</tr>
<tr>
<td>Via FTP Download</td>
<td>The publishing server sends the release using FTP.</td>
</tr>
<tr>
<td></td>
<td>When you select FTP, the server prompts you for information required to use FTP:</td>
</tr>
<tr>
<td></td>
<td><strong>Release Name</strong>: Name assigned to the release, for example Beta Release of WmExample Package.</td>
</tr>
<tr>
<td></td>
<td><strong>Remote Server Host</strong>: Name of the machine on which the publishing server resides.</td>
</tr>
<tr>
<td></td>
<td><strong>Remote FTP Port</strong>: FTP port on the publishing server through which the publisher will send the package.</td>
</tr>
<tr>
<td></td>
<td><strong>Remote User Name</strong>: User that the subscriber uses to log into the publishing server.</td>
</tr>
<tr>
<td></td>
<td><strong>Remote Password</strong>: Password of the user that the subscribing server uses to log into the subscribing server.</td>
</tr>
</tbody>
</table>

5 Install the package. For instructions, see “Installing a Package Published by Another Server” on page 225.
to the Replicators ACL. In addition, the subscriber must supply other connection information, such as listening port.

The following procedures describe how to request a subscription.

**Note:** The following procedure is for adding a subscriber from a subscribing server. If you want to set up a subscription on a publishing server, see “Adding Subscribers from a Publishing Server” on page 208.

**Important!** If you request a subscription to a package that does not exist on the specified server, or if that server does not own the package (i.e., it is a subscriber of the package), you will receive an error message, and the publishing server does not process your subscription.

To subscribe to a package from another server

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click **Subscribing**.
3. Click **Subscribe to Remote Package**.
4. Type the name of the package in the **Package** field. Be sure to type the name exactly as it is specified on the publishing server, using the same combination of upper- and lower-case characters.
5. Enter the information in the following fields to set up your request:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publisher Alias</td>
<td>Alias assigned to the publisher. The alias definition tells the subscriber how to connect to the publishing server to register for a subscription. The alias contains connection information such as host name or IP address. If you have not already defined an alias for this publisher, click the link to go to the Remote Servers screen. From this screen you can set up an alias for the publisher. See “Setting Up Aliases for Remote SAP BC servers” on page 85 for more information.</td>
</tr>
<tr>
<td>Local Port</td>
<td>Port number on which the subscriber listens for the publisher to send the package. This setting determines whether the publisher uses HTTP or HTTPS.</td>
</tr>
</tbody>
</table>

**Important!** If you want the publisher to use SSL when sending the package to the subscriber, you must specify an HTTPS port here.
### Field Description

**Local User Name**
User as which the publisher will log into the subscriber.

This user must belong to a user group that is assigned to the Replicators ACL. If you delete the user or change its association with the Replicators ACL, the publisher cannot send this package to the subscriber.

**Local Password**
Password for the local user name.

**Notification Email**
Email address to notify when the publishing server releases a package or a package is delivered.

**Automatic Pull**
Specifies whether the subscribing server is to automatically pull the package from the publisher when a new release becomes available.

If you select *Yes*, you must also specify the email address of a user on an email server to which the publishing server should send a service-invocation email.

The subscribing server, through an email port, periodically checks this email address for a service-invocation email. When the subscribing server processes the email, it pulls the package.

The service invocation-email contains a call to a service that runs on the subscribing server and loads the package to the subscribing server's Inbound directory.

For automatic pull to work, you must set up an email port to listen at the automatic pull address (described below).

For information about setting up an email port, see “Configuring Ports” on page 66.

**Automatic Pull Email**
E-mail address to which the publishing server is to send a service-invocation e-mail when a new release of the package becomes available.

Use a different email address for the notification and service-invocation emails. For example, send notification emails to `package_notifications@mymailserver.com` and service invocation e-mails to `package_autopulls@mymailserver.com`.

For automatic pull to work, you must set up an email port to listen at this address.

For information about setting up an email port, see “Configuring Ports” on page 66.
Updating Your Subscription Information

Use this procedure to update information about your subscription, such as the user name or password on the subscribing server.

To update your subscription information

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of in the navigation area, click Subscribing.
3. Click Update and Unsubscribe from Remote Package.
4. Click Edit in the Update column for the package you want to update.
5. To change subscription information, enter information in the appropriate fields below:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>Package for which you want to change subscription information. You can change the package to another package if you do not already subscribe to or publish the new package. This restriction exists because you cannot both subscribe to and publish the same package.</td>
</tr>
<tr>
<td>Publisher Alias</td>
<td>Alias assigned to the publisher. The alias definition tells the subscriber how to connect to the publishing server to register for a subscription. The alias contains connection information such as host name or IP address. If you have not already defined an alias for this publisher, click the link to go to the Remote Servers screen. From this screen you can set up an alias for the publisher. See “Setting Up Aliases for Remote SAP BC servers” on page 85 for more information.</td>
</tr>
<tr>
<td>Local Port</td>
<td>Port number on which the subscriber listens for the publisher to send the package. This setting determines whether the publisher uses HTTP or HTTPS. Important! If you want the publisher to use SSL when sending the package to the subscriber, you must specify an HTTPS port here.</td>
</tr>
</tbody>
</table>

Note: The publishing server must be running at the time you add the subscription.
### Copying Packages from One Server to Another

**Note:** When the publisher connects to the subscriber, the publisher uses its default certificate (specified on its Security Settings screen). Make sure the port you specify here can accept that certificate.

### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local User Name</strong></td>
<td>User as which the publisher will log into the subscriber. This user must belong to a user group that is assigned to the Replicators ACL. If you delete the user or change its association with the Replicators ACL, the publisher cannot send this package to the subscriber.</td>
</tr>
<tr>
<td><strong>Local Password</strong></td>
<td>Password for the local user name.</td>
</tr>
<tr>
<td><strong>Notification Email</strong></td>
<td>Email address to notify when the publishing server releases a package or a package is delivered.</td>
</tr>
<tr>
<td><strong>Automatic Pull</strong></td>
<td>Specifies whether the subscribing server is to automatically pull the package from the publisher when a new release becomes available. If you already have automatic pull configured and want to turn it off, select No. Then go to the Automatic Pull Email field and delete the email address there. If you want to configure your server for Automatic Pull, select Yes. You must also specify the email address of a user on an email server to which the publishing server should send a service-invocation email. The subscribing server, through an email port, periodically checks this email address for a service-invocation email. When the subscribing server processes the email, it pulls the package. The service invocation-email contains a call to a service that runs on the subscribing server and loads the package to the subscribing server's Inbound directory. For automatic pull to work, you must set up an email port to listen at the automatic pull address (described below). For information about setting up an email port, see “Configuring Ports” on page 66.</td>
</tr>
</tbody>
</table>

For information about setting up an email port, see “Configuring Ports” on page 66.
CHAPTER 11 Managing Packages

Field | Description
--- | ---
Automatic Pull Email | E-mail address to which the publishing server is to send a service-invocation e-mail when a new release of the package becomes available.

Use a different email address for the notification and service-invocation emails. For example, send notification emails to `package_notifications@mymailserver.com` and service invocation e-mails to `package_autopulls@mymailserver.com`.

For automatic pull to work, you must set up an email port to listen at this address.

For information about setting up an email port, see “Configuring Ports” on page 66.

Note: The publishing server must be running at the time you add the subscription.

6 Click **Submit Changes**.

7 The server updates the information on both the subscribing and publishing servers.

**Canceling a Subscription**

When you cancel a subscription, the server sends your cancellation notice to the publishing server. The publishing server removes your server from the subscription list for the specified package. If the publisher is not running when you cancel your subscription, the next time the publisher tries to send the package to your server, the publisher is informed of the cancellation and automatically deletes the subscription from its list of subscribers.

Note: If a subscriber removes a subscription initiated by the publisher, the subscribing server removes the subscription from its subscriptions list, but the subscription is not immediately removed from the publisher’s list. Instead, the next time the publishing server tries to send the package to the subscriber, the publisher is notified of the removal and then deletes the subscription from the publisher’s list.

To cancel your subscription to a package on another server

1 Open the Server Administrator if it is not already open.

2 In the Packages menu of the navigation area, click **Subscribing**.

3 Click **Unsubscribe to Remote Package**.
Copying Packages from One Server to Another

SAP BC Administration Guide 4.8

Installing a Package Published by Another Server

When another server publishes a package to your server, you need to install the published package.

If you install a package that has the same name as an existing package on your server, your server copies the original package to <sapbc>/server/replicate/salvage before it installs the new one. This lets you easily revert to the previous version if you are dissatisfied with the new package. For information about reverting to the earlier version, refer to “Recovering a Package” on page 200.

You can select whether you want the server to immediately activate the package after it installs it. If you do not select to activate the package, the server copies the package to the packages directory, but it is not available for clients to use. To make this package available for clients, you must manually activate. For more information, refer to “Activating a Package” on page 198.

Important! Make sure that packages you install come from a legitimate source, such as a replication from another server. If you are not sure, check with the developers in your organization to verify that an authorized person updated the package. Unknown packages might contain code that could damage your server.

To install a package that was published from another server

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. Click Install Inbound Packages.
4. Select the package you want to install from the Package name drop down list.
5. If you want to make the package available immediately following installation, check the Enable upon installation checkbox in the Option field.
6. Click Install.
Managing Services

- About Services ................................................................. 228
- Fully-Qualified Service Names ........................................ 228
- Finding Information about Services and Folders .................... 229
- Working with Services ..................................................... 231
- Running Services When Packages Are Loaded, Unloaded, or Replicated .................. 232
- Running Services in Response to Specific Events ..................... 233
- Scheduling Services to Execute at Specified Times .................. 234
CHAPTER 12 Managing Services

About Services

A service is a server-resident unit of functionality that clients can invoke. A service might be an entire application or used as part of a larger application. There are several types of services—flow services, Java services, C/C++ services, and WIDL services.

You can create all flow services using the SAP BC Developer. You can create database
flow services from the Server Administrator as well. You can also use the Developer to
create Java services, or use your own development environment to create Java and C/C++
services. Although you cannot use the Developer to create WIDL services, the server still
executes WIDL services that were created with previous versions of the Developer. For
more information about the types of services and how to create them, refer to the SAP BC
Developer Guide.

You can designate one or more services in a package as a startup, shutdown, or replication
service. A startup service is a service that the server automatically executes when a
package is loaded. A shutdown service is a service that the server automatically executes
when a package is unloaded. A replication service is a service that the server
automatically executes when a package is replicated.

To improve the performance of services, you can have the server cache the service results.
Then, when the server receives subsequent requests for the service, it returns the cached
results rather than executing the service. For more information, see “Caching Service
Results” on page 103.

Fully-Qualified Service Names

The fully-qualified service name is comprised of two parts—a folder identifier and the
service name. The folder identifier consists of one or more folder names. The service name
is a single name of the service.

Use a folder to group related services together. When a folder contains other folders, the
nested folders are called subfolders. For example, if you have several services that involve
financial information, you might create a folder named “Finance” to hold the services.
Within the financial services, there might be services that are for personal finances. You
might create a subfolder named “Personal” to hold those services.

Use any name for the service name. For example, if one of the financial services obtains
stock quotes, you might name the service, “StockQuote.”

To specify a fully-qualified service name, identify the folder portion, then a colon (:), then
the service name:

folder:service

For example, if the “StockQuote” service is in the “Finance” folder, the fully-qualified
service name is:

Finance:StockQuote
If the folder portion identifies more than one folder, separate each folder name with a period.

\texttt{folder.subfolder1.subfolder2:service}

For example, if the “HomeLoan” service is in the “Personal” folder, which is contained in the “Finance” folder, the fully-qualified service name is:

Finance.Personal:HomeLoan

The \textit{fully-qualified name of each service must be unique within the server}. In addition, the fully-qualified name of a service cannot be the same as the fully-qualified name of any specification or record that resides on the server.

\textbf{Note:} The \texttt{watt.server.illegalNSChars} setting in the \texttt{server.cnf} file (which is located in the <\texttt{sapbc}\server\config> directory) defines the characters that you cannot use when naming folders and services. To view or change this setting, use the \texttt{Settings>Extended} screen from the Server Administrator as described on page “Working with Extended Configuration Settings” on page 100.

\section*{Package Names and Service Names}

The relationship between the package name and the folder name can cause confusion. The name of the package to which a service belongs has no bearing on the names of the services and folders it contains. Nor does it affect how it is referenced by a client application. For example, if you move a service called “Personnel:GetDeptNames” from a package called “Admin” to a package called “EmployeeData” you will not affect client applications that reference that service—it will still be referenced by the name “Personnel:GetDepNames.”

Because the fully-qualified name of each service must be unique within the server, you cannot have two identically named services in two different packages on the same server.

\section*{Finding Information about Services and Folders}

This section describes how to list the services (and folders) on your server and display information about a specific service.

\section*{Listing Folders and Services}

The \texttt{Folders and Services} screens of the Server Administrator list the services that reside on your server and the folders with which they are associated.

To list folders and services

1. Open the Server Administrator if it is not already open.
2 From the Packages menu in the navigation area, click Management.

3 Click Browse Folders.

4 To view the contents of a folder, click the folder name. The server displays another Folder List screen. For the selected folder, the server displays the subfolders followed by the services.

   You can continue to click on folder names to view subfolders and services in selected folders.

Displaying Information about a Service

The Service Information for ServiceName screen displays a variety of information about a selected service.

To display information about a service

1 Open the Server Administrator if it is not already open.

2 From the Packages menu in the navigation area, click Management.

3 From the Package List, click the package whose services you want to view.

4 Click Browse Services in packagename.

5 Click the name of the service for which you want to display information.

   The server displays a screen that contains the following sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Info</strong></td>
<td>Identifies</td>
</tr>
<tr>
<td></td>
<td>The service by the name of the folder in which the service is contained and the service name.</td>
</tr>
<tr>
<td></td>
<td>The name of the package with which the service is associated.</td>
</tr>
<tr>
<td></td>
<td>The type of service—Flow, Java, C/C++, or WIDL.</td>
</tr>
<tr>
<td></td>
<td>Whether the service is stateless or not.</td>
</tr>
<tr>
<td><strong>Universal Name</strong></td>
<td>Specifies the namespace name and the local name, if any.</td>
</tr>
<tr>
<td><strong>Java-Specific Parameters</strong></td>
<td>For a Java service, identifies the Java class name and method name for the service.</td>
</tr>
<tr>
<td><strong>Access Control</strong></td>
<td>Identifies the ACL assigned to the service, if any. For information about ACLs and services, see “Controlling Access to Services and Files with ACLs” on page 123.</td>
</tr>
</tbody>
</table>
Working with Services

You can perform the following tasks that act on services.

Manually Adding a Service to the Server

If you have Java or C/C++ services that were not created using the Developer, you must manually add them to the server using the jcode utility. See “Building Java Services with Your Own IDE” in the SAP BC Developer Guide for more information.

Testing Services

You can test the operation of a service. This allows you to quickly and easily verify the operation of a service and test it with special-case input values.

Note: The Developer offers a more robust environment for testing services.

To test a service

1. Open the Server Administrator if it is not already open.
2. From the Packages menu in the navigation area, click Management.
3. From the Package List, click the package whose service you want to test.
4. Click Browse Services in packagename.
5. Click on the name of the service you want to test.
6. To test the service, click Test servicename.
   The server displays the Test ServiceName screen.
7. If you want to test the service with input values, fill in the required input information in the Assign Input Values section of the screen and click Test( with inputs).
If you want to test the service without specifying input values, click **Test (without inputs)**.

**Running Services When Packages Are Loaded, Unloaded, or Replicated**

To have the server automatically execute a prescribed set of operations each time the server loads or unloads a package from memory or replicates a package, you can identify startup, shutdown, and replication services. This section provides an overview of startup, shutdown, and replication services.

To identify these services you must use the Developer. See the *SAP BC Developer Guide* for instructions.

**What Is a Startup Service?**

A startup service is one that the server automatically executes when it loads a package. The server loads a package:

- At server initialization (if the package is enabled)
- When someone uses the Server Administrator to reload a package
- When someone uses the Server Administrator to enable a package

Startup services are useful for generating initialization files or assessing and preparing (e.g., setting up or cleaning up) the environment before the server loads a package. However, you can use a startup service for any purpose. For example, you might want to execute a time-consuming service at startup so that its cached result is immediately available to client applications.

**What Is a Shutdown Service?**

A shutdown service is one that the server automatically executes when it unloads a package from memory. If a package is in memory, the server unloads the package:

- At server shutdown or restart
- When someone uses the Server Administrator to disable the package
- Before the server removes the package from memory when someone uses the Server Administrator to reload a package

Shutdown services are useful for executing clean-up tasks such as closing files and purging temporary data. You could also use them to capture work-in-progress or state information before a package unloads.
What Is a Replication Service?

A replication service is one that the server automatically executes when it prepares to release or archive a package. The service executes when the administrator clicks the Create Release or Create Archive button on the Package Publishing screen.

Replication services provide a way for a package to persist state or configuration information so that this is available when the package is activated on the remote server.

Guidelines for Using Startup, Shutdown, and Replication Services

Keep the following guidelines in mind when using startup, shutdown, and replication services.

- When you create a startup or shutdown service, you must register that service in the package with which it will be used. When you create a replication service, you can register any valid service from any loaded package on the server, including the current package itself.
- Because services in a package are not made available to clients until that package's startup services finish executing, you should avoid implementing startup services that access busy remote servers. They will delay the availability of other services in that package.
- You may assign one or more startup services to a package; however, you cannot specify the order in which they will execute. If you have a series of operations that must execute in a specific order, encode the entire sequence within a single service or have a startup service invoke others.

See the SAP BC Developer Guide for instructions.

Running Services in Response to Specific Events

The Event Manager runs on the server, monitoring it for events. An event is a specific action that the Event Manager recognizes and an event handler can react to. An event handler is a service a developer writes to perform an action when a specific event occurs. The Event Manager recognizes a number of different events. For example, an alarm event occurs when the SAP BC Server throws an exception regarding the status or health of the server. The server generates alarm events when a user cannot log on to the server, a port cannot be started, a user is denied access to a port, and so on.

Developers control the Event Manager through the Developer. The server saves information for event types and event subscriptions in the eventcfg.bin file. This file is generated the first time you start an SAP BC Server and is located in the following directory: <sapbc>\server\config. There is no need for you to work with this file directly.
For more information about using the Event Manager, refer to the *SAP BC Developer Guide*.  

**Scheduling Services to Execute at Specified Times**

Use the server's scheduling function to schedule services to execute at times you specify. Services that you schedule are referred to as *user tasks*.  

You can view a list of and update the scheduling options for scheduled user tasks. After the server completes all scheduled executions of a service, it removes the user task from the list. You can also cancel a scheduled user task before the server completes all scheduled executions or temporarily suspend the task's execution.  

In addition to the scheduled user tasks that you set up, the server schedules *system tasks* that it performs for normal system operation. You can view, but not update or cancel, the scheduled system tasks.  

**Note:** You can also perform scheduling by using a set of built-in services. See the *SAP BC Built-In Services Guide* for more information.  

**Scheduling a User Task**

To schedule a user task, you specify:

- **Fully-qualified service name.** To indicate the service that you want the server to execute, you specify the fully-qualified name of the service. For information about specifying service names, see “Fully-Qualified Service Names” on page 228.  

- **User name that you want the server to use when running the service.** The server runs the service as if the user you specify is the authenticated user that invoked the service. If the service is governed by an ACL, be sure to specify a user that is allowed to invoke the service.  

- **Whether you want the scheduled user task to persist after a restart of the server.** Select this option if you want the server to maintain the scheduled user task in the event that the server is restarted before it completes all scheduled executions. When the server is restarted, it will continue to execute the service at the scheduled time(s). If you do not select this option, the user task is removed from the system at server shutdown and the server no longer executes the user task when the server is restarted.  

- When and how often you want the service to run.  
  
  **Once.** The server executes the service a single time.  
  
  **Repeating.** The server executes the service repeatedly at an interval you specify.  
  
  **Complex.** The server runs the service on the day(s) and at the time(s) that you specify either during a specified date range or indefinitely.
Scheduling Services to Execute at Specified Times

Whether or not you want the scheduled user task to run on any SAP BC Server in the cluster. Select this option if you have set up clustering (not recommended) and want the task to run anywhere in your cluster of SAP BC Servers.

Using the Once Option

When you schedule a user task using the **Once** option, the server executes the service one time on the date and at the time that you specify. After the server executes the service at the scheduled time, it removes the user task from the list of scheduled user tasks.

If you indicate that you do *not* want the user task to persist after a restart and the server is restarted before the scheduled execute time, the server will not execute the service.

Using the Repeating Option

When you schedule a user task using the **Repeating** option, the server executes the service for the first time immediately after you add the user task. The server continues to execute the service at the interval you specify. You specify the interval in seconds. For example, if you want the server to execute the service every 24 hours, specify 86400 seconds for the interval.

By default, when you schedule a task to repeat, the server bases the repeat interval on when the task starts. For example, if you specify a repeat interval of 3600 seconds, the server starts the task and schedules the next execution for 3600 seconds after the previous execution began. There may be times however when you want to specify that subsequent executions are to begin after the previous execution has ended. For example, suppose your repeat interval is 60 seconds. If a task sometimes takes longer than 60 seconds to complete, base the interval on when the task ends so that the executions do not overlap. To base the repeat interval on when a task ends, click **Repeat from end of invocation** on the Schedule New Task screen.

With the **Repeating** option, there is no end date or time associated with a repeating user task. The server continues to execute the service until you cancel the scheduled user task, or if the user task does not persist after restart, until the server is restarted. If you want to specify an end date or time, use the **Complex** option, described below.

Using the Complex Option

The **Complex** option offers the greatest flexibility for specifying when you want the server to execute the service.

Specify any combination of the following settings to indicate when and how often you want the server to execute the service:
### Setting Indications...

<table>
<thead>
<tr>
<th>Setting</th>
<th>Indicates...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Start Date</strong></td>
<td>The date on which you want the server to execute the service for the first time. Use the format YYYY/MM/DD to specify the date. If you leave this field blank, the server executes the task at the first date specified by the remaining settings.</td>
</tr>
<tr>
<td><strong>Start Time</strong></td>
<td>The time at which you want the server to begin executing the service. Use the format HH:MM:SS to specify the time (using a 24-hour clock). If you leave this field blank, the server uses 00:00:00 (midnight).</td>
</tr>
<tr>
<td><strong>End Date</strong></td>
<td>The date on which you want the server to execute the service for the last time. Use the format YYYY/MM/DD to specify the date. If you leave this field blank, the server executes the service for an indefinite period of time.</td>
</tr>
<tr>
<td><strong>End Time</strong></td>
<td>The time on the last date at which you want the server to execute the service. Use the format HH:MM:SS to specify the time (using a 24-hour clock). If you leave this field blank, the server uses 00:00:00 (midnight).</td>
</tr>
<tr>
<td><strong>Months</strong></td>
<td>The months (January through December) that you want the server to execute the service.</td>
</tr>
<tr>
<td><strong>Days</strong></td>
<td>The days of the months (0 through 31) that you want the server to execute the service.</td>
</tr>
<tr>
<td><strong>Weekly Days</strong></td>
<td>The days of the week (Sunday through Saturday) that you want the server to execute the service.</td>
</tr>
<tr>
<td><strong>Hours</strong></td>
<td>The hours of the days that you want the server to execute the service.</td>
</tr>
<tr>
<td><strong>Minutes</strong></td>
<td>The minute of the hour that you want the server to execute the service.</td>
</tr>
</tbody>
</table>

The server combines all your selections to determine when to execute the service. If you do not select an item in one of the above settings, the server assumes all items for the selection. For example, if you do not specify a month, the server assumes you want the service to execute every month. If you do not select any items for any of the settings, the server assumes you want the service to execute every month, every day, all week days, every hour, and every minute; in other words, the server executes the service every minute from the time you add the task.

If you use the **Start Date** and **End Date** to specify a date range, the server executes the service at the scheduled times until the end of the time period. At the end of the time period, the server removes the user task from the list of scheduled user tasks. If you indicate that you do not want the user task to persist after a restart, the server will not continue to execute the service if the server is restarted before the schedule time period elapses.

If you do not specify an **End Date** to specify a date range, the server executes the service for an indefinite period of time. The scheduled user task remains in the system until you cancel the scheduled user task, or if the user task does not persist after restart, until the server is restarted.
The following shows examples of how to use the Complex option settings:

<table>
<thead>
<tr>
<th>If you want the service to execute...</th>
<th>For this setting:</th>
<th>Specify...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The 28th day of every month at midnight for the year 2000.</td>
<td>Start Date</td>
<td>2000/01/01</td>
</tr>
<tr>
<td></td>
<td>Start Time</td>
<td>00:00:00 (or leave blank)</td>
</tr>
<tr>
<td></td>
<td>End Date</td>
<td>2000/12/31</td>
</tr>
<tr>
<td></td>
<td>End Time</td>
<td>00:00:00 (or leave blank)</td>
</tr>
<tr>
<td></td>
<td>Months</td>
<td>no selection</td>
</tr>
<tr>
<td></td>
<td>Month Days</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>Week Days</td>
<td>no selection</td>
</tr>
<tr>
<td></td>
<td>Hours</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Minutes</td>
<td>0</td>
</tr>
<tr>
<td>Every Monday in the months of January, February, and March at 2:30 p.m. for an indefinite period of time.</td>
<td>Start Date</td>
<td>leave blank</td>
</tr>
<tr>
<td></td>
<td>Start Time</td>
<td>leave blank</td>
</tr>
<tr>
<td></td>
<td>End Date</td>
<td>leave blank</td>
</tr>
<tr>
<td></td>
<td>End Time</td>
<td>leave blank</td>
</tr>
<tr>
<td></td>
<td>Months</td>
<td>January, February, March</td>
</tr>
<tr>
<td></td>
<td>Month Days</td>
<td>no selection</td>
</tr>
<tr>
<td></td>
<td>Week Days</td>
<td>Monday</td>
</tr>
<tr>
<td></td>
<td>Hours</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Minutes</td>
<td>30</td>
</tr>
<tr>
<td>Every hour of every Tuesday of the month of June, 2000.</td>
<td>Start Date</td>
<td>2000/06/01</td>
</tr>
<tr>
<td></td>
<td>Start Time</td>
<td>00:00:00 (or leave blank)</td>
</tr>
<tr>
<td></td>
<td>End Date</td>
<td>2000/06/30</td>
</tr>
<tr>
<td></td>
<td>End Time</td>
<td>00:00:00 (or leave blank)</td>
</tr>
<tr>
<td></td>
<td>Months</td>
<td>June</td>
</tr>
<tr>
<td></td>
<td>Month Days</td>
<td>no selection</td>
</tr>
<tr>
<td></td>
<td>Week Days</td>
<td>Tuesday</td>
</tr>
<tr>
<td></td>
<td>Hours</td>
<td>no selection</td>
</tr>
<tr>
<td>If you want the service to execute...</td>
<td>For this setting</td>
<td>Specify...</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>Every minute of every hour of every Tuesday of the month of June, 2000.</td>
<td>Start Date</td>
<td>2000/06/01</td>
</tr>
<tr>
<td></td>
<td>Start Time</td>
<td>00:00:00 (or leave blank)</td>
</tr>
<tr>
<td></td>
<td>End Date</td>
<td>2000/06/30</td>
</tr>
<tr>
<td></td>
<td>End Time</td>
<td>00:00:00 (or leave blank)</td>
</tr>
<tr>
<td></td>
<td>Months</td>
<td>June</td>
</tr>
<tr>
<td></td>
<td>Month Days</td>
<td>no selection</td>
</tr>
<tr>
<td></td>
<td>Week Days</td>
<td>Tuesday</td>
</tr>
<tr>
<td></td>
<td>Hours</td>
<td>no selection</td>
</tr>
<tr>
<td></td>
<td>Minutes</td>
<td>no selection</td>
</tr>
</tbody>
</table>
Scheduling Services to Execute at Specified Times

To schedule the execution of a service

1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Scheduler.
3. Click Create a scheduled task.
4. Set the Service Information parameters as follows:

   For this parameter  Specify...
   folder:subfolder:service  The fully-qualified service name of the service you want the server to execute.
   Run As User  The user name you want the server to use when running the service.
   Persistence  Whether you want the server to maintain this user task in the event that the server is restarted. Check the Persist after restart checkbox if you want the server to execute the service at the scheduled time(s) if the server is restarted.
   Clustering  Whether you want the task to run anywhere in your cluster of SAP BC Servers.

5. Select Run Once, Repeating (Simple), or Repeating (Complex) to indicate when and how often you want the server to execute the service.

   If you select...  Specify...
   Run Once  The date on which you want the server to execute the service in the Date field. Use the format YYYY/MM/DD to specify the date. For example, if you want the server to execute the service on March 11, 2000, specify 2000/03/11.

   The time at which you want the server to execute the service in the Time field. Use the format HH:MM:SS to specify the time (using a 24-hour clock). For example, if you want the server to execute the service at 1:00:00 a.m., specify 1:00:00; if you want the server to execute the service at 1:00:00 p.m., specify 13:00:00.

   For more information about using this option, see “Using the Once Option” on page 235.
CHAPTER 12  Managing Services

If you select… Specify…

Repeating (Simple) The number of seconds that you want the server to wait between executions of the service in the Interval field.

If you want the server to wait for a service to complete execution before it starts the next scheduled execution of the service, check Repeat from end of invocation. For example, suppose the GetData service is scheduled to run every minute, but sometimes takes longer than that to complete. By default, the server will start the next execution even though the previous one has not yet completed. If you check the Repeat from end of invocation box, the server will wait for the service to complete before beginning its next scheduled execution.

For more information about using this option, see “Using the Repeating Option” on page 235.

Repeating (Complex) When you want the first execution of this service by specifying a beginning date and time in the Start Date and Start Time fields. For Start Date, use the format YYYY/MM/DD. For Start Time, use the format HH:MM:SS (using a 24-hour clock). For example, if you want the time period in which to run this service to begin be on May 3, 2000 at 1:00:00 p.m., specify 2000/05/03 for Start Date and 13:00:00 for Start Time. If you omit the Start Date, the first execution occurs on the first date as indicated by the Run Mask parameters. If you omit Start Time, the server uses 00:00:00 (midnight).

When you want the last execution of this service to run by specifying an ending date and time in the End Date and End Time fields. For End Date, use the format YYYY/MM/DD. For the End Time, use the format HH:MM:SS (using a 24-hour clock). For example, if you want the time period in which to run this service to end on June 4, 2000 at 2:00:00 a.m., specify 2000/06/04 for End Date and 02:00:00 for End Time. Omitting End Date indicates that you want this service to execute for an indefinite period of time. If you omit End Time, the server uses 00:00:00 (midnight).

Use the Run Mask parameters to indicate when you want the server to execute the service. For examples of setting these parameters, see “Using the Complex Option” on page 235.

For more information about using this option, see “Using the Complex Option” on page 235.

6 Click Save Tasks.
Scheduling Services to Execute at Specified Times

Viewing Scheduled User Tasks
Perform the following procedure to view the user tasks you have scheduled.

To view scheduled user tasks
1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Scheduler.

Updating Scheduled User Tasks
Perform the following procedure to change the scheduling parameters for scheduled user tasks.

To update a scheduled user task
1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Scheduler.
3. Click the service name for the user task you want to update.
4. Update the scheduling options for the selected user task. For information about the options you can specify, see “Scheduling a User Task” on page 234.
5. Click Update Tasks.

Suspending Scheduled User Tasks
Perform the following procedure to suspend all scheduled executions of a service. When you suspend a user task, it remains scheduled, but does not execute until you resume its execution. If a suspended task has been marked Persist after restart, meaning the server maintains the task in the event the server restarts, the task remains suspended when the server restarts. If a task expires while suspended, the server removes it from the task list.

To suspend a scheduled user task
1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click Scheduler.
3. Locate the task in the Services list, and click the icon in the Active column to suspend the task. The server displays a screen to confirm you want to suspend the task. Click OK.
The server replaces the ✔️ icon with **Suspended** to indicate that the task is now suspended.

### Resuming Suspended Scheduled User Tasks

Perform the following procedure to resume all scheduled executions of a task that has been suspended.

**To resume execution of a suspended user task**

1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click **Scheduler**.
3. Locate the task in the **Services** list, and click **Suspended** in the Active column to activate the task. The server displays a screen to confirm you want to resume the task. Click **OK**.

The server replaces **Suspended** with the ✔️ icon to indicate that the task is available to execute again.

### Canceling Scheduled User Tasks

Perform the following procedure to cancel a user task before all scheduled executions of the service are complete.

**To cancel a scheduled user task**

1. Open the Server Administrator if it is not already open.
2. In the Server menu of the navigation area, click **Scheduler**.
3. Click the ☒ icon in the **Remove** column for the user task you want to cancel. The server issues a prompt to verify that you want to cancel the user task. Click **OK**.

### Viewing the Scheduled System Tasks

The server needs to perform system tasks periodically, such as expiring sessions and rotating the log files. The server schedules these tasks. Perform the following procedure to view the scheduled system tasks.

**To view the scheduled system tasks**
1 Open the Server Administrator if it is not already open.
2 In the Server menu of the navigation area, click Scheduler.
3 Click View system tasks.

The server displays the System Tasks screen. It lists the names of each scheduled task, the next date and time the server is to execute the task, and how often (Interval) the server is to execute the task.
Configuring Guaranteed Delivery

- About Guaranteed Delivery .................................................. 246
- Configuring the Server for Guaranteed Delivery .................. 247
- Administering Guaranteed Delivery ................................. 251
- Shutting Down Guaranteed Delivery .................................. 251
- Reinitializing Guaranteed Delivery .................................... 251
- Testing Guaranteed Delivery .............................................. 253
- Specifying an E-Mail Address and SMTP Server for Error Messages .......... 254
About Guaranteed Delivery

Use the guaranteed delivery capabilities of the SAP BC Server to ensure guaranteed one-time execution of services.

The SAP BC Server guaranteed delivery capabilities ensure the following occur despite transient failures:

- Requests to execute services from clients are delivered to the server
- Services are executed once, and only once
- Responses from the execution of the services are delivered to the client

The SAP BC guaranteed delivery capabilities protect against transient failures that might occur on the network, in the client, or on the server. A transient failure is a failure that can correct itself during a specified period of time. If a request cannot be delivered to the server due to a transient failure, the request is resubmitted; if the problem has corrected itself, the request is successfully delivered on a subsequent attempt. You determine what constitutes a transient error by specifying a time-to-live (TTL) period for a guaranteed delivery transaction and, optionally, the number of times a transaction should be retried.

Because an SAP BC Server can act as either a server or a client in a guaranteed delivery transaction, the guaranteed delivery capabilities of the server handle both inbound transactions and outbound transactions. When a client invokes a service on a server, the server is acting as a server. If a service uses guaranteed delivery to invoke a service on another SAP BC Server, the server that invokes the service is the client.

The guaranteed delivery capabilities allow you to build robust, transaction-based client applications without having to embed complex error handling code to respond to transient failures.

Important! Use the guaranteed delivery capabilities with stateless (i.e., atomic) transactions because state information cannot be maintained from one request to the next. As a result, guaranteed delivery capabilities cannot be used with multi-request conversational services.
Configuring the Server for Guaranteed Delivery

This section describes configuration settings that the SAP BC Server uses for guaranteed delivery transactions. Most of the settings have defaults. In general, you will want to use the defaults; however, you can specify alternate settings in the server.cnf server configuration file. You can change these settings by using the Settings>Extended screen of the Server Administrator as described on page “Working with Extended Configuration Settings” on page 100.

There are settings for both inbound and outbound guaranteed delivery transactions.

Settings Shared by Both Inbound and Outbound Transactions

watt.server.txMail
Use the watt.server.txMail setting to specify the e-mail address of an administrator to notify when guaranteed delivery capabilities are disabled due to error (for example, if the server encounters a disk full condition). An example of using this setting is watt.server.txMail=SAPBCAdmin@YourCompany.com.

There is no default for this setting.

watt.server.smtpServer
Use the watt.server.smtpServer setting to specify the domain name (e.g., purple.sap.com) or IP address (e.g., 132.906.19.22) of the SMTP server you want the SAP BC Server to use when sending an e-mail about an error during guaranteed delivery. An example of using this setting is watt.server.smtpServer=132.906.19.22

There is no default for this setting.

When an administrator receives an e-mail notification of an error, the administrator should correct the problem, then use the Server Administrator to reinitialize guaranteed delivery capabilities. For instructions on how to reinitialize guaranteed delivery, refer to “Reinitializing Guaranteed Delivery” on page 251.

Settings for Inbound Transactions

For inbound transactions, the server maintains a job store of transactions and the status of each. Periodically, the server sweeps the job store to remove expired transactions; that is, to remove transactions that have an elapsed time-to-live (TTL) period. For inbound requests, the client must specify the TTL for a transaction.

In addition to the job store, the server maintains an audit-trail log of all operations it performs for inbound transactions.
The following describes the inbound transaction settings you can configure.

<table>
<thead>
<tr>
<th>You can configure:</th>
<th>Using this setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where the server maintains the job store</td>
<td>watt.server.tx.jobdir</td>
</tr>
<tr>
<td>How often the server sweeps the job store to remove expired transactions</td>
<td>watt.server.tx.sweepTime</td>
</tr>
<tr>
<td>How the server updates the status of PENDING transactions when a heuristic failure occurs</td>
<td>watt.server.tx.heuristicFailRetry</td>
</tr>
<tr>
<td>Where the server maintains the audit-trail log</td>
<td>watt.server.tx.logfile</td>
</tr>
</tbody>
</table>

**watt.server.tx.jobdir**

Use the `watt.server.tx.jobdir` setting to specify the directory in which the server maintains the job store for inbound transactions. The server registers each guaranteed delivery transaction it receives in the job store. It uses the job store to manage guaranteed delivery inbound transactions.

The default is: `logs\jobsin`

**watt.server.tx.sweepTime**

Use the `watt.server.tx.sweepTime` setting to specify the number of seconds between sweeps (clean up) of the job store of inbound transactions. The server sweeps the job store to remove expired transactions.

The default is: 60 seconds

**watt.server.tx.heuristicFailRetry**

Use the `watt.server.tx.heuristicFailRetry` setting to indicate whether the server is to re-execute services for transactions in the job store that are PENDING when the server is restarted after a failure. If a transaction is PENDING, the service began but did not complete execution when the server failed.

Because the server cannot determine the exact status of a service request, the server considers the guaranteed transaction to have encountered a heuristic failure. You can configure the server to respond to heuristic failures as appropriate. The default `watt.tx.heuristicFailRetry` setting causes the server to execute a service at least one time at the risk of re-executing it a subsequent time after a heuristic failure. Alternatively, you can reconfigure the setting to guarantee that a service is executed at most one time at the risk of not executing a service due to a heuristic failure.

If the `watt.tx.heuristicFailRetry` setting is true, the server resets the transaction status from PENDING to NEW, and the server will retry the service. When the setting is true, a request to execute a service can only fail if the transaction expires before the server executes the service. (The client specifies the settings that indicate when a transaction expires.)
If the `watt.tx.heuristicFailRetry` setting is false, the server resets the transaction status from PENDING to FAIL to indicate the heuristic failure; the server does not retry the service. When the setting is false, a request to execute a service can fail due to a heuristic failure or due to the transaction expiring.

The default is: true

`watt.server.tx.logfile`

Use the `watt.server.tx.logfile` setting to specify the file in which the server maintains an audit-trail log of all operations it processes for inbound guaranteed delivery transactions.

The default is: `logs\txin.log`

## Settings for Outbound Transactions

You can disable the use of guaranteed delivery for outbound transactions. However, if you allow guaranteed delivery for outbound transactions, the server maintains a separate job store for the transactions. Similar to the inbound job store, the server keeps the status of each transaction in the outbound job store. If a service request fails, the server waits a specified amount of time before resubmitting the request. The server periodically processes the job store to identify transactions that it needs to submit.

The server maintains a thread pool to service pending outbound requests. You can configure how many client threads the server should maintain in the thread pool.

The server also maintains a separate audit-trail log of all operations it performs for outbound transactions.

The following describes the settings you can configure.

<table>
<thead>
<tr>
<th>You can configure</th>
<th>Using this setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whether you want to disable guaranteed delivery for outbound transactions.</td>
<td><code>watt.tx.disabled</code></td>
</tr>
<tr>
<td>Where the server maintains the job store.</td>
<td><code>watt.tx.jobdir</code></td>
</tr>
<tr>
<td>The default TTL value for outbound transactions.</td>
<td><code>watt.tx.defaultTTLMins</code></td>
</tr>
<tr>
<td>How long the server should wait before resubmitting failed requests.</td>
<td><code>watt.tx.retryBackoff</code></td>
</tr>
<tr>
<td>How often the server processes the job store to identify transactions that it needs to submit.</td>
<td><code>watt.tx.sweepTime</code></td>
</tr>
</tbody>
</table>
You can configure:  
Using this setting

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>watt.tx.disabled</td>
<td>Use the <code>watt.tx.disabled</code> setting to specify that you want to disable the use of guaranteed delivery for outbound requests. By default, the server allows the use of guaranteed delivery for outbound transactions. The default is: <code>false</code></td>
</tr>
<tr>
<td>watt.tx.jobdir</td>
<td>Use the <code>watt.tx.jobdir</code> setting to specify the directory in which the server maintains a job store of outbound transactions. The server registers each guaranteed delivery transaction it requests in the job store. It uses the job store to manage outbound guaranteed delivery transactions. The default is: <code>logs\jobsout</code></td>
</tr>
<tr>
<td>watt.tx.defaultTTLMins</td>
<td>Use the <code>watt.tx.defaultTTLMins</code> setting to specify the default time-to-live (TTL) value for outbound guaranteed delivery transactions. Specify the number of minutes you want the server to maintain outbound transactions in the job store when a service initiating an outbound transaction does not specify a TTL value. The default is: <code>30</code></td>
</tr>
<tr>
<td>watt.tx.retryBackoff</td>
<td>Use the <code>watt.tx.retryBackoff</code> setting to specify the number of seconds to wait after a service request failure before the Job Manager resubmits the request to execute the service to the SAP BC Server. The default is: <code>60</code></td>
</tr>
<tr>
<td>watt.tx.sweepTime</td>
<td>Use the <code>watt.tx.sweepTime</code> setting to specify the number of seconds between sweeps of the job store of outbound transactions. The server sweeps the job store to identify transactions that it needs to submit. The default is: <code>60</code></td>
</tr>
<tr>
<td>watt.tx.jobThreads</td>
<td>Use the <code>watt.tx.jobThreads</code> setting to specify the number of client threads you want to make available in a thread pool to service pending requests. The default is: <code>5</code></td>
</tr>
</tbody>
</table>
watt.tx.logfile
Use the watt.tx.logfile setting to specify the file in which the server maintains an audit-trail log of all operations it processes for outbound guaranteed delivery transactions.

The default is: logs\txout.log

## Administering Guaranteed Delivery

When you initialize the server, it initializes guaranteed delivery capabilities. You can use the Server Administrator to shut down, reinitialize, and test guaranteed delivery.

### Shutting Down Guaranteed Delivery

You can shut down and re-enable guaranteed delivery capabilities without having to shut down the server.

You might want to shut down guaranteed delivery to perform some administration functions, such as correcting configuration errors or starting a new audit-trail log. (To start a new audit-trail log, move or rename the existing log—the server automatically starts a new log if one does not already exist.)

**To shut down guaranteed delivery**

1. Open the Server Administrator if it is not already open.
2. In the **Packages** menu of the navigation area, click **Management**.
3. In the list of packages, click **WmRoot**.
4. Click **Browse Services in WmRoot**.
5. In the list of services, click **wm.server.tx:shutdown**.
6. Click **Test shutdown**. The server displays the test screen for the wm.server.tx:shutdown service.
7. Click **Test (without inputs)**. The server disables the guaranteed delivery capabilities for inbound transactions.

### Reinitializing Guaranteed Delivery

Reinitialize guaranteed delivery if it becomes disabled. This section describes the procedures to reinitialize guaranteed delivery for inbound transactions and outbound transactions.
Inbound Transactions

If you shut down the guaranteed delivery capabilities to correct a configuration problem or to make an administrative change, you can reinitialize guaranteed delivery using the Server Administrator.

You can also use this procedure to reinitialize guaranteed delivery if it becomes disabled due to an error (for example, because of a disk full condition or if the server could not locate the job store). Reinitialize guaranteed delivery after you correct the problem.

To reinitialize guaranteed delivery for inbound transactions

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. In the list of packages, click WmRoot.
4. Click Browse Services in WmRoot.
5. In the list of services, click wm.server.tx:init.
6. Click Test init. The server displays the test screen for the wm.server.tx:init service.
7. Click Test (without inputs). The server reinitializes the guaranteed delivery capabilities for inbound transactions.

Outbound Transactions

If guaranteed delivery capabilities for outbound transactions become disabled due to an error (for example, because of a disk full condition or if the server could not locate the job store), use this procedure to reinitialize guaranteed delivery after you correct the problem.

To reinitialize guaranteed delivery for outbound transactions

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. In the list of packages, click WmRoot.
4. Click Browse Services in WmRoot.
5. In the list of services, click wm.server.tx:resetOutbound.
6. Click Test resetOutbound. The server displays the test screen for the wm.server.tx:resetOutbound service.
7. Click Test (without inputs). The server reinitializes the guaranteed delivery capabilities for outbound transactions.
Testing Guaranteed Delivery

You can invoke a service from the Server Administrator to ensure that the configuration is correct and guaranteed delivery is functioning properly.

To test guaranteed delivery

1. Open the Server Administrator if it is not already open.
2. In the Packages menu of the navigation area, click Management.
3. In the package list, click WmRoot.
4. Click Browse Services in WmRoot.
5. Click wm.server.tx:start.
6. Click Test wm.server.tx:start. Type the number of minutes for the transaction to live in the ttl (time-to-live) field. Then, click Test (with inputs). The server returns a tid (transaction id). Copy the tid; you will need it in the next two steps. Click your browser’s back button until you return to the list of services in package WmRoot.
7. In the list of services, click wm.server.tx:execute.
8. Click Test execute. The server displays the test screen for the wm.server.tx:execute service. Fill in the input information:
   1. Paste the tid (transaction id) in the tid field.
   2. Type the folder name for the service you want to invoke in the ifc field (e.g., wm.server).
   3. Type the service name for the service you want to invoke in the svc field (e.g., ping)

    **Note:** Use a service that does not require input.

    Then, click Test (with inputs). The server invokes the service.
9. Click your browser’s back button until you return to the list of services in package WmRoot.
10. Click wm.server.tx:end.
11. Click Test end. The server displays the test screen for the wm.server.tx:end. Paste the tid (transaction id) in the tid field. Then, click Test (with inputs).
CHAPTER 13 Configuring Guaranteed Delivery

Specifying an E-Mail Address and SMTP Server for Error Messages

When you configure guaranteed delivery, you must specify the e-mail address to which the SAP BC Server can issue an error message if guaranteed delivery becomes disabled. In addition, you must specify the domain name or IP address of the SMTP server you want to handle these e-mails.

To set the e-mail address and SMTP server using the Server Administrator

1. Open the Server Administrator if it is not already open.
2. In the Settings menu of the navigation area, click Logging.
3. Click Edit Logging Settings.
4. Type the e-mail address for the administrator to whom you want the server to send error notification in the Transaction Email field in the Email Notification section of the screen.
5. Type the domain name (e.g., purple.sap.com) or IP address (e.g. 132.906.19.22) of the SMTP server you want the SAP BC Server to use.
6. Click Save Changes.
CHAPTER 14

Configuring Access to Database Systems

- Database Connection Information .................................................. 256
- Pre-Loading JDBC Drivers at Server Initialization ............................ 256
- Making the Java Classes for JDBC Drivers Accessible ....................... 257
- Identifying Database Aliases .......................................................... 257
- Updating Information for Database Aliases ..................................... 261
- Deleting Database Aliases ............................................................... 262
- Monitoring Connection Pools ......................................................... 262
Database Connection Information

The server requires information about databases so it can connect to them. You specify database connection information by configuring database aliases. You must configure database aliases if you want to create database flow services using the Server Administrator.

The following lists the information you specify for a database alias:

- **The type of database and where it is located.** JDBC uses a URL naming scheme to locate and identify databases. Each JDBC driver responds to a slightly different syntax for these URLs. Consult the documentation for the driver you are using for details. If you are using the JDBC-ODBC bridge driver, the URL is jdbc:odbc:your-datasource. For the SAPDB driver, the URL is jdbc:sapdb://<your host>/<your database>.

- **The user name and password required to connect to the database.** If the database requires a user name and password to connect to it, you must supply the server with the user name and password it should supply.

- The JDBC driver. You identify the fully-qualified Java class name of the JDBC driver to use for this connection (for example, com.company.jdbc.Driver). This class must be in the server’s classpath.

The server uses the database connection information to connect to a database when:

- You use the Server Administrator to display information about database tables when creating a database service (Generate from table). The server must connect to the database to get information about the structure of the database.

- The server executes a database service that was created using the Server Administrator.

*Note:* For information about creating services that access a database, refer to the *SAP BC Developer Guide*.

Pre-Loading JDBC Drivers at Server Initialization

You can have the server preload the JDBC drivers when it initializes. When the JDBC drivers are loaded at initialization, it increases the performance of the first run of the first database service that requires a specific JDBC driver. If the server does not have a JDBC driver loaded when a database service requires it, the server must load the JDBC driver when the database server executes. Also, if you preload JDBC drivers, the server displays the list of drivers in a drop down list on the screens that require a JDBC driver name. When the drop down list is available, you can select a driver from it rather than typing in the name of a driver.

To have the server load JDBC drivers when it initializes, specify a comma-delimited list of JDBC drivers that you want preloaded in the watt.server.jdbc.driverList field in the...
Identifying Database Aliases

server.cnf file. If this field does not exist in the server.cnf file, you can add it. The server.cnf file is located in the <sapbc>\server\config directory. To change this setting use the Settings>Extended screen from the Server Administrator as described on page “Working with Extended Configuration Settings” on page 100.

Making the Java Classes for JDBC Drivers Accessible

The server requires access to the Java classes for each JDBC driver that it is to use. You need to place the Java classes in a location that the server can access. Typically, you place the Java classes in the server’s classpath.

To place the classes in the server’s classpath, place the .zip or .jar file containing the classes in the <sapbc>\server\lib\jars directory. If the jars subdirectory does not exist, create it. The server will automatically add the .zip or .jar libraries to its classpath.

If a patch you are unzipping or “unjarring” overwrites existing classes, be sure to back up the existing classes in case the patch does not work.

Note: If you are using an ODBC data source on a Windows platform, you do not need to change the server’s classpath. If you are using Sun’s JRE (the default installed with the SAP BC Server), the JDBC-ODBC bridge driver is sun.jdbc.odbc.JdbcOdbcDriver. If you are using Microsoft’s JView, the JDBC-ODBC bridge driver is com.ms.jdbc.odbc.JdbcOdbcDriver. In either case, an ODBC data source corresponding to your database must exist.

If the server does not have access to the Java classes for the JDBC driver, users receive the following error when the server attempts to connect to the database:

Couldn’t make connection: No suitable driver

If the driver loads successfully but cannot connect to the database for some reason, the resulting error message usually includes some driver-specific codes or messages. The most common source of such errors is an invalid user name or password on the database. Consult your driver documentation for details.

Identifying Database Aliases

Identify a database alias to specify the connection information that the server must supply to connect to a database. Set up aliases if:

- You want developers to be able to define database flow services using the SAP BC Server Administrator.
- You want to maintain all database connection parameters in one place to allow for easier maintenance. For example, you can easily update the information if a password changes or a database moves.
To better manage requests for database connections you can specify that the server use *database connection pools*. With database connection pools in effect, the SAP BC Server controls the number of database connections it makes to a database, preventing the database from exceeding ITS connection limit and rejecting requests. If a request exceeds a database connection pool’s limit, the server blocks it and tries it again later.

See “Database Connection Pooling” on page 261 for more information about how database connection pools work and how to set them up.

**To identify a database alias**

Use the following procedure to identify database aliases to your SAP BC Server:

1. Open the Server Administrator if it is not already open.
2. In the *Adapters* menu of the navigation area, click *Database*. The server displays a database management screen.
3. Click *Alias Management* in the navigation area.
4. Click *Add*. The server displays the *New DB Alias* screen.
5. Set the *Details* parameters as follows:
Identifying Database Aliases

For this parameter  Specify...

**Alias**
The alias name that you want to use for the database. You can give the database any alias name you want. There is no restriction on the characters you can specify for or a limit to the length of the alias name.

**DB URL**
The URL for the database, for example, jdbc:odbc:Support.

**DB Username**
The user name that the server must supply to log into the specified database. If a user name and password are not required, leave this field blank.

**DB Password**
The password that the server must supply to log into the selected database. If a user name and password are not required, leave this field blank.

**DB Driver**
The name of the Java class for the JDBC driver.

If the server has any drivers currently loaded, the server displays a **Loaded Drivers** field that has a drop down list containing the drivers that are loaded. If you select a driver from the list, the server places the driver name in the **DB Driver** field.

If you do not specify a driver, the server uses the default driver specified in the `watt.server.jdbc.defaultDriver` parameter in the server.cnf file. If the server.cnf file does not contain the `watt.server.jdbc.defaultDriver` field, the server uses the Sun JDBC-ODBC bridge.

To see the value of the default driver, use the Settings>Extended screen from the Server Administrator as described on “Working with Extended Configuration Settings” on page 100.

**Important!** Be sure the server has access to the Java classes for the driver. For more information about making the Java classes available, see “Making the Java Classes for JDBC Drivers Accessible” on page 257.

If you are using one of the following JDBC-ODBC bridges on a Windows platform, the server has access to the Java classes for the driver:

- If you are using Sun’s JVM, the driver name is `sun.jdbc.odbc.JdbcOdbcDriver`.
- If you are using Microsoft’s Jview, the driver name is `com.ms.jdbc.odbc.JdbcOdbcDriver`.
For this parameter Specify...

Minimum Connections

The minimum number of connections to maintain in the database connection pool. The server creates these connections when it first receives a request to connect to the database. If more connections are needed, the server creates additional connections until the value for **Maximum Connections**, described below, is reached. The default is 0.

If the pool for this alias has already been created, connections are not filled up to the new minimum value immediately. They are filled up step by step when new requests are arriving which need a new connection.

See “Database Connection Pooling” on page 261 for more information.

Maximum Connections

The maximum number of connections to maintain in the database connection pool. When the number of connection requests reaches this value, the server blocks the requests. The default is 1.

See “Database Connection Pooling” on page 261 for more information.

Expiration Time (ms)

The amount of time the server waits before discarding an inactive connection from the database connection pool. The server only discards connections if the pool contains more than the minimum number of connections. Afterwards, if more connections are needed than exist in the pool, the server creates new ones. The default expiration time is 60000 milliseconds.

See “Database Connection Pooling” on page 261 for more information.

Test SQL

**Note:** Make sure that the statement is valid for this database. Otherwise the pool cannot be created. Choose a resource-saving statement.

Optional. You can enter an SQL statement which is used to ping the database. This helps the BC to identify broken connections.

6 Click **Submit**.
Database Connection Pooling

For greater scalability SAP BC uses database connection pooling. With this feature, the server creates a pool of connections for each database defined to the SAP BC Server. The server maintains these connections, creating and dropping them as needed.

The pool controls the number of connections a server can have to the database at any given time. For example, if your database allows 50 connections and you have five servers that connect to the database, you can limit each server’s database connection pool to 10. This way, the number of connections to the database never exceeds 50 and no requests will be rejected due to lack of available slots. If the number of connection requests to a server exceeds 10, that server will block the request and try it again later. The request will not be rejected.

Without database connection pooling, each server would obtain as many connections to the database as requested until the database reaches its limit of 50. Subsequent requests will be rejected by the database until the number of database connections drops down below 50.

Database connection pooling applies to all databases defined to the server. In other words, the server will maintain a connection pool for each database for which an alias is defined. You can, however, control the characteristics of each pool individually. For example, you might specify more connections for a large, busy database than for a small, relatively quiet one.

Updating Information for Database Aliases

If the connection information changes for a database, update the database alias information to identify the new connection information.

To update information for a database alias

1. Open the Server Administrator if it is not already open.
2. In the Adapters menu of the navigation area, click Database. The server displays a database management screen.
3. Click Alias Management in the navigation area.
4. From the Current Data Sources field, select the database alias for the database you want to update.
5. Click Edit. The server displays the Edit Alias Information screen.
6. Set the Details parameters. For information about what to specify for each of the fields, refer to the procedure in “Identifying Database Aliases” on page 257.
7. Click Submit.
Deleting Database Aliases

When you no longer need access to a database, you can delete the database alias for the database.

To delete a database alias

1. Open the Server Administrator if it is not already open.
2. In the Adapters menu of the navigation area, click Database. The server displays a database management screen.
3. Select the database alias you want to delete from Current Data Sources.
4. Click Delete. The server issues a prompt to verify that you want to delete the database alias. Click OK to delete the database alias.

Monitoring Connection Pools

In 4.8, new DSP’s have been introduced for monitoring the connection pools.

To view connection pool information

1. Open the Server Administrator if it is not already open.
2. In the Adapters menu of the navigation area, click Database. The server displays a database management screen.
3. Click Connection Pool Info.

You now see all currently existing DB-Pools (or none if no pool exists). The Pool configuration parameters are displayed, and some current usage parameters:

- **Current Connections**: how many connections are opened
- **Available Connections**: how many connections are available for new requests
- **Waiting Requests**: the number of the currently waiting requests
- **Average Waiting Time**: average waiting time of the currently waiting requests (meaning the average of the wait start values!)
- **Pool Age**: the age of the pool (the pool is created after BC startup not immediately, but when the first DB request occurs)
- **Last Expiry Run**: when was the last expiry run? This closes unused connections which have expired.
You can follow the JDBC-URL hyperlink to get more detailed information about the pool, including statistics counter and connection details:

- **CreatedConnections**: how many connections were physically created since pool start?
- **ClosedValidConnections**: how many connections were removed normally?
- **ClosedExpiredConnections**: how many connections were removed because they expired?
  Note: Only available connections can expire!
- **ClosedStaleConnections**: how many connections were detected as stale? Stale means: the testSQL statement did not succeed. This can give hints about stability problems with the network/DB connection
- **ClosedIllegalStateConnections**: how many connections could not be put into clearTransaction status?
- **AllocatedConnections**: how many connections were allocated since pool start?
- **ReleasedValidConnections**: how many connections were released since pool start?
- **ReleasedLostSessionConnections**: how many connections were released because their session did not exist anymore? This can give a clue for missing error handling in a user service
- **StartedWaitRequests**: how many requests started to wait for an available connection since pool start?
- **EndedWaitRequests**: how many requests ended to wait for an available connection since pool start?
- **AverageWaitingTime**: average waiting time for all requests which returned from wait. Avoid rounding issues
- **ReleasedNotInPoolExceptions**: how many connections were tried to be removed, but not found in the pool? This is a clue for incorrect use of the pool
- **EmptySessionExceptions**: how many empty session calls? Connections can only be created with a non-null Service.getSession() object
- **TimeoutInGetConnExceptions**: how many requests timed out while trying to get a connection?
- **InterruptedExceptions**: how many requests were interrupted while trying to get a connection?
- **ConnectionDetails**: show some interesting connection attributes, e.g. the active SQL statement (if any). This can give hints about DB lock problems.
SAP BC Server Deployment Checklist

- Introduction ......................................................... 266
- STAGE 1: Installation .................................................. 266
- STAGE 2: Basic Configuration ....................................... 266
- STAGE 3: Setting Up Users, Groups, and ACLs .................. 268
- STAGE 4: Publishing Packages ...................................... 269
- STAGE 5: Configuring Database Connections .................... 270
- STAGE 6: Installing Run-Time Classes ........................... 270
- STAGE 7: Setting Up Security ....................................... 271
- STAGE 8: Startup and Test .......................................... 272
- STAGE 9: Archive Sources .......................................... 272
Introduction

This appendix contains a useful checklist for setting up your SAP BC Server. It describes the steps to perform to put an SAP BC Server into production. The process is comprised of several stages. You should complete one stage before advancing to the next.

STAGE 1: Installation

Complete the following steps to install, run, and test the SAP BC Server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Install the Integration Server.</strong> Follow the instructions in <em>webMethods Integration Server and webMethods Developer Installation Guide</em> to install and test the Integration Server.</td>
</tr>
</tbody>
</table>
| 2    | **Change default passwords.** Use the Server Administrator to assign new passwords to the following user accounts:  
  - The “Administrator” user account.  
  - The “Replicator” user account.  
  - The “Developer” user account.  
  For instructions on how to change passwords, refer to “Changing Passwords and Password Requirements” on page 50. |

STAGE 2: Basic Configuration

Use the Server Administrator to configure the way in which the server will send outbound requests, accept inbound requests, expire sessions, and issue error messages.
### STAGE 2: Basic Configuration

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Set up the ports.</strong> Use the <strong>Ports</strong> screen to specify the ports on which the server will listen for requests.</td>
<td><strong>Tip!</strong> If you will receive HTTP and/or HTTPS requests on multiple ports, you may want to disable all but one port (the one you will use to interact with the Server Administrator) until the server is ready for production. For instructions on how to set up and disable ports, see “Configuring Ports” on page 66.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Specify the proxy servers.</strong> Use the <strong>Proxy Servers</strong> screen to specify the proxy server(s) (if any) through which this server will issue outbound requests. Specify which URLs (if any) can bypass the proxy server. For instructions on how to specify proxy servers and bypass lists, see “Specifying a Third-Party Proxy Server for Outbound Requests” on page 83.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Configure session timeouts.</strong> Use the <strong>Resources</strong> screen to set the timeout value you want the server to use. For instructions, see “Setting the Session Timeout Limit” on page 96.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>Specify the error message recipients and an SMTP server.</strong> Use the <strong>Logging</strong> screen to specify the e-mail address where you want the server to send error messages when an exception (a critical server error or a binding failure) occurs and the name of the SMTP server to use for this purpose. For instructions, see “Working with Log Files” on page 89.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Set the logging parameters.</strong> Edit the logging parameters on the <strong>Logging</strong> screen and in the server.cnf file to specify the amount of detail you want the server to record in the audit log and the way in which you want it to rotate the log files. For instructions, see “Working with Log Files” on page 89.</td>
<td></td>
</tr>
</tbody>
</table>
STAGE 3: Setting Up Users, Groups, and ACLs

Use the Server Administrator to identify user accounts, groups, and access control lists (ACLs) to provide appropriate levels of access to the services that will run on this server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify service security requirements. Services are implicitly blocked from access by anyone other than Administrators and Developers. Determine what level of access is required, whether limited to one group of users, all authenticated users, or even unauthenticated users, and apply the appropriate ACL to the service.</td>
</tr>
<tr>
<td>2</td>
<td>Create user IDs and groups or configure an external directory. If you have secure services, identify users and/or client applications that are authorized to access those services and create groups that contain the authorized members. If your site uses an external directory (either LDAP or NIS), you can configure the server to access the user and group information from the external directory. For instructions for creating user IDs, see “Adding User Accounts” on page 48. For instruction for creating groups, see “Adding Groups” on page 57. For instructions for using an external directory, see “Using an External Directory (LDAP or NIS)” on page 175.</td>
</tr>
<tr>
<td>3</td>
<td>Create ACLs. Create the ACLs needed to meet your services’ security requirements and assign the groups you have created to these ACLs. For instructions, see “Creating ACLs” on page 125.</td>
</tr>
<tr>
<td>4</td>
<td>Identify backup administrators. Select one or two users who can act as a backup administrator when the primary administrator is unavailable. Use the Users and Groups screen to add these users to the “Administrators” group. For instructions on how to grant a user administrator privileges, see “Controlling Who Can Configure and Manage the Server” on page 108.</td>
</tr>
</tbody>
</table>
STAGE 4: Publishing Packages

Install and configure the packages that will run on this server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Install services on the server.</strong> Use one of the following methods to publish your services to the production server:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Method 1.</strong> Use the <strong>Packages Publishing</strong> screen to replicate the packages from the development server to the production server. For instructions, see “Copying Packages from One Server to Another” on page 201.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Method 2.</strong> Use the Server Administrator of the publishing server to create a zip file containing each package you want to publish; then:</td>
</tr>
<tr>
<td></td>
<td>1 Copy the zip file to following directory on the target server:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;sapbc&gt;\server\replicate\inbound</code></td>
</tr>
<tr>
<td></td>
<td>2 Use the <strong>Packages Management</strong> screen to install each package.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Configure the services on the server.</strong> Ensure that each service is enabled. Then, configure the following operating parameters for each:</td>
</tr>
<tr>
<td></td>
<td>- <strong>ACL assignment</strong> For instructions, see “Assigning ACLs to Services and Folders” on page 130.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Caching parameters</strong> For instructions, see the <em>webMethods Developer User’s Guide</em>.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Output Template Assignment</strong> For instructions, see the <em>webMethods Developer User’s Guide</em>.</td>
</tr>
<tr>
<td></td>
<td>- <strong>XML binding</strong> For instructions, see the <em>webMethods Developer User’s Guide</em>.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Delete unnecessary services.</strong> Delete the WmSamples package, which contains services that are not needed to support a production server. For instructions, see “Deleting a Package” on page 200.</td>
</tr>
</tbody>
</table>
STAGE 5: Configuring Database Connections

If your server will execute services against databases, take the following steps to configure the connection between the SAP BC Server and each database.

For more information about completing this stage, see “Configuring Access to Database Systems” on page 255.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Make sure the target database is accessible from the SAP BC Server. Ping the database server from the SAP BC Server to ensure the connection between the two servers is operational.</td>
</tr>
<tr>
<td>2</td>
<td>Verify that the required user accounts are set up on the target database system. Make sure that the user IDs, passwords, and database privileges that the SAP BC Server will use are set up on the target database. Consult with the database administrator to verify this.</td>
</tr>
</tbody>
</table>
| 3    | Install JDBC Drivers. Obtain the JDBC classes zip or jar file from the database vendor. Copy the zip or jar file to a device or directory that your SAP BC Server can access.  
    **Note:** If you use JDBC to access an ODBC driver, install and configure the ODBC driver according to the vendor’s instructions. |
| 4    | Update the classpath. Unzip or unjar the Java classes for the JDBC driver in the <sapbc>\server\lib\classes directory. For more information, see “Making the Java Classes for JDBC Drivers Accessible” on page 257. |
| 5    | Create a database alias for each target database. Use the Adapters screen to create a database alias name for each data source required by your services. |

STAGE 6: Installing Run-Time Classes

If your services use run-time classes beyond those provided by Java or the SAP BC Server (e.g., CORBA or MQ Series classes), install those classes on the server.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Install run-time classes. Obtain the zip or jar file from the vendor, and then copy the zip or jar file to a device or directory that your SAP BC Server can access.</td>
</tr>
<tr>
<td>2</td>
<td>Update the classpath. Update the classpath statement in the server.sh or server.bat file so that it points to the directory in which you have installed the run-time classes.</td>
</tr>
</tbody>
</table>
### STAGE 7: Setting Up Security

Take the following steps to ensure that the security measures you want to use are in place.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Check passwords.</strong> Verify that the Administrator and Replicator passwords have been changed from the default values assigned by SAP.</td>
</tr>
</tbody>
</table>
| 2    | **Edit the index.html file to prevent access to Server Administrator.** If you want to prevent a user from inadvertently accessing the Server Administrator, edit the following file:  

```
<sapbc>\server\packages\Default\pub\index.html
```

Change the SRC in the `<frame src="/WmRoot/index.dsp">` tag to some innocuous page you have created (perhaps one that displays an error message with alternate links).  

Note that if you implement this safeguard, you will not be able to invoke the Server Administrator in the standard way (i.e., simply connecting to the SAP BC Server's listening port). Instead, you will need to type the Server Administrator's complete URL as shown below:

```
http://Server:Port/WmRoot/index.dsp
```

where:

- **Server** is the name of the SAP BC Server, and  
- **Port** is the port on which it listens for HTTP requests. |
| 3    | **Check user accounts.** Verify that all user accounts have passwords as required. |
| 4    | **Check ACL assignments.** Verify that all secure services have correct ACL assignments. |
| 5    | **Check proxy server settings.** Verify that proxy server settings and bypass list are correct. |
| 6    | **Restrict access.** Configure allow/deny lists to restrict inbound requests as necessary. |
| 7    | **Install and configure digital certificates.** If you are deploying an SSL-enabled server, install the server's cert.der and privkey.der files in the following directory:  

```
<sapbc>\server\config\n```

Then, use the **Certificates** screen to configure X.509 features.  

For information about setting up the server to use SSL, see “Securing Communications with the Server” on page 111.
STAGE 8: Startup and Test

Start the server and test services to ensure that they operate as expected.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Verify that ports are enabled. If you disabled the ports to prevent access to the server during the setup phase, use the Ports screen to enable them now. <strong>Tip!</strong> After you enable a port, ping it to verify that it is operational.</td>
</tr>
<tr>
<td>2</td>
<td>Restart the server. Use the Server Administrator to restart the server to ensure all settings that you have made are in effect. For instructions, see “Restarting the SAP BC Server” on page 36.</td>
</tr>
<tr>
<td>3</td>
<td>Test services. Perform tests to ensure that user/client applications can access the server successfully. <strong>Note:</strong> During this test you may also want to verify that your current license will accommodate the expected concurrency demands on this server. Contact SAP to increase number of licensed sessions if necessary.</td>
</tr>
<tr>
<td>4</td>
<td>Go Live!</td>
</tr>
</tbody>
</table>

STAGE 9: Archive Sources

Archive a master copy of the packages on the server and the source files that were used to build them.
### STAGE 9: Archive Sources

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Copy the contents of the server\packages directory to another device for backup and archival purposes.</td>
</tr>
<tr>
<td>2</td>
<td>Archive a copy of all the source files that went into producing the services deployed on this server.</td>
</tr>
</tbody>
</table>
Server Configuration Parameters

- Introduction ................................................................. 276
- watt.debug ................................................................. 276
- watt.debug2 ................................................................. 276
- watt.net ................................................................. 277
- watt.security ................................................................. 279
- watt.server ................................................................. 280
- watt.tx ................................................................. 290
Introduction

This appendix contains a description of the parameters you can specify in the server configuration file (server.cnf), which is located in the <sapbc>/server/config directory. Typically you will use the Settings>Extended screen from the Server Administrator to update this file, but there might be times when you need to edit the file directly using a text editor. If you edit the file directly, you should first shut down the SAP BC Server before updating the file. After you make the changes, restart the server.

The server uses default values for many of the parameters. If a parameter has a default, it is listed with the description of the parameter. Many of these parameters are set as you administer the SAP BC Server using the Server Administrator.

watt.debug.

watt.debug.level
Sets level of debugging information recorded in the server’s log file. The default is 4.

watt.debug.logfile
Specifies the name of the file to receive debugging information. The default is logs\server.log.

watt.debug2.

watt.debug2.facList
Specifies a comma delimited list of enabled facilities for which the server logs information. The facilities are numbered. Specify the numbers for the facilities. The default is 999, which indicates the server is to log information for all facilities. Specify 1000 to prohibit the server from logging information for any service.

To view the names of facilities, use the Log Settings screen of the Server Administrator to enable and disable facilities for which you want the server to log information.

watt.debug2.logstringfile
Specifies the name (without the extension .txt) for the dictionary file that contains error codes and facilities. The default is lib\logstr (English Version).

watt.mail.

watt.mail.debug
Turns on the protocol trace for SMTP connections (pub.client:smtp) and for IMAP/POP3 connections (EmailListener).
watt.net.

watt.net.ftpAcceptTimeout
This parameter is used in two locations in the Business Connector:

a) If the BC is acting as an FTP client (pub.client.ftp:* Services) and the FTP client is using "active transfer mode", then this timeout specifies, how long the BC should wait for the FTP server to open the data connection.

b) If the BC is acting as an FTP server (FTPListener) and the FTP client is using "passive transfer mode", then this timeout specifies, how long the BC should wait for the FTP client to open the data connection. Specified in seconds.

watt.net.ftpConnTimeout
watt.net.ftpDataConnTimeout
These parameters specify FTP Listener timeout. By default, FTP connection timeout is set to 900 seconds and FTP data connection timeout is set to 60 seconds. The value specified is in milliseconds, e.g. watt.net.ftpConnTimeout=900000.

watt.net.ftpDebug
If set to true, turns on debugging for the FTP Session Sweeper as well as protocol trace (control connection) for FTP client.

watt.net.ftpServerDebug
If set to true, turns on protocol trace (control connection) for FTP Listener.

watt.net.ftpSweepInterval
Defines the time interval (in seconds), how often the FTP Session Sweeper should check for idle FTP client connections.

watt.net.localhost
Sets localhost name for the BC. Use this, if the machine has multiple network cards.

watt.net.maxRedirects
Specifies the maximum number of HTTP redirects to allow before throwing an I/O exception. The default is 5.

watt.net.proxyHost
Specifies the host that this server should use for outbound HTTP requests. There is no default.

watt.net.proxyPass
Specifies the password to use for authentication with the HTTP proxy host. There is no default.

watt.net.proxyPort
Specifies the port number on the proxy host to use for outbound HTTP requests. There is no default.
watt.net.proxySkipList
Specifies a list of domain names for which the SAP BC Server should not use proxy servers. The default is localhost.

watt.net.proxyUser
Specifies the user name to use for authentication with the HTTP proxy host. There is no default.

watt.net.retries
 Specifies the number of times to retry a server that times out. This can be overridden by the client. The default is 0.

watt.net.secureProxyHost
Specifies the host that this server should use for outbound HTTPS requests. There is no default.

watt.net.secureProxyPass
Specifies the password to use for authentication with the HTTPS proxy host. There is no default.

watt.net.secureProxyPort
Specifies the port number on the proxy host to use for outbound HTTPS requests. There is no default.

watt.net.secureProxyUser
Specifies the user name to use for authentication with the HTTPS proxy host. There is no default.

watt.net.ssl.client.handshake.maxVersion
Max ssl client handshake version.

watt.net.ssl.client.handshake.minVersion
Min ssl client handshake version.

watt.net.ssl.client.strongcipheronly
If set to true, BC allows only strong encryption.

watt.net.timeout
Specifies the number of seconds the server waits for an HTTP request to be fulfilled before the request times out. The default is 0.

watt.net.userAgent
Specifies the value the server uses in the HTTP User Agent request header when it requests a Web document from a Web server. The default is Mozilla/4.0 [en] (WinNT; I).

watt.net.useCookies
Accept (“true”) or deny (“false” or null) cookies when communicating with Web servers. It is almost never a good idea to turn this off. Defaults to “true.”
**watt.repo.**

**watt.repo.retryWait**
Applies only if the repository uses a JDBC database. Specifies the length of time in milliseconds the repository server is to pause between attempts to connect to the repository database. The default is 30000.

**watt.repo.retryCount**
Applies only if the repository is a JDBC database. Specifies the number of times the repository server is to try connecting to the repository database. The default is 5.

**watt.security.**

**watt.security.caCert**
Comma-separated list of the path and file names of SAP BC's certificate chain. List the files in the following order: intermediate certificates 1-n, if present, and finally certificate of the Certificate Authority (CA) that issued the SAP BC Server's digital certificate. The default is config\cacert.der.

**watt.security.CADir**
Specifies the path name of a directory (relative to the server home) that contains the digital certificates of CAs that your SAP BC Server trusts, for example config\cas. When you indicate that you want the server to request client certificates (watt.server.requestCerts), the server automatically presents the list of certificates in this directory to the client when it submits its own certificate. There is no default.

**watt.security.cert.wmChainVerifier.checkEntireChain**
Checks entire certificate chain.

**watt.security.certProvider**
Certification provider class.

**watt.security.privateKey**
Specifies the path and file name of the file that contains the RSA private key associated with the SAP BC Server's digital certificate. The default is config\privkey.der.

**watt.security.signedCert**
Specifies the path and file name of the file containing the SAP BC Server's digital certificate. The default is config\cert.der.

**watt.security.ssl.cacheClientSessions**
Disabling the caching of client sessions enables the proper use of multiple credentials to the same server.

**watt.security.ssl.ignoreExpiredChains**
Specifies whether the SAP BC Server ignores expired CA certificates in a certificate chain it receives from an Internet resource (i.e., a Web server, another SAP BC Server). To have the SAP BC Server ignore expired CA certificates and allow SSL connections when a certificate is expired, set the watt.security.ssl.ignoreExpiredChains setting to
“true.” Note that this is less secure than denying connections when a certificate is expired. The default is “false.” For more information about this setting, see “When the SAP BC Server Is an SSL Server” on page 112.

**watt.server.**

**watt.server.auditLog**

Specifies whether audit events are generated. You can specify the following values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>persvc</td>
<td>Services generate audit events according to their Audit Level setting. This is the default setting.</td>
</tr>
<tr>
<td>off</td>
<td>Services do not generate audit events.</td>
</tr>
<tr>
<td>brief</td>
<td>All services generate audit events and pass compact sets of input data to the event handler unless their Audit Level is set to verbose; in which case, they pass complete sets of input data.</td>
</tr>
<tr>
<td>verbose</td>
<td>All services generate audit events and pass complete sets of input data to the event handler. This option should only be used when circumstances truly demand it. The overhead of copying the pipeline before and after every service can degrade server performance.</td>
</tr>
</tbody>
</table>

**Important!** The settings in `watt.server.auditLog` override the individual service Audit Level settings.

**watt.server.cache.flushMins**

Specifies how often (in minutes) the server sweeps the cache to remove expired cache entries and to prefetch cache service entries. The default is 10 minutes.

**watt.server.cache.gcMins**

Specifies how often (in minutes) the server sweeps the cache to perform garbage collection. The default is 60 minutes.

**watt.server.cache.isPersistent**

Specifies whether you want server cache to be persistent (“true”) or not (“false”). The default is “true”.

**watt.server.cache.prefetchUser**

User which BC should use when executing prefetched services.

**watt.server.cache.storageThresholdKB**

Specifies the size (in kilobytes) of the persistent server cache threshold. There is no default.
watt.server.clientTimeout
Specifies the amount of time (in minutes) after which an idle user session times out. The default is 10.

watt.server.compile
Specifies the compiler command the SAP BC Server uses to compile Java services that are developed using the Developer. This compiler command is also used from the jcode utility. By default, the server uses javac -classpath {0} -d {1} {2}.

watt.server.compile.unicode
Specifies the compiler command the SAP BC Server uses to compile Java services that are stored in Unicode encoding. This compiler command is also used from the jcode utility. By default, the server uses javac -encoding Unicode -classpath {0} -d {1} {2}. This setting works with the Sun JDK compiler.

watt.server.cronCatchup
Specifies whether the Server should attempt to catch up persisted scheduled jobs upon startup. When set to true, the Server will try to run these jobs at a constant rate until they are caught up. Use this property carefully! This activity consumes significant server resources. The default is false.

watt.server.cronMaxThreads
Maximum thread pool size for the System Tasks Scheduler.

watt.server.cronMinThreads
Minimum thread pool size for the System Tasks Scheduler.

watt.server.dateStampFmt
Specifies the date format to use in log files. You can use any format that is supported by the Java class java.text.SimpleDateFormat. For example, to display the date with the format 1999-12-08 14:44:33, specify yyyy-MM-dd hh:mm:ss.

watt.server.db.blocktimeout
Specifies the maximum time in milliseconds to block when waiting for a connection. A timeout of -1 means an indefinite wait. The absence of this property means an indefinite wait. This property is global to all pools.

Examples: watt.server.db.blocktimeout=20000
           watt.server.db.blocktimeout=-1

watt.server.db.compatibilityMode
This feature enables 4.7 style of DB Connection Handling. Use this only if you have compatibility problems with the new 4.8 DB Connection Pool implementation. Note that this switch will be removed in a following release.

watt.server.defaultCountry
Country for session locale.

watt.server.defaultLanguage
Language for session locale.
watt.server.defaultPackage
If BC cannot resolve a URL, it tries to use the URL relative to the configured default package. You can define here which package name this should be. Also, if you define a different package here, you will be redirected to the start of this package when you enter `<bc-host>:<bc-port>`. You can still reach the main page of BC with `<bc-host>:<bc-port>/WmRoot` in that case.

watt.server.displayDirectories
Flag to allow/disallow displaying the contents of the pub directories of Packages via HTTP.

watt.server.email.from
Specifies the "from" address for log email sent by the Business Connector Server. Add the property set it equal to the desired "from" address.

watt.server.email.processReplyEmails
The Email Listener always rejects emails when their subject starts with "Re:" which are reply emails. There may be implementations however where these emails need to processed. The property should be set to true to process reply emails.

watt.server.email.processReturnedEmails
Enables the email listener to accept returned email messages. When set to 'true,' the email listener passes returned messages to the global service, if one is defined, or to the default service.

watt.server.errorMail
Specifies the e-mail address of administrator to notify when the server encounters an internal fault. There is no default.

watt.server.eventHandlerUser
Event handlers can have a user other than Administrator invoke them if for some reason those handlers needed to be protected by an ACL which does not grant Administrator access.

watt.server.eventMgr.maxThreads
Maximum event manager threads.

watt.server.eventMgr.minThreads
Minimum event manager threads.

watt.server.fileEncoding
Specifies the encoding the server is to use when reading and writing text files. This setting has no effect on files stored as Unicode. The default is 8859_1.

watt.server.hideRuntimeEnvironment
Flag to allow/disallow displaying information about the used Server Runtime Environment in the BC About section. If you do not want to expose this information, switch it off.

watt.server.hostAllow
Specifies the name of the host that is allowed service. There is no default.
watt.server.httplog
If this entry is found in the extended resource settings, create logfile http.log.

watt.server.hostDeny
Specifies the name of the host that is denied service. There is no default.

watt.server.I18NLogging
Logging in internationalized format.

watt.server.illegalNSChars
Specifies the characters that you cannot use when naming a package, folder or service. The default is ?-#&@^!%*:$./'",=)(|}{\[><.

watt.server.inetaddress
Specifies the IP address of the network interface card (NIC) on which the server is to listen for incoming requests. By default, on multiple IP machines, the SAP BC Server listens on all available IPs. To limit the machine to listen on a single IP, specify its address on this parameter.

watt.server.invokeDirective
Specifies an alternate invoke directive (in HTTP URL).

watt.server.java.unicode
Specifies whether the source code for Java services is stored in Unicode encoding. The default is “false.” Set this value to “true” if the source code contains characters that cannot be rendered in the server’s native encoding.

watt.server.jdbc.defaultDriver
Specifies the name of the Java class for the driver you want to use to connect to databases when no driver name is supplied for a database alias. The default is the driver name for Sun’s JVM: sun.jdbc.odbc.JdbcOdbcDriver.

watt.server.jdbc.driverList
Specifies a comma-delimited list of JDBC drivers you want the server to load when it initializes. There is no default.

watt.server.jdbc.loginTimeout
The WmDB package calls the setLoginTimeout method of the JDBC java.sql.DriverManager class to prevent the JDBC getConnection method from blocking indefinitely if the database host is not available. The value specified is in seconds. By default, setLoginTimeout() will be called with a timeout of 600 seconds (10 minutes).

watt.server.jdbc.moreResults
Determines whether to enable multiple result sets from a stored procedure (not all drivers support that).

watt.server.jdbcLogFile
Name of JDBC logfile.

watt.server.jdbcLogging
Enables/disables jdbc logging.
**watt.server.keepAliveTimeout**  
Keep-alive timeout.

**watt.server.key**  
Specifies the license key for the server. There is no default.

**watt.server.ldap.attr.member**  
Specifies the name of the variable (e.g., member) from which the SAP BC Server is to get the group membership information for a user ID. There is no default.

**watt.server.ldap.attr.password**  
Specifies the name of the variable (e.g., password) from which the SAP BC Server is to get the password for a user ID. This value must match the name of the attribute in which the LDAP server expects to receive passwords. There is no default.

**watt.server.ldap.attr.userid**  
Specifies the name of the variable (e.g., uid) from which the SAP BC Server is to get the User ID. This value must match the name of the attribute in which the LDAP server expects to receive user IDs. There is no default.

**watt.server.ldap.directoryRoot**  
Specifies the distinguished name (DN) that defines the top entry of your LDAP directory hierarchy (e.g., o=SAP, c=US).

**watt.server.ldap.groupFilter**  
Specifies a filter that the SAP BC Server uses to narrow down the selection of groups within the group root that is specified by the `watt.server.ldap.groupRoot` parameter. For more information about LDAP search filters, consult RFC 2254: Search Filters for LDAP. To set a group filter, type the name of the filter—e.g., `(objectclass=rfc822mailgroup)` (objectclass=homeGrownGroup). There is no default.

**watt.server.ldap.groupRoot**  
Specifies the name of the root where group entries are located, relative to the specified directory root, which is identified by the `watt.server.ldap.directoryRoot` parameter. Specify the name of the group root entry (e.g., ou=Groups). There is no default.

**watt.server.ldap.retryCount**  
**watt.server.ldap.retryWait**  
Whenever there are connection exceptions, the Server will automatically try to reconnect to LDAP server. These server properties allow you to customize the retry behavior.

- `watt.server.ldap.retryCount`: default value 10
- `watt.server.ldap.retryWait`: default value 30000 ms

**watt.server.ldap.security.credential**  
Specifies the password the SAP BC Server uses when submitting a request to the LDAP server. There is no default.

**watt.server.ldap.security.principal**  
Specifies the distinguished name that identifies the user that the SAP BC Server uses when submitting a request to the LDAP server. There is no default.
watt.server.ldap.server
Specifies the URL of the LDAP server. Specify the complete URL for your LDAP server. There is no default.

watt.server.ldap.userRoot
Specifies the name of the root where user entries are located, relative to the specified directory root, which is identified by the watt.server.ldap.directoryRoot parameter. Specify the name of the user root entry (e.g., ou=People). There is no default.

watt.server.licenses
Specifies the number of licenses. The default is 1.

watt.server.logEncoding
Specifies the encoding the server is to use when writing log files. The default is your JVM’s file.encoding property.

watt.server.loginFailureLimit
Login failure limit. If more than watt.server.loginFailureLimit denials, send e-mail to to the person responsible for internal.

watt.server.logRotate
Specifies how the server rotates the log files. Specify “rename” to have the server rename the log files to include the date and time it closes the logs and keep the log files in the same directory. Specify “move” to have the server create a new subdirectory with a name that represents the date and time it closes the log files and move the closed log files to the new directory. If this parameter is not included in the server.cnf file, the server uses “rename.”

watt.server.logRotateFirstInterval
Specifies the number of milliseconds after the server is started when the server is to rotate the log files for the first time. If this parameter is not included in the server.cnf file, the server rotates the log files for the first time at 00:00 (midnight).

watt.server.logRotateInterval
Specifies the number of milliseconds after the first log rotation when the server is to continue to rotate the log files. If this parameter is not included in the server.cnf file, the server continues to rotate the log files every 24 hours.

watt.server.logRotateMoveDirectory
Directory to store the rotated logfiles.

watt.server.oldkey
Specifies the license key that was in use prior to the current key. There is no default.
watt.server.net.chunkSize
Chunk size for HTTP response.

watt.server.netEncoding
Specifies the encoding the server is to use when reading and writing text to the network. This setting has no effect on text that is explicitly encoded in a particular encoding. The default is 8859_1.

watt.server.new.http.session.context
Context reuse causes HTTP session information to be reused. Previously, context was being reused for HTTP client operations. If this property is set to true, then a new context will be created for HTTP operations.

watt.server.nis.domain
NIS domain.

watt.server.nis.server
NIS server.

watt.server.nis.timeout
NIS timeout.

watt.server.noObjectURL
Specifies the URL to which the server redirects a request after three attempts to log in to the Server Administrator have failed because the server cannot find the document the user is requesting. The default is for the server to display an HTML screen saying “No such object.”

watt.server.noAccessURL
Specifies the URL to which the server is to redirect a request after three attempts to log in to the Server Administrator have failed because the user does not have access to the requested document. The default is for the server to display an HTML screen saying “Access denied.”

watt.server.ns.backupNodes
Specifies whether services are removed completely when they are deleted. When set to true, service node.ndf files will be renamed to node.bak when they are deleted. The default is false.

watt.server.port
Specifies the port number of the SAP BC Server’s primary port. The default is 5555.

watt.server.portQueue
Specifies the size of the port queue for HTTP and HTTPS ports. The port queue is the number of outstanding inbound connections that are queued in the TCP/IP stack. The default is 65534. If your server runs on AS/400, set this number to 511.

watt.server.recordToDocument.bufferSize
Size for string buffer.
watt.server.revInvoke.allowDirective
Specifies list of proxy ports (separated by commas) that should only be allowed to use HTTP invocations.

watt.server.revInvoke.connectionCheck
Checks reverse invoke connections every x minutes.

watt.server.revInvoke.errorSvcName
Fully qualified name of the service to run if your filtering service encounters errors
(folder.folder2:serviceName).

watt.server.revInvoke.filterSvcContentTypes
Name of content types for which you want the filtering service to run. If the content type specified by the client matches any in the list, the reverse invoke SAP BC Server runs the filtering service. You can specify multiple content types separated by commas.

watt.server.revInvoke.filterSvcName
Fully qualified name of the service to run if your filtering service encounters errors
(folder.folder2:serviceName).

watt.server.requestCerts
Specifies whether the SAP BC Server requests a digital certificate from clients that connect to it through SSL. Set this parameter to “true” if you want the server to request certificates. Set it to “false” if you do not want the server to request certificates. The default is false.

watt.server.saveConfigFiles
Flag for backing up config files on startup

watt.server.scheduler.allowDisabledUsers
If an administrator schedules a task for a user, and that user’s account is subsequently disabled, the scheduled task will continue to execute. This behavior can be useful (such as having a “dummy user” used for running scheduled tasks, but which cannot log in), but can also present a risk if not anticipated. After you have installed this CoreFix, you can set the “watt.server.scheduler.allowDisabledUsers” extended setting to "false" to disallow this behavior. If the setting is “true” (the default), then the previous behavior, which allows for the "dummy user“ option described above, will continue.

watt.server.scheduler.maxThreads
Specifies the maximum number of threads in the thread pool the server uses for the scheduler. This number controls the maximum number of jobs that can run concurrently on the server. If this maximum number is reached, the server waits until jobs complete and return threads to the pool before running more jobs. The default is 10.

watt.server.securePort
Specifies the port number of the SAP BC Server’s primary secured listening port. The default is 0.

watt.server.securePortList
Specifies the additional port numbers on which the SAP BC Server listens for HTTPS requests. The list is comma delimited. There is no default.
watt.server.serverLogFmt
Specifies the format and contents of log messages in the server log file. The available formatting options are:

| 0 | Error code  |
| 1 | Facility code |
| 2 | Severity character |
| 3 | Error message |
| 4 | Sequence number |
| 5 | Date and time stamp |
| 6 | Product group |
| 7 | Product component |

The default setting is:

```
  watt.server.serverLogFmt={4}  [{6}{7}.{1}.{0,number,0000}] {3}
```

The resulting format is:

```
```

To change the format to show a date and timestamp, you could specify:

```
  watt.server.serverLogFmt={5}  [{6}{7}.{1}.{0,number,0000}] {3}
```

The resulting format is:

```
```

**Note:** Do not remove the product group [6], product component [7], Error code [0], or facility code [1] (that is, B2BSERV:0028:0012) from the message display. These fields are useful to SAP Customer Care when resolving problems.

watt.server.serviceMail
Specifies the e-mail address of an administrator to notify when a service no longer binds to a target site correctly. There is no default.

watt.server.session.logfile
Session log file.

watt.server.smtpServer
Specifies the SMTP server to use for server error logging and service error e-mail notification. There is no default.

watt.server.SOAP.useMultiReference
If set to true (default) the SOAP coder uses multi-references, where values can be identified and linked via an ID, otherwise (single reference mode) all values are embedded in all locations.
**watt.server.SOAP.validateSOAPMessage**  
Flag for validating soap messages.

**watt.server.SoapRPC.debug**  
Prints soap debug information.

**watt.server.SoapRPC.verbose**  
Prints soap verbose information.

**watt.server.stats.avgTime**  
Specifies the time period (in seconds) for which performance metrics are averaged. The default is 10.

**watt.server.stats.logfile**  
Specifies the name of the file to receive statistics. The default is `logs\stats.log`.

**watt.server.stats.pollTime**  
Specifies the number of seconds between updates of statistics loggings. The default is 10.

**watt.server.threadPool**  
Specifies the maximum number of threads that the server maintains in the thread pool that it uses to run services. If this maximum number is reached, the server waits until services complete and return threads to the pool before running more services. The default is 75.

**watt.server.threadPoolMin**  
Specifies the minimum number of threads that the server maintains in the thread pool that it uses to run services. When the server starts, the thread pool initially contains this minimum number of threads. The server adds threads to the pool as needed until it reaches the maximum allowed, which is specified by the `watt.server.threadPoolMin` setting. The default is 10.

**watt.server.txMail**  
Specifies the e-mail address of an administrator to notify when guaranteed delivery capabilities are disabled due to an error (for example, if the SAP BC Server encounters a disk full condition or if the audit-trail log is full). There is no default.

**watt.server.tx.heuristicFailRetry**  
Specifies whether the SAP BC Server is to re-execute services for guaranteed delivery transactions in the job store that are PENDING when the SAP BC Server is restarted after a failure. If a transaction is PENDING, the service began execution before the SAP BC Server failed.

If the setting is TRUE, the SAP BC Server resets the transaction status from PENDING to NEW, and the service will be re-executed. If the setting is FALSE the SAP BC Server resets the transaction status from PENDING to FAIL to indicate the heuristic failure, and the service will not be re-executed. The default is “true”.

**watt.server.tx.logfile**  
Specifies the file in which the server maintains an audit-trail log of all operations it processes for inbound guaranteed delivery transactions. The default is `logs\txin.log`. 
**APPENDIX B  Server Configuration Parameters**

- **watt.server.tx.jobdir**: Specifies the directory in which the server maintains the job store for inbound guaranteed delivery transactions. The default is \logs\jobsin.

- **watt.server.tx.sweepTime**: Specifies the number of seconds between sweeps (clean up) of the job store for inbound guaranteed delivery transactions. The server sweeps the job store to remove expired transactions. The default is 60.

- **watt.server.WSDL.debug**: Flag for enabling WSDL debug information.

- **watt.server.wsdl.separateXSD**: Allows you to inline the XSD schemas into the main WSDL document body instead of generating a separate .xsd file. The Property is set to "false" by default.

- **watt.server.xml.enforceEntityRef**: Set this property to "false" to prevent the Business Connector from throwing an exception when the XML parser detects a malformed entity.

**watt.tx.**

- **watt.tx.defaultTTLMins**: Specifies the default time-to-live (TTL) value for outbound guaranteed delivery transactions. Specify the number of minutes you want the server to maintain outbound transactions in the job store when a service initiating an outbound transaction does not specify a TTL value. The default is 30.

- **watt.tx.disabled**: Disables guaranteed delivery. Services which will use this feature then will throw an exception.

- **watt.tx.jobdir**: Specifies the directory in which the SAP BC Server maintains the job store of outbound guaranteed delivery transactions. The default is \logs\jobsout.

- **watt.tx.jobThreads**: Specifies the number of client threads you want to make available in a thread pool to service pending requests in the outbound guaranteed delivery job store. The default is 5.

- **watt.tx.logfile**: Specifies the path and file in which the SAP BC Server maintains an audit-trail log of all the outbound guaranteed delivery transaction operations it requests. The default is \logs\txout.log.

- **watt.tx.retryBackoffTime**: Specifies the number of seconds to wait after a service request failure before the Job Manager resubmits the request to execute the service to the SAP BC Server. The default is 60.
**watt.tx.sweepTime**
Specifies the number of seconds between sweeps of the job store of outbound guaranteed delivery transactions. The server sweeps the job store to identify transactions that it needs to submit. The default is 60.
Server Log Files and JVM Information

- The Audit Log ................................................................. 294
- The Error Log ................................................................. 294
- The Server Log ................................................................. 295
- The Session Log ................................................................. 297
- The Transaction Logs ......................................................... 298
- JVM Information ................................................................. 299
APPENDIX C  Server Log Files and JVM Information

The Audit Log

The Audit Log (audit.log) maintains an audit trail for all services that the server executes. For each service, the server adds two records that contain information about the service. It adds one when it begins execution; this record includes the timestamp of when execution began. It adds the second record when the service terminates; this record includes the timestamp of when the service terminated.

Note: For security purposes, you should regularly review your audit logs to check for security breaches. In addition, you should regularly back them up in case the machine where the logs are kept becomes unavailable.

Example of audit.log

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Session identifier of the session that invoked the service</th>
<th>Thread identifier of the thread that invoked the service</th>
<th>Name of the service</th>
<th>Description of the audit event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-02-03-08:05:22</td>
<td>--- (Start)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:30</td>
<td>23878555569 00134db3 wm.server.ui:getMenus begin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:31</td>
<td>23878555569 00134db3 wm.server.ui:getMenus ok</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:33</td>
<td>23878555569 00134db3 wm.server.query:getStats begin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:34</td>
<td>23878555569 00134db3 wm.server.query:getStats ok</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:36</td>
<td>23878555569 00134db3 wm.server.ui:getMenus begin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:37</td>
<td>23878555569 00134db3 wm.server.ui:getMenus ok</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:39</td>
<td>23878555569 00134db3 wm.server.access:userList begin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001-02-03-08:05:40</td>
<td>23878555569 00134db3 wm.server.access:userList ok</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The date format shown in the sample above is not the default format. The default format is a hexadecimal representation of the system time. For instructions on changing to a more readable format, see “Specifying the Date Format to Use in Log Files” on page 93.

The Error Log

The Error log (error.log) maintains information about all errors and exceptions that have occurred in the server system.
The Server Log

Example of error.log

<table>
<thead>
<tr>
<th>Timestamp and exception</th>
<th>Stack trace that shows the call sequence leading to the exception</th>
</tr>
</thead>
</table>

Note: The date format shown in the sample above is not the default format. The default format is a hexadecimal representation of the system time. For instructions on changing to a more readable format, see “Specifying the Date Format to Use in Log Files” on page 93.

The Server Log

The Server Log (server.log) maintains information about the version of the SAP BC Server that is running and operations the server has performed. You can use this information for general debugging purposes.
## Example of server.log

<table>
<thead>
<tr>
<th>Log Sequence</th>
<th>Facility Information</th>
<th>Log Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>1</td>
<td>Wed Mar 14 13:16:42 EST 2001</td>
</tr>
<tr>
<td>00001 SRVINI0001C</td>
<td>2</td>
<td>SAP BC Server 4.0 Build 671</td>
</tr>
<tr>
<td>00002 SRVINI0002C</td>
<td>3</td>
<td>License Manager started</td>
</tr>
<tr>
<td>00003 SRVINI0008C</td>
<td>1</td>
<td>State Manager started</td>
</tr>
<tr>
<td>00004 SRVINI0010C</td>
<td>2</td>
<td>Service Manager started</td>
</tr>
<tr>
<td>00005 SRVINI0012C</td>
<td>3</td>
<td>Cache Manager started</td>
</tr>
<tr>
<td>00006 SRVINI0009C</td>
<td>1</td>
<td>WIDL Service Manager started</td>
</tr>
<tr>
<td>00007 SRVINI0004C</td>
<td>2</td>
<td>Flow Service Manager started</td>
</tr>
<tr>
<td>00008 SRVINI0002C</td>
<td>3</td>
<td>Package Manager started</td>
</tr>
<tr>
<td>00009 SRVINI0011C</td>
<td>1</td>
<td>Package Replicator Manager started</td>
</tr>
<tr>
<td>0000A PKGSES0001C</td>
<td>2</td>
<td>Loading packages</td>
</tr>
<tr>
<td>0000B PKGSES0005C</td>
<td>3</td>
<td>Loading WmRoot package</td>
</tr>
<tr>
<td>0000C PKGSES0005C</td>
<td>1</td>
<td>Loading WmDB package</td>
</tr>
<tr>
<td>0000D PKGSES0005C</td>
<td>2</td>
<td>Loading WmSamples package</td>
</tr>
<tr>
<td>0000E PKGSES0005C</td>
<td>3</td>
<td>Loading WmPublic package</td>
</tr>
<tr>
<td>0000F PKGSES0005C</td>
<td>1</td>
<td>Loading WmPartners package</td>
</tr>
<tr>
<td>00010 PKGSES0005C</td>
<td>2</td>
<td>Loading Default package</td>
</tr>
<tr>
<td>00011 PKGSES0012C</td>
<td>3</td>
<td>WmRoot: Startup service (wm.server.net.email:init)</td>
</tr>
<tr>
<td>00012 PKGSES0012C</td>
<td>1</td>
<td>WmRoot: Startup service (wm.server.net.ftp:init)</td>
</tr>
<tr>
<td>00013 PKGSES0012C</td>
<td>2</td>
<td>WmRoot: Startup service (wm.server.net.http:init)</td>
</tr>
<tr>
<td>00014 PKGSES0012C</td>
<td>3</td>
<td>WmRoot: Startup service (wm.server.net.https:init)</td>
</tr>
<tr>
<td>00015 PKGSES0012C</td>
<td>1</td>
<td>WmRoot: Startup service (wm.server.net.s MIME:init)</td>
</tr>
<tr>
<td>00016 PKGSES0012C</td>
<td>2</td>
<td>WmRoot: Startup service (wm.server.net.socket:init)</td>
</tr>
<tr>
<td>00017 PKGSES0012C</td>
<td>3</td>
<td>WmRoot: Startup service (wm.server.net.sslsocket:init)</td>
</tr>
<tr>
<td>00018 PKGSES0012C</td>
<td>1</td>
<td>WmRoot: Startup service (wm.server.remote:init)</td>
</tr>
<tr>
<td>00019 PKGSES0012C</td>
<td>2</td>
<td>WmRoot: Startup service (wm.server.schedule:init)</td>
</tr>
<tr>
<td>0001A PKGSES0012C</td>
<td>3</td>
<td>WmRoot: Startup service (wm.server.security.revInvoke:init)</td>
</tr>
<tr>
<td>0001B PKGSES0012C</td>
<td>1</td>
<td>WmRoot: Startup service (wm.server.tx:init)</td>
</tr>
<tr>
<td>0001C TRXJOB0011O</td>
<td>2</td>
<td>Start</td>
</tr>
<tr>
<td>0001D PKGSES0012C</td>
<td>3</td>
<td>WmRoot: Startup service (wm.server.event:startEventAgentPort)</td>
</tr>
<tr>
<td>0001E PKGSES0012C</td>
<td>1</td>
<td>WmDB: Startup service (wm.server.db:startup)</td>
</tr>
<tr>
<td>0001F PKGSES0012C</td>
<td>2</td>
<td>WmPublic: Startup service (pub.ldap:init)</td>
</tr>
<tr>
<td>00020 PKGSES0012C</td>
<td>3</td>
<td>WmPartners: Startup service (wm.PartnerMgr.gateway.admin:startup)</td>
</tr>
<tr>
<td>00021 XTNSTO0015W</td>
<td>1</td>
<td>init: xtnStore: (entries=0, freeBlocks=0, maxFree=0, bytes=0, sync=true, file=packages\WmPartners\config\xtn.log)</td>
</tr>
<tr>
<td>00022 SRVINI0005C</td>
<td>2</td>
<td>Port Manager started</td>
</tr>
<tr>
<td>00023 SRVINI0001C</td>
<td>3</td>
<td>Cache Sweeper started</td>
</tr>
<tr>
<td>00024 B2BSRV0002C</td>
<td>1</td>
<td>Initialization complete.</td>
</tr>
</tbody>
</table>

### Column Details

<table>
<thead>
<tr>
<th>Column</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log Sequence</td>
<td>Six-digit sequence number in hexadecimal</td>
</tr>
<tr>
<td>Facility Information</td>
<td>Code that describe the facility or component that added the log message and the log message identifier, for example: SRVINI0001C</td>
</tr>
</tbody>
</table>

1. Six-character abbreviation of the facility or component that issued the log message.
2. Four-digit message number of the log message.
The Session Log

The Session Log (session.log) maintains information about all sessions in the server system. The SAP BC Server creates a session when a developer connects to the server from the SAP BC Developer or a server client connects to the server to execute services. Each record in the Session Log includes a timestamp indicating when a user established a connection with the SAP BC Server.

Example of session.log

<table>
<thead>
<tr>
<th>Column</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>One-digit severity code of the log message:</td>
</tr>
<tr>
<td>C</td>
<td>Critical</td>
</tr>
<tr>
<td>D</td>
<td>Debug</td>
</tr>
<tr>
<td>E</td>
<td>Error</td>
</tr>
<tr>
<td>V</td>
<td>Verbose</td>
</tr>
<tr>
<td>W</td>
<td>Warning</td>
</tr>
</tbody>
</table>

Log Message | Message that indicates the action the server performed.

Note: The date format shown in the sample above is not the default format. The default format is a hexadecimal representation of the system time. For instructions on changing to a more readable format, see “Specifying the Date Format to Use in Log Files” on page 93.
The Transaction Logs

The Transaction Logs (txin.log and txout.log) maintain information about guaranteed delivery transactions. They contain a record of all critical operations that the server performs for each guaranteed delivery transaction. The server maintains one log for inbound guaranteed delivery transactions (txin.log) and a separate log for outbound transactions (txout.log).

You can use these logs in conjunction with the appropriate job store to identify specific information about a transaction. For example, if a transaction fails, you can use the log to determine how far the transaction got before failing so you can take appropriate measures to recover a transaction.

**Note:** The server contains a job store for inbound transactions and a separate job store for outbound transactions.

The server uses the watt.server.txlogfile configuration setting in the server.cnf file for the location of the job store of inbound transaction. If this setting is not in the server.cnf file, the server maintains the log in the `<sapbc>\server\logs\txin.log` file.

The server uses the watt.txlogfile configuration setting in the server.cnf for the location of the job store of outbound transaction. If this setting is not in the server.cnf file, the server maintains the log in the `<sapbc>\server\logs\txout.log` file.

To view or change these settings, use the **Settings>Extended** screen of the Server Administrator as described in “Working with Extended Configuration Settings” on page 100.
Example of tx.log

<table>
<thead>
<tr>
<th>2001-02-03-06:02:10</th>
<th>--- (Start) ---</th>
<th>When the server begins logging</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001-02-03-06:02:11</td>
<td>START TXJobManager</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:12</td>
<td>CREATE tid=1711915921524310 (NEW)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:14</td>
<td>JOB SUCCESS tid=1711915921524310 (DONE)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:15</td>
<td>END tid=1711915921524310 (DONE)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:16</td>
<td>CREATE tid=1711915921524481 (NEW)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:17</td>
<td>JOB SUCCESS tid=1711915921524481 (DONE)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:19</td>
<td>END tid=1711915921524481 (DONE)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:21</td>
<td>CREATE tid=1711915921524502 (NEW)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:23</td>
<td>JOB SUCCESS tid=1711915921524502 (DONE)</td>
<td></td>
</tr>
<tr>
<td>2001-02-03-06:02:24</td>
<td>END tid=1711915921524502 (DONE)</td>
<td></td>
</tr>
</tbody>
</table>

Timestamp | Operation the server performed | Status of the transaction 
--- | --- | --- 
Each line contains a tid, the transaction id for a transaction. All lines with the same tid are for the same request.

Note: The date format shown in the sample above is not the default format. The default format is a hexadecimal number. For instructions on changing to a more readable format, see “Specifying the Date Format to Use in Log Files” on page 93.

JVM Information

You can view and modify various properties of the Java Virtual Machine that runs the BC via Server -> JVM on the left menu bar of the Server Administrator.

For the standard (Sun) JVM trace information is provided. This information essentially is a thread dump that lists the stack trace for each thread.

If the BC was started with the SAP JVM additional data is made available apart from the standard trace information.

The JVM screen menu shows the following entries:

Trace Information (Standard JVM)

Trace information can be viewed.

Trace Information (SAP JVM)

Trace information is logged in several files. All available trace files are listed. Files can be viewed and deleted (the currently open file may not be deletable depending on the Operating System). A new trace file is opened each time the BC is started. The trace information is divided into a "Stack/DLL Trace" and a "Class/GC Trace". The "Full Trace"
combines the two. Whenever one of these three buttons is pressed the respective trace information is appended to the open trace file.

**Memory Information**

A concise overview of memory-related parameters is shown.

**VM Information**

VM parameters such as arguments, flags etc. are listed.

**Thread Information**

An overview of all threads and their essential properties (including a stack trace) is displayed. In essence, this information presents the Stack part of the Stack/DLL trace in a different format, enriched with additional thread-specific data.

**Debugger Control**

The remote debugging facility can be viewed and modified. The current debugging state is shown. A debugger can be attached (i.e. listeners can be started on the specified range of ports) and thus debugging can be enabled, or the debugger may be detached, thus effectively disabling debugging.

**Note:** For information on how to connect to one of these listeners and conduct a remote debugging session, see the manual of your Java development environment (e.g. Eclipse).
Wireless Communication with the SAP BC Server

- How Does the SAP BC Server Communicate with Wireless Devices? .................. 302
- Using URLs for Wireless Access to the SAP BC Server ................................. 304
- WML and HDML Samples ................................................................. 306
The SAP BC Server can receive requests from and send responses to Internet-enabled wireless devices. A wireless device requests information from the SAP BC Server by using a URL. The responses sent by the server contain WML (Wireless Markup Language) content or HDML (Handheld Device Markup Language) content. Examples of wireless devices that the SAP BC Server can communicate with include Internet-enabled wireless phones and Internet-enabled personal digital assistants.

You might want to use a wireless device to communicate with the SAP BC Server to:

- Check inventory levels at your company or at a supplier.
- Place an order or check the status of an existing order.
- Receive order confirmation for an order submitted with a wireless device.
- Send or receive notification to alert subscribers to trade fulfillments of security price changes.
- Collect statistics about your SAP BC Server by using event handlers that send information to wireless devices.
- Request an HDML or WML page stored on the SAP BC Server.

You access the SAP BC Server from a wireless device by entering a URL in the Web browser of wireless device. The URL can invoke a service on the SAP BC Server or can request a WML or HDML page stored on the SAP BC Server.

**How Does the SAP BC Server Communicate with Wireless Devices?**

The SAP BC Server communicates with wireless devices by means of a wireless gateway. The wireless gateway (sometimes called a WAP gateway) converts a request from a wireless device to an HTTP request. The wireless gateway also converts the HTTP response from the SAP BC Server to a format understood by the Web browser or micro-browser on the wireless device.

The following diagram illustrates how the SAP BC Server communicates with an Internet-enabled wireless device.
How Does the SAP BC Server Communicate with Wireless Devices?

Communication Between the SAP BC Server and a Wireless Device

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A user requests a URL using a Web browser on a wireless device such as a wireless phone or a personal digital assistant (PDA). The URL indicates the service to be invoked or identifies the requested WML or HDML page. The wireless device sends an encoded request to the wireless gateway.</td>
</tr>
<tr>
<td>2</td>
<td>The wireless gateway (such as a Phone.com’s Up.Link Server or Nokia Active Server) decodes the request from the wireless device, creates an HTTP or HTTPS request (depending on what is specified in the URL) for the specified URL, and sends it to the SAP BC Server.</td>
</tr>
</tbody>
</table>
| 3     | The SAP BC Server does one of the following depending on what the user requested in the URL:  
  - Executes the service specified in the URL and inserts the service results into the assigned WML or HDML output template.  
  - OR -  
  - Retrieves the WML or HDML page requested in the URL. |
| 4     | The SAP BC Server sends an HTTP or HTTPS response to the wireless gateway. |
| 5     | The wireless gateway removes the HTTP or HTTPS header from the response and sends an encoded response containing the HDML or WML content to the wireless device. The Web browser on the wireless device decodes the response and displays the WML or HDML results. |

For more information about wireless gateways and wireless protocol, see www.wapforum.org.
Using URLs for Wireless Access to the SAP BC Server

To use a wireless device to access information or invoke services on the SAP BC Server, you need to use the device's Web browser to enter or select a URL. The following sections explain how to invoke a service with a URL and how to request a WML or HDML page with a URL.

**Note:** Some Web browsers for wireless devices place limitations on the length of a URL that a user can enter or select. Make sure any WML or HDML pages that you create for use with wireless devices are compliant with browser requirements.

**Note:** To minimize the amount of user input and therefore reduce the possibilities for input errors, embed hyperlinks to URLs in the WML or HDML page.

## Invoking a Service with a URL

You can use a URL to invoke a service from an Internet-enabled wireless device. You can request a URL by entering the URL into the Web browser directly or by selecting a link for the URL that is embedded into a HDML or WML page. In either case, the URL needs to be in the following format:

```
http://localhost:5555/invoke/folderName.subFolderName/serviceName?variable=value&variable=value
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identifies the name and port number for the SAP BC Server on which the service you want to invoke resides.</td>
</tr>
</tbody>
</table>

**Important!** For wireless access, the server name (localhost) must be a registered domain name—that is, the server needs to be accessible via the Internet.

**Important!** Many wireless gateways use port 80 as the default registered port number. If you want to use a different port number, make sure to register the server name and port number with the wireless gateway. (For security reasons, SAP discourages using port numbers below 1024. For more information, see “Configuring Ports” on page 66.)

| 2    | Specifies the required keyword “invoke”, which tells the SAP BC Server that the URL identifies a service that is to be invoked. |
Using URLs for Wireless Access to the SAP BC Server

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Identifies the folder in which the service to invoke resides. Separate subfolders with periods. This field is case-sensitive. Be sure to use the same combination of upper and lower case letters as specified in the folder name on the SAP BC Server.</td>
</tr>
<tr>
<td>4</td>
<td>Identifies the service that you want to invoke. This field is case-sensitive. Be sure to use the same combination of upper and lower case letters as specified in the service name on the SAP BC Server.</td>
</tr>
<tr>
<td>5</td>
<td>Specifies the input values for the service. Specify a question mark (?) before the input values. The question mark signals the beginning of input values. Each input value is represented as a variable=value pair. The variable portion is case-sensitive. Be sure to use the same combination of upper and lower case letters as specified in your service. If your service requires more than one input value, separate each variable=value pair with an ampersand (&amp;).</td>
</tr>
</tbody>
</table>

**Note:** Only specify this part of the URL when using the HTTP GET method.

For more information about invoking a service with a URL, see “Building a Browser Based Client” in the SAP BC Developer Guide.

**Note:** If you use the URL to invoke a service, make sure that the service applies the output to the appropriate template type (WML or HDML). For more information about creating output templates, see Building Output Templates and DSPs.

### Requesting a WML or HDML Page with a URL

You can use an Internet-enabled wireless device to request a WML or HDML page stored on the SAP BC Server. By entering a URL in the Web browser of a wireless device or by selecting a hyperlink to a URL, you can access any WML or HDML page stored in the following directory:

```
< SAPBC >\server\packages\packageName\ pub
```

Where `packageName` is the name of the package in which the WML or HDML file is saved.
The URL you enter in the Web browser needs to adhere to the following format:

```
http://localhost:5555/packageName/pub/fileName
```

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identifies the name and port number for the SAP BC Server on which the file you want to request resides.</td>
</tr>
<tr>
<td></td>
<td><strong>Important!</strong> For wireless access, the server name (localhost) must be a registered domain name—that is, the server needs to be accessible outside via the Internet.</td>
</tr>
<tr>
<td></td>
<td><strong>Important!</strong> Many wireless gateways use port 80 as the default registered port number. If you want to use a different port number, make sure to register the server name and port number with the wireless gateway. (For security reasons, SAP discourages using port numbers below 1024. For more information, see “Configuring Ports” on page 66.)</td>
</tr>
<tr>
<td>2</td>
<td>Identifies the package in which the WML or HDML file you want to request resides.</td>
</tr>
<tr>
<td>3</td>
<td>Specifies the pub directory. WML and HDML files that can be served to wireless devices need to reside in this directory.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You do not need to specify the pub directory. SAP BC Server automatically looks in pub for the requested file if no directory is specified.</td>
</tr>
<tr>
<td>4</td>
<td>Identifies the file you want to request.</td>
</tr>
</tbody>
</table>

For example, the following URLs access the hello.wml file from the pub directory for the Wireless package:


### WML and HDML Samples

The SAP BC Server provides sample services, WML files, HDML files, and output templates that you can use to view how wireless devices communicate with the SAP BC Server. These files are located in the sample.wireless folder in the WmSamples package. These samples provide examples of services that you might create to enable wireless devices to...
order products, view order history, obtain SAP BC Server statistics, and request the current date and time.

For more information about using these samples, see the following file:

<sapbc>\server\packages\WmSamples\pub\WAPDemo.htm
Index

A
accelerator, hardware 117
Access Control Lists (ACLs)
  ACL used when none assigned to service 124, 128
Administrators 124
Anonymous 124
assigning to files 131
assigning to folders 128
assigning to services 130
creating 125
Default 124
deleting 127
description of use 123
Developers 124
how they work with services 127
inherited 128
Internal 124
predefined 124
protecting use of remote server aliases 86
removing from services 130
removing protection from files 132
Replicators 124
updating 125
access file, using to control access to files 131
accessing
  any Web document for package 196
  external user and group information 177
  home page for package 196
ACLs. See Access Control Lists (ACLs)
activating
  packages 198
adding
  Access Control Lists (ACLs) 125
  additional ports 69
  administrators 109
  aliases for remote servers 87
database aliases 257
developers 110
groups 57
  packages 198
  port restrictions 133, 134
  services to server manually 231
  subscribers to packages 209
  user accounts 48
  users to a group 57, 58
Administrator user account 47
administrators
  adding alternate administrators 21
  defining 109
  defining external users as 183
e-mail address for guaranteed delivery 247
password for predefined user account 21
predefined ACL, description 124
predefined group, description 55
predefined user account, description 21, 47
receiving messages, overview 21
responsibilities 20
role 20
SMTP server for e-mail address for guaranteed delivery 247
specifying who receives messages 92
Administrators ACL 124
Administrators group 55
aliases, database
  adding 257
deleting 262
  information to supply for 256
  updating 261
aliases, remote servers
deleting 89
identifying 87
testing connection 88
updating 89
Allow By Default
access to services through a port 164
port IP access (custom) 81
port IP access (global) 80
anonymous
  predefined ACL, description 124
Anonymous ACL 124
Anonymous group 56
API
   related documentation 16
architecture, SAP BC Server 24
archiving packages 201
AS/400 systems, port queue size 286
assigning
   Access Control Lists (ACLs) to files the server can serve 131
   Access Control Lists (ACLs) to folders 128
   Access Control Lists (ACLs) to services 130
audit log
   date format 93
   description 90, 294
   example 294
   field descriptions 294
   location of 91
   log rotation 93
audit-trail logging
   description of log 90, 294
   overview 29
   writing log to screen 34
authenticating
   basic authentication 141
   client certificates 136
   customizing authentication with pluggable module 141
   description 135
   reasons server cannot use client certificates 137
   registering alternate authentication processor 145
   unregistering alternate authentication processor 146
   using NT Challenge/Response 168
   using user names and passwords (basic) 141
   using user names and passwords with NT Challenge/Response 168
   when invalid password supplied 141
   when invalid user name supplied 141
   when it occurs 135
   when user name not supplied 141
   when using external directories 183
   when using LDAP 183
   when using NIS 183
   authentication module
      creating 144
   automatic pull facility 217

B
B2B services. See services blocking incoming requests to server 77

C
C/C++ services, adding to server manually 231
caching service results
   overview 104
   resetting for all services 105
   resetting for single service 106
   viewing statistics 106
canceling
   package subscriptions 224
   when services scheduled to execute 242
CA.s. See certificate authorities (CAs)
certificate authorities (CAs)
   certificates to validate client certificates 138
   requesting digital certificate from 115
certificate signing request (CSR), creating 115
Certificate Toolkit 115
certificates, digital
   certificates required to validate client certificates 138
   description of use 113
   obtaining certificate authority's 116
   reasons server cannot use to authenticate 137
   requesting for server 115
   using for authentication 136
Challenge/Response (NT)
   activating 169
deactivating 169
   using to authenticate 168
changing
   Access Control Lists (ACLs) 125
   aliases for remote servers 89
database aliases 261
license key 64
membership for groups 60
password for user accounts 51
passwords 50
primary port 76
   when services scheduled to execute 241
Index

checking if server is running 35
checklists
  configuring database connections 270
  configuring server 266
  deploying the server 266
  implementing SSL 114, 121
  installing server 266
  installing services 269
  security 271
  setting up user accounts, groups, and ACLs 268
classpath, updating servers 257
client certificates
  certificates required to validate 138
  description 136
  information required to use 138
  presenting multiple 121
  reasons server cannot use to authenticate 137
closing log files 93
code subdirectory 190
command line
  parameters for starting server 32
  starting server from 32
communications with server, securing with SSL 111
configuration settings
  bypass list for proxy servers 85
  controlling who can set 108
  date format in log files 93
  descriptions 276
  guaranteed delivery 247
  how long to keep inactive sessions 96
  how much information in server log 91
  how to set 42
  how to store closed log files 93
  LDAP 178
  license keys 64
  log file rotation 93
  NIS 180
  overriding when starting server 32
  ports 66
  proxy servers 83
  server.cnf file 42
  SMTP server to use for e-mail 92
when to close log files 93
who receives error messages 92
configuring
  additional ports 69
  bypass list for proxy servers 85
  checklist 266
  controlling who can configure the server 108
  date format in log files 93
  description of all settings 276
  different primary port 76
  guaranteed delivery 247
  how long to keep inactive sessions 96
  how much information in server log 91
  how to store closed log files 93
  log file rotation 93
  ports 66
  proxy servers 83
  server 63
  SMTP server to use for e-mail 92
  SSL 117
  SSL, checklist 114, 121
  SSL, required information 115
  use of LDAP or NIS 177
  user account to use 47
  when to close log files 93
  who receives error messages 92
controlling
  access to services and files 123
  access to services by port 132
  server SSL security level by port 132
  who can configure the server 108
  who can develop services 110
conventions used in this document 14, 15
copying packages
  ACL used 124
  group used 56
  how to 212
  publisher tasks 208
  requesting subscriptions to packages 219
  subscriber tasks 217
  to other servers 201
  user account used 48
creating
a release of a package 212
Access Control Lists (ACLs) 125
certificate signing request (CSR) 115
packages 198
packages distribution files 212
CSR (certificate signing request), creating 115

D
database aliases
adding 257
deleting 262
information to supply for 256
updating 261
database connections
checklist for configuring 270
information to supply 256
making JDBC driver classes accessible 257
preloading JDBC drivers at server initialization 256
specifying database location 259
specifying JDBC driver 259
debug mode of the server 33
Default ACL 124
Default user account 47
defining
Access Control Lists (ACLs) 125
administrators 109
database aliases 257
developers 110
groups 57
packages 198
subscribers to packages 209
user accounts 46
deleting
Access Control Lists (ACLs) 127
aliases for remote servers 89
database aliases 262
groups 60
packages 200
ports 77
subscribers to packages 211
user accounts 49
Deny By Default
access to services through a port 133, 164

port IP access (custom) 82
port IP access (global) 79
Developer
predefined user account to use 48
privilege required to access server from 110
Developer user account 48
developers
defining 110
defining external users as 184
predefined ACL, description 124
predefined group, description 56
predefined user account, description 48
Developers ACL 124
Developers group 56
digital certificates
See also certificates
certificates required to validate client certificates 138
client certificates for authentication 136
description of use 113
obtaining certificate authority's 116
reasons server cannot use to authenticate 137
requesting 115

disabling
guaranteed delivery for outbound requests 250
packages 199
ports 77
users 54
displaying
active sessions 35, 65
documentation for packages 196
folders 229
license key 64
licensed session limit 65
log files 96
log information on screen 34
membership for groups 60
package information 193
package subscriptions 218
packages residing on your server 192
service information 230
service statistics 106
services 229
subscribers to packages 208
when services scheduled to execute 241
when system tasks execute 242
whether packages are enabled/disabled 193
whether packages are loaded/unloaded 192
distribution files for packages
  creating 212
  sending 212
DMZ
  running a reverse invoke Integration Server in 147
doc subdirectory 190
documentation
  conventions used 14
  for your packages 196
  printing 18
  related manuals 16
  viewing 17
E
e-mail
  address for messages when guaranteed delivery fails 92, 247, 254
  address for messages when internal errors occur 92
  address for messages when output binding fails 92
  SMTP server to use 92
  SMTP server to use for messages when guaranteed delivery fails 247, 254
e-mail ports
  assigning 69
  configuring 66
Enabled icon, color descriptions 193
enabling
  packages 199
  ports 78
  users 54
encryption
  improving efficiency of 117
error log
  date format 93
  description 90, 294
  example 295
  field descriptions 295
  location of 91
  log rotation 93
  viewing 96
errors
  log 90, 294
  specifying who receives messages 92
Event Manager 233
eventcfg.bin file 233
events
  running services in response to 233
Everybody group 56
executing
  replication services 233
  services 27
  services at scheduled times 234
  shutdown services 232
  startup services 232
external directories
  accessing users and passwords in 141
  considerations for user accounts and groups 181
  granting users access to services and files 185
  granting users administrator privileges 183
  granting users developer privileges 184
  how server authenticates 183
  how server uses 176
  overview 176
  to stop using 181
  when server accesses 177
external groups
  assigning administrator privileges 183
  assigning developer privileges 184
  assigning to ACLs 185
  configuring server to use 177
  how server uses 176
  when server accesses 177
external user accounts
  assigning access to services and files 185
  assigning administrator privileges 183
  assigning developer privileges 184
  configuring server to use 177
  how server uses 176
  when server accesses 177
F
factory class
creating 143
files, removing Access Control List (ACL) protection from 132
filtering service
setting up on a reverse invoke Integration Server 158
firewall
running an internal server behind 147
folders
assigning Access Control Lists (ACLs) to 128
description 228
listing 229
FTP ports, assigning 69
full release
of a package 203

gateway. See wireless gateway
group name
description 55
specifying for groups 55
groups
adding 57
adding users to 57, 58
administrator privileges 55
Administrators 55
Anonymous 56
changing membership 60
considerations when using external directories 181
defining 55
deleting 60
developer privileges 56
Developers 56
externally-defined 176
group name 55
overview 46
predefined 55
privileges that can be shared 55
purpose 46
replicator privileges 56
Replicators 56
settings 55
specify members of 55
specify users that belong to 47
viewing membership 60
guaranteed delivery
administering 251
audit-trail log for inbound transactions 249
audit-trail log for outbound transactions 251
configuring 247
description of 246
disabling outbound transactions 250
e-mail address and SMTP server for error notification 254
e-mail address for error notification 247
handling heuristic failures 248
handling restart after a failure 248
how often to cleanup inbound job store 248
inbound job store, description 247
location of inbound job store 248
location of outbound 250
reinitializing for inbound transactions 252
reinitializing for outbound transactions 252
requests to other servers 246
retry after server failure 248
retry wait 250
server failure 248
service threads 250
shutting down 251
SMTP server for error notification 247
specify how long transactions active 250
submitting outbound transactions 250
testing 253
thread pool 250
transaction logs 90, 298

Handheld Device Markup Language. See HDML (Handheld Device Markup Language)
hardware acceleration 117
HDML (Handheld Device Markup Language) 302, 305
HDML pages
accessing with wireless devices 305
heuristic failures, guaranteed delivery
specifying how to handle 248
home page for packages 196
hosts
allowing inbound requests 79, 82
deny ing inbound requests 80
HTTP ports
assigning 69
changing to or from primary port 76
HTTP proxy server
bypass list 85
configuring 83
HTTPS ports
assigning 69
changing to or from primary port 76
HTTPS proxy server
bypass list 85
configuring 83

I
IMAP mail server
configuring port for 66
inherited Access Control Lists (ACLs) 128
inner firewall
running an internal server behind 147
installing
package published from another server 225
run-time classes 270
server 266
services, checklist 269
Internal ACL 124
internal server
setting up in a reverse invoke configuration 160
IP access to ports
customizing for individual ports 81
setting globally 78

J
Java services
adding to server manually 231
JDBC drivers
making classes accessible 257
preloading at server initialization 256
specifying in database alias 259
job store
for inbound guaranteed delivery transactions 247
how long outbound transactions active 250
location of inbound 248
location of outbound transactions 250
removing expired inbound transactions 248
submitting outbound transactions 250
JVM
checking version when copying a package 206
JVM Information 299

K
key pair used for SSL
description 115
obtaining 115

L
LDAP
accessing users and passwords in 141
assigning groups to ACLs 185
configuration settings 178
configuring server to use 178
considerations for user accounts and groups 181
granting users access to services and files 185
granting users administrator privileges 183
granting users developer privileges 184
how server authenticates 183
how server uses 176
when server accesses 177
license keys
changing 64
description 64
licensed sessions 65
renewal reminders 65
renewing 65
viewing 64
viewing licensed session limit 65
viewing number of active sessions 65
when seession limit reached 65
listening ports
overview 24
listing
active sessions 35
folders 229
log information on screen 34
packages residing on your server 192
services 229
Loaded? icon, color descriptions 192
loading packages 198
locking out
  users 54
log files
  audit log 90, 294
  closing 93
  date format 93
  error log 90, 294
  location 91
  log rotation 93
  save closed files in multiple directory 94
  save closed files in single directory 93
  server log 90, 295
  session log 90, 297
  transaction logs 90, 298
  types 90
  viewing 96
logging
  closing log files 93
  date format 93
  location of log files 91
  log rotation 93
  overview 29
  save closed log files in multiple directory 94
  save closed log files in single directory 93
  types of log files 90
  writing log to screen 34
logging out of Server Administrator 42

M
manifest file
  for packages 191
messages
  guaranteed delivery messages 92
  internal error messages 92
  service error messages 92
  SMTP server to use 92
  specify who receives when errors occur 92

N
naming services 228
NIC

O
output bindings, messages when they fail 92
overview
  audit-trail logging 29
  caching service results 29, 104
  external directories 176
  guaranteed delivery 246
  package replication 201
  ports 24
  proxy servers 27
  receiving administrator messages 21
  SAP BC Server 24
  server security 28, 108
  services 228
  user and groups 46

P
packages
  ACL for package replication 124
  activating 198
  archive 201
  archiving 201
  canceling subscriptions to 224
  code subdirectory 190
  specifying which one server is to listen on for incoming requests 283

NIS
  assigning groups to ACLs 185
  configuration settings 180
  configuringservvertouse 180
  considerations for user accounts and groups 181
  granting users access to services and files 185
  granting users administrator privileges 183
  granting users developer privileges 184
  how server authenticates 183
  how server uses 176
  when server accesses 177
  ns subdirectory 190
NT Challenge/Response
  activating 169
  deactivating 169
  using to authenticate 168
Index

copying to another server 201
creating 198
delete 200
deleting 200
description 188
determining whether enabled/disabled 193
determining whether loaded/unloaded 192
directory structure 189
disabling 199
displaying documentation for 196
displaying subscriptions to 218
doc subdirectory 190
Enabled icon color descriptions 193
enabling 199
full vs. patch release 203
group for package replication 56
guidelines for package replication 207
home page 196
information you can view 191
installing published package 225
Loaded? icon color descriptions 192
location 189
making available 199
manifest file 191
ns subdirectory 190
predefined 188
prohibiting access to 199
pub subdirectory 190
publishing to other servers 212
recovering 200
release 202
reloading 198
replicating 201, 212
residing on your server 192
retrieving automatically 217
retrieving manually 217
safe delete 200
sample services 189
subscribing to 219
tasks you can perform 197
templates subdirectory 190
user account for package replication 48
viewing information about 193
who can subscribe to 207
widls subdirectory 190
partial release
of a package 203
passwords
changing 50, 51
description 47
predefined Administrator user account 21
requirements 50
specifying in user accounts 47
patch release
of a package 203
PDF, viewing 17
pluggable module
as alternate authentication processor 141
customizing authentication with 141
POP3 mail server
configuring port for 66
port queue size, lowering for AS/400 286
ports
adding additional 69
Allow By Default access to services 164
changing primary 76
configuring 66
controlling access to services through 132
controlling IP access to 78
controlling SSL security level of 132
deleting 77
Deny By Default access to services 133, 164
disabling 77
enabling 78
overview 24
reasons to add additional 69
reasons to change primary 76
resetting access to 135
ports, listening
adding additional 69
Allow By Default access to services 164
changing primary 76
configuring 66
deleting 77
Deny By Default access to services 133, 164
disabling 77
enabling 78
overview 24
reasons to change primary 76
resetting access to 135
preventing
access to packages 199
hosts that can connect to server 80
use of ports 77
printing
this guide 18
private/public key pair used for SSL
description 115
obtaining 115
privileges
administrator, description 108
administrator, granting 109
administrator, granting when using external directory 183
developer, description 110
developer, granting 110
developer, granting when using external directory 184
replicator 56
that groups can share 55
protocols
e-mail 66
FTP 66
HTTP 66
HTTPS 66
SOCK 66, 148
SSLSOCK 66, 148
proxy port
on reverse invoke Integration Server, definition 148
proxy servers
bypassing 85
configuring 83
overview 27
pub subdirectory 190
publishing packages
creating the distribution file 212
guidelines 207
how to 212
identifying recipients (subscribers) 209
installing published package 225
removing recipients (subscribers) 211
requesting subscriptions 219
sending the distribution file 212
sending the release 212
updating subscriber information) 210
who can publish 207
who can subscribe 207
publishing servers
creating package distribution file 212
displaying subscribers 208
publishing packages 212
sending package release 212
tasks 208
who can publish 207
pulling a package
automatically 217
manually 217
R
receiving administrator messages 21
recovering packages 200
registering
alternate authentication processor 145
registration port
on reverse invoke Integration Server, definition 148
reinitializing
guaranteed delivery for inbound transactions 252
guaranteed delivery for outbound transactions 252
releases
creating 212
full vs. patch 203
sending 202, 212
reloading packages 198
remote servers
identifying aliases 85
removing
Access Control Lists (ACLs) 127
Access Control Lists (ACLs) from services 130
database aliases 262
groups 60
packages 200
ports 77
scheduled execution of service 242
subscribers to packages 211
user accounts 49
renewing license key 65
replicating packages
  ACL 124
  group 56
  guidelines 207
  how to 212
  overview 201
  publisher tasks 208
  Replicator user account 207
  Replicators ACL 207
  Replicators group 207
  requesting subscriptions to packages 219
  subscriber tasks 217
  user account 48
  who can subscribe 207
replication services
  what they are 233
  Replicator user account 48
  Replicators ACL 124
  Replicators group 56
resetting
  access to ports 135
  cache for all services 105
  cache for single service 106
restarting
  guaranteed delivery for inbound transactions 252
  guaranteed delivery for outbound transactions 252
  reasons for restarting server 36
  server 36
restricting
  access to Server Administrator 108
  access to server from Developer 110
  access to services and files 123
  access to services by port 132
  hosts that can connect to server 79, 82
resuming scheduled execution of service 242
retry guaranteed delivery 248
reverse invoke
  overview 147
  setting up 149
  setting up a filtering service 158
role
  of administrator 20
  of SAP BC Server 24
rotation of log files 93
round robin
  method for selecting connection to internal server in reverse
  invoke configuration 148
run-time classes, installing 270

S
SAP BC
  recovery after hardware or software failure 36
SAP BC Server
  accessing with URLs 304
  architecture 24
  audit-trail logging, overview 29
  configuration settings 276
  configuring 63
  date format in logs. 93
  debug mode 33
  deploying 266
  how it maintains log files 93
  how SSL is used 112
  identify hosts that can connect 79, 82
  installing 266
  license keys 64
  log showing server information 90, 295
  overview 24
  preventing use of port 77
  process for executing services 27
  process when starting 34
  rejecting connections from specified hosts 80
  requesting digital certificate 115, 116
  restarting 36
  retrieving data for services 27
  role 24
  security overview 28, 108
  setting up aliases for remote servers 85
  specify who receives messages when errors occur 92
  starting from command line 32
  starting on UNIX 32
  testing installation 272
  using with wireless gateways 302
  when it authenticates clients 135
when to restart 36
wireless devices, communicating with 302
wireless devices, using with 302
scheduling execution of services
canceling scheduled user task 242
changing scheduled times 241
complex scheduling options 235
description 234
examples of complex scheduling options 237
execute one time 235
execute repeated times 235
how often to execute 234, 235
resuming scheduled user task 242
suspending scheduled user task 241
system tasks 234
user tasks 234
viewing scheduled times 241
viewing when system tasks execute 242
Secure Sockets Layer (SSL)
background information 112
certificate signing request (CSR) 115
checklist to implement 114, 121
client certificates 136
configuring server to use 117
controlling security level by port 132
how server uses 112
information required to implement SSL 115
information required to use client certificates 138
obtaining certificate authority's certificate 116
obtaining private/public key pair 115
private key pair 115
requesting digital certificate 115
use of certificates 113
use of digital certificates 113
security
checklist 271
checklist to implement SSL 114, 121
controlling access to services and files 123
controlling access to services by port 132
controlling SSL security level by port 132
controlling who can access the server 123
controlling who can develop services 110
information required to implement SSL 115
overview 28, 108
securing server communications 111
sending
distribution files 212
releases 212
server. See SAP BC Server
Server Administrator
controlling access to 108
description 20, 40
how to use 41
logging out of 42
picture of 41
starting 40
server log
configuring how much information is recorded 91
date format 93
description 90, 295
element 296
field descriptions 296
location of 91
log rotation 93
viewing 96
server security. See security
server.cnf
description of settings 276
guaranteed delivery settings 247
how to set configuration settings 42
location 276
updating from Server Administrator 100
services
ACL used when no assigned ACL 124, 128
assigning Access Control Lists (ACLs) to 130
caching results, overview 104
caching service results overview 29
canceling scheduled user task 242
changing scheduled times of execution 241
controlling access by port 132
controlling who can access 123
controlling who can develop 110
deleting its Access Control List (ACL) 127
denying access to external users 185
fully-qualified names 228
granting access to external users 185
guaranteeing delivery of requests to server 246
guaranteeing delivery of responses from services 246
guidelines for using startup/shutdown/replication 233
how Access Control Lists (ACLs) work 127
how server executes 27
how server retrieves data for 27
how to control access to 123
information to schedule user tasks 234
inherited Access Control Lists (ACLs) 128
invoking with URLs 304
listing 229
manually adding to server 231
naming 228
overview 228
removing Access Control Lists (ACLs) from replication 233
resetting cache for all services 105
resetting cache for single service 106
resuming scheduled user task 242
running in response to specific events 233
samples in WmSamples 189
scheduling to execute 234
shutdown 232
startup 232
suspending scheduled user task 241
tasks you can perform 231
testing 231
viewing information about 230
viewing scheduled times of execution 241
viewing service statistics 106
viewing when system tasks execute 242
when replication services are executed 233
when shutdown services are executed 232
when startup services are executed 232
sessions
how long to keep inactive sessions 96
log file 90, 297
maximum number allowed per license 65
stopping all 35
timeout limit 96
viewing 35
sessions, licensed
limit 65
viewing active 65
viewing limit 65
shutdown services
guidelines for use 233
what they are 232
shutting down
guaranteed delivery 251
SAP BC Server 35
server with restart 36
SMTP address for messages when guaranteed delivery fails 247
SMTP server
specifying for error messages generated during guaranteed delivery processing 254
to use for e-mail when sending notification of errors 92
SOCK
protocol for unencrypted transfer in reverse invoke configuration 148
SSL. See Secure Sockets Layer (SSL)
SSLSOCK
protocol for encrypted transfer in reverse invoke configuration 148
starting
command line parameters 32
guaranteed delivery for inbound transactions 252
guaranteed delivery for outbound transactions 252
Server Administrator 40
server from command line 32
server on UNIX 32
server, overriding configuration settings 32
server, process 34
startup services
guidelines for use 233
what they are 232
statistics log
date format 93
sessions
how long to keep inactive sessions 96
log file 90, 297
maximum number allowed per license 65
stopping all 35
timeout limit 96
viewing 35
sessions, licensed
limit 65
viewing active 65
viewing limit 65
shutdown services
guidelines for use 233
what they are 232
shutting down
guaranteed delivery 251
SAP BC Server 35
server with restart 36
SMTP address for messages when guaranteed delivery fails 247
SMTP server
specifying for error messages generated during guaranteed delivery processing 254
to use for e-mail when sending notification of errors 92
SOCK
protocol for unencrypted transfer in reverse invoke configuration 148
SSL. See Secure Sockets Layer (SSL)
SSLSOCK
protocol for encrypted transfer in reverse invoke configuration 148
starting
command line parameters 32
guaranteed delivery for inbound transactions 252
guaranteed delivery for outbound transactions 252
Server Administrator 40
server from command line 32
server on UNIX 32
server, overriding configuration settings 32
server, process 34
startup services
guidelines for use 233
what they are 232
statistics log
date format 93
sessions
how long to keep inactive sessions 96
log file 90, 297
maximum number allowed per license 65
stopping all 35
timeout limit 96
viewing 35
sessions, licensed
limit 65
viewing active 65
viewing limit 65
shutdown services
guidelines for use 233
what they are 232
shutting down
guaranteed delivery 251
SAP BC Server 35
server with restart 36
SMTP address for messages when guaranteed delivery fails 247
SMTP server
specifying for error messages generated during guaranteed delivery processing 254
to use for e-mail when sending notification of errors 92
SOCK
protocol for unencrypted transfer in reverse invoke configuration 148
SSL. See Secure Sockets Layer (SSL)
SSLSOCK
protocol for encrypted transfer in reverse invoke configuration 148
starting
command line parameters 32
guaranteed delivery for inbound transactions 252
guaranteed delivery for outbound transactions 252
Server Administrator 40
server from command line 32
server on UNIX 32
server, overriding configuration settings 32
server, process 34
startup services
guidelines for use 233
what they are 232
statistics log
date format 93
location of  91
log rotation  93
stopping
  active sessions  35
  server  35
  server with restart  36
use of external directories  181
subscribing servers
  canceling subscriptions  224
  displaying  208
  identifying  209
  installing published package  225
  removing  211
  requesting subscriptions  219
  tasks  217
  updating information for  210, 222
  who can subscribe  207
subscribing to packages
  displaying current subscriptions  218
  guidelines  207
  how to  219
  manually pulling current subscriptions  218
  updating subscription information)  222
  who can subscribe  207
subscriptions
  canceling  224
  displaying  218
  installing published package  225
  pulling  218
  requesting from a remote server  219
suspending scheduled execution of service  241
system tasks
  description  234
  viewing when scheduled to execute  242
T
templates subdirectory  190
testing
  connection to remote servers  88
  guaranteed delivery  253
  installation of server  272
  services  231
The  24

This  266
time to live (TTL), specifying for guaranteed delivery  250
timeout limits
  sessions  96
transaction logs
  date format  93
  description  90, 298
  example  299
  field descriptions  299
  inbound guaranteed delivery transactions  249
  location of  91
  log rotation  93
  outbound guaranteed delivery transactions  251
TTL (time to live), specifying for guaranteed delivery  250
typographical conventions  14, 15
U
unauthenticated users
  predefined group, description  56
UNIX
  starting SAP BC Server  32
unregistering
  alternate authentication processor  146
updating
  Access Control Lists (ACLs)  125
  aliases for remote servers  89
  database aliases  261
  license key  64
  membership for groups  60
  subscriber information  210
  subscription information  222
  when services scheduled to execute  241
URLs
  accessing the server with  304
  invoking services with  304
  using to access HDML or WML pages  305
user accounts
  account to configure and manage server  47
  account to use when connecting from Developer  48
  adding  48
  Administrator  21
  considerations when using external directories  181
  deleting  49
Index

description 47
externally-defined 176
group membership 47
overview 46
password 47
predefined 47
purpose 46
settings 46
user name 47
using to authenticate (basic) 141
using to authenticate with NT Challenge/Response 168
when client does not supply a user name 47
user name
externally-defined 176
specifying in user account 47
using to authenticate (basic) 141
using to authenticate with NT Challenge/Response 168
user tasks
canceling scheduled execution 242
changing when scheduled to execute 241
complex scheduling options 235
examples of complex scheduling options 237
information to schedule services 234
resuming scheduled execution 242
schedule to execute one time 235
schedule to execute repeated times 235
suspending scheduled execution 241
viewing when scheduled to execute 241
users
authenticating 135
disabling 53, 54
enabling 53, 54
locking out 53
Using Tokens to Counter XSRF Attacks 170

V

versions
checking when a package is copied 206
viewing
active sessions 35, 65
documentation for packages 196
documentation in PDF format 17
folders 229
license key 64
licensed session limit 65
log files 96
membership for groups 60
package information 193
package subscriptions 218
packages residing on your server 192
service information 230
service statistics 106
services 229
subscribers to packages 208
when services scheduled to execute 241
when system tasks execute 242
whether packages are enabled/disabled 193
whether packages are loaded/unloaded 192
viewing this document in PDF format 17

W

WAP gateway. See wireless gateway
watt.debug.level 276
watt.debug.logfile 276
watt.debug2.facList 276
watt.debug2.logstringfile 276
watt.net.maxRedirects 277
watt.net.proxyHost 277
watt.net.proxyPass 277
watt.net.proxyPort 277
watt.net.proxySkipList 277
watt.net.proxyUser 277
watt.net.proxyList 278
watt.net.proxyUser 278
watt.net.retries 278
watt.net.secureProxyHost 278
watt.net.secureProxyPass 278
watt.net.secureProxyPort 278
watt.net.secureProxyUser 278
watt.net.timeout 278
watt.net.useCookies 278
watt.net.userAgent 278
watt.security.caCert 279
watt.security.CADir 279
watt.security.privateKey 279
watt.security.signedCert 279
watt.security.ssl.ignoreExpiredChains 279
watt.server.cronCatchup 281
writing a user-defined program to run on the reverse invoke
  Integration Server  148

Z
zip files for packages
  creating  212
  sending  212